Univerza v Ljubljani
Fakulteta za elektrotehniko

Marko Meža

Samodejna interpretacija rezultatov predtransfuzijskih testiranj na slikah gelskih kartic

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Jurij F. Tasič

Ljubljana, 2007

Kazalo

Slike

Tabele

Zahvala

Zahvalil bi se rad vsem, ki so me pri mojem raziskovalnem delu usmerjali in mi pomagali. Posebej bi se rad zahvalil mentorju prof. dr. Juriju Tasiču.

Pomembne nasvete in pomoč sem prejel od marsikoga. Za pomoč se zahvaljujem doc. dr. Andreju Koširju in doc. dr. Primožu Rožmanu. Za koristne debate in pomoč pri reševanju problemov se zahvaljujem tudi ostalim sodelavcem iz Laboratorija za digitalno obdelavo signalov slik in videa.

Zahvaljujem se osebju Zavoda za transfuzijsko medicino, s katerim smo skupaj razvili in v prakso vpeljali sistem za telekonzultacije v transfuzijski medicini. Z uporabo sistema so mi zagotovili pomembne podatke za izgradnjo učne in testne množice.

Moje raziskovalno delo je omogočilo Ministrstvo za šolstvo znanost in šport RS, ki je financiralo moj podiplomski študij, in Fakulteta za elektrotehniko v Ljubljani, ki mi je nudila primerno okolje za delo.

Posebna zahvala pa velja mojim staršem, ki sta me vedno spodbujala in mi bila na voljo, ko sem ju potreboval. Brez njiju te disertacije ne bi bilo.

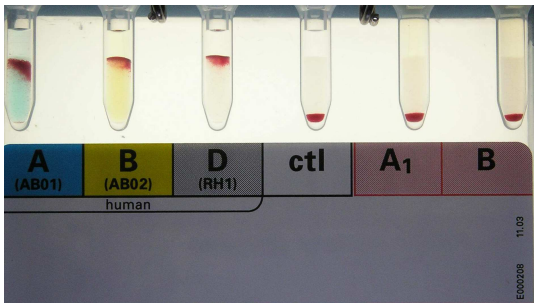
Povzetek

V disertaciji smo raziskali, kako uporabiti algoritme strojnega učenja za samodejno interpretacijo rezultatov predtransfuzijskih preiskav z gelsko metodo. Naš cilj je bila izbira kombinacije algoritmov strojnega učenja, s katerimi smo zgradili celoten model interpretacije rezultatov predtransfuzijskih preiskav. Model smo zgradili z algoritmi strojnega učenja na podlagi diagnostičnih podatkov in pripadajočih odločitev specialistov transfuzijske medicine.

Ukvarjali smo se s predtransfuzijskimi preiskavami, ki se izvajajo z gelsko metodo. Gelska metoda je osnovana na zaznavanju stopnje jakosti aglutinacije eritrocitov z različnimi reagenti [12][13]. Aglutinati, nastali pri reakciji, se pri centrifugiranju skozi inerten gel zaustavijo v gelu [13]. Metoda se izvaja z gelskimi karticami; To so plastične kartice z vdelanimi 6 kolonami z gelom in reagenti. Fotografija gelske kartice je predstavljena na sliki 1. V postopku preiskave specialist vizualno pregleda centrifugirano gelsko kartico s produkti reakcije in interpretira rezultat preiskave. Postopek interpretacije preiskave je sestavljen iz dveh korakov. V prvem koraku specialist za vsako od 6 kolon določi stopnjo jakosti aglutinacije. V drugem koraku pa na podlagi kombinacije stopenj jakosti aglutinacije v posameznih kolonah določi končno interpretacijo preiskave.

Zgradili smo sistem, ki posnema postopek interpretacije, ki ga izvajajo specialisti transfuzijske medicine. Sistem za samodejno interpretacijo smo zgradili iz dveh modelov. Prvi model modelira določanje stopnje jakosti aglutinacije v vsaki od šestih kolon gelskih kartic. Vhodni podatki v ta model so slike posameznih kolon. Drugi model modelira določanje dokončne interpretacije preiskave. Vhodni podatki v ta model so vektorji, katerih elementi so določene stopnje jakosti aglutinacije za vsako od šestih kolon gelske kartice. Postopek, ki ga posnemata modela, je predstavljen na sliki 2.

Za gradnjo modelov s postopki strojnega učenja potrebujemo učno in testno množico podatkov. Z učno množico smo modele samodejne interpretacije zgradili, s testno pa preverili njihovo učinkovitost. Učno in testno množico smo zgradili iz podatkov, ki smo



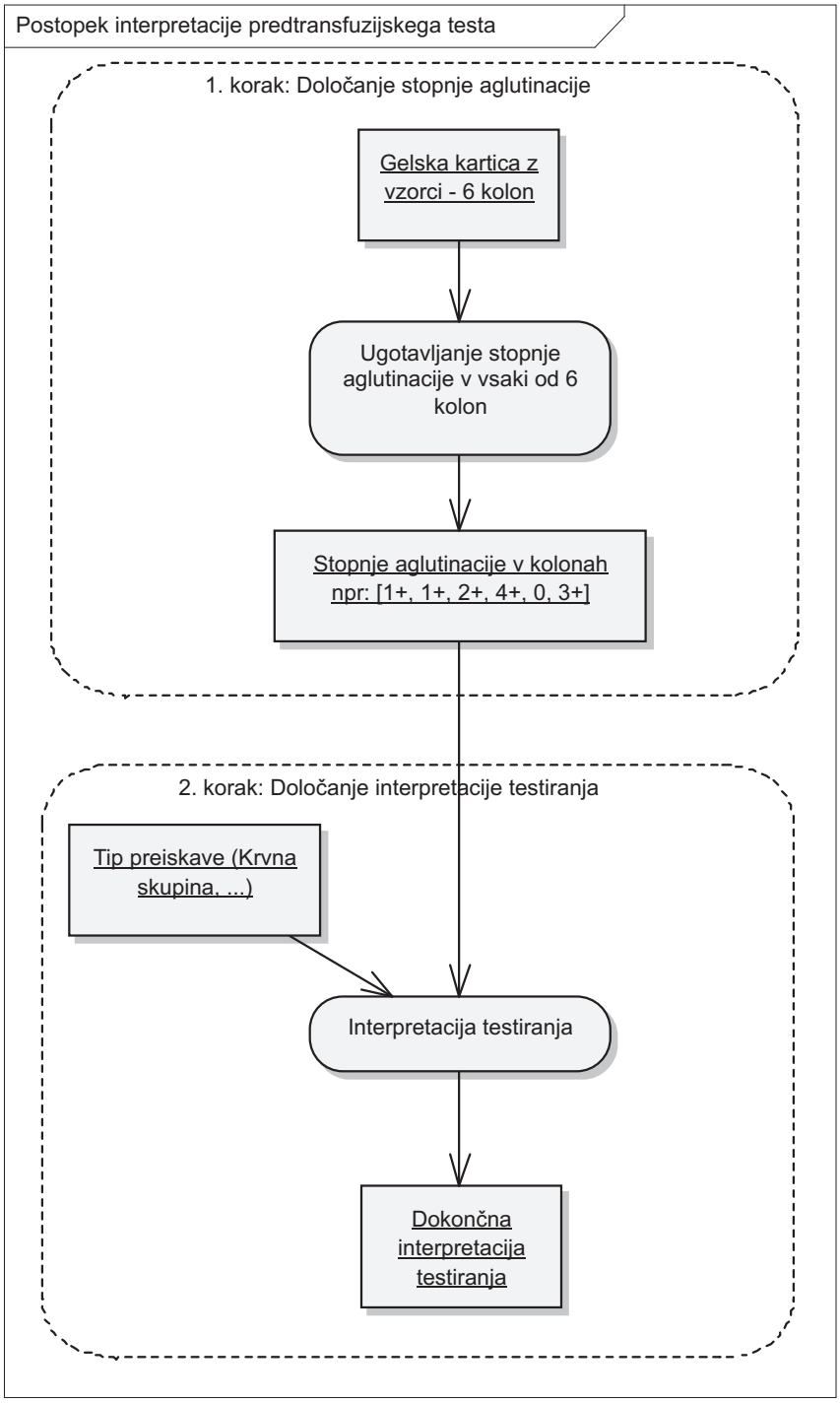
Slika 1: Fotografija gelske kartice za določanje krvne skupine z vzorci po končani reakciji in centrifugiranju.

jih pridobili iz sistema za telekonzultacije v transfuzijski medicini. Sistem smo razvili in uvedli v transfuzijsko prakso transfuzijskih oddelkov bolnišnic v Sloveniji. Specialisti transfuzijske medicine lahko z uporabo sistema na daljavo interpretirajo predtransfuzijske preiskave. Podatki, ki so jih specialisti uporabili za interpretacijo preiskav in pripadajoče interpretacije, se beležijo v podatkovni bazi. Ti podatki so predstavljali osnovo za gradnjo učne in testne množice podatkov, ki smo ju uporabili za razvoj sistema za samodejno interpretacijo rezultatov predtransfuzijskih preiskav.

V disertaciji smo opisali postopke, s katerimi smo iz delujočega sistema za telekonzultacije zajeli podatke. Zajeti podatki so bili v surovi obliki. Preden smo jih lahko obdelali s postopki strojnega učenja, smo jih morali predobdelati. S predobdelavo smo iz njih izluščili za nas koristno informacijo. Rezultat predobdelave podatkov so vektorji lastnosti, ki učinkovito zapišejo za nas koristno informacijo.

Iz sistema zajeti podatki so bile slike gelskih kartic z določenimi stopnjami jakosti aglutinacije za vsako od kolon in pripadajočimi interpretacijami. Slike in pripadajoče stopnje jakosti aglutinacije smo uporabili za gradnjo modela določanja stopnje jakosti aglutinacije. Stopnje jakosti aglutinacije in pripadajoče interpretacije smo uporabili za gradnjo modela za določanje dokončne interpretacije.

Za določanje stopnje jakosti aglutinacije je pomembna porazdelitev eritrocitov po višini posameznih kolon, saj je ta neposredno povezana z iskano stopnjo jakosti aglutinacije [13]. Za ugotovitev porazdelitve v posameznih kolonah smo morali iz slik gelskih kartic najprej razpoznati področja posameznih kolon. Iz slik kolon smo izluščili zanimiva področja, ki vsebujejo eritrocite. Izluščili smo jih s segmentacijskimi algoritmi. Uporabljenim algoritmom so za osnovo služile lastnosti barvnih prostorov, v katere smo preslikali slike. V



Slika 2: Interpretacija predtransfuzijske preiskave v dveh korakih. V prvem je določena stopnja jakosti aglutinacije v vsaki od 6 kolon. V drugem je na podlagi stopenj jakosti aglutinacije in tipa preiskave določena dokončna interpretacija preiskave.

okviru raziskav, zajetih v doktorsko disertacijo smo razvili in preizkusili enajst različnih segmentacijskih algoritmov za segmentacijo eritrocitov v slikah kolon in ugotovili, da smo najboljše rezultate dobili z uporabo nelinearnega filtriranja in kombinacije posameznih komponent RGB slike.

Iz segmentiranih slik kolon smo izračunali vektorje porazdelitve, iz teh pa vektorje lastnosti. Vektorji lastnosti na zgoščen način predstavljajo informacijo o stopnji jakosti aglutinacije v slikah kolon. Vektorji porazdelitve predstavljajo porazdelitev eritrocitov po višini kolon. Vektorji porazdelitve so dolgi tipično 400 elementov. Vektorje porazdelitve smo skrajšali in iz njih izračunali vektorje lastnosti tako, da smo vektorjem porazdelitve zmanjšali dimenzijo s približno 400 elementov na 1 do 20 elementov. Za ta izračun smo uporabili in primerjali dve metodi. To sta metodi – metoda analize osnovnih komponent PCA, opisana v literaturi [14], in metoda z zrnjenjem ZRNI, opisana v podpoglavju 4.3.3.2.

Z navedenimi segmentacijskimi algoritmi in različno parametriziranima metodama izračuna vektorjev lastnosti smo generirali 352 podatkovnih naborov vektorjev lastnosti s pripisanimi stopnjami jakosti aglutinacije za množico 182 kolon. Te podatkovne nabore smo uporabili za gradnjo modela določanja stopnje jakosti aglutinacije v kolonah. Za gradnjo modela za določanje stopnje jakosti aglutinacije smo uporabili in preverili 49 algoritmov strojnega učenja. Uporabljeni algoritmi so našteti v tabeli 4.1. Uporabo in preverjanje algoritmov smo izvedli z orodjem WEKA [5]. Delovanje posameznih algoritmov smo primerjali tako, da smo dobljene modele preizkusili z navzkrižno validacijo [2][4]. Navzkrižno validacijo smo uporabili zato, ker je nabor podatkov, zajetih iz sistema, vseboval premalo vzorcev za učinkovito generiranje ločene testne in učne množice. Z metodo navzkrižne validacije smo iste podatke učinkovito uporabili za gradnjo modelov in njihovo testiranje. Cena, ki smo jo plačali za to, je bil veliko daljši postopek validacije, saj smo vsak model zgradili in preizkusili 10-krat. Pri vsaki gradnji in preizkusu smo uporabili različne dele množice vzorcev, ki je bila na voljo. Rezultate posameznih preizkusov posameznega modela smo po navodilih postopka navzkrižne validacije [2] povprečili.

Rezultate navzkrižne validacije modelov smo zabeležili v matriko pravilnih in napačnih razvrstitev. Iz te matrike smo za grobo izbiro posameznih kombinacij segmentacijskih algoritmov, metod za izračun vektorjev lastnosti in algoritmov strojnega učenja izračunali delež pravilno razvrščenih. Delež pravilno razvrščenih je skalarna vrednost in pove, v kolikšnem deležu preizkusov s posameznimi vzorci je model deloval pravilno.

Postopek strojnega učenja smo ponovili za gradnjo drugega modela, modela interpretacije stopenj jakosti aglutinacije v dokončno interpretacijo predtransfuzijske preiskave.

Uporabili smo prijeme, opisane za gradnjo modelov določanja stopnje jakosti aglutinacije. Izgubna predobdelava podatkov za gradnjo tega modela ni bila potrebna, ker so zajeti podatki iz sistema za telekonzultacije že bili nominalni in diskretni. Posamezne vzorce namreč predstavljajo vektorji, dolžine 6. Elementi teh vektorjev so določene stopnje jakosti aglutinacije pripadajočih kolon obravnavane gelske kartice. Različne modele interpretacije smo zgradili z uporabo 49 algoritmov strojnega učenja in jih preizkusili z navzkrižno validacijo. Za vsako napoved izbranega modela smo zabeležili delež pravilno razvrščenih.

Model določanja stopnje jakosti aglutinacije in model določanja dokončne interpretacije rezultata smo združili in izračunali deleže uspešnosti za posamezne interpretacije, ki sta jih dala združena modela. Na podlagi analize deležev uspešnosti združenih modelov smo podali izbiro najučinkovitejše kombinacije segmentacijskih algoritmov, metod za izračun vektorjev lastnosti, algoritmov strojnega učenja za gradnjo prvega modela in algoritmov strojnega učenja za gradnjo drugega modela.

Ključne besede

- samodejna interpretacija
- strojno učenje
- model določanja stopnje jakosti aglutinacije eritrocitov
- model interpretacije predtransfuzijske preiskave na osnovi stopenj jakosti aglutinacije eritrocitov
- gelska metoda
- telekonzultacije v transfuzijski medicini

Abstract

In the following doctoral dissertation we have described our research on the machine learning algorithms for the automatic pre-transfusion test interpretation. Analyzed pre-transfusion tests were carried out by means of a gel-card agglutination detection method. The main goal of the research work was the selection of machine learning algorithms suitable for building a pre-transfusion test interpretation model. We have built that test interpretation model using machine learning algorithms, based on a combination of pre-transfusion test diagnostic data and interpretations of that test, determined by transfusion medicine specialists.

We have focused our research on the area of pre-transfusion tests which are performed using the gel method. The gel method is based on the detection of the agglutination level of red blood cells with different reagents [12][13]. In the proces, the agglutinated red blood cells are separated from the non-agglutinated red blood cells by means of centrifuging the reaction products through the inert sephadex gel. The method is performed by using special gel-cards. Gel-cards are plastic cards with six micro-tubes embedded into them. A photo, representing a gel-card is shown on Figure 3. Interpretation of the test involves visual observation and interpretation of the micro-tubes content after reaction and centrifuging procedure. It consists of two steps. In the first step, the transfusion medicine specialist determines the agglutination strength for each of the six micro-tubes. In the second step, based on previously determined agglutination strengths, the specialist determines the final pre-transfusion test interpretation.

We have built a system, which mimics the interpretation process as carried out by blood transfusion specialists. We have used two models for building the system. The first one models the agglutination strength determination in each of the six micro-tubes. It takes pictures of the micro-tubes, containing blood as input. The second model models the final pre-transfusion test interpretation. It takes the vectors of determined agglutination strengths as input. We have illustrated the whole process in Figure 4.

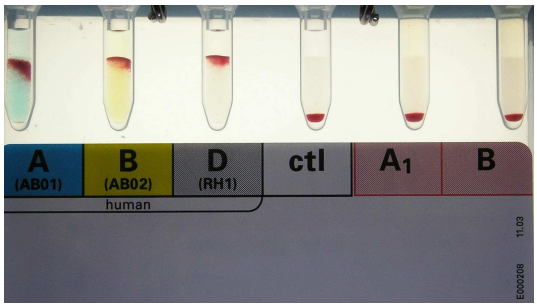


Figure 3: Image of gel-card for determination of human blood type. The blood samples have been administered, the reaction occurred and the gel-card was centrifuged.

Machine learning algorithms require training and test data-sets for the development of the models. Using the training data-set is required for model construction, whereas the test data-set is required for the validation of the model performance. We have captured the training and test data-sets from the blood transfusion teleconsulting system, which we have developed and integrated into the blood transfusion practice of the transfusion wards in the hospitals in Slovenia. Using this system, the transfusion medicine specialists remotely interpret pre-transfusion tests. Data, used for these interpretations is recorded in the system's database. We have built our training and test data-sets using this data.

In this dissertation we have described the data capture process from the live blood transfusion teleconsulting system. The captured data was in its raw form. We had to preprocess the captured data in order for it to be in a suitable form for processing with the machine learning algorithms. During the preprocessing process, we extracted the information, useful for the interpretation process in the form of feature vectors.

The data, captured from the teleconsulting system consisted of the gel-card images with determined agglutination strengths for each of the six micro-tubes and corresponding final test interpretations. We have used these images and determined agglutination strengths for the construction of the agglutination strength determination model. We have used determined agglutination strengths and final result interpretations for the construction of the final interpretation determination model.

The distribution of red blood cells across the height of the micro-tubes [13] determines the agglutination strength of that micro-tube. To asses the distribution in each micro-tube, we had to extract the image of each micro-tube from the image of the gel-card. To extract the image, we had to determine the exact location of micro-tubes on the gel-card

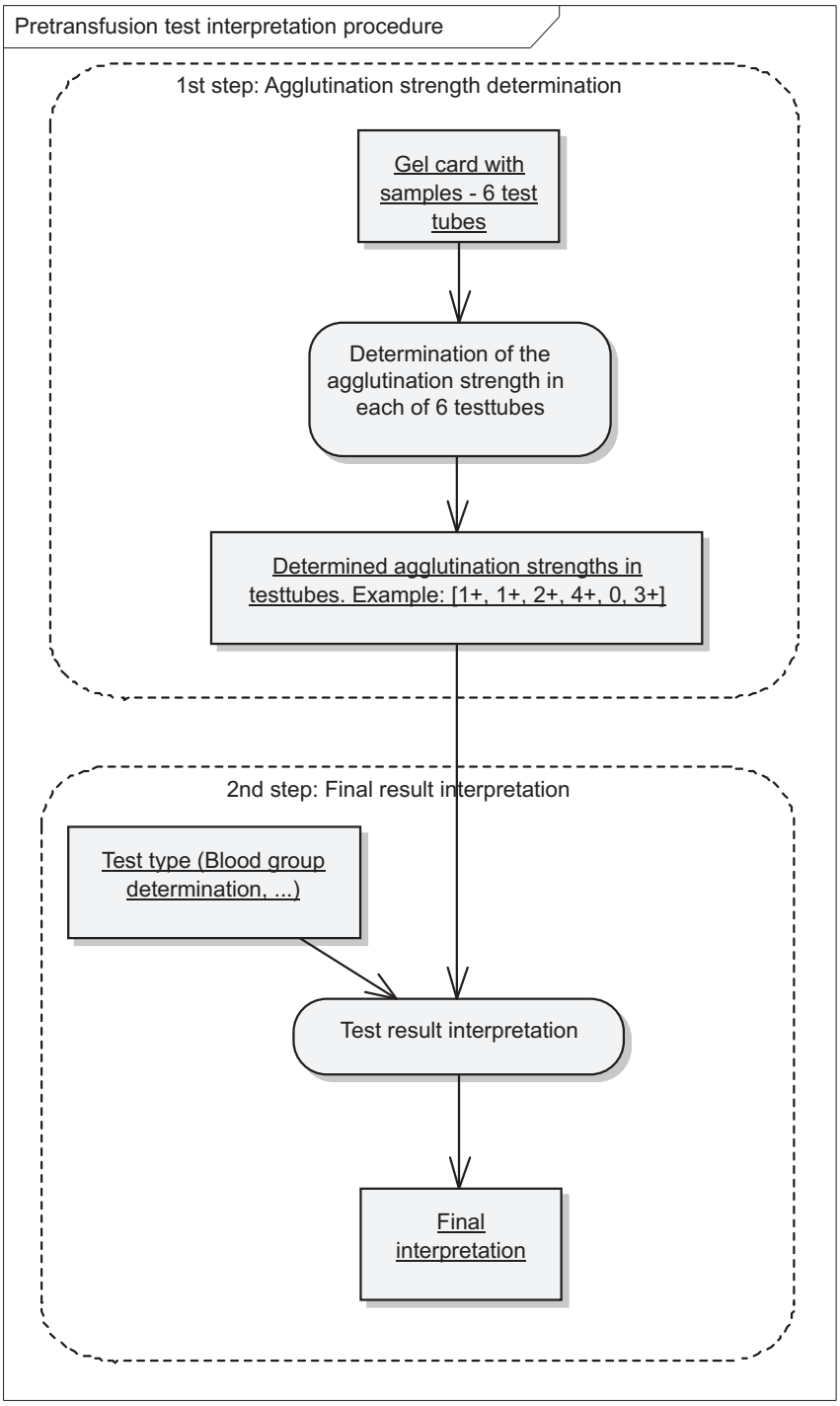


Figure 4: Two step pre-transfusion test interpretation. The first step is the agglutination strength determination for each of six micro-tubes. The second step is the final pre-transfusion test interpretation, based on the agglutination strengths and test type.

images. After we extracted the micro-tube images, we extracted the areas, containing the red blood cells. We accomplished this by means of segmentation algorithms. All used segmentation algorithms were based on different color space features. We developed and evaluated eleven segmentation algorithms and concluded, that we obtain the best results when using non-linear filtering and the combination of different color channels of the image in the RGB color space.

In the next stage of the procedure we computed the distribution vectors from the segmented images. Distribution vectors describe the distribution of blood cells across the micro-tube height. Distribution vectors are typically 400 elements long. We used these distribution vectors to compute feature vectors, which in a condensed manner represent information about agglutination strength of the observed micro-tube. The typical length of feature vectors was 1 to 20 elements. We used and compared two different methods for this purpose. These methods are Principal Component Analysis (PCA), described in [14] and the granulation method – ZRNI, described in chapter 4.3.3.2.

Using the above mentioned segmentation algorithms and feature vector calculation algorithms, we have generated 352 data-sets, containing feature vectors and corresponding agglutination strengths. Each data-set consisted of information, describing 182 micro-tubes. These data-sets were used for the construction of the agglutination strength determination model. We tested 49 different machine learning algorithms for the construction of the model. Algorithms are listed in the table 4.1. The machine learning algorithms were used and tested using the WEKA data-mining suite. We used the 10-fold cross-validation method [2][4] to test and compare the performance of the models. Cross-validation was used, because the available data-set was not large enough to construct suitable training and test data-sets. Using cross-validation, we were able to effectively use available data-sets for both the construction and testing of the models. The price we had to pay, was a longer validation process, because, we had to build each model 10 times (10 folds), using a different part of the available data-set each time. The final result of the model performance was obtained by averaging of the results, obtained in each fold, as prescribed by the author in [2].

The results of each validation were recorded into the confusion matrix. Data recorded in the confusion matrix served for a coarse selection of combination of the segmentation, feature extraction and machine learning algorithms. We made a coarse selection, based on precision parameters, calculated from the confusion matrices. Precision is a scalar, representing the fraction of correctly classified samples among all classifications.

In the next step, we repeated the use of machine learning algorithms for the construction of the model for the second step – the model of the final pre-transfusion test interpretation determination. We used the same principles as we had for the construction of the agglutination strength determination model. Since the data obtained from the system for teleconsulting was already in a form, suitable for use with machine learning algorithms, there was no need for lossy data preprocessing. Each sample employed in this step is represented by vectors that are six elements long. Each element of this vector represents agglutination strength of the corresponding micro-tube of the observed gel-card. We built 49 models of the final pre-transfusion test interpretation determination, using 49 different machine algorithms. We tested obtained models by using the cross-validation method. We used the cross-validation results to compute the precision for each of the interpretations proposed by the models.

In the final step of the research we combined the agglutination strength determination model and the final pre-transfusion test interpretation determination model. We calculated the precision of interpretations, obtained by the combined models. We used the combinations of coarsely selected agglutination strength determination models and final pre-transfusion test interpretation determination models. Based on this calculation we decided on the selection of the best combination of the segmentation algorithm, feature extraction algorithm, and machine learning algorithms for the construction of the first and second model.

Key words

- automatic interpretation
- machine learning
- agglutination strength determination model
- final test result determination model based on agglutination strength combination
- gel method
- pre-transfusion testing
- teleconsulting in the blood transfusion medicine

Poglavje 1

Uvod

1.1 Predtransfuzijske preiskave

Pred vsako transfuzijo krvi je potrebno izvesti obvezne predtransfuzijske serološke preiskave. Preiskave se izvajajo z uporabo seroloških diagnostičnih metod. Za vse potrebne predtransfuzijske preiskave so primerne diagnostične metode, ki se uvrščajo v skupino preiskav z gelsko metodo [12][13]. Najpogosteje je uporabljenih naslednjih pet preiskav:

- določanje krvne skupine AB0 in RhD bolnika in dajalca,
- indirektni Coombsov test bolnika,
- direktni Coombsov test bolnika,
- navzkrižni preizkus (kri bolnika in dajalca),
- specifikacija protiteles bolnika.

Osnova za gelsko metodo je zaznavanje imunske reakcije med protitelesi in antigeni, ki se odraža kot aglutinacija (zlepljenje) eritrocitov [12]. Postopek izvajanja gelske metode je natančno predpisan [12]. Izvaja se z uporabo standardnih diagnostičnih pripomočkov, kamor spadajo reagenti, gelske kartice, naprave za doziranje krvi in centrifugiranje ter ostali pribor.

Zadnji korak v postopku preiskave je odčitavanje in interpretacija rezultatov preiskave. Ob predpostavki, da se preiskava izvede v skladu s predpisanimi standardi in z uporabo

predpisane opreme, sta odčitavanje in interpretacija rezultatov ključna za pravilno izvedeno preiskavo. Odčitavanje in interpretacijo lahko izvedejo ustrezno usposobljeni specialisti transfuzijske medicine.

1.1.1 Razpršenost zahtev, malo primerov/center, malo specialistov

Storitve transfuzijske medicine se v Sloveniji opravljajo na desetih oddelkih za transfuzijo krvi pri bolnišnicah in na Zavodu Republike Slovenije za transfuzijsko medicino (ZTM). Zavod za transfuzijsko medicino predstavlja osrednji laboratorij za področje predtransfuzijskih preiskav in je kot tak tudi najbolje opremljen. Dnevno se v Sloveniji za transfuzijo izda nekaj sto enot krvi, od tega polovica na oddelkih za transfuzijo krvi, preostala polovica pa na ZTM. Za vsako izdano enoto krvi se izvedejo predtransfuzijske serološke preiskave.

Ker na ZTM opravijo večino predtransfuzijskih preiskav, so zaradi pospešitve in poenostavitve preiskav opremljeni z opremo, ki omogoča delno avtomatizirano izvajanje predtransfuzijskih preiskav, vključno z odčitavanjem očitnih rezultatov. V primeru, da rezultati preiskave niso jasni, obstoječi sistem zahteva posredovanje specialista transfuzijske medicine.

Za zagotavljanje dejavnosti transfuzijske službe je v Sloveniji potrebnih vsaj 11 nenehno dežurnih zdravnikov specialistov transfuzijske medicine. Zaradi pomanjkanja kadrov in različnega števila obdelanih primerov med posameznimi ustanovami niha tudi kakovost opravljenih storitev [1].

Zato se je pojavila potreba po optimizaciji postopka predtransfuzijskih preiskav. Namen optimizacije postopka je izboljšanje kakovosti izvajanja predtransfuzijskih preiskav. Kakovost predtransfuzijskih preiskav naj bo enako kakovostna na vseh oddelkih, ki nudijo storitve transfuzijske medicine. Zaradi pomanjkanja ustrezno usposobljenega osebja je potrebno optimizirati tudi delo osebja.

1.2 Vpeljava telekonzultacijskega sistema

Do nedavnega so se v mejnih primerih odčitavanja in interpretacije rezultatov predtransfuzijskih preiskav z gelsko metodo manj izkušeni zdravniki s področja transfuzijske medicine ali medicinsko osebje posvetovali z bolj izkušenimi na tem področju s pomočjo telefonske

konzultacije in kurirske izmenjave problematičnih vzorcev krvi.

V slovensko transfuzijsko službo smo uvedli pilotni sistem za telekonzultacije v transfuzijski službi. Z uporabo sistema je mogoče na daljavo opraviti najzahtevnejši korak v postopku predtransfuzijskih preiskav – odčitavanje in interpretacijo rezultatov preiskave [15]. Na ta način smo znatno olajšali in pospešili dostop do ekspertize osrednjega laboratorija vsem transfuzijskim oddelkom po državi. Sistem omogoča prenos in hrambo vseh podatkov, potrebnih za interpretacijo rezultatov predtransfuzijskih preiskav z gelsko metodo.

V okviru novih konceptov je bila postavljena logistična zasnova sistema, ki predvideva novo delovno mesto dežurnega konzultanta specialista transfuzijske medicine. Dežurni konzultant bo na voljo 24 ur na dan in bo po potrebi nudil storitve strokovne interpretacije rezultatov preiskav, ki se izvajajo kjerkoli po državi. Možnost 24-urne takojšnje telekonzultacije z usposobljenim konzultantom in drugimi strokovnjaki s področja transfuzijske medicine znatno skrajša postopek predtransfuzijskih preiskav. Skrajšanje trajanja postopka predstavlja izboljšavo kakovosti storitev transfuzijske službe, ki je še posebno očitna v primerih, ko na transfuzijskem oddelku ni prisotnega zdravnika specialista transfuzijske medicine. Če so obravnavani primeri urgentne narave in je potrebna hitra interakcija med dežurnim konzultantom in konzultirajočim, sistem omogoča vzpostavitev videokonferenčne povezave in delo v realnem času.

1.3 Samodejna interpretacija rezultatov predtransfuzijskih preiskav

Z namenom poenostavite dela specialistov transfuzijske medicine smo načrtovali sistem za samodejno interpretacijo rezultatov predtransfuzijskih preiskav. Sistem za samodejno interpretacijo specialistu na podlagi analize slik gelskih kartic in na podlagi tipa preiskave določi in predlaga stopnje jakosti aglutinacije posameznih kolon in interpretacijo le-teh v dokončno interpretacijo preiskave. Sistem se interpretacije preiskav nauči iz predhodnih odločitev specialistov, zabeleženih v sistemu za telekonzultacije v transfuzijski medicini. Pri postopku interpretacije posnema delo specialista, ki dela v dveh korakih. Specialist v prvem koraku oceni stopnjo jakosti aglutinacije v vsaki od šestih kolon gelske kartice. V drugem koraku pa na podlagi kombinacije jakosti stopenj aglutinacije v šestih kolonah gelske kartice določi dokončno interpretacijo preiskave.

Razvoj in raziskave sistema za samodejno interpretacijo predtransfuzijskih preiskav, opravljenih z gelsko metodo, predstavljata jedro doktorske disertacije. Sistem se bo kot modul vključil v obstoječi sistem za telekonzultacije in bo nudil podporo medicinskemu strokovnemu osebju pri interpretaciji preiskav. Na podlagi primerov določenih interpretacij predtransfuzijskih preiskav, ki so jih določili specialisti transfuzijske medicine z uporabo sistema za telekonzultacije v transfuzijski medicini, bo modul adaptivno izboljševal in popravljal model za samodejno interpretacijo rezultatov predtransfuzijskih preiskav, izpeljanih z gelsko metodo. Sistem bo pri svojem delovanju posnemal postopek, ki ga za interpretacijo izvedejo specialisti transfuzijske medicine.

1.3.1 Kratek opis delovanja sistema za samodejno interpretacijo

Sistem deluje v dveh fazah. V prvi fazi, fazi učenja, sistem na podlagi diagnostičnih podatkov, ki jih med svojim delom kot vprašanja vnašajo konzultirajoči, in interpretacij preiskav, ki jih kot odgovore vnesejo specialisti transfuzijske medicine, gradi model interpretacije preiskav. V drugi fazi, fazi interpretacije, sistem na podlagi analize diagnostičnih podatkov in v prvi fazi naučenega modela interpretacije predlaga interpretacije preiskav.

1.3.1.1 Faza učenja

Vhodni podatki v prvo fazo, fazo učenja, so:

- slike gelskih kartic, na katerih je jasno vidna vsebina kolon s centrifugiranimi rezultati reakcije med vzorci in reagenti,
- določene stopnje jakosti aglutinacije kolon gelske kartice,
- tip preiskav,
- dokončne interpretacije preiskav.

1.3.1.2 Faza interpretacije preiskav

Vhodni podatki v drugo fazo, fazo interpretacije preiskav, so:

- slike gelskih kartic, na katerih je jasno vidna vsebina kolon s centrifugiranimi rezultati reakcije med vzorci in reagenti,
- tip preiskave.

1.3.2 Pridobivanje učne in testne množice

Telekonzultacijski sistem predstavlja vir učnih in testnih podatkov za razvoj sistema za samodejno interpretacijo predtransfuzijskih preiskav. Konzultacije se izvajajo v primerih nejasnih in z vidika interpretacije težavnih primerih. Te primere razreši specialist osrednjega laboratorija. Rešitve primerov in njihovi vhodni podatki se shranjujejo v sistemu. Zato smo imeli dostop do vira z vidika strojnega učenja kakovostnih podatkov za gradnjo učne množice z enega mesta. Ta vir se z uporabo sistema za telekonzultacije nenehno dopolnjuje, kar omogoča, da lahko gradimo vedno boljšo in popolnejšo učno množico, ki je osnova za gradnjo učinkovitih modelov interpretacije rezultatov.

1.3.3 Validacija sistema

Validacija in preizkušanje sistema predstavlja pomemben korak v izdelavi le tega. Da je sistem za samodejno interpretacijo primeren za delo v praksi, potrebujemo podatke o zanesljivosti njegovega delovanja. Potrebno je vedeti, v kolikšni meri lahko zaupamo predlaganim rezultatom. Zanesljivost delovanja je predstavljena kot verjetnost, da je sistem za samodejno interpretacijo rezultat preiskave napovedal pravilno. Poimenovali smo jo delež uspešnosti.

Naš model interpretacije rezultatov predtransfuzijskih preiskav deluje v dveh korakih. Posamezna koraka predstavljata dva samostojna modela. Sistem smo sestavili iz dveh zaporedno vezanih modelov. Prvi model modelira določanje stopnje jakosti aglutinacije v kolonah. Drugi model, ki sledi prvemu, določi na podlagi določenih stopenj jakosti aglutinacije v kolonah dokončno interpretacijo rezultata predtransfuzijske preiskave. Oba modela smo razvili neodvisno drug od drugega. Za vsakega smo ocenili delež pravilno napovedanih rezultatov. Iz deležev pravilno napovedanih rezultatov posameznih modelov smo ocenili delež pravilnih končnih interpretacij – delež uspešnosti.

1.3.4 Obstoječi postopki samodejne interpretacije

Trenutna na trgu dostopna oprema DiaMed-ID Maestro, ki jo proizvaja podjetje DiaMed omogoča delno samodejno odčitavanje in interpretacijo predtransfuzijskih preiskav. Oprema je sposobna odčitati očitne rezultate, pri neočitnih pa je potrebno posredovanje specialistov transfuzijske medicine. Oprema, ki omogoča omenjeno funkcionalnost, je precej draga in si jo lahko privošči le osrednji transfuzijski laboratorij, ki dnevno ob-

dela veliko primerov predtransfuzijskih preiskav z gelsko metodo. Ostali laboratoriji pa postopke predtransfuzijskih preiskav še vedno izvajajo ročno.

Napravo za samodejno interpretacijo rezultatov je potrebno ob postavitvi v laboratorij najprej kalibrirati in nastaviti delovne parametre. Naprava je na podlagi nastavitev parametrov kalibracije sposobna določati posamezne stopnje jakosti aglutinacije eritrocitov v posameznih kolonah. Za določanje stopnje jakosti aglutinacije je področje vsake kolone na gelskih karticah razdeljeno na 5 oken. Zgornje okno je za pozitivne rezultate, spodnje za negativne, vmesna za vmesne. Algoritem ugotavlja prisotnost aglutinatov v teh oknih in na podlagi prisotnosti aglutinatov določi stopnjo jakosti aglutinacije.

Pri postopku kalibracije naprave je potrebno definirati položaje in velikost treh področij kolon gelskih kartic. Ko so v obravnavani gelski kartici določene vse kolone, interpretira naprava glede na tip preiskave kombinacijo kolon iz pravilnostne tabele posamezne preiskave v rezultat preiskave.

Podrobnosti o delovanju sistema niso javno dostopne, ker jih podjetje DiaMed-ID skriva kot poslovno skrivnost. Predstavljene podatke smo pridobili iz reklamnega materiala podjetja DiaMed-ID in z razgovori z uporabniki te opreme.

1.4 Kratek pregled vsebine

V 2. poglavju – *Opis problema* smo podali opis predtransfuzijskih preiskav na področju Slovenije. Opisali smo gelsko metodo, na kateri smo osnovali naše delo. Dejstvo, da je rezultate preiskave z gelsko metodo mogoče zajeti v obliki slike, omogoča interpretacijo rezultatov na daljavo in tudi izdelavo sistema za samodejno interpretacijo le-teh. Identificirali smo dva osnovna problema, določanje stopnje jakosti aglutinacije v kolonah gelskih kartic in dokončno interpretacijo rezultatov predtransfuzijskih preiskav.

V 3. poglavju – *Obstoječe rešitve* smo opisali obstoječe stanje na področju transfuzijske medicine s poudarkom na predtransfuzijskih preiskavah. Podali smo ozadje, ki omogoča pridobivanje podatkov za gradnjo sistema za samodejno interpretacijo predtransfuzijskih preiskav. Predstavili smo pregled osnov algoritmov strojnega učenja, ki smo jih uporabili in preizkusili. Podali smo definicijo osnovne terminologije in načine za predstavitev naučenega znanja. Pregled osnovnih pristopov strojnega učenja zajema klasifikacijo, asociiranje, rojenje in numerično napovedovanje. Za primerjavo delovanja metod strojnega učenja nujno potrebujemo metode za ocenjevanje njihove uspešnosti. Zato smo predstavili tudi načine za ocenjevanje učinkovitosti modelov, zgrajenih z algo-

ritmi strojnega učenja.

V 4. poglavju – *Sistem za samodejno interpretacijo* smo podrobno razdelali problematiko, povezano z izdelavo sistema za samodejno interpretacijo. Obravnavali smo celotno pot razvoja. Ta pot je sestavljena iz spoznavanja s problemom in pregledom ozadja problema, zajemom in registracijo podatkov, izbiro ustreznih algoritmov strojnega učenja, izdelavo modelov sistema interpretacije preiskav in njihovo validacijo.

V 5. poglavju – *Rezultati eksperimentov* sledi predstavitev rezultatov eksperimentalnega dela. Eksperimentirali smo s kombinacijo različnih metod segmentacije, načini izračunov vektorjev lastnosti in metodami strojnega učenja. Predstavili smo rezultate grobe izbire najoptimalnejših kombinacij metod in rezultate izbire kombinacije najoptimalnejše kombinacije. Podali smo tudi oceno deleža uspešnosti celotnega postopka za samodejno intepretacijo preiskav za določanje krvne skupine.

Sledi zaključek in diskusija, ki smo ji podali v 6. poglavju – *Zaključek*. V zaključku smo navedli sklepne misli, predstavili pa smo tudi ideje za nadaljnje delo. Podali smo tudi seznam izvirnih prispevkov znanosti.

Poglavje 2

Opis problema

V tem poglavju smo predstavili ozadje in motivacijo za razvoj sistema za samodejno interpretacijo predtransfuzijskih seroloških preiskav. Podali smo opis gelske metode, uporabljenih diagnostičnih pripomočkov ter sistema, ki omogoča interpretacijo rezultatov predtransfuzijskih preiskav na daljavo. Razvoj in izdelava sistema za samodejno interpretacijo predstavlja reševanje več različnih problemov. Ti problemi zajemajo spoznavanje s področjem dela, pregled ozadja, zajem in registracijo podatkov, izbiro ustreznih algoritmov strojnega učenja, s katerimi smo zgradili modele sistema za samodejno interpretacijo predtransfuzijskih preiskav, in validacijo zgrajenih modelov.

2.1 Predstavitev problema

Kot smo že predstavili v uvodu, se storitve transfuzijske medicine Sloveniji opravljajo na desetih oddelkih za transfuzijo krvi in na Zavodu Republike Slovenije za transfuzijsko medicino (ZTM). Dnevno se izda nekaj sto enot krvi, od tega polovica na oddelkih za transfuzijo krvi po državi, preostala polovica pa na ZTM. Za vsako izdano enoto krvi se izvedejo predtransfuzijske serološke preiskave. Za zagotavljanje dejavnosti transfuzijske službe je potrebnih vsaj enajst nenehno dežurnih zdravnikov specialistov transfuzijske medicine. Zaradi pomanjkanja kadrov in različnega števila obdelanih primerov med posameznimi ustanovami niha tudi kakovost opravljenih storitev [1].

V slovensko transfuzijsko službo smo uvedli pilotni sistem za telekonzultacije v transfuzijski službi. Sistem omogoča nudenje ekspertize strokovnjakov transfuzijske medicine na daljavo. Z uporabo sistema je mogoče na daljavo opraviti najzahtevnejši korak v postopku predtransfuzijskih preiskav – odčitavanje in interpretacijo rezultatov preiskave

[15].

Specialisti transfuzijske medicine dnevno rutinsko interpretirajo mnogo predtransfuzijskih preiskav. Postopek interpretacije lahko poenostavimo z uvedbo sistema za samodejno interpretacijo rezultatov predtransfuzijskih preiskav. Sistem naj se kot modul vgradi v sistem za telekonzultacije v transfuzijski medicini in naj osebju predlaga interpretacije predtransfuzijskih preiskav. Sistem naj se postopka samodejne interpretacije nauči na podlagi analize rešenih primerov predtransfuzijskih preiskav, ki so jih rešili specialisti transfuzijske medicine.

2.2 Cilji raziskav doktorske disertacije

Cilj raziskav, zajetih v doktorsko disertacijo, je bila gradnja sistema za samodejno interpretacijo predtransfuzijskih preiskav, opravljenih z gelsko metodo. Sistem naj posnema postopek interpretacije predtransfuzijskih preiskav, ki ga opravljajo specialisti transfuzijske medicine. Postopek naj posnema z modeli interpretacije preiskav, zgrajenimi z algoritmi strojnega učenja.

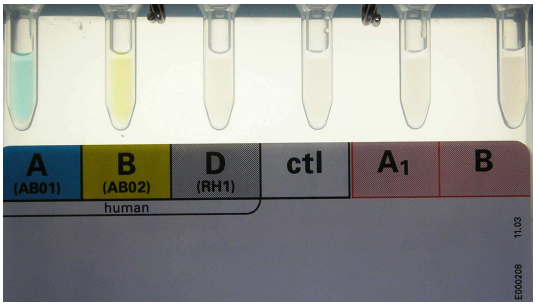
Modele naj zgradi na osnovi kombinacije diagnostičnih podatkov in pripadajočih interpretacij preiskav, ki so jih določili specialisti transfuzijske medicine.

2.3 Opis predtransfuzijskih preiskav

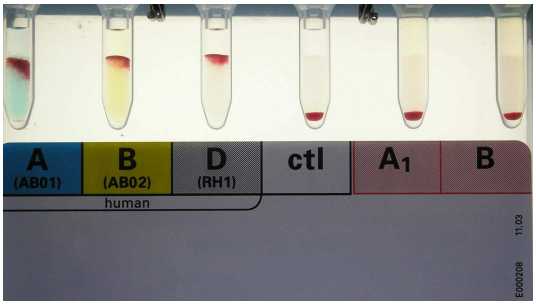
Najpogostejše predtransfuzijske serološke preiskave v Republiki Sloveniji (RS) zajemajo 5 različnih preiskav: določanje krvne skupine AB0 in RhD bolnika in dajalca, indirektni Coombsov test test bolnika, direktni Coombsov test test bolnika, navzkrižni preizkus (kri bolnika in dajalca) in specifikacijo protiteles bolnika [16]. Preiskave se izvajajo z gelsko metodo [12][13].

2.3.1 Gelska metoda

Predtransfuzijske serološke preiskave so osnovane na zaznavanju reakcij med antigeni na eritrocitih s protitelesi. Za podrobnejši opis glejte podpoglavje 3.1.2. Pri preiskavah je ključnega pomena natančnost odčitavanja in interpretiranje reakcije preiskave. Natančnost je še posebej pomembna v primerih, ko je reakcija šibka. Metoda se izvaja z uporabo gelskih kartic. To so plastične kartice, v katere je vdelanih šest kolon. Slika 2.1 prikazuje



(*a*)



(*b*)

Slika 2.1: Fotografija prazne gelske kartice (a) in fotografija gelske kartice za določanje krvne skupine z vzorci po končani reakciji in centrifugiranju (b).

primer prazne gelske kartice (a) in gelske kartice z vzorci krvi (b).

V primeru, da je v posamezni koloni reakcija potekla, se eritrociti v tej koloni zlepijo v mrežo – aglutinat. Interpretacija preiskave poteka v dveh korakih. V prvem koraku je potrebno za vsako kolono določiti stopnjo jakosti aglutinacije. Stopnja jakosti aglutinacije je povezana s porazdelitvijo aglutinatov po volumnu kolon [12][13]. Ugotavlja se šest različnih stopenj jakosti aglutinacije, ki se jih označuje z oznakami NEG, 1+, 2+, 3+ 4+, DCP. V naši obravnavi smo dodatno ugotavljali tudi, če kolona ne vsebuje krvi. Natančnejša razlaga določanja stopnje jakosti aglutinacije je podana v podpoglavju 3.1.2.1. V drugem koraku, dokončni interpretaciji preiskave, specialist interpretira nabor določenih stopenj jakosti aglutinacije v posameznih kolonah v končno interpretacijo preiskave. Za vsako od petih različnih tipičnih preiskav obstaja končni nabor možnih rezultatov preiskave in preiskavi lasten način določanja dokončne interpretacije. Seznam možnih interpretacij za vsako od petih preiskav je podan v podpoglavju 3.1.2.2.

2.4 Sistem za samodejno interpretacijo predtransfuzijskih preiskav

Sistem za samodejno interpretacijo smo sestavili iz dveh modelov, ki posnemata prvi in drugi korak interpretacije preiskave z gelsko metodo.

2.4.1 Namen sistema za samodejno interpretacijo predtransfuzijskih preiskav

Namen sistema za samodejno interpretacijo je podpora strokovnemu osebju s predlaganjem interpretacije predtransfuzijskih preiskav. Dokončno odločitev o interpretaciji in izdaji izvida bo moral kljub predlogu sistema pregledati in odobriti specialist transfuzijske medicine. Pri tem mu bo v pomoč predlagana interpretacija z oceno deleža uspešnosti, s katero bo ocenjena vsaka predlagana interpretacija. Delež uspešnosti je število v intervalu [0..1], ki pove, v kolikšni meri lahko specialist zaupa predlagani interpretaciji. Za vsako interpretacijo, ki jo bo predlagal sistem za samodejno interpretacijo, bo na podlagi rezultatov validacije v postopku intepretacije uporabljenih modelov ocenjen delež uspešnosti.

Sistem je poleg predlaganja interpretacij uporaben tudi za kontrolo napak. S spremljanjem dela specialista in vzporednim napovedovanjem interpretacije preiskav le-te primerja z interpretacijami, ki jih je določil specialist. V primeru razhajanja bo sistem specialista opozoril, da se je pri interpretaciji preiskave morda zgodila napaka. Specialist bo lahko interpretacijo preiskav po opozorilu podrobneje pregledal in se odločil o pravilni interpretaciji. Sistem je uporaben tudi kot učni pripomoček v postopku izobraževanja specialistov transfuzijske medicine.

2.4.2 Gradnja modelov sistema z algoritmi strojnega učenja

Sistem smo zgradili iz modelov, ki modelirajo interpretacijo predtransfuzijskih preiskav na način, kot to počnejo specialisti transfuzijske medicine. Specialisti interpretirajo preiskave v dveh korakih. V prvem koraku določijo stopnjo jakosti aglutinacije v posameznih kolonah gelskih kartic. Temu koraku sledi drugi korak, v katerem na podlagi kombinacije stopenj jakosti aglutinacije v posameznih kolonah določijo rezultat preiskave. Postopek interpretacije smo posnemali z dvema modeloma. Prvi model, ki modelira prvi korak, je model določanja stopnje jakosti aglutinacije. Drugi model, ki modelira drugi korak, pa je

model določanja dokončne interpretacije preiskave.

Za gradnjo obeh modelov smo morali izbrati in uporabiti najprimernejši algoritem strojnega učenja. Kljub poznavanju področja izbira primernega algoritma ni bila očitna. Zato smo za gradnjo modelov po predlogu avtorjev literature [2][17] uporabili različne algoritme strojnega učenja in nastale modele primerjali med sabo. Algoritem, ki je bil uporabljen za gradnjo najuspešnejšega modela, je najprimernejši.

2.4.3 Evaluacija posameznih modelov interpretacije

Za primerjavo učinkovitosti modelov interpretacije in metod strojnega učenja, ki smo jih uporabili za njihovo gradnjo, smo potrebovali metodo za njihovo primerjavo. Uporabili smo preizkus modela z navzkrižno validacijo in zapis rezultatov testa modela v matriko pravilnih in napačnih razvrstitev. Iz matrike pravilnih in napačnih rezultatov smo izračunali več skalarnih parametrov, ki govore o učinkovitosti obravnavanega modela. Potrebno je bilo izbrati pravi parameter, na podlagi katerega smo med sabo primerjali učinkovitost posameznih modelov.

2.4.4 Evaluacija kombinacije modelov interpretacije

Ko smo imeli na voljo delujoče modele in smo poznali njihovo uspešnost, smo modele kombinirali med sabo. Za vsako dokončno interpretacijo smo kombinirali rezultate večkratne uporabe posameznih modelov. Ker je potrebno v prvem koraku določiti stopnjo jakosti aglutinacije v šestih kolonah gelske kartice, smo model določanja stopnje jakosti aglutinacije uporabili šestkrat. Določene stopnje jakosti aglutinacije smo kombinirali v vektor, ki je predstavljal vhod v drugi korak interpretacije – model dokončne interpretacije rezultatov. Potrebno je bilo oceniti delež uspešnosti dokončne interpretacije, do katere smo prišli s šestkratno uporabo modela določanja stopnje jakosti aglutinacije v kolonah in uporabo modela za določanje dokončne interpretacije. Ocenili smo jo iz podatkov o uspešnosti posameznih uporabljenih modelov, pridobljenih v postopkih njihove validacije. Na podalgi deleža uspešnosti posamezne kombinacije modelov, uporabljenih za napovedovanje dokončne interpretacije, smo ugotovili najučinkovitejšo kombinacijo algoritmov.

2.4.5 Pridobivanje podatkov za učno in testno množico

Rezultate preiskave z gelsko metodo je mogoče fotodokumentirati s fotografiranjem gelske kartice [18]. Primerno kvalitetna slika gelske kartice vsebuje dovolj informacij, da lahko specialist transfuzijske medicine določi interpretacijo preiskave na osnovi te slike. Ker je slike gelskih kartic mogoče preprosto prenašati, je interpretacija preiskave z gelsko metodo izvedljiva tudi na daljavo. To dejstvo nam je omogočilo izdelavo telekonzultacijskega sistema kategorije shrani in obdelaj (ang. store and forward) [19].

Podatke, ki se zbirajo v sistemu za telekonzultacije, smo uporabili za gradnjo in učne in testne množice, ki smo ju potrebovali za gradnjo modelov interpretacije z algoritmi strojnega učenja. Zakonodaja RS predpisuje varovanje osebnih podatkov. Zato je v praksi delujoč sistem za telekonzultacije zaprt in do njega nimamo neposrednega dostopa. Za potrebe dostopa do podatkov iz sistema za telekonzultacije smo razvili modul, ki zajame potrebne podatke iz sistema in jih shrani v datoteko. Podatki so anonimni in vsebujejo le podatke, potrebne za gradnjo modulov za samodejno interpretacijo predtransfuzijskih preiskav. Datoteko s podatki nam je posredovala oseba, ki je pooblaščena za delo na sistemu za telekonzultacije.

Podatke iz datoteke je bilo potrebno predobdelati, da so bili primerni za obdelavo z algoritmi strojnega učenja. Predobdelava je zajemala razpoznavanje področij kolon na slikah gelskih kartic in segmentacijo slik posameznih kolon z namenom iskanja slikovnih elementov, ki predstavljajo eritrocite. Iz segmentiranih slik smo izračunali porazdelitve aglutinatov po višini kolon. Iz porazdelitev smo izračunali vektorje lastnosti, ki smo jih, zapisane v ustrezno urejene podatkovne nabore, uporabili za gradnjo modela določanja stopnje jakosti aglutinacije. Iz datoteke smo prebrali in v ustrezno obliko zapisali tudi podatke za gradnjo modela določanja dokončne interpretacije preiskave.

Poglavje 3

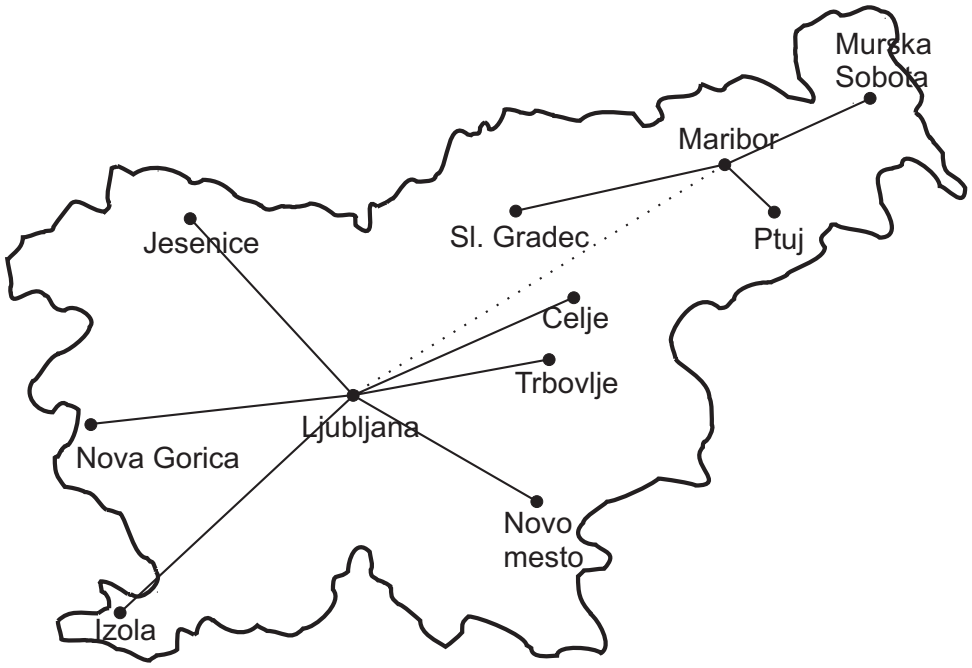
Obstoječe rešitve in orodja za razvoj sistema za samodejno interpretacijo predtransfuzijskih preiskav

V pričujočem poglavju smo opisali obstoječe stanje na področju predtransfuzijskih preiskav z gelsko metodo. Pri tem smo se osredotočili na sistem za telekonzultacije v transfuzijski medicini. Ta predstavlja vir podatkov, ki smo jih potrebovali za izdelavo modelov interpretacije preiskav. S sistemom za samodejno interpretacijo rezultatov bomo nadgradili sistem za telekonzultacije. V nadaljevanju poglavja smo predstavili pregled osnovnih pojmov in elementov metod strojnega učenja, ki so primerne za gradnjo modelov interpretacije predtransfuzijskih preiskav. Predstavili smo osnovne ideje, na podlagi katerih delujejo uporabljeni algoritmi strojnega učenja. Za reševanje našega problema je bil pomemben dostop do podatkov, s katerimi smo gradili učne in testne množice. Dostop do podatkov smo si omogočili z izdelavo sistema za telekonzultacije v transfuzijski medicini.

3.1 Opis obstoječega stanja na transfuziji v RS

3.1.1 Organizacija transfuzijske službe, nadgrajene s sistemom za telekonzultacije

Transfuzijske ustanove na podlagi transfuzijske anamneze bolnika in predtransfuzijskih preiskav, opravljenih iz vzorca krvi bolnika, pripravijo ustrezno komponento krvi [20]. Transfuzijske ustanove v Sloveniji so ZTM v Ljubljani (centralna transfuzijska ustanova),



Slika 3.1: Organizacija transfuzijske službe v Sloveniji [1].

Oddelek za transfuziologijo in imunohematologijo v Mariboru (regijska ustanova) in devet bolnišničnih oddelkov za transfuziologijo [21]. Predvideno je, da bosta centra v Ljubljani in Mariboru skrbela za storitve telekonzultiranja za bolnišnične oddelke. Slika 3.1 prikazuje organizacijo transfuzijske službe v Sloveniji. V centrih v Ljubljani in Mariboru so konzultanti, ki svetujejo konzultirajočim. Uporabniki sistema za telekonzultacije so razdeljeni v dve skupini. V prvi so konzultirajoči, v drugi pa konzultanti. Uporabniki lahko med sabo poljubno komunicirajo [18].

Konzultirajoči so dežurni zdravniki na transfuzijskih oddelkih, laboratorijski tehniki in medicinske sestre. Konzultanti pa so dežurni zdravniki, specialisti transfuzijske medicine [20]. Z vidika sistema za telekonzultacije zajema delo dežurnega zdravnika izvajanje laboratorijskih preiskav, zajem in posredovanje laboratorijskih podatkov, oblikovanje vprašanj za konzultanta in končno opredelitev o izdaji krvi. Delo konzultanta pa zajema nadzor sistema vprašanja/odgovori, sprejem vprašanj in podatkov, strokovno konzultacijo in odgovarjanje dežurnim zdravnikom [18].

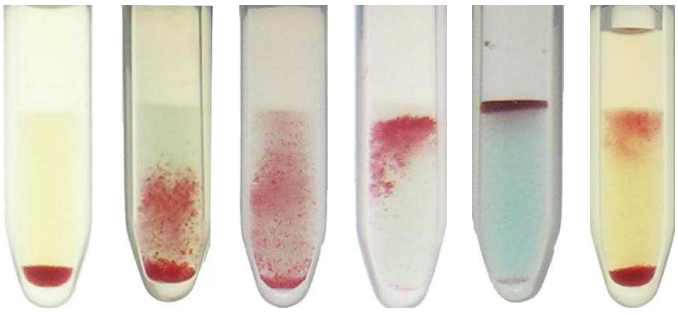
Predtransfuzijske preiskave se izvajajo z uporabo gelske metode.

3.1.2 Opis gelske metode

Predtransfuzijske serološke preiskave so osnovane na zaznavanju reakcij med antigeni na eritrocitih s protitelesi. Te preiskave se običajno izvajajo s t.i. aglutinacijskimi testi v slanih ali makromolekularnih medijih z nemodificiranimi ali z encimi obdelanimi eritrociti ob uporabi določenih potenciatorjev (ang. potentiators) aglutinacije, kot antiglobulinski (Coombsov) serum ali polikationi (ang. polycations)[12]. Pri preiskavah je ključnega pomena natančnost odčitavanja in interpretiranje reakcije preiskave. Natančnost je še posebej pomembna v primerih, ko je reakcija šibka. Za zagotavljanje dobrih interpretacij mora specialist odčitati in interpretirati reakcije kmalu (nekaj ur) po končani reakciji. Gelska metoda preiskave je bila razvita z namenom standardizacije aglutinacije in z namenom fiksiranja aglutinatov, kar omogoča preprostejše odčitavanje. Gelska metoda je občutljiva, a hkrati preprosta za uporabo [13]. Metoda se izvaja z uporabo t.i. gelskih kartic. Gelske kartice so plastične kartice, v katere je vdelanih šest kolon, dolžine 15 mm in premera 2 mm. Slika 2.1 prikazuje prazno gelsko kartico (a) in gelsko kartico z vzorci krvi (b).

Za vsak tip preiskave se uporablja specifičen tip gelske kartice. V grobem se tipi gelskih kartic delijo na nevtralne in specifične kartice. Nevtralne kartice vsebujejo le gel brez reagentov, v specifičnih pa je gelu dodan tudi reagent. V nadaljevanju je opisan postopek z nevtralnimi karticami. Z vidika strojne interpretacije rezultatov preiskav je postopek enak tudi pri uporabi specifičnih kartic.

V zgornjem delu kolon v gelskih karticah se s pipeto zmeša vzorec preiskovanih eritrocitov in diagnostičnih reagentov. Po določenem času med vzorcem in reagentom reakcija poteče, delno poteče ali pa ne poteče. V primeru, da je reakcija potekla, se eritrociti zlepijo v mrežo – aglutinat. Če reakcija ni potekla, ostanejo eritrociti nepovezani. V nadaljevanju postopka sledi ugotavljanje, ali je reakcija v posameznih kolonah potekla in v kolikšni meri je potekla. Mera, ki govori o tem, v kolikšni meri je reakcija potekla, je stopnja jakosti aglutinacije, ki se jo določa za vsako posamezno kolono posebej. Da določimo stopnjo jakosti aglutinacije, je potrebno gelske kartice z vzorci najprej centrifugirati. Specifična gostota gela v kolonah je manjša od specifične gostote eritrocitov, zato med centrifugiranjem prosti eritrociti potonejo skozi gel in se naberejo na dnu kolon (negativen rezultat). Eritrociti, ki so povsem zlepljeni v aglutinat, zaradi velikosti le-tega ne prodrejo skozi gel in ostanejo na vrhu gela (pozitiven rezultat). Delno zlepljene celice pa se glede na velikost aglutinatov porazdelijo po volumnu kolon [12][13].



Slika 3.2: Slike kolon gelskih kartic z različnimi stopnjami jakosti aglutinacije.

Interpretacija rezultatov preiskave je sestavljena iz dveh korakov. Prvi korak je ocenjevanje stopnje jakosti aglutinacije v posameznih kolonah. Drugi korak je končna interpretacija opravljene predtransfuzijske preiskave.

3.1.2.1 Ugotavljanje stopnje jakosti aglutinacije

Specialist transfuzijske medicine oceni stopnjo aglutinacije v vsaki koloni na podlagi porazdelitve eritrocitov po višini kolone in jo razvrsti v enega od šestih razredov. Ti razredi so [12]:

- negativno (NEG, 0)
- 1+
- 2+
- 3+
- 4+
- dvojna celična populacija (DCP)

Rezultat negativno pomeni popolno odsotnost aglutinacije (vse celice so na dnu kolon), 4+ pa najvišjo stopnjo jakosti aglutinacije (vse celice so na vrhu gela v kolonah). Rezultat DCP pomeni, da je del celic na dnu kolone, del celic pa na vrhu gela. Dopuščena je tudi možnost, da v koloni ni vzorca, v tem primeru je kolona označena kot prazno. Primeri slik kolon gelskih kartic z različnimi stopnjami algutinacije so predstavljeni na sliki 3.2.

3.1.2.2 Dokončna interpretacija rezultata preiskave

V nadaljevanju postopka preiskave specialist interpretira nabor določenih stopenj jakosti aglutinacije v posameznih kolonah v končni rezultat preiskave. Za vsako od petih različnih tipičnih preiskav obstaja končni nabor možnih rezultatov preiskave. Za naštete tipe preiskav so možni sledeči rezultati:

- **Določanje krvne skupine AB0 RhD:** 0-NEG, 0-NEG Du+, 0-POZ, A-NEG, A-NEG Du+, A-POZ, B-NEG, B-NEG Du+, B-POZ, AB-NEG, AB-NEG Du+, AB-POZ
- **Indirektni Coombsov test (ICT):** POZ, NEG
- **Direktni Coombsov test (DCT):** POZ, NEG
- **Navzkrižni preizkus (NP):** POZ, NEG
- **Specifikacija protiteles (PT):** anti-D, anti-C, anti-c, anti-E, anti-e, anti-K, anti-k, anti-Fya, anti-Fyb, anti-Jka, anti-Jkb, anti-M, anti-N, anti-Lea, anti-Leb, anti-P1, anti-I, anti-Cw, anti-S, anti-s, anti-i, anti-Lua, anti-Lub, anti-Kpa, anti-Kpb, NI PT

3.2 Sistem za telekonzultacije

3.2.1 Namen sistema – problemi, ki jih rešuje

Sistem za telekonzultacije v transfuzijski medicini smo vzpostavili z namenom zmanjšanja potrebe po velikem številu specialistov transfuzijske medicine in z namenom poenotenja kakovosti storitev transfuzijske medicine v vseh ustanovah v državi. Sistem omogoča povezavo dežurnih zdravnikov in specialistov transfuzijske medicine vseh transfuzijskih ustanov. S tem je omogočena izmenjava podatkov, potrebnih za postavljanje diagnoz. Le-ti so: pacientovi medicinski podatki, podatki, pridobljeni iz predhodnih obravnav, in rezultati preiskav [15][22][23][24][25].

3.2.2 Delovanje sistema za telekonzultacije

Z uporabo sistema za telekonzultacije postavljajo bolnišnični zdravniki (konzultirajoči) vprašanja o strokovnih problemih konzultantom v transfuzijskem centru. Pri dvoumnih

laboratorijskih rezultatih zdravnik uporabi sistem za telekonzultacije in se o njem posvetuje z dežurnim v transfuzijskem centru, konzultantom. Konzultirajoči se na podlagi urgence obravnavanega primera odloči za način telekonzultacije. Če primer ni nujen, konzultirajoči o primeru sestavi vprašanje iz podatkov o pacientu in diagnostičnih podatkov, pridobljenih z gelsko metodo. Sistem doda podatke o morebitnih pacientovih predhodnih rezultatih, ki jih pridobi iz sistema DATEC. Sistem uvrsti vprašanje na konzultantov seznam odprtih zadev. Dežurni konzultant v transfuzijskem centru po vrsti obdela in odgovori na vprašanja s seznama, ki so prispela iz različnih ustanov [15][26].

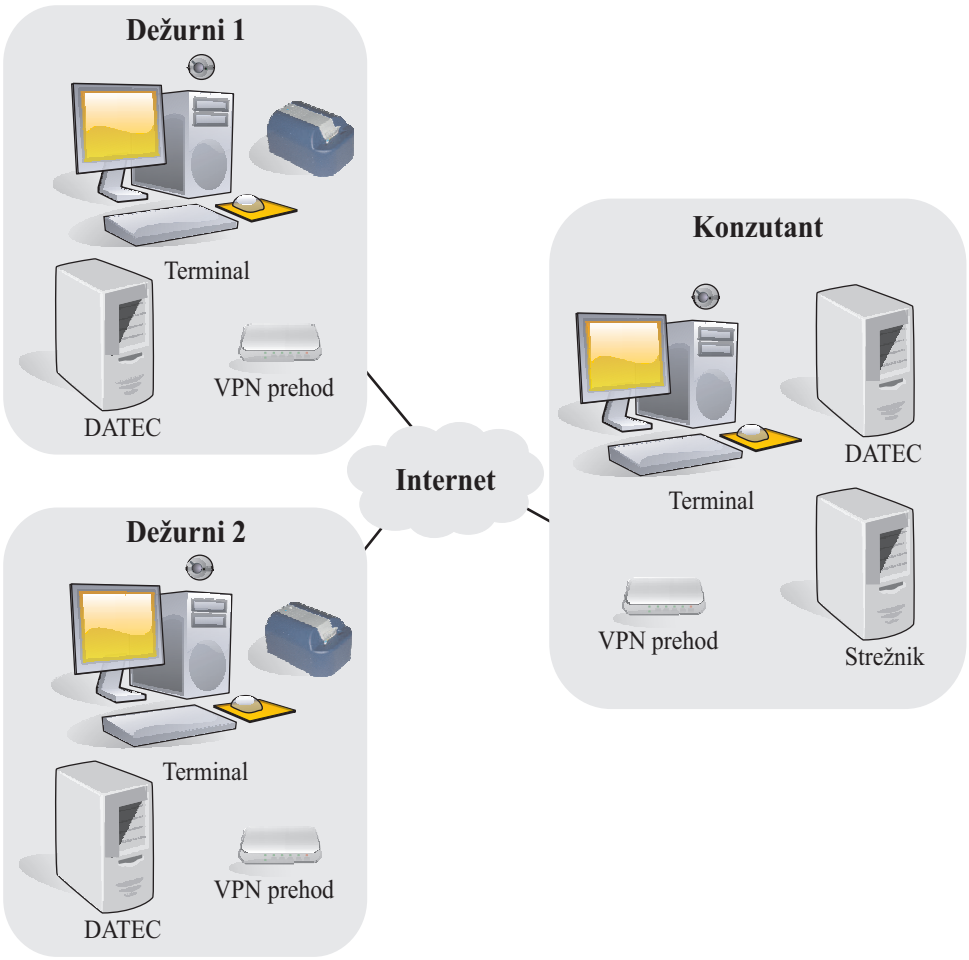
Če je obravnavani primer nujen, zahteva konzultirajoči telekonzultacijo z dežurnim konzultantom v živo. Sistem vzpostavi povezavo s prostim dežurnim konzultantom. Ko konzultant sprejme telekonzultacijsko sejo, se med konzultantom in zdravnikom vzpostavi videokonferenčna zveza, ki omogoča konzultacijo v realnem času. Konzultantu so tudi v tem primeru na voljo vsi podatki, potrebni za postavitev diagnoze. Že obstoječi so zajeti avtomatsko iz obstoječega informacijskega sistema, diagnostične podatke pa vnese konzultirajoči. Dodatna prednost je možnost sprotnega izvajanja vseh potrebnih aktivnosti po navodilih konzultanta [27].

Slika 3.3 prikazuje idejno zasnovo arhitekture sistema za telekonzultacijo. V transfuzijskem centru je dežurni specialist transfuzijske medicine – konzultant, ki odgovarja na vprašanja dežurnih zdravnikov ali ostalih konzultirajočih. V transfuzijskem centru je tudi centralni strežnik sistema. Zdravniki na oddelkih uporabljajo za telekonzultacijo terminale – osebne računalnike z ustrezno dodatno strojno in programsko opremo. Dodatna strojna oprema so naprave za zajem slik gelskih kartic. Slika 3.7 prikazuje napravo za zajem slik gelskih kartic. Poleg omenjenega so vsi terminali opremljeni z opremo, ki omogoča videokonferenčno povezavo. Komunikacija med posameznimi terminali in strežnikom za telekonzultacijo poteka po javnem omrežju internet. Varne povezave med posameznimi elementi sistema so zagotovljene z uporabo prehodov VPN (Virtual Private Network), ki vse elemente povežejo v navidezno zasebno omrežje.

3.2.3 Vzpostavljanje sistema

3.2.3.1 Uporabniške zahteve

V RS zagotavlja storitve transfuzije krvi dve veliki transfuzijski ustanovi in devet bolnišničnih krvnih bank. Ta mreža ustanov zagotavlja krvne produkte in obvezne predtransfuzijske serološke preiskave za vse institucije, ki to potrebujejo. Zagotavljanje teh



Slika 3.3: Zasnova sistema za telekonzultacije s konzultantom in dvema dežurnima lokacijama. Osebje na lokacijah Dežurni 1 in Dežurni 2 s pomočjo sistema zastavlja konzultantu vprašanja, na katera specialist konzultant z uporabo sistema odgovarja.

storitev zahteva nenehno prisotnost vsaj enega specialista s področja imunohematologije v vsaki od ustanov [16]. Z namenom racionalizacije zahtev po obveznih predtransfuzijskih seroloških preiskavah smo identificirali uporabniške zahteve sistema za telekonzultacije v transfuzijski medicini, ki bi izboljšal pogoje, način in učinkovitost dela na tem področju.

Telekonzultacijski sistem naj omogoča:

- zajem slik gelskih kartic visoke ločljivosti, ki jih je mogoče elektronsko prenašati in shranjevati;
- povečevanje in zmanjševanje slike gelske kartice na zaslonu z namenom podrobnega opazovanja aglutinatov v kolonah gelskih kartic;
- 24-urno dostopnost ekspertize referenčnega laboratorija za vse oddaljene bolnišnične oddelke, kjer nudijo storitve transfuzije krvi;
- interakcijo med laboratoriji in referenčnimi laboratoriji v realnem času za urgentne primere;
- shrani in posreduj način telemedicinskega sistema za neurgentne primere [19];
- izmenjavo vseh podatkov o pacientih in krvodajalcih;
- stalno povezavo z nacionalno podatkovno bazo krvodajalcev in pacientov za zagotavljanje tranfuzijskih in anamnestičnih podatkov;
- videokonferenčno avdio/video povezavo med uporabniki sistema;
- popolno sledljivost vseh postopkov, izpeljanih s sistemom;
- zasnovo, ki omogoča razširitve sistema;
- zanesljivo, varno in kodirano izmenjavo podatkov;
- izvedbo v skladu z mednarodnimi standardi.

3.2.4 Gradniki sistema

Na podlagi uporabniških zahtev smo razvili in izdeleli sistem za telekonzultacije. Sistem je razdeljen v več medsebojno povezanih modulov. Zasnova je vidna na sliki 3.3. Osnovni moduli sistema za telekonzultacije v transfuzijski medicini so:

- terminali,
- namenska strojna oprema za zajem slik gelskih kartic – Gelscope32,
- programska aplikacija na odjemalcu,
- programska aplikacija na strežniku,
- programska in strojna oprema za videokonferenčno zvezo,
- podatkovna baza,
- komunikacijski modul z DATEC obstoječim informacijskim sistemom,
- strežnik,
- omrežje z elementi za kriptiranje in varnost.

V nadaljevanju smo podali osnovne opise posameznih modulov, uporabljenih za gradnjo sistema za telekonzultacije.

3.2.4.1 Programska aplikacija na odjemalcu

Aplikacija na odjemalcu je bila razvita v programskem jeziku Java. Slika uporabniškega vmesnika aplikacije je predstavljena na sliki 3.4. Aplikacija skrbi za interakcijo z uporabniki. Na začetku od uporabnika zahteva, da se le-ta indentificira in prijavi v sistem. Aplikacija v nadaljevanju na podlagi tipa prijavljenega uporabnika (konzultant, konzultirajoči, tehnik, administrator) ponudi različen nabor funkcionalnosti. Konzultirajočemu omogoča dodajanje novih sej, pregledovanje sej in vzpostavljanje videokonferenčne povezave.

Dodajanje nove seje poteka tako, da uporabnik v napravo Gelscope32 vstavi gelsko kartico. Aplikacija sliko gelske kartice samodejno zajame in prikaže na zaslonu. Uporabnik jo lahko povečuje in se premika po njej, kar omogoča natančno opazovanje detajlov na sliki gelske kartice. V nadaljevanju postopka uporabnik s čitalcem črtne kode odčita številko vzorca krvi. Aplikacija s prebrano številko črtne kode izvede poizvedbo o pacientovih podatkih v DATEC in jih prikaže na zaslonu. Nato uporabnik izbere tip preiskave in s čitalcem črtne kode odčita črtno kodo gelske kartice. Sledi vnos vprašanja in izbira konzultanta, na katerega bo seja naslovljena. Ko je vnos podatkov zaključen, pošlje uporabnik sejo konzultantu.

Konzultant prejme sejo z vprašanjem. Na voljo mu je slika gelske kartice, ki jo lahko poljubno povečuje in se po njej pomika, da si lahko podrobno ogleda vsebino kolon v gelski kartici. Na voljo so mu tudi podatki, pridobljeni iz sistema DATEC. Po končani analizi konzultant določi stopnjo jakosti aglutinacije v vsaki od kolon in končni rezultat preiskave. Lahko vnese tudi tekst odgovora. Ko konča z vnosom podatkov, sejo zaključi.

Aplikacija opozarja konzultante in konzultirajoče o prispelih sejah s prikazom opozoril na zaslonih in s pošiljanjem SMS sporočil na GSM telefone.

3.2.4.2 Programska aplikacija na strežniku

Strežniška komponenta sistema za telekonzultacije je napisana v programskem jeziku Java. Le-ta teče na aplikacijskem strežniku Tomcat [28] kot Java servlet. Odjemalci z njo komunicirajo s sporočili po protokolu http.

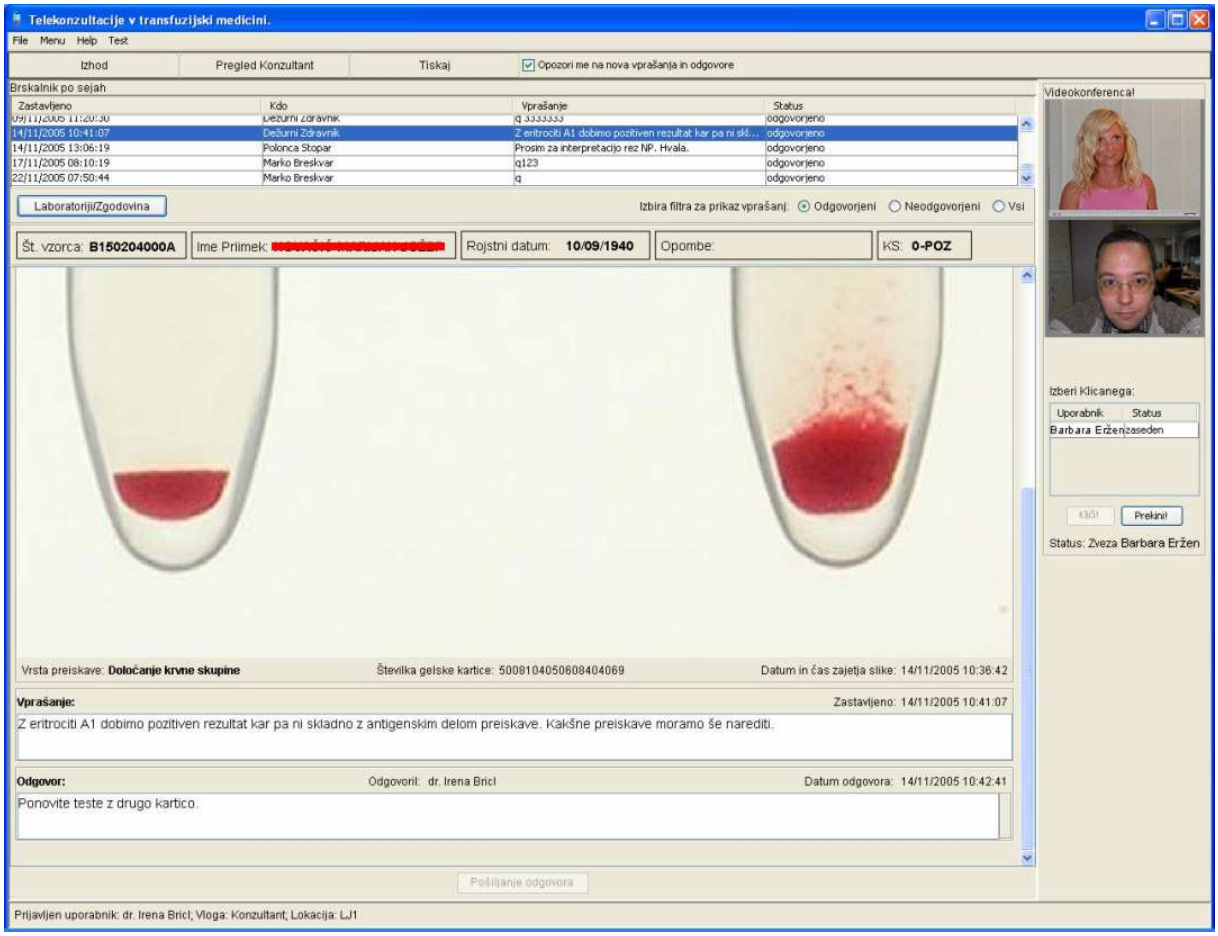
3.2.4.3 Programska in strojna oprema za videokonferenčno zvezo

Modul za videokonferenčno povezavo je v celoti napisan v programskem jeziku Java. Za razvoj je bil uporabljen paket JMF – Java media framework [29]. Videokonferenčna povezava se vzpostavi neposredno med komunicirajočima. Strežnik pri vzpostavitvi sodeluje le s posredovanjem podatkov o komunicirajočih. Vzpostavi se dvosmerna avdio/video povezava po protokolu UDP [30]. Za kodiranje povezave je uporabljen standard H.323 [31]. Terminali so za podporo videokonferenčne povezave opremeljeni s slušalkami z mikrofonom in web kamero. Modul za videokonferenco je integriran v programsko aplikacijo na odjemalcu.

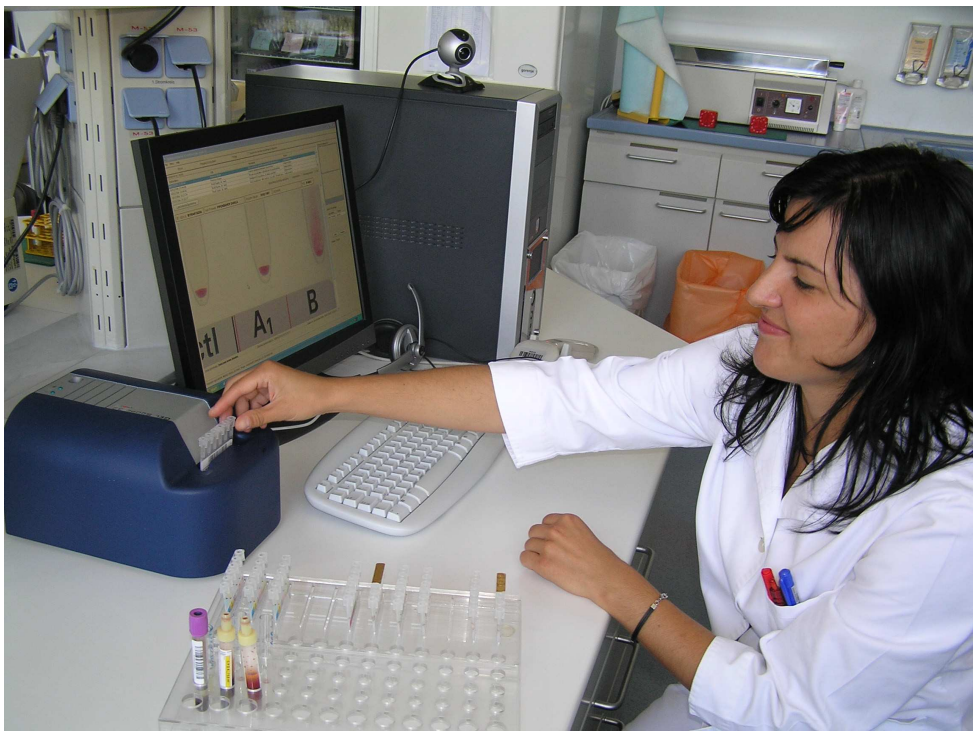
3.2.4.4 Podatkovna baza

Za razvoj sistema za telekonzultacije je bila izbrana odprto-kodna relacijska podatkovna baza MySQL [32]. Podatkovna baza teče na istem strežniku kot strežniška aplikacija. Aplikaciji nudi storitvi hrambe in dostopa do podatkov, ki jih le-ta potrebuje za delovanje. S podatkovno bazo preko vmesnika krmilnikov baze s strežniško aplikacijo komuniciramo z jezikom SQL (Structured Query Language). SQL je najpogostejši standardizirani jezik za opisovanje poizvedb v podatkovnih bazah. Že od leta 1986 ga definira standard ANSI/ISO SQL [32].

Za razvoj in izvedbo strežniške aplikacije je uporabljena tehnologija Java [33], zato so bili za razvoj komunikacije med bazo in strežniško aplikacijo uporabljeni krmilniki JDBC



Slika 3.4: Uporabniški vmesnik aplikacije za telekonzultacije v transfuzijski medicini. Na sliki vidimo glavno okno aplikacije, ki vsebuje 20x povečano sliko kolon na gelski kartici in osnovne pacientove podatke, pridobljene iz sistema DATEC. Vidimo tudi okno z vzpostavljeno videokonferenčno sejo.



Slika 3.5: Delovno mesto konzultirajočega v bolnišničnem laboratoriju. Konzultirajoča je z napravo Gelscope32 ravnokar zajela sliko predtransfuzijskih preiskav na gelski kartici. Slika je takoj vključena v telekonzultacijski sejo skupaj s podatki o pacientu, pridobljenimi iz DATEC. Konzultirajoči lahko kadarkoli med procesom vzpostavi s konzultantom videokonferenčno povezavo in se z njim o problemu posvetuje v realnem času.

(Java Database Connectivity) [34].

V podatkovni bazi so shranjeni avtorizacijski podatki uporabnikov, podatki, potrebni za samo delo sistema, podatki o postavljenih vprašanjih in odgovorih ter podatki o imenih datotek, ki vsebujejo slike gelskih kartic. Posamezne slike gelskih kartic so shranjene vsaka v svoji datoteki z ustreznim imenom. Vsi dogodki v sistemu se beležijo v bazi podatkov. Zapisi so opremljeni s podatkom o času nastanka in osebi, ki je dogodek sprožila. Slovenska zakonodaja predpisuje sledljivost postopka transfuzije in arhiviranje dokumentacije, kar je z opisano rešitvijo zagotovljeno.

3.2.4.5 Strežnik

Na strežnikih teče operacijski sistem Linux s potrebnimi programskimi moduli. Aplikacijski strežnik, ki streže aplikacijo, je web servlet strežnik Tomcat. Na strežniku teče tudi podatkovna baza MySQL. Na strežnik je priključen GSM modul, za pošiljanje SMS sporočil. Strežnik je nameščen na Zavodu za transfuzijsko medicino. Nameščen je v strežniški sobi, ki je primerno varovana in klimatizirana. Napajanje strežnika je izvedeno preko sistema za neprekinjeno napajanje. Za izdelavo varnostnih kopij je na strežnik priključena tračna enota, na katero se vsak dan izvede varnostno kopiranje podatkov s strežnika.

3.2.4.6 Omrežje z elementi za kriptiranje in varnost

Omrežje, po katerem komunicira sistem za telekonzultacije v transfuzijski medicini, je javno IP omrežje internet. Dostop do omrežja zagotavljajo komercialni ponudniki dostopa do omrežja internet preko ADSL. Z namenom doseganja večje zanesljivosti dostop do omrežja internet na vseh lokacijah zagotavljata dva različna ponudnika interneta. Za varno komunikacijo med posameznimi vozlišči omrežja je poskrbljeno z uporabo tehnologije VPN – navideznih virtualnih omrežij [35], ki poskrbi za ustrezno kriptiranje prometa, ki se izmenjuje med vozlišči.

3.2.4.7 Terminali

V sistemu so kot terminali uporabljeni osebni računalniki. Za namene sistema za telekonzultacije v transfuzijski medicini so izbrani primerno zmogljivi osebni računalniki, opremljeni z ustrezno programsko in strojno opremo. Na terminalih teče operacijski sistem Windows XP, z dodatkom programske tehnologije za poganjanje v Javi napisanih

aplikacij JRE – Java run time environment. Pomembno je, da so terminali opremljeni z zmogljivimi monitorji, ki omogočajo primerno kakovosten prikaz slik gelskih kartic. Terminalom je poleg standardne opreme za osebne računalnike dodan čitalec črtne kode, web kamera, slušalke z mikrofonom in naprava za zajem slik gelskih kartic Gelscope32. Terminali so opremljeni tudi s kakovostnim barvnim tiskalnikom.

3.2.4.8 Namenska strojna oprema za zajem slik gelskih kartic – Gelscope32

Ker so za uspešno odčitavanje rezultatov preiskav potrebne kakovostne barvne fotografije gelskih kartic, smo za zajem slik gelskih kartic izdelali namensko strojno in programsko opremo. Da so rezultati posameznih fotografiranj primerljivi, mora biti postopek fotografiranja gelskih kartic normaliziran in ponovljiv. Fotografije morajo biti brez odbleskov ter brez geometrijskih in barvnih popačenj. Fotografirane kartice morajo biti enakomerno osvetljene, goriščna razdalja fotografiranja pa vedno enaka. Za poenostavitev nadaljnje obdelave mora biti pri fotografiranju zagotovljen vedno enak položaj gelske kartice. Uporaba naprave za zajem slik gelskih kartic mora biti preprosta. Slika 3.7 [18] prikazuje "rentgenski pogled" naprave in fotografijo izdelane naprave za zajem slik gelskih kartic – napravo Gelscope32.

Naprava Gelscope32 je namenjena zajemu slik gelskih kartic na uporabniku prijazen način. Gelscope32 je preko vmesnika USB in RS232 priključen na osebni računalnik. Ob vstavitvi gelske kartice naprava samodejno zajame sliko le te. Z uporabo naprave je zagotovljena ponovljivost in medsebojna primerljivost zajetih slik. Ponovljivost zagotavljajo vedno enake razmere osvetljevanja kartice, zajem slik gelskih kartic s fiksno postavljenimi parametri optike ter ostalimi parametri zajema.

Za zajem slik gelskih kartic je v napravo Gelscope32 vgrajena komercialna kamera široke potrošnje. Na ta način smo dosegli sprejemljivo nizko ceno naprave, ki kljub nizki ceni omogoča zajem dovolj kakovostnih fotografij. Za uporabljeno kamero proizvajalec ponuja razvojno okolje, ki omogoča razvoj aplikacije na osebnem računalniku, ki preko USB vmesnika komunicira s kamero v napravi in nadzoruje njene funkcije. Za namen krmiljenja kamere in prenos zajetih slik na terminal – osebni računalnik smo razvili vmesnik, ki komunicira s kamero. Vmesnik smo vključili v aplikacijo za telekonzultacije.

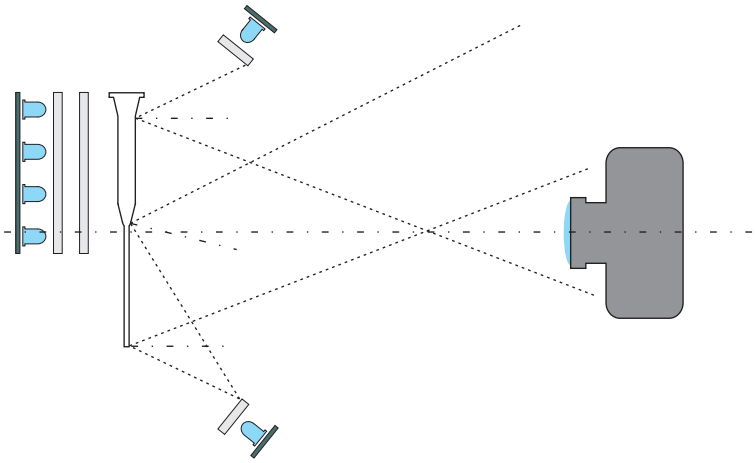
Za osvetljevanje gelske kartice v napravi smo uporabili svetlobna telesa, izdelana iz belih LED, z difuzorji svetlobe. Ker ohišje naprave onemogoča, da bi zunanja svetloba osvetljevala gelsko kartico, so edini vir osvetlitve svetlobna telesa v napravi. Na ta način smo dosegli konstantne osvetlitvene razmere za vse slike. Gelska kartica je iz prozorne

plastike. Sestavljena je iz dveh zanimivih področij, ki morata biti razločno zajeti na slikah. Ti področji sta – področje s kolonami in področje nalepke z napisi. Naprava Gelscope32 je zasnovana tako, da osvetli gelsko kartico iz treh smeri. Za opazovanje vsebine kolon je primerna osvetlitev s presvetlitvijo vsebine. Za opazovanje napisov na nalepkah gelskih kartic pa je potrebna tudi osvetlitev od spredaj. Za presvetlitev gelske kartice smo svetlobno telo postavili za gelsko kartico. Vir svetlobe v tem svetlobnem telesu so bele LED, razporejene v matriko 4 x 6. Neposredno pred to matriko je difuzor, sestavljen iz dveh kosov mlečno belega 3 mm debelega pleksi stekla. Razdalja med kosoma pleksi stekla je 1 cm. S takšno sestavo svetlobnega telesa dobimo enakomerno ploskovno svetlobno telo, ki enakomerno presvetli gelsko kartico in v gelsko kartico vdelane kolone.

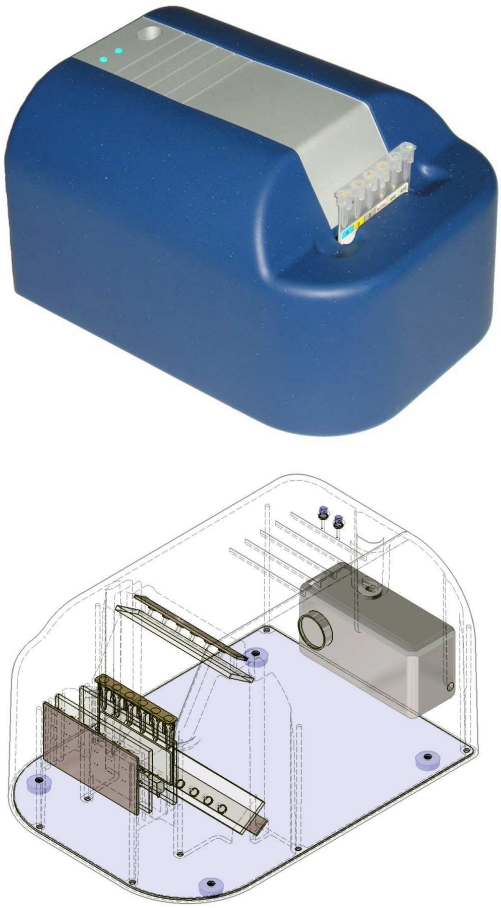
Za osvetlitev od spredaj sta uporabljeni 2 svetlobni telesi. Le-ti sta sestavljeni iz belih LED in difuzorja iz pleksi stekla. Svetlobni telesi morata biti postavljeni tako, da na gladki površini gelske kartice ne tvorita odbleskov, ki bi bili vidni s kamero. Tej zahtevi je ugodeno, če svetlobni telesi osvetljujeta gelsko kartico pod dovolj ostrim kotom, da se njuna slika na gladki površini gelske kartice ne odbije v objektiv kamere. Na sliki 3.6 je narisan shematski prerez naprave Gelscope32 s poudarkom na svetlobnih telesih, gelski kartici in kameri. Na levi strani slike je zadnje svetlobno telo: matrika LED z dvema difuzorjema. Proti desni sledi gelska kartica ter dve sprednji osvetljevali z difuzorjema. Na skrajni desni strani slike je kamera. Sprednji osvetljevali morata biti dovolj odmaknjeni od optične osi objektiva, da se neposredni odboji osvetljeval ne odbijejo od gladkih površin gelske kartice v objektiv. Na sliki 3.6 so narisani koti najneugodnejših odbojev s črtkano črto. Vidimo, da je postavitev gelske kartice, sprednjih osvetljeval in kamere taka, da se odboji ne odbijejo v objektiv kamere.

Celotno napravo krmili mikrokrmilnik družine AVR, ki skrbi za sporočanje statusa naprave osebnemu računalniku preko vmesnika RS232. Skrbi tudi za obveščanje uporabnika o poteku zajema slik gelskih kartic preko, statusnih LED, vgrajenih v ohišje naprave. Dodatna podporna vezja v napravi skrbijo za napajanje vseh sklopov naprave ter prilagoditev napetostnih nivojev pri komunikaciji.

Rezultat zajema slik gelskih kartic z napravo Gelscope32 so barvne digitalne JPEG stisnjene fotografije v velikosti 2048 x 1536 slikovnih elementov. Po ustrezni digitalni obdelavi so zajete fotografije dovolj kakovostne za postavljanje diagnoz.



Slika 3.6: Shematski prerez naprave Gelscope32: Postavitev svetlobnih teles, gelske kartice in kamere. Črtkano so označeni odboji sevetlobe svetilnih teles od površine gelske kartice.



Slika 3.7: Fotografija izdelane naprave Gelscope32 in rentgenski pogled.

3.2.4.9 Komunikacijski modul z DATEC obstoječim informacijskim sistemom

Trenutno je transfuzijska služba v Sloveniji že podprta z informacijskim sistemom DATEC, ki je bil leta 1990 razvit na Zavodu Republike Slovenije za transfuzijsko medicino [36][37]. Informacijski sistem DATEC je kombinacija baze podatkov, ki vsebuje osebne in medicinske podatke pacientov, in tekstovno-grafičnega vmesnika VT100 [38] za dostop do teh podatkov. DATEC teče na samostojnih strežnikih v vseh laboratorijih, v katerih se nudijo storitve transfuzije krvi. Sistem teče na operacijskem sistemu UNIX. Dostop do vmesnika za dostop do podatkov je omogočen preko protokola Telnet.

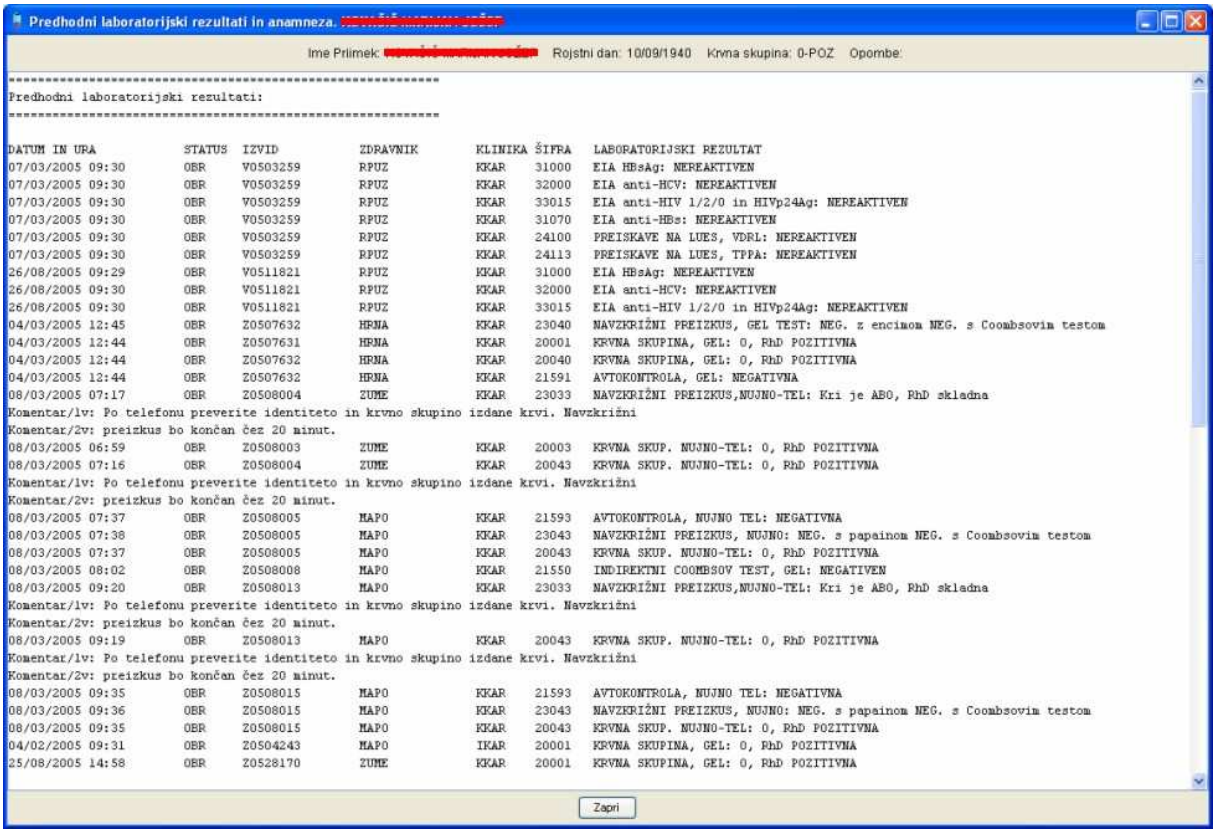
Sistem za telekonzultacije omogoča, da se v telekonzultacijsko sejo na transparenten način vključijo že obstoječi pacientovi matični, medicinski podatki in podatki o zgodovini transfuzijskih posegov iz sistema DATEC. Ti podatki so na voljo konzultantu, ki obdeluje dano sejo. Konzultant potrebuje dostop do teh podatkov, da lahko pravilno ukrepa v mejnih primerih.

Komunikacijski modul je del aplikacije na odjemalcu in ustrezne konfiguracije omrežja, ki omogoča promet med terminali in strežniki sistema DATEC. Modul s sistemom DATEC komunicira preko protokola Telnet in FTP. Komunikacija zajema pošiljanje zahteve po podatkih za posameznega pacienta po protokolu Telnet. Strežnik zahtevo obdela in zahtevane podatke shrani v datoteko. To datoteko modul z uporabo protokola FTP prenese in iz nje prebere podatke o pacientu in njegovi transfuzijski zgodovini. Na sliki 3.8 je vidno okno aplikacije na odjemalcu, ki vsebuje podatke o pacientu, pridobljene iz sistema DATEC.

V pravkar opisanem sistemu za telekonzultacije v transfuzijski medicini se zbirajo podatki, ki jih uporabimo za gradnjo modela interpretacije rezultatov, ki predstavlja osnovo sistema za samodejno interpretacijo rezultatov predtransfuzijskih preiskav. Podatke obdelamo z metodami strojnega učenja, kot je predstavljeno v nadaljevanju.

3.3 Pregled metod strojnega učenja, primernih za gradnjo modelov samodejne interpretacije

Metode strojnega učenja predstavljajo posebno tehniko analize podatkov. Področja, ki vključujejo strojno učenje, so poleg mnogih drugih tudi rudarjenje podatkov, prepoznavanje vzorcev, analiza slik in bioinformatika [39][40].



Slika 3.8: Podatki, pridobljeni iz sistema DATEC, kot so na voljo uporabnikom sistema za telekonzultacije. Zaradi varstva podatkov je na sliki skrito ime obravnavanega pacienta.

Za učinkovito obravnavo algoritmov strojnega učenja smo povzeli definicijo osnovne terminologije, ki opisuje osnovne gradnike metod. Definirali smo pomen *konceptov*, *vzorcev* in *atributov*. Podali smo tudi načine predstavitve naučenega znanja. Obstaja nekaj glavnih podatkovnih struktur, ki jih lahko uporabimo, da predstavimo naučeno znanje. V besedilu smo predstavili *pravilnostne tabele*, *odločitvena drevesa*, *klasifikacijska pravila*, *asociacijska pravila*, in *roje*. Opisu podatkovnih struktur sledi opis štirih glavnih skupin metod strojnega učenja.

3.3.1 Povzetek osnovne terminologoje

V pričujočem podpoglavju smo povzeli pregled osnovne terminologije, ki je potrebna za razpravo o metodah strojnega učenja.

3.3.1.1 Koncept (ang. concept)

Koncepti predstavljajo dele znanja, ki sestavljajo model delovanja danega opazovanega sistema. Z modelom poizkušamo čim bolje posnemati delovanje opazovanega sistema.

Postopki strojnega učenja se v grobem delijo na naslednje pristope: **klasificiranje, asociiranje, rojenje in numerično napovedovanje.** Neodvisno od uporabljenega pristopa strojnega učenja poizkušamo pri vsakem postopku na različne načine izluščiti in spoznati koncepte opazovanega sistema. Rezultati postopkov strojnega učenja so opisi konceptov (ang. concept description) [2]. Skupek opisov konceptov predstavlja model opazovanega sistema.

3.3.1.2 Vzorec (ang. instance, feature, example)

Vzorec je osnovna samostojna entiteta iz nabora podatkov. Vhod v postopek strojnega učenja je nabor vzorcev. Vsak vzorec predstavlja individualen neodvisen primer koncepta, ki se ga želimo naučiti. Vsak, posamezen vzorec je sestavljen iz vrednosti določenih atributov. Vzorec lahko predstavimo kot vektor atributov. V matriko urejeni vektorji atributov – vzorci predstavljajo nabor podatkov [2].

3.3.1.3 Atribut (ang. attribute), značilka, lastnost

Atribut predstavlja element vzorca, ki nosi o njem določeno informacijo [2][39]. Vsak vzorec, ki predstavlja vhod v postopek strojnega učenja je, definiran z vrednostmi na

fiksiranem in preddefiniranem naboru atributov. Vrednosti atributov določenega vzorca so lahko numerične ali nominalne. Nominalni atributi lahko zavzamejo le vrednosti iz končnega nabora vrednosti.

3.3.2 Predstavitev naučenega znanja

Z metodami strojnega učenja je potrebno naučeno znanje pravilno predstaviti za nadaljnjo uporabo. To storimo z ustrezno izbrano podatkovno strukturo, ki predstavlja naučeno znanje. Običajno so podatkovne strukture za predstavitev naučenega znanja implicitno definirane z izbiro metode strojnega učenja. V nadaljevanju so predstavljene različne podatkovne strukture, s katerimi se srečujemo pri strojnem učenju [2]. Te podatkovne strukture so *pravilnostne tabele*, *odločitvena drevesa*, *klasifikacijska pravila*, *asociativna pravila* in *roji*. V literaturi o metodah strojnega učenja avtorji navajajo tudi strukture za numerično napovedovanje, vendar ti postopki pri reševanju naših problemov niso uporabni, zato jih v nadaljevanju teksta ne bomo omenjali.

3.3.2.1 Pravilnostne tabele

Pravilnostne tabele (ang. truth table), včasih poimenovane tudi odločitvene tabele, so najosnovnejši način predstavitve rezultatov metod strojnega učenja. V pravilnostnih tabelah je znanje predstavljeno tako, da so v njih našteti vsi možni vzorci – kombinacije možnih vrednosti posameznih atributov s pripadajočimi rezultati. Vsaki kombinaciji atributov – vzorcu je dodana vrednost tega vzorca (razvrstitev vzorca v razred). Ko poizkušamo ugotoviti vrednost neznanega vzorca, v odločitveni tabeli poiščemo vzorec, katerega kombinacija vrednosti atributov je enaka vrednostim atributov opazovanega vzorca, in odčitamo pripadajočo vrednost [2].

3.3.2.2 Odločitvena drevesa

Odločitvena drevesa (ang. decision tree) so podatkovne strukture v obliki dreves z vozlišči in listi. Listi predstavljajo končno odločitev – razvrstitev vzorca v razred [41]. Odločitvena drevesa so učinkovit način predstavitve naučenega znanja. Primer vizualizacije odločitvenega drevesa je predstavljen na sliki 3.12.

Vozlišča v odločitvenih drevesih predstavljajo razvejišča v strukturi drevesa. Vozlišča predstavljajo testiranje vrednosti določenega atributa vzorca in odločanje o nadaljnji poti po drevesu na podlagi te vrednosti. Opazovanju vrednosti v vozlišču sledi premik po

drevesu do naslednjega vozlišča ali lista. Običajno je vrednost atributa primerjana s konstantno vrednostjo. Pri določenem tipu dreves vozlišča lahko predstavljajo tudi primerjanje vrednosti več kot enega atributa, kot tudi uporabo določene funkcije nad vrednostmi enega ali več atributov opazovanega vzorca. Listi drevesa predstavljajo dokončne klasifikacije vzorcev ali nabor klasifikacij vzorcev, ki so uspeli z vrednostmi atributov po drevesu priplezati do lista. Neznani vzorec se klasificira tako, da se z vrednostmi njegovih atributov spustimo po drevesu. Začnemo v korenskem vozlišču in na podlagi rezultata testiranja v vozlišču predpisanega atributa pot nadaljujemo po veji, ki jo določa rezultat testa. Na podlagi kriterija v posameznem doseženem vozlišču nadaljujemo pot po drevesu. Ko z vzorcem priplezamo do konca drevesa – do lista, mu pripišemo razred – rezultat, ki je pripisan temu listu [17].

Če ima testirani atribut nominalno vrednost, je število vej, ki izhajajo iz tega vozlišča, običajno enako številu možnih vrednosti tega atributa. V tem primeru ta atribut vzorca v nadaljnjih vozliščih ne bo več obravnavan. Včasih je v vozlišču vrednost opazovanega atributa razdeljena v podmnožice, katerih velikost je večja kot ena. Drevo se potem v tem vozlišču deli na toliko vej, kolikor podmnožic vrednosti atributa obstaja. V tem primeru bo vrednost obravnavanega atributa v nadaljnjih vozliščih verjetno še testirana [17].

Če je obravnavani atribut numerična vrednost, potem do odločitve za eno od dveh možnosti vozlišča običajno pride na podlagi tega, če je vrednost atributa večja ali manjša od predhodno definirane konstante. Alternativna možnost je vejenje v tri veje. Možnih načinov za vejenje je več. Če je nabor vrednosti atributa iz množice celih števil, potem se lahko za eno od treh možnosti odločimo na podlagi primerjave *je manjše*, *je večje*, *je enako*. Če so vrednosti atributa iz množice realnih števil, potem je namesto primerjave *je enako* boljša definicija intervala in primerjava vrednosti atributa *je manjše od intervala*, *je znotraj intervala*, *je večje od intervala*. Numerične vrednosti atributov so na poti drevesa od korenin do lista večkrat testirane. Običajno vsakič z drugo konstanto [17].

3.3.2.3 Klasifikacijska pravila

Klasifikacijska pravila (ang. classification rule) so popularna alternativa odločitvenim drevesom. Klasifikacijska pravila so sestavljena iz nabora testov, primerljivih s testi v vozliščih odločitvenih dreves. Vsako pravilo je sestavljeno iz določenega nabora testov. Ko za dani vzorec ugotavljamo, če zanj velja določeno klasifikacijsko pravilo, izvedemo zanj vse teste, ki jih predpisuje to pravilo. Rezultate teh testov združimo z logičnim operatorjem IN. Če so bili vsi testi pravila pravilni, potem je zaključek testiranja, da ta

testirani vzorec spada v razred, definiran z ravnokar uporabljenim pravilom. Klasifikacijska pravila je mogoče preprosto prebrati iz odločitvenih dreves tako, da se sprehodimo do vseh listov po drevesu in zapisujemo posamezne teste v prehojenih vozliščih [2].

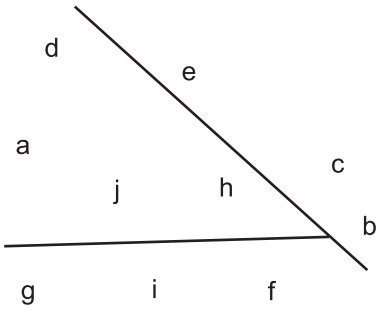
3.3.2.4 Asociacijska pravila

Asociacijska pravila (ang. association rule) se bistveno ne razlikujejo od klasifikacijskih pravil. Lahko pa, za razliko od klasifikacijskih pravil, napovedo tudi atribute vzorcev in ne samo njihovih dokončnih razvrstitev v razrede. Asociacijska pravila opisujejo poljubne povezave med atributi vzorcev in rezultati. Možnih kombinacij asociacijskih pravil je veliko. Različna asociacijska pravila opisujejo različne zakonitosti, ki jim je podvržen obravnavani nabor podatkov in v generalnem napovedujejo različne stvari. Ker je lahko tudi iz zelo malega nabora podatkov izpeljanih mnogo različnih asociacijskih pravil, se je dobro omejiti na tista, ki se nanašajo na razumno velik del obravnavanih vzorcev in imajo razumno visoko stopnjo pravilnosti v napovedih za obravnavane vzorce. Pri obravnavi asociacijskih pravil se ukvarjamo z dvema kazalnikoma njihove uporabnosti – *pokritjem* in *natančnostjo*. Pokritje asociacijskega pravila je število vzorcev, ki jih asociacijsko pravilo napoveduje pravilno. Natančnost tega pokritja pa je izražena kot delež vseh pravilno napovedanih vzorcev med vsemi, na katere se določeno pravilo nanaša.

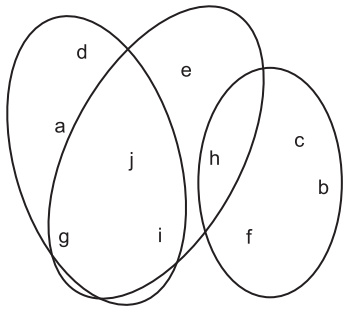
3.3.2.5 Roji

V primerih, ko so za namene strojnega učenja uporabljeni algoritmi rojenja (ang. clustering) se sistem uči rojev namesto klasifikacije. Roji so podatkovna struktura v obliki diagrama, ki opisuje naučeno znanje na način, kako posamezni vzorci pripadajo določenim rojem. V najpreprostejšem primeru predstavitve to predstavlja slikanje vzorcev na 2-dimenzionalno ravnino in delitev te ravnine na področja, ki so jim pripisane oznake posameznih rojev. Za primer glejte sliko 3.9 a. Kompleksnejša je razdelitev N-dimenzionalnega prostora v področja, katerim pripadejo vzorci. V pričujočem delu je znanje v primeru uporabe algoritmov rojenja predstavljeno na ta način. Obstajajo tudi druge metode predstavitve naučenega. Določeni algoritmi rojenja dopuščajo možnost, da vzorci pripadajo več kot enemu roju. V tem primeru je znanje podano z medsebojno se pokrivajočimi področji v prostoru. Primer je predstavljen na sliki 3.9 b. Določeni algoritmi rojenja pripišejo posameznim vzorcem verjetnosti, da le-ti spadajo v določen razred. Te vrednosti so za vsak vzorec podane tabelarično. Primer je predstavljen na sliki 3.9 c. V skupino algorit-

mov rojenja spadajo tudi algoritmi, katerih izvajanje da hierarhično podatkovno strukturo razredov. Ta je zgrajena tako, da v vsaki globini opazovanja razdeli prostor vzorcev na podprostore. Ta metoda predstavitve je podana na sliki 3.9 d. Posamezni vzorci so v najglobljem nivoju povsem razdrobljeni, ko pa se pomikamo višje po strukturi, so posamezni vzorci in razredi združeni skupaj.



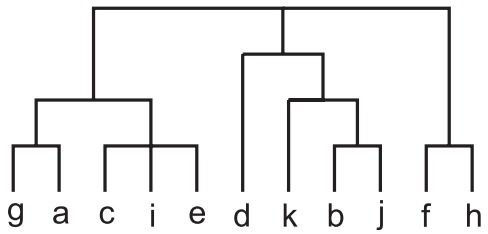
(a)



(b)

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

(c)



(d)

Slika 3.9: Različni načini predstavitve rojev – znanja, naučenega z metodami rojenja [2].

3.3.3 Pristopi strojnega učenja

V literaturi zasledimo štiri osnovne pristope k strojnemu učenju. Avtorji [2] svetujejo, da se za postopke strojnega učenja najprej preizkusi najpreprostejše algoritme posameznih pristopov, ker z uporabo le-teh običajno dobimo presenetljivo dobre rezultate. Teh rezultatov z uporabo veliko kompleksnejših algoritmov ne izboljšamo bistveno [2]. Posamezni pristopi strojnega učenja za določen nabor podatkov delujejo različno dobro. Običajno je nemogoče napovedati, kateri pristop bo za dani nabor podatkov najučinkovitejši. Zato

je najučinkovitejši pristop za gradnjo učinkovitega modela, ki je pogojen z izbiro za dani problem *optimalnega* algoritma strojnega učenja, povezan z empiričnim izbiranjem najbolje delujočega algoritma iz danega nabora metod strojnega učenja. V osnovi se pristopi strojnega učenja delijo na naslednje skupine:

1. Klasifikacija (ang. classification): Z uporabo pristopov iz te skupine metod strojnega učenja se nauči konceptov razvrščanja vzorcev v predifinirane razrede.

2. Asociiranje (ang. association): Z uporabo pristopov iz te skupine metod strojnega učenja iščemo vse povezave med atributi vzorcev. Pri tem niso izvzete povezave, ki niso neposredno potrebne za razvrščanje.

3. Rojenje (ang. clustering): Pri teh metodah poizkušamo vzorce združevati v skupine na podlagi določenih podobnosti in razlik teh vzorcev.

4. Numerično napovedovanje (ang. numeric prediction): Model, generiran s temi metodami, poizkuša napovedati numerično vrednost in ne razreda, kateremu pripada obravnavani vzorec.

V nadaljevanju so predstavljene posamezne skupine pristopov strojnega učenja. Pristopom so dodani primeri algoritmov strojnega učenja, razloženi na vzorčnem naboru podatkov.

3.3.3.1 Klasifikacija

Pri uporabi metod strojnega učenja iz te skupine predpostavimo, da imamo na voljo preddefiniran nabor razredov, v katere moramo razvrstiti naše neznane vzorce. Metode so primerne za oba koraka našega problema. Primerne so za določanje stopnje jakosti aglutinacije v kolonah na način, da posamezne kolone gelskih kartic razvrščajo v razrede, opisane s stopnjo jakosti aglutinacije. Vsi možni razredi kolon glede na stopnjo jakosti aglutinacije eritrocitov v njih so namreč znani vnaprej. Prav tako so metode iz te skupine primerne za razvrščanje nabora razvrščenih kolon v dokončno interpretacijo preiskave. Za vsako skupino testov namreč obstaja končen in vnaprej znan nabor interpretacij.

Delovanje metod bomo ilustrirali na demonstracijskem podatkovnem naboru, ki je predstavljen v tabeli 3.1.

A_1	A_2	A_3	A_4	*Rezultat*
1	1	1	2	B
1	1	1	1	B
2	1	1	2	A
3	2	1	2	A
3	3	2	2	A
3	3	2	1	B
2	3	2	1	A
1	2	1	2	B
1	3	2	2	A
3	2	2	2	A
1	2	2	1	A
2	2	1	1	A
2	1	2	2	A
3	2	1	1	B

Tabela 3.1: Demonstracijski podatkovni nabor. Povzet in popravljen iz [2].

3.3.3.1.1 1R Najpreprostejša metoda za ugotavljanje klasifikacijskega pravila iz nabora vzorcev je imenovana 1R. Metoda generira enostopenjsko odločitveno drevo, ki je izraženo v obliki nabora pravil, ki testirajo le en atribut vzorca. Metoda 1R je preprosta in hitra, ki pogosto daje presenetljivo dobre rezultate. Osnovna ideja metode je sledeča: zgradimo pravila, ki testirajo samo en atribut vzorcev in razvrščajo vzorce v razrede samo na podlagi rezultata tega testa. Vsaka razvejitev pripade različni vrednosti atributa.

Algoritem 1R izbere pravila za razvrščanje na sledeči način: za razvrščanje uporabi razred, ki se v učnem naboru podatkov pojavi največkrat. Ko so izbrana pravila, je potrebno za vsako pravilo ugotoviti, kako dobro deluje za vsako pravilo izračuna delež napačnih (ang. error rate). Izvajanje nad vsakim atributom generira različen nabor pravil, in sicer po eno pravilo za vsako možno vrednost atributa. V nadaljevanju izvajanja algoritem med sabo primerja dobljene deleže napačnih za vsak nabor pravil za vsak atribut in izbere najboljšega. V tabeli 3.2 je ilustriran postopek izbire atributa, primernega za klasifikacijo. Vidimo, da doseže metoda najboljše rezultate takrat, ko se odloči za razvrščanje na podlagi atributov A_1 ali A_3.

3.3.3.1.2 Statistično modeliranje – Naivni Bayes Metoda 1R za odločanje o razvrščanju uporabi le en atribut vzorca. Izbere tistega, s katerim deluje razvrščanje najbolje. Druga, tudi preprosta metoda za izračun odločitve, v kateri razred spada obravnavani vzorec za razvrščanje, uporabi vse atribute vzorca. V tabeli 3.3 je prikazano, kolikokrat se v naboru podatkov, podanih v tabeli 3.1, za vsako vrednost posameznega

	Atribut	Pravila	Napake	Skupaj napake
1	A_1	$1 \rightarrow B$	2/5	4/14
		$2 \rightarrow A$	0/4	
		$3 \rightarrow A$	2/5	
2	A_2	$1 \rightarrow B$	2/4	5/14
		$2 \rightarrow A$	2/6	
		$3 \rightarrow A$	1/4	
3	A_3	$1 \rightarrow B$	3/7	4/14
		$2 \rightarrow A$	1/7	
3	A_4	$2 \rightarrow A$	2/8	5/14
		$1 \rightarrow B$	3/6	

Tabela 3.2: Primer generiranja pravil 1R iz podatkov učne množice, podane v tabeli 3.1. Povzeto in popravljeno iz [2].

atributa $(A_1, ..A_4)$ pojavi posamezna vrednost rezultata. Iz zgornje polovice tabele lahko preberemo, da se rezultat A pri vrednosti atributa $A_1 = 1$ pojavi dvakrat, rezultat B pa se pri vrednosti atributa $A_1 = 1$ pojavi trikrat. Iz spodnje polovice tabele pa lahko preberemo delež pojavljanja posameznega rezultata v vseh vzorcih pri dani vrednosti opazovanega atributa. V vseh vzorcih se rezultat A pojavi 9-krat, rezultat B pa 5-krat.

A_1			A_2			A_3			A_4			*Rez*	
	A	B		A	B		A	B		A	B	A	B
1	2	3	1	2	2	1	3	4	2	6	2	9	5
2	4	0	2	4	2	2	6	1	1	3	3		
3	3	2	3	3	1								
1	2/9	3/5	1	2/9	2/5	1	3/9	4/5	2	6/9	2/5	9/14	5/14
2	4/9	0/5	2	4/9	2/5	2	6/9	1/5	1	3/9	3/5		
3	3/9	2/5	3	3/9	1/5								

Tabela 3.3: Primer generiranja pravil za statistično modeliranje iz podatkov učne množice, podane v tabeli 3.1. Povzeto in popravljeno iz [2].

A_1	A_2	A_3	A_4	Rezultat
1	3	1	1	?

Tabela 3.4: Neznani vzorec. Povzeto in popravljeno iz [2].

Za neznani vzorec, podan v tabeli 3.4, izračuna metoda verjetnost za rezultat A, kot je zapisano v enačbah 3.1. Najprej izračuna verjetnosti za pojav vsakega od možnih rezultatov. Za vsak rezultat iz tabele 3.3 prebere delež pojavljanja rezultata pri dani vrednosti opazovanega atributa. Za neznani vzorec iz tabele 3.4 v primeru atributa $A_1 = 1$ metoda ugotovi, da se rezultat A pojavi v 2/9 primerov. Vse deleže množi med sabo in

dobi verjetnost za rezultat A. To stori tudi za ostale možne rezultate. Neznanemu vzorcu pripiše rezultat, ki ima največjo verjetnost. V našem primeru je to rezultat B. Dobljene verjetnosti normalizira, tako da vsota vseh znaša 1, kot je ilustrirano v 3. in 4. vrstici enačbe 3.1. Vidimo, da je verjetnost, da je pravi rezultat rezultat B, skoraj štirikrat tolikšna, kot verjetnost, da je pravi rezultat A.

$$\begin{aligned} P(A) &= 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053 \\ P(B) &= 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206 \\ Pr(A) &= \frac{0.0053}{0.0053 + 0.0206} = 20.5\% \\ Pr(B) &= \frac{0.0206}{0.0053 + 0.0206} = 79.5\% \end{aligned} \tag{3.1}$$

Ta preprosta in intuitivna metoda je osnovana na Bayesovem izreku pogojne verjetnosti [42]. Verjetnosti dogodka H ob pogoju, da se je dogodek E zgodil, pravimo pogojna verjetnost dogodka H ob pogoju E in jo označimo s $P(H/E)$. Verjetnost, da sta se hkrati zgodila dogodka A in B, označimo s $P(EH)$. Pogojno verjetnost dogodka H ob pogoju E izračunamo, kot je zapisano v enačbi 3.2.

$$P(H/E) = \frac{P(EH)}{P(E)}, P(E) \neq 0. \tag{3.2}$$

Bayesov izrek trdi:

$$P(E/H) = \frac{P(E/H)P(H)}{P(E)}. \tag{3.3}$$

Če upoštevamo zapis v enačbi, rečemo, da s H označen dogodek da je pravilen rezultat A. Dogodek E pa je posebna kombinacija vrednosti atributov, kot je podana v tabeli 3.4. Dogodke, da se v vzorcu pojavi posamezna vrednost posameznega objekta, označimo z E_1, E_2, E_3, E_4. Pogojne verjetnosti, da se zgodi dogodek E_x ob pogoju H, pa $P(E_x/H)$. Opazujmo pogoj $H = A$. Ker predvidevamo, da so posamezni atributi vzorca med sabo neodvisni, je njihova kombinirana verjetnost pridobljena s produktom pogojnih verjetnosti posameznih atributov za opazovani rezultat. V enačbi 3.4 je zapisana splošna enačba za $P(A/E)$ in enačba, v kateri so vrednosti $P(E_x/A)$ nadomeščene z vrednostmi, ugotovljenimi iz podatkov tabele 3.1 ter podani v tabeli 3.3.

$$P(A/E) = \frac{P(E_1/A)P(E_2/A)P(E_3/A)P(E_4/A)P(A)}{P(E)}$$
$$P(A/E) = \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{P(E)} \tag{3.4}$$

Ko bomo izračunali še vrednosti za ostale možne rezultate in jih normirali, tako da bo vsota vseh verjetnosti ena, bodo imenovalci izginili.

Opisana metoda se imenuje Naivni Bayes (ang. Naive Bayes), ker je osnovana na Bayesovem pravilu in naivno privzema medsebojno neodvisnost posameznih atributov vzorca. Naivni Bayes običajno deluje zelo dobro, še posebej v primerih, ko so za sestavo vzorcev izbrani atributi, ki nosijo veliko informacije in posledično doprinašajo k veliki ločljivosti posameznih vzorcev.

Problem z zgoraj opisano metodo se pojavi, če v učni množici podatkov manjka določena vrednost določenega atributa. Če se pojavi vzorec s to v učni množici manjkajočo vrednostjo, bo člen, ki opisuje verjetnost tega dogodka, 0. Ker so s to vrednostjo množene vse ostale vrednosti, bo na koncu verjetnost za ta dogodek enaka 0. In to ne glede na to, da so lahko ostale vrednosti zelo velike. Problem demonstriramo v predstavljenem vzorčnem problemu tako, da opazujemo rezultat B pri vrednosti atributa $A_1 = 2$. Glejte tabelo 3.3. V tem primeru bo verjetnost vedno 0. Dotično slabost metode se odpravi na preprost način. Najpreprostejša rešitev je prištevanje vrednosti 1 številu pojavov določenega rezultata za določeno vrednost določenega atributa. Ta popravek povzroči, da tudi v primeru, ko se v učni množici določena vrednost določenega atributa ni nikoli pojavila, se tej vrednosti atributa za določen rezultat pripiše mala, od nič različna verjetnost. V našem primeru bi bile tako upoštevane vrednosti v zgornjem delu tabele 3.3 atributa A_1 za rezultat B $(4, 1, 3)$, namesto $(3, 0, 2)$. Pripadajoče preračunane verjetnosti v spodnjem delu tabele pa $(4/8, 1/8, 3/8)$, namesto $(3/5, 0/5, 2/5)$.

Tehnika prištevanja 1 vsakemu štetju je standarden prijem in se v literaturi pojavlja pod imenom *Laplaceov estimator*. Po navedbah v literaturi [2] deluje izredno dobro. Ne obstaja pa noben poseben razlog, da bi rezultatom prištevali natančno 1. Namesto 1 lahko prištejemo poljubno malo konstanto μ. V našem primeru bi izračun posameznih verjetnosti izgledal takole:

$$\frac{3 + \mu/3}{5 + \mu}, \frac{0 + \mu/3}{5 + \mu}, \frac{2 + \mu/3}{5 + \mu}. \tag{3.5}$$

Z velikostjo μ definiramo pomembnost začetnih vrednosti posameznih možnih vrednosti atributov pri izračunu verjetnosti. V našem primeru je število različnih vrednosti

atributa A_1 3. Ko smo prišteli številu preštetih pojavov vrednost 1, smo izbrali vrednost $\mu = 1/3$. Torej je pomembnost začetnih vrednosti vsake možne vrednosti sledeča: $A_1 = 1, p_i = 1/3; A_1 = 2, p_i = 1/3; in A_1 = 3, p_i = 1/3$. Če je izbran velik μ, je pomembnost velika in v učno množico dodani vzorci počasi spreminjajo znanje sistema, če pa je vrednost μ mala, pa je predhodna pomembnost mala. Prav tako ni nujno, da razdelimo μ na enake dele. Lahko uporabimo sledečo formulacijo:

$$\frac{3 + \mu p_1}{5 + \mu}, \frac{0 + \mu p_2}{5 + \mu}, \frac{2 + \mu p_3}{5 + \mu}, \tag{3.6}$$

pri čemer velja

$$p_1 + p_2 + p_3 = 1. \tag{3.7}$$

3.3.3.2 Klasifikacija: Gradnja odločitvenih dreves

Metode tega tipa so v literaturi navedene kot *topdown* metode [17]. Te metode začno obdelavo s celo množico podatkov in jo postopno delijo, dokler delitev ni tako fina, da so rezultati deljenja posamezni razredi.

Problem gradnje odločitvenih dreves je mogoče predstaviti rekurzivno. Najprej je potrebno izbrati enega od atributov in z njim začeti testiranje v *korenskem vozlišču* (ang. root node). Iz korenskega vozlišča so izpeljane veje za vsak možen rezultat testiranja v tem vozlišču. Testiranje v vozlišču razdeli množico vzorcev v podmnožice – za vsak rezultat testa z opazovanim atributom eno. V nadaljevanju se ta postopek v vsaki veji rekurzivno ponavlja. Obravnava se samo tiste vzorce, ki z vrednostmi priplezajo do trenutno opazovanega vozlišča. V trenutku, ko je vsem vzorcem, ki priplezajo do opazovanega vozlišča, pripisan isti rezultat, se postopek razvijanja tega dela drevesa ustavi. Potrebno se je odločiti za način, kako za določen nabor podatkov v določenem koraku rekurzivnega postopka izbrati atribut, ki bo testiran v določenem vozlišču in uporabljen za nadaljnje vejenje drevesa.

3.3.3.2.1 ID3 in izpeljanke V nadaljevanju je opisan postopek, ki ga za gradnjo dreves uporablja algoritem ID3 in njegove izpeljanke [2]. Po ogledu demonstracijske učne množice, podane v tabeli 3.1, ugotovimo, da lahko gradnjo drevesa pričnemo na štiri različne načine. Na voljo imamo namreč štiri atribute. Možne delitve po prvi iteraciji so predstavljene na sliki 3.10 a – d. Potrebno se je odločiti, katera možnost je najboljša.

Za vsako vejo je prikazano število vzorcev s pripadajočimi rezultati a in b. Vsaka veja, do katere priplezajo vzorci z istimi pripadajočimi rezultati (vsi a ali vsi b), predstavlja končni list drevesa. Od tu dalje drevesa ni potrebno več vejiti. Z različnim zaporedjem izbiranja atributov za posamezna vozlišča vplivamo na potrebno število vozlišč drevesa za doseganje končnih listov drevesa. Za opis našega sistema želimo dobiti čim manjša drevesa, zato želimo, da po drevesu z vzorci hitro priplezamo do končnih listov. Da lahko dosežemo ta kriterij, je potrebno definirati mero čistosti vsakega vozlišča. V nadaljevanju na podlagi te mere izbiramo atribute, ki dajo najbolj čista hčerinska vozlišča. Mera čistosti (ang. measure of purity), ki jo predlaga avtor literature [2] je informacija in se meri v bitih. V povezavi z vozliščem drevesa predstavlja pričakovano količino informacije, ki bi bila potrebna, da se vzorec, ki je prišel do tega vozlišča klasificira v rezultat a ali b. Izračunamo jo na osnovi števila rezultatov a in b v vozlišču. Postopek izračuna informacije je podan v podpoglavju 3.3.3.2.1 – Izračun informacije.

Če si kot primer ogledamo drevo na sliki 3.10 a, ugotovimo, da je število rezultatov a in b naslednje: $[2, 3], [4, 0], [3, 2]$. Informacija teh vozlišč je sledeča:

$$\begin{aligned} I([2,3]) &= 0,971\ bit \\ I([4,0]) &= 0\ bit \\ I([3,2]) &= 0,971\ bit \end{aligned} \tag{3.8}$$

Lahko izračunamo povprečno informacijo teh vrednosti, pri čemer upoštevamo število vzorcev, ki dosežejo vsako vejo – prvo in tretjo vejo doseže pet vzorcev, drugo štirje vzorci:

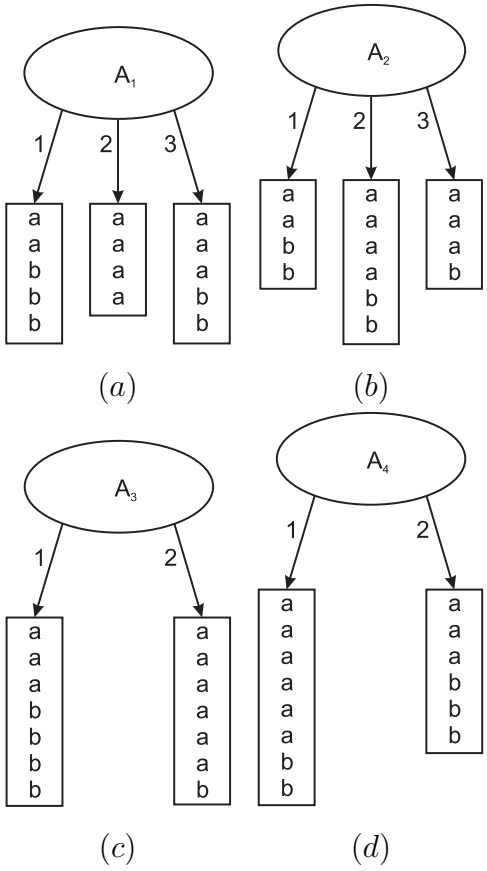
$$I([2,3],[4,0],[3,2]) = (5/14) \times 0.971 + (4/14) \times 0 + (5/14) \times 0.971 = 0.693\ bit. \tag{3.9}$$

Ta številka predstavlja pričakovano količino informacije, ki je potrebna, da se vzorec razvrsti v razred, če je uporabljena drevesna struktura, podana v sliki 3.10 a.

Preden smo generirali drevesne strukture, predstavljene na sliki 3.10, smo imeli na voljo učno množico, ki je bila sestavljena iz 9 rezultatov a in 5 rezultatov b. Informacija te odločitve je:

$$I([5,9]) = 0,940\ bit. \tag{3.10}$$

Zato je drevo na sliki 3.10 a odgovorno za doprinos informacije (ang. information gain) G:



Slika 3.10: Štori dreves za posamezne atribute $A_1..A_4$ za podatke iz tabele 3.1. Povzeto in popravljeno iz [2].

$$G(A_1) = I([9,5]) - I([2,3],[4,0],[3,2]) = 0,940 - 0,693 = 0,247\ bit. \tag{3.11}$$

To lahko interpretiramo kot informacijsko vrednost, če se odločimo in v vozlišču vejimo po atributu A_1.

V nadaljevanju izračunamo doprinos informacije za preostale atribute in se odločimo, da drevo razvejimo po tistem, ki ima največji G:

$$\begin{aligned} G(A_1) &= 0,247\ bit, \\ G(A_2) &= 0,029\ bit, \\ G(A_3) &= 0,152\ bit, \\ G(A_4) &= 0,048\ bit. \end{aligned}$$

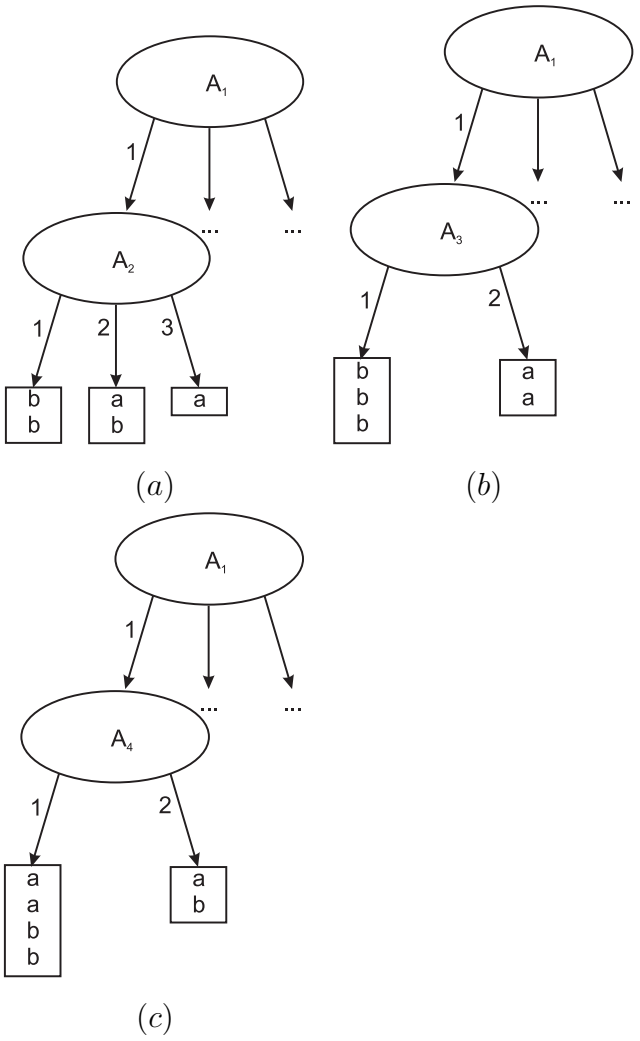
Ugotovimo, da je pri izbiri vejenja po atributu A_1 vrednost največja. Druga najboljša izbira je vejenje po atributu A_3. Ko se odločimo za atribut, s postopkom rekurzivno nadaljujemo. Na sliki 3.11 a – c so prikazane možnosti vejenja za atribut $A_1 = 1$. Ker smo atribut A_1 že porabili, nam ostanejo le še trije, ki jih lahko uporabimo za vejenje. Doprinos informacije za te atribute je sledeč:

$$\begin{aligned} G(A_2) &= 0,571\ bit, \\ G(A_3) &= 0,971\ bit, \\ G(A_4) &= 0,020\ bit. \end{aligned}$$

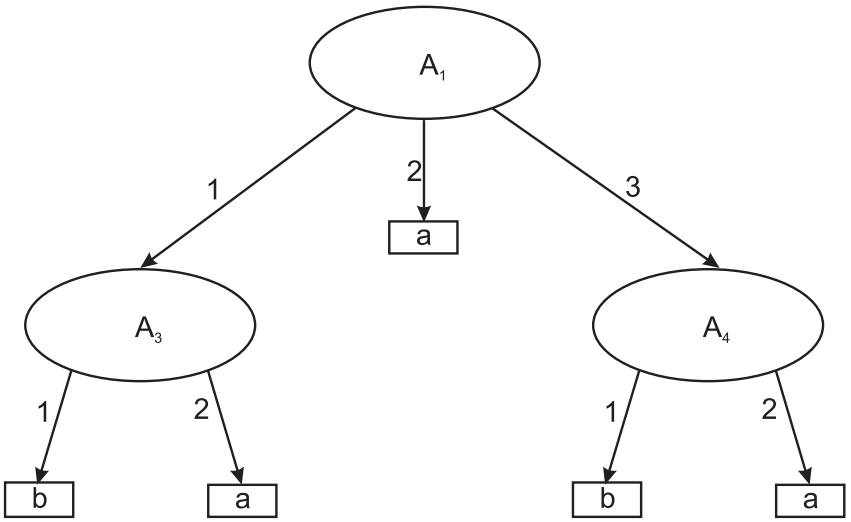
Zato za vozlišče izberemo vejenje po atributu A_3. S tem smo v tej veji dosegli končni list in s tem je ta veja drevesa zaključena. Z uporabo pravkar opisanega postopka za ostale vrednosti in atribute pridemo do odločitvenega drevesa, predstavljenega na sliki 3.12.

Izračun informacije Informacija je mera čistosti posameznih dreves [2]. Zahteve za to mero so sledeče:

- Ko je število rezultatov a ali b enako nič, je velikost informacije 0.
- Ko je število rezultatov a ali b enako, je velikost informacije največja.



Slika 3.11: Drugi korak v gradnji drevesa za demonstracijski problem. Za osnovno vejo je bil izbran atribut A_1. Podane so možne vejitve za posamezne atribute $A_2..A_4$ za podatke iz tabele 3.1. Povzeto in popravljeno iz [2].



Slika 3.12: Odločitveno drevo za razvrščanje podatkov, podanih v tabeli 3.1. Povzeto in popravljeno iz [2].

- Mera za informacijo odločitve mora upoštevati možnost, da določeno odločitev naredimo v enem ali več korakih. V obeh primerih mora biti vrednost velikosti informacije enaka.

Mera mora biti uporabna tudi v primerih, ko je število možnih rezultatov – razredov večje od 2. Mera se nanaša na količino informacije, ki je pridobljena z opravljeno odločitvijo. Odločitve so lahko storjene v enem koraku ali pa v več korakih. Količina informacije, vključene v odločitve, pa je v obeh primerih enaka. Tako je lahko odločitev, povezana z izračunom informacije

$$I([2,3,4]), \tag{3.12}$$

narejena v dveh korakih. Najprej se odločimo, ali je ta odločitev prvi primer ali eden od preostalih dveh primerov:

$$I([2,7]). \tag{3.13}$$

V nadaljevanju izračunamo mero informacije za ostali dve odločitvi:

$$I([3,4]) \tag{3.14}$$

V določenih primerih druga odločitev ne bo potrebna – to je v primerih, ko se izkaže, da je bila storjena prva odločitev. Če to upoštevamo, sledi

$$I([2,3,4]) = I([2,7]) + (7/9) \times I([3,4]). \tag{3.15}$$

Mera za informacijo je informacijska entropija H[43]. Podana je s sledečo enačbo:

$$H(p_1, p_2, \ldots, p_n) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \ldots - p_n \log_2 p_n \tag{3.16}$$

Ker je uporabljen logaritem z osnovo 2, je enota *bit*. Argumenti $p_1, \ldots p_n$ v enačbi 3.16 so normirani, da njihova vsota znaša ena. Primer:

$$I([2,3,4]) = H(2/9, 3/9, 4/9) \tag{3.17}$$

Večstopenjske odločitve lahko v splošnem zapišemo kot:

$$H(p,q,r) = H(p, q+r) + (q+r) \cdot H\left(\frac{q}{q+r}, \frac{r}{q+r}\right) \tag{3.18}$$

pri čemer velja:

$$p + q + r = 1 \tag{3.19}$$

3.3.3.3 Klasifikacija: Konstruriranje pravil z algoritmi s pokrivanjem

Ravnokar opisani algoritmi za generiranje dreves so osnovani na ideji deli in vladaj. Delujejo od zgoraj navzdol – na celem naboru podatkov poizkušajo najti način, kako posamezne vzorce najbolje razdeliti v posamezne razrede. Alternativen pristop je pristop, pri katerem se za vsak razred vprašamo, kateri od vzorcev v dani razred spadajo in kateri ne. S tem postopkom v vsakem koraku izvajanja zgradimo pravilo, ki pokrije del vzorcev. Izvajanje algoritmov, osnovanih na tem pristopu algoritmov, zaradi njihove narave ne vodi do odločitvenih dreves, marveč do nabora pravil. Ker se z dodajanjem pravil trudimo čim bolje pokriti vzorce z istim rezultatom označimo te algoritme z imenom algoritmi s pokrivanjem [2][17][44].

Algoritmi s pokrivanjem delujejo tako, da pravilu dodajajo teste, s katerimi izboljšujejo natančnost pravila. Algoritem deli in vladaj dodaja pravila drevesu, ki ga gradi z namenom maksimiranja ločljivosti med razredi. Vsak od teh algoritmov temelji na iskanju atributa, po katerem se izvaja deljenje. Algoritmi s pokrivanjem pa izbirajo test – par

(atribut, vrednost) na tak način, da maksimirajo verjetnost želene klasifikacije. Želimo, da vsak dodaten test popravi pravilo tako, da z njim pokrijemo čim več vzorcev pravega razreda in izključimo čim več vzorcev ostalih razredov. Če novo pravilo pokrije t vzorcev, od katerih p pripada pravemu razredu, jih $t-p$ pripada ostalim razredom in predstavljajo napake. Nove teste je potrebno izbirati na tak način, da z njihovo izbiro maksimiziramo razmerje p/t.

Postopek lahko vizualiziramo v 2-D prostoru, v katerem so predstavljeni posamezni vzorci. Preprost primer je predstavljen na sliki 3.13 a. Najprej zgradimo pravilo, ki pokriva vzorce iz razreda a. Za prvi test v pravilu razdelimo prostor vzorcev vertikalno, kot je prikazano na sliki 3.13 (a) – srednja. Pravilo je torej sledeče:

```
Če x > 1.2 potem razred = a
```

Če na množici vzorcev uporabimo to pravilo, le-to ne deluje zadovoljivo, saj zajame tudi precej vzorcev, ki pripadajo razredu b. Zato dodamo temu pravilu nov test, ki že razdeljen prostor vzorcev ponovno razdeli na način, kot je prikazano na sliki 3.13 (a) – desna. Popravljeno pravilo se glasi:

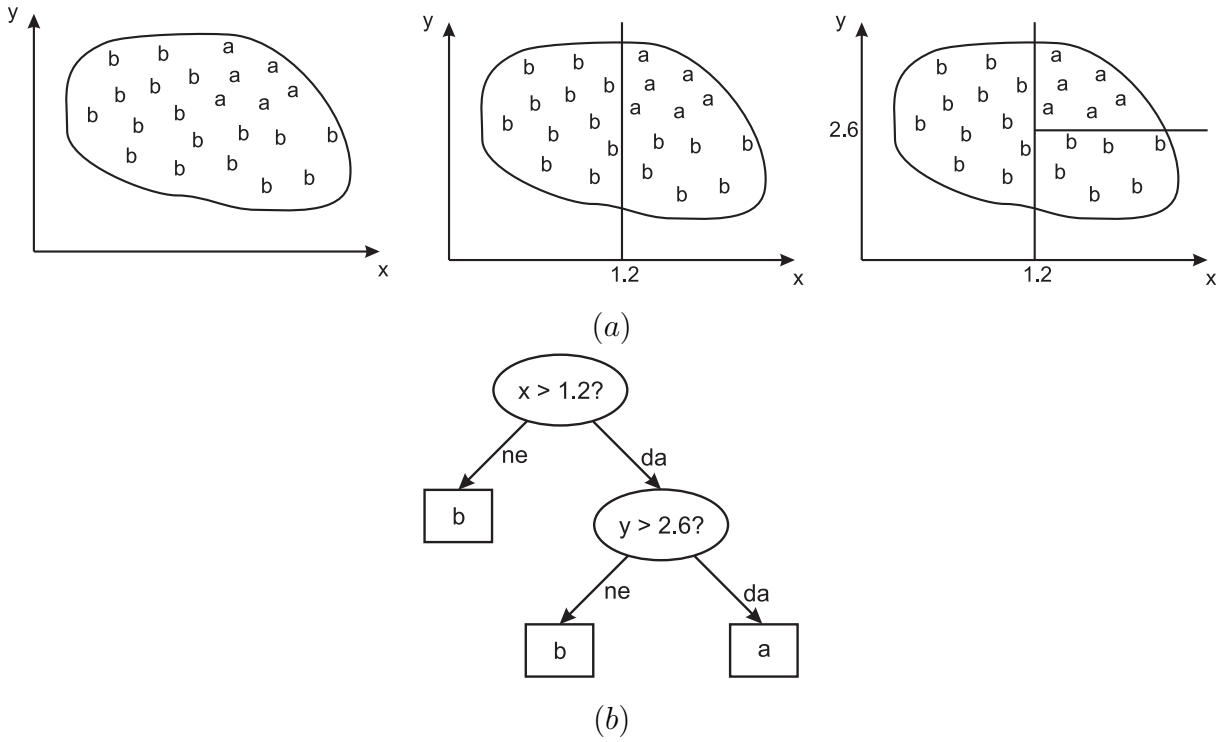
```
Če x > 1.2 in y > 2.6 potem razred = a
```

Tako sestavljeno pravilo pokriva vse vzorce, ki pripadajo razredu a. Z istim postopkom pridemo do dveh pravil, ki pokrivata vzorce iz razreda b:

```
Če x ≤ 1.2 potem razred = b
Če x > 1.2 in y ≤ potem razred = b
```

Na sliki 3.13 (b) je za primerjavo predstavljeno odločitveno drevo, ki opiše isto razvrščanje vzorcev, podanih na sliki 3.13 (a) – levo.

3.3.3.3.1 Primerjava pravil in dreves Algoritem deli in vladaj, ki deluje na istem podatkovnem naboru kot algoritem s pokrivanjem, bo zelo verjetno delal na precej podoben način. Verjetno bo podatkovni prostor razdelil po atributu x na mestu $x = 1.2$. Razlika med algoritmom deli in vladaj in algoritmom s pokrivanjem v tej točki je v tem, da se bo algoritem s pokrivanjem ukvarjal le s pokrivanjem enega razreda, algoritem deli in vladaj pa bo zgradil drevo, ki se nanaša na vse razrede. Druga delitev pri algoritmu deli in vladaj bo verjetno izvedena po atributu y $y = 2.6$. Izvajanje algoritma deli in vladaj za dani primer je odločitveno drevo, ki je predstavljeno na sliki 3.13 (b).



Slika 3.13: Algoritem s pokrivanjem (a) in odločitveno drevo za isti problem (b). Povzeto in popravljeno iz [2].

3.3.3.3.2 Preprost algoritem s pokrivanjem – PRISM V nadaljevanju je na primeru opisano delovanje preprostega algoritma PRISM [2]. Na voljo imamo nabor podatkov o načinu predpisovanja kontaktnih leč. Možni so trije izidi: priporočene so mehke leče, priporočene so trde leče, nošenje leč se odsvetuje. Leče se predpiše na podlagi opazovanja štirih parametrov: starosti pacienta, kratkovidnosti/daljnovidnosti, prisotnosti astigmatizma ter solzenja oči. Podatki so predstavljeni v tabeli 3.5.

Starost	Daljnovidnost/ Kratkovidnost	Astigmatizem	Solzenje	Priporočene leče
nizka	kratkovidnost	ne	zmanjšano	nobene
nizka	kratkovidnost	ne	normalno	mehke
nizka	kratkovidnost	da	zmanjšano	nobene
nizka	kratkovidnost	da	normalno	trde
nizka	daljnovidnost	ne	zmanjšano	nobene
nizka	daljnovidnost	ne	normalno	mehke
nizka	daljnovidnost	da	zmanjšano	nobene
nizka	daljnovidnost	da	normalno	trde
srednja	kratkovidnost	ne	zmanjšano	nobene
srednja	kratkovidnost	ne	normalno	mehke
srednja	kratkovidnost	da	zmanjšano	nobene
srednja	kratkovidnost	da	normalno	trde
srednja	daljnovidnost	ne	zmanjšano	nobene
srednja	daljnovidnost	ne	normalno	mehke
srednja	daljnovidnost	da	zmanjšano	nobene
srednja	daljnovidnost	da	normalno	nobene
visoka	kratkovidnost	ne	zmanjšano	nobene
visoka	kratkovidnost	ne	normalno	nobene
visoka	kratkovidnost	da	zmanjšano	nobene
visoka	kratkovidnost	da	normalno	trde
visoka	daljnovidnost	ne	zmanjšano	nobene
visoka	daljnovidnost	ne	normalno	mehke
visoka	daljnovidnost	da	zmanjšano	nobene
visoka	daljnovidnost	da	normalno	nobene

Tabela 3.5: Demonstracijski podatkovni nabor: podatki o načinu predpisovanja kontaktnih leč. Povzeto iz [2].

Za začetek si oglejmo generiranje pravila, ki bo pokrilo sledeče:

```
Če ? potem priporočene = trde
```

Za neznani test ? imamo na voljo 9 možnosti:

```
Starost = mlada                                    2/8
```

```
Starost = srednja                              1/8
Starost = visoka                               1/8
Daljnovidnost/Kratkovidnost = kratkovidnost 3/12
Daljnovidnost/Kratkovidnost = daljnovidnost 1/12
Astigmatizem = ne                              0/12
Astigmatizem = da                              4/12
Solzenje = zmanjšano                           0/12
Solzenje = normalno                            4/12
```

Deleži, pripisani testu, povedo število pravilnih napovedi tega testa. Ker opazujemo priporočilo za trde leče, je pravilen rezultat trde. V prvi iteraciji izberemo za generacijo pravila test, ki napove pravilni rezultat v največjem deležu napovedi. V našem primeru se to zgodi v dveh primerih. Odločimo se za sledeče pravilo:

```
če Astigmatizem = da
potem Priporočene leče = trde
```

To pravilo ni posebno natančno, saj pravilno razvrsti le 4 od 12 vzorcev. V tabeli 3.6 so prikazani vzorci, ki jih pokrije to pravilo.

Starost	Daljnovidnost/ Kratkovidnost	Astigmatizem	Solzenje	Priporočene leče
nizka	kratkovidnost	da	zmanjšano	nobene
nizka	kratkovidnost	da	normalno	trde
nizka	daljnovidnost	da	zmanjšano	nobene
nizka	daljnovidnost	da	normalno	trde
srednja	kratkovidnost	da	zmanjšano	nobene
srednja	kratkovidnost	da	normalno	trde
srednja	daljnovidnost	da	zmanjšano	nobene
srednja	daljnovidnost	da	normalno	nobene
visoka	kratkovidnost	da	zmanjšano	nobene
visoka	kratkovidnost	da	normalno	trde
visoka	daljnovidnost	da	zmanjšano	nobene
visoka	daljnovidnost	da	normalno	nobene

Tabela 3.6: Delni podatki o načinu predpisovanja kontaktnih leč pri izbranem atributu *Astigmatizem = da*. Povzeto iz [2].

Pravilo izboljšamo na sledeči način:

```
če Astigmatizem = da in
```

```
?
potem Priporočene leče = trde
```

Za neznani test ? imamo po pregledu tabele 3.6 na voljo 7 možnosti:

```
Starost = mlada                                  2/4
Starost = srednja                                1/4
Starost = visoka                                 1/4
Daljnovidnost/Kratkovidnost = kratkovidnost      3/6
Daljnovidnost/Kratkovidnost = daljnovidnost      1/12
Solzenje = zmanjšano                             0/6
Solzenje = normalno                              4/6
```

Očitno je, da bomo v nadaljevanju gradnje pravila izbrali test *Solzenje = normalno*. Rezultat je pravilo:

```
če Astigmatizem = da in
Solzenje = normalno
potem Priporočene leče = trde
```

Starost	Daljnovidnost/ Kratkovidnost	Astigmatizem	Solzenje	Priporočene leče
nizka	kratkovidnost	da	normalno	trde
nizka	daljnovidnost	da	normalno	trde
srednja	kratkovidnost	da	normalno	trde
srednja	daljnovidnost	da	normalno	nobene
visoka	kratkovidnost	da	normalno	trde
visoka	daljnovidnost	da	normalno	nobene

Tabela 3.7: Delni podatki o načinu predpisovanja kontaktnih leč pri izbranih atributih *Astigmatizem = da in Solzenje = normalno*. Povzeto iz [2].

Tabela 3.7 prikazuje vzorce, ki jih pokriva do sedaj zgrajeno pravilo. Deleži za naslednji test so sledeči:

```
Starost = mlada                                  2/2
Starost = srednja                                1/2
Starost = visoka                                 1/2
Daljnovidnost/Kratkovidnost = kratkovidnost      3/3
Daljnovidnost/Kratkovidnost = daljnovidnost      1/3
```

Do sedaj smo med sabo primerjali le deleže pravilnih napovedi, ki jih izbira določenega testa uvede v pravilo. Pri izbiri testa pa je pomembno tudi pokritje dela množice, ki ga pokrije pravilo z izbranim testom. Zato v nadaljevanju izberemo test, ki pokrije 3 vzorce *Daljnovidnost/Kratkovidnost = kratkovidnost*. Pravilo je torej sledeče:

```
če Astigmatizem = da in
Solzenje = normalno in
Daljnovidnost/Kratkovidnost = kratkovidnost
potem Priporočene leče = trde
```

Pravilo pokriva le tri od štirih primerov, v katerih so priporočene trde kontaktne leče. Zato v nadaljevanju postopka iz tabele 3.5 izbrišemo te tri primere in ponovimo postopek z začetnim pravilom.

```
Če ? potem Priporočene leče = trde
```

Po izvajanju postopka pridemo do pravila:

```
Če Starost = nizka in
Astigmatizem = da in
Solzenje = normalno
potem Priporočene leče = trde
```

S kombinacijo pravkar definiranih pravil so pokriti vsi vzorci za priporočene trde leče. V nadaljevanju postopka je potrebno definirati še pravila za priporočilo mehkih leč in priporočilo nobenih leč.

3.3.3.4 Asociiranje

Asociacijska pravila so podobna klasifikacijskim pravilom. Od klasifikacijskih pravil se razlikujejo v tem, da ne podajajo strogo povezave med vrednostmi atributov posameznega vzorca in njegovim rezultatom, marveč govore o poljubni povezavi med vrednostmi atributov in njegovih rezultatov. Do njih lahko pridemo z uporabo že opisanih algoritmov za gradnjo dreves in z algoritmi za konstruiranje pravil s pokrivanjem. Algoritmi za izračun pravil delujejo tako, da zgradijo pravila, ki za dan nabor atributov napovedo rezultat. Za izračun asociacijskih pravil je potrebno pognati omenjene algoritme z vsemi kombinacijami atributov in rezultatov vzorcev tako, da se vsi atributi in rezultati vzorcev pojavijo

	Skrčeni vektorji dolžine ena	Število pojavov
1	A_1=1	5
2	A_1=2	4
3	A_1=3	5
4	A_2=3	4
5	A_2=2	6
6	A_2=1	4
7	A_3=2	7
8	A_3=1	7
9	A_4=1	6
10	A_4=2	8
11	Rez=A	9
12	Rez=B	5

Tabela 3.8: Na en element skrčeni vektorji za podatkovni nabor, podan v tabeli 3.1. Povzeto in popravljeno iz [2].

kot atributi in kot rezultati. Kombinacija poljubne kombinacije atributov in rezultatov lahko napoveduje vrednosti poljubne kombinacije atributov in rezultatov. Tako na primer pravilo 1 v tabeli 3.12 napoveduje Rez = A v če $A_3 = 2$ in $A_4 = 2$; pravilo 5 pa napoveduje $A_4 = 2$ in Rez = A če $A_3 = 2$. Iskanje vseh asociacijskih pravil je zelo obširen postopek, ki da zelo obširno množico asociacijskih pravil. Zato je potrebno to množico zožiti na podlagi pokrivanja in natančnosti posameznih pravil.

Če se odločimo, da nas zanimajo pravila z veliko stopnjo pokrivanja, se lahko lotimo iskanja pravil na sledeči način: Najprej zapišemo vse vzorce in vzorcem pripadajoče rezultate kot posamezne vektorje. Rezultate posameznih vzorcev obravnavamo kot dodatni atribut vzorca. V nadaljevanju iz nabora vzorcev generiramo posamezne skrčene vektorje tako, da za kombinacijo posameznih atributov in njihovih vrednosti zapišemo, kolikokrat se pojavijo v originalnem naboru podatkov. Pri tem se omejimo na kombinacije, ki se pojavijo v vsaj vnaprej izbranem številu vzorcev. V demonstracijskem primeru se odločimo za 2 vzorca. Primer na en element skrčenih vektorjev za podatkovni nabor, podan v tabeli 3.1, je predstavljen v tabeli 3.8. Primeri na dva, tri in štiri elemente skrčenih vektorjev so predstavljeni v tabelah 3.9, 3.10 in 3.11.

Ko imamo na voljo skrčene vektorje z zahtevanim pokritjem, sledi pretvorba skrčenih vektorjev v pravila. Pri pretvorbi v pravila obdržimo le tista, ki dosegajo določeno stopnjo natančnosti. Na osnovi določenih skrčenih vektorjev bodo generirali več pravil, na osnovi določenih pa nobenega. Tako na primer na podlagi skrčenega vektorja številka 38 iz tabele 3.10 generiramo sedem pravil, ki jih podamo v tabeli 3.12.

Vrednosti na desni strani tabele govore o številu skrčenih podatkovnih naborov za

	Skrčeni vektorji dolžine dve	Število pojavov
1	A_1=1, A_2=2	2
2	A_1=1, A_2=1	2
3	A_1=1, A_3=2	2
4	A_1=1, A_3=1	3
5	A_1=1, A_4=1	2
6	A_1=1, A_4=2	3
7	A_1=1, Rez=A	2
...	...	...
12	A_1=2, A_4=1	2
...	...	...
47	A_4=2, Rez=B	2

Tabela 3.9: Na dva elementa skrčeni vektorji za podatkovni nabor, podan v tabeli 3.1. Povzeto in popravljeno iz [2].

	Skrčeni vektorji dolžine tri	Število pojavov
1	A_1=1, A_2=2, A_3=1	2
2	A_1=1, A_2=1, Rez=B	2
3	A_1=1, A_3=2, Rez=A	2
4	A_1=1, A_3=1, A_4=2	2
5	A_1=1, A_3=1, Rez=B	3
6	A_1=1, A_4=2, Rez=B	2
7	A_1=2, A_2=1, A_4=2	2
...	...	...
12	A_1=2, A_4=2, Rez=A	2
...	...	...
38	A_3=2, A_4=2, Rez=A	4
39	A_3=1, A_4=2, Rez=B	2

Tabela 3.10: Na tri elemente skrčeni vektorji za podatkovni nabor, podan v tabeli 3.1. Povzeto in popravljeno iz [2].

	Skrčeni vektorji dolžine štiri	Število pojavov
1	A_1=1, A_2=2, A_3=1, Rez=B	2
2	A_1=1, A_3=1, A_4=2, Rez=B	2
3	A_1=2, A_2=1, A_4=2, Rez=A	2
4	A_1=3, A_2=2, A_4=2, Rez=A	2
5	A_1=3, A_3=1, A_4=2, Rez=A	2
6	A_2=3, A_3=1, A_4=2, Rez=A	2

Tabela 3.11: Na štiri elemente skrčeni vektorji za podatkovni nabor, podan v tabeli 3.1. Povzeto in popravljeno iz [2].

	Pravilo	Natančnost
1	Če A_3=2 in A_4=2 potem Rez=A	4/4
2	Če A_3=2 in Rez=A potem A_4=2	4/6
3	Če A_4=2 in Rez=A potem A_3=2	4/6
4	Če A_3=2 potem A_4=2 in Rez=A	4/7
5	Če A_4=2 potem A_3=2 in Rez=A	4/8
6	Če Rez=A Potem A_3=2 in A_4=2	4/9
7	Če – potem A_3=2 in A_4=2 in Rez=A	4/12

Tabela 3.12: Asociacijska pravila, generirana iz skrajšanega vektorja 38 iz tabele 3.10. Povzeto in popravljeno iz [2].

katere držijo vsi trije pogoji, deljeno s številom podatkovnih naborov, za katere drži napovedani rezultat ali kombinacija rezultatov. Ta vrednost predstavlja delež vseh skrčenih podatkovnih naborov, za katere pravilo drži – natančnost pravila. Ker se odločimo, da želimo, da pravila dosegajo 100 % natančnost, je primerno samo pravilo 1. V tabeli 3.13 je predstavljenih nekaj asociacijskih pravil za podatkovni nabor, podan v tabeli 3.1. Predstavljena so pravila, ki pokrijejo najmanj dva vzorca in dosegajo 100 % natančnost. Logični operator v predstavljenih pravilih je IN. V literaturi [2] je podan učinkovit način generiranja asociacijskih pravil.

	Pravilo	Rezultat	Pokritje	Natančnost
1	$A_3 = 2$, $A_4 = 2$	Rez = A	4	100 %
2	$A_2 = 3$,	$A_3 = 2$	4	100 %
3	$A_1 = 2$,	Rez = A	4	100 %
4	$A_2 = 3$, Rez = A	$A_3 = 2$	3	100 %
5	$A_1 = 3$, $A_4 = 2$	Rez = A	3	100 %
6	$A_1 = 3$, Rez = A	$A_4 = 2$	3	100 %
7	$A_1 = 1$, $A_3 = 1$	Rez = B	3	100 %
8	$A_1 = 1$, Rez = B	$A_3 = 1$	3	100 %
9	$A_2 = 3$, $A_4 = 2$	$A_3 = 2$, Rez = A	2	100 %
10	$A_2 = 3$, $A_3 = 2$, $A_4 = 2$	Rez = A	2	100 %
11	$A_2 = 3$, $A_4 = 2$, Rez = A	$A_3 = 2$	2	100 %
12	$A_1 = 3$, $A_3 = 2$ $A_4 = 2$	Rez = A	2	100 %
13	$A_1 = 3$, $A_3 = 2$, Rez = A	$A_4 = 2$	2	100 %
14	$A_1 = 3$, $A_2 = 2$, $A_4 = 2$	Rez = A	2	100 %
15	$A_1 = 3$, $A_2 = 2$, Rez = A	$A_4 = 2$	2	100 %
16	$A_2 = 2$, $A_4 = 2$, Rez = A	$A_1 = 3$	2	100 %
17	$A_1 = 2$, $A_2 = 1$	$A_4 = 2$, Rez = A	2	100 %
18	$A_1 = 2$, $A_4 = 2$	$A_2 = 1$, Rez = A	2	100 %
19	$A_2 = 1$, Rez = A	$A_1 = 2$, $A_4 = 2$	2	100 %
20	$A_1 = 2$, $A_2 = 1$, $A_4 = 2$	Rez = A	2	100 %
21	$A_1 = 2$, $A_2 = 1$, Rez = A	$A_4 = 2$	2	100 %
22	$A_1 = 2$, $A_4 = 2$, Rez = A	$A_2 = 1$	2	100 %
23	$A_2 = 1$, $A_4 = 2$, Rez = A	$A_1 = 2$	2	100 %
24	$A_4 = 2$, Rez = B	$A_1 = 1$, $A_3 = 1$	2	100 %
25	$A_1 = 1$, $A_3 = 1$, $A_4 = 2$	Rez = B	2	100 %
26	$A_1 = 1$, $A_4 = 2$, Rez = B	$A_3 = 1$	2	100 %
27	$A_3 = 1$, $A_4 = 2$, Rez = B	$A_1 = 1$	2	100 %
28	$A_1 = 1$, $A_2 = 1$	$A_3 = 1$, Rez = B	2	100 %
29	$A_2 = 1$, Rez = B	$A_1 = 1$, $A_3 = 1$	2	100 %
30	$A_1 = 1$, $A_2 = 1$, $A_3 = 1$	Rez = B	2	100 %
31	$A_1 = 1$, $A_2 = 1$, Rez = B	$A_3 = 1$	2	100 %
...	...	...	...	...
58	$A_1 = 1$, $A_2 = 1$	$A_3 = 1$	2	100 %

Tabela 3.13: Asociacijska pravila. Povzeto in popravljeno iz [2].

3.3.3.5 Rojenje

Metode rojenja spadajo v t.i. nenadzorovane metode strojnega učenja. To so metode, pri katerih se model prilagaja opazovanim vzorcem. Od nadzorovanih metod se ločijo po dejstvu, da od njih ni pričakovanega *a priori* rezultata, ampak nabor vhodnih podatkov razdelijo v podatkom lastne skupine – roje. Metode nenadzorovanega učenja obravnavajo nabor vhodnih podatkov kot nabor naključnih spremenljivk, na katerem se potem med postopkom zgradi model, ki opisuje, na kakšen način posamezni vzorci spadajo skupaj [45]. Med metode nenadzorovanega strojnega učenja spadajo tudi metode iskanja rojev v množici vzorcev.

3.3.3.5.1 Metode iskanja rojev v množici vzorcev Metode rojenja so algoritmi za klasifikacijo neoznačenega nabora podatkov v različne podmnožice – roje. Postopki obravnavani nabor podatkov razdelijo na roje na tak način, da so si elementi v posameznih podmnožicah med sabo na nek predefiniran način čim bolj podobni, elementi, ki pa pripadajo različnim rojem, pa se med sabo na isti način kar najbolj razlikujejo [40] [46] [44]. Mehka definicija metod rojenja je: "Proces organizacije objektov v skupine, katerih člani so si podobni na nek način"[46]. Za ugotavljanje podobnosti med elementi se pogosto uporablja razdalja med elementi množice. Ko v prostoru definiramo razdaljo med elementi prostora, postane ta prostor *prostor z metriko*. Razdalja med elementi množice X je preslikava, definirana v enačbi 3.20, ki za dani par elementov te množice x in y množice da realno, nenegativno število.

Definicija preslikave razdalje je sledeča:

$$d : X \times X \rightarrow R \tag{3.20}$$

Pri čemer je R nabor realnih števil. Za preslikavo razdalje za spremenljive x, y, z velja sledeče:

$$\begin{array}{lll} 1: & d(x,y) \geq 0 & \text{Nenegativnost} \\ 2: & d(x,y) = 0 \Leftrightarrow x = 0 & \\ 3: & d(x,y) = d(y,x) & \text{Simetrija} \\ 4: & d(x,z) \leq d(x,y) + d(y,z) & \text{Trikotniška neenakost} \end{array} \tag{3.21}$$

3.3.3.5.2 Tipi algoritmov rojenja Algoritmi rojenja se delijo na algoritme, osnovane na teoriji grafov, na hierarhične algoritme, na delitvene algoritme in na algoritme z nevronskimi omrežji [39].

3.3.3.5.3 Algoritmi na osnovi grafov Algoritmi temeljijo na predstavitvi vzorcev z minimalnim vpetim drevesom in na iskanju ločenih poddreves [39].

3.3.3.5.4 Hierarhični algoritmi Algoritmi iščejo roje na zaporeden način, in sicer tako, da pri postopku uporabljajo predhodno definirane roje, ki jih glede na medsebojno podobnost postopno združujejo ali razdružujejo med sabo. Delijo se v dve skupini glede na način iskanja rojev: združevalni (ang. bottom up) ali delilni (ang. top down) [40].

Pri hierarhičnih združevalnih algoritmih predstavlja na začetku izvajanja algoritma vsak posamezen vzorec en roj. V vsakem koraku algoritmi združijo dva roja, ki sta najbolj podobna, v novi, večji roj. Ko je doseženo želeno število rojev, se postopek združevanja ustavi [39].

Pri hierarhičnih razdruževalnih algoritmih se postopek začne z enim rojem, ki vsebuje vse vzorce. Algoritem postopno deli roj na več manjših rojev na tak način, da se vzorci v posameznih novo definiranih rojih med sabo kar najbolj razlikujejo [47].

3.3.3.5.5 Delitveni algoritmi Delitveni algoritmi iskanja rojev temeljijo na razbitju množice vzorcev v podmnožice – roje na tak način, da doseže izbrana kriterijska funkcija optimalno vrednost [39]. Algoritmi temeljijo na začetnem razbitju množice vzorcev v roje in na prestavljanju vzorcev iz enega roja v drugega, če to prispeva k izboljšanju vrednosti kriterijske funkcije. Pogosto uporabljan algoritem, ki pripada tej skupini, je algoritem K-Means.

3.3.3.5.6 Metoda K-tih povprečij Algoritem metode K-tih povprečij (ang. K-Means) je preprost in računsko nezahteven algoritem. Algoritem metode K-tih povprečij pripiše roju vzorec, katerega centroid je najbližje vzorcu. Bližino ocenimo z uporabo Evklidove razdalje (ang. Euclidian distance) [17]. Centroid je točka, katere koordinate so izračunane kot aritmetična sredina koordinat vseh vzorcev v obravnavanem roju. Algoritem deluje na sledeči način [48]:

1. Izberi število rojev K.
2. Naključno generiraj K rojev, izračunaj centroide ali neposredno generiraj K točk, ki služijo kot začetni centroidi rojev.
3. Pripiši vse vzorce najbližjemu centroidu.

4. Ponovno izračunaj nove centroide.

5. Ponavljaj koraka 3 in 4, dokler se centroidi ne spreminjajo več bistveno.

Slabost algoritma metode K-tih povprečij je v tem, da je potrebno vnaprej poznati število iskanih rojev in da različna izbira začetnih točk centroidov pripelje do različnih rezultatov, kar pa za reševanje našega problema ne predstavlja ovire, saj poznamo število rojev [44]. Poznamo namreč število stopenj jakosti aglutinacije in tudi število dokončnih interpretacij za posamezne predtransfuzijske preiskave.

3.3.3.6 Numerično napovedovanje

Do sedaj opisane metode delujejo z nominalnimi atributi. V primeru, da imamo na voljo vzorce, katerih atributi so numerične vrednosti, lahko do sedaj opisane metode uporabimo tako, da teste metod prilagodimo, da le-ti izvajajo teste na numeričnih vrednostih, ali pa numerične vrednosti diskretiziramo in jim pripišemo nominalne vrednosti. Obstajajo pa tudi metode strojnega učenja, ki delujejo neposredno z numeričnimi vrednostmi atributov.

3.3.3.6.1 Numerično napovedovanje: Linearna regresija Ko je rezultat vzorca ali razred, ki mu vzorec pripada, numeričen in so vsi atributi vzorca numerični, je linearna regresija prva metoda, ki jo je vredno preiskusiti. Osnovna ideja linearne regresije je v tem, da se rezultat vzorca izrazi kot linearna kombinacija atributov vzorca, uteženih z utežmi:

$$x = w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k \tag{3.22}$$

Pri čemer je x razred ali rezultat, $a_1, a_2, \ldots, a_k$ so vrednosti atributov, $w_1, w_2, \ldots, w_k$ pa so uteži.

Uteži so izračunane iz učnega nabora podatkov. Za vsak vzorec iz učne množice zapišemo svojo linearno kombinacijo uteži in vrednosti atributov. Kot primer za 1. vzorec učne množice zapišemo sledečo enačbo:

$$w_0 + w_1 a_1^{(1)} + w_2 a_2^{(1)} + \ldots + w_k a_k^{(1)} = \sum_{j=0}^{k} w_j a_j^{(1)} \tag{3.23}$$

Zapisana vrednost predstavlja napovedano in ne resnične vrednosti prvega vzorca. Zanima nas razlika med resnično in napovedano vrednostjo opazovanega vzorca. Postopek

linearne regresije predstavlja izbiranje $k+1$ uteži w_j na tak način, da je vsota kvadratov razlike med napovedanimi in resničnimi vrednostmi za učno množico najmanjša. Če imamo v učni množici n vzorcev, je vsota kvadratov razlik sledeča:

$$\sum_{i=1}^{n} \left(x^{(i)} - \sum_{j=0}^{k} w_j a_j^{(i)} \right)^2 \tag{3.24}$$

S postopkom optimizacije izberemo uteži na tak način, da je vsota kvadratov razlik, podana v enačbi 3.23, najmanjša. Po postopku optimizacije imamo na voljo nabor uteži, s katerimi znamo oceniti vrednost novih vzorcev.

Linearna regresija je preprosta metoda numerične predikcije, vendar v primeru, ko podatki izražajo nelinerne lastnosti, ne deluje najbolje.

3.3.4 Ocenjevanje učinkovitosti metod strojnega učenja

Za razvoj učinkovitega postopka strojnega učenja potrebujemo metode za evaluacijo modelov, ki jih dobimo z uporabo algoritmov strojnega učenja. Na ta način lahko med sabo primerjamo različne metode strojnega učenja in izberemo najboljšo. Za primerjavo posameznih metod strojnega učenja potrebujemo sistematičen način za oceno delovanja in primerjavo posameznih metod.

Rezultat izvajanja algoritma strojnega učenja je naučeni model obravnavanega sistema. Med sabo primerjamo učinkovitost delovanja različnih modelov in s tem algoritmov strojnega učenja, ki smo jih uporabili za gradnjo teh modelov. Za učenje in testiranje modela sistema imamo v večini primerov omejen podatkovni nabor – nabor vzorcev z rezultati. Ta nabor podatkov moramo uporabiti za učenje in testiranje modela. Za učinkovito učenje moramo model naučiti s kar največ učnimi vzorci. Skupek učnih vzorcev imenujemo učna množica. Fazi učenja modela sledi faza testiranja le-tega. Model testiramo s testno množico podatkov. Testna množica podatkov je sestavljena iz vzorcev podatkov, opremljenih s pripadajočimi rezultati. Pri testiranju modela z modelom izračunamo rezultate vzorcev iz testne množice. Dobljene rezultate primerjamo z znanimi rezultati. Za učinkovit test je potrebno model strojnega učenja preizkusiti s kar največ testnimi vzorci. Pri testiranju je potrebno uporabiti podatke, ki niso bili uporabljeni za gradnjo modela. Na ta način dobimo rezultate, ki realno napovedujejo obnašanje modela na neznanih podatkih [2].

3.3.4.1 Mere učinkovitosti metod strojnega učenja – razvrščanja

Ker je nabor rezultatov, ki jih v našem primeru napovedujejo modeli strojnega učenja, končen in diskreten, lahko delo modelov obravnavamo kot razvrščanje vektorjev lastnosti v razrede C_i. Razredi so označeni z oznakami ω_i. Posamezni vzorec – vektor atributov označimo z $\vec{x}$. Dejstvo, da vzorec x pripada razredu C_i, zapišemo kot:

$$\vec{x} \in C_i \tag{3.25}$$

Dogodek razvrščanja posameznega vzorca označimo z $\delta(\vec{x})$. Pri dogodku razvrščanja pripišemo vzorcu X oznako razreda ω_i. Definicija je podana v enačbi 3.26:

$$\delta(\vec{x}) = \omega_i|_{\vec{x} \in C_k} \tag{3.26}$$

Pravilno razvrščanje označimo:

$$\delta(\vec{x}) = \omega_i|_{\vec{x} \in C_k} \quad i = k \tag{3.27}$$

Ker v našem modulu pričakujemo neidealnosti, je mogoče, da naš model napačno razvrsti vzorec $\vec{x}$. Dogodek napačnega razvrščanja označimo kot:

$$\delta(\vec{x}) = \omega_i|_{\vec{x} \in C_k} \quad i \neq k \tag{3.28}$$

Rezultati zanesljivosti modela razvrščanja vektorjev lastnosti v posamezne razrede so predstavljeni v matriki pravilnih in napačnih razvrstitev (ang. confussion matrix). Matrika je podana v tabeli 3.14. Njene dimenzije so $M \times M$, pri čemer je M število razredov. Posamezna mesta v matriki predstavljajo število dogodkov razvrstitev posameznega vzorca v posamezen razred [49][4]. V matriki je na mestih (i, k) zapisano število dogodkov razvrstitve vzorcev v posamezne razrede pri preverjanju metode. Dogodki so definirani z enačbo 3.26. Dogodek pravilne razvrstitve se zgodi v primeru $i = k$. Število teh dogodkov je zabeleženo v diagonalnih elementih. Vsota vseh diagonalnih elementov matrike predstavlja število vseh pravilnih razvrstitev, vsota vseh ostalih elementov pa predstavlja število vseh napačnih razvrstitev.

Za vsak razred iz matrike pravilnih in napačnih razvrstitev izračunamo sledeče parametre [4][49]:

- Delež vektorjev lastnosti $\vec{x} \in C_i$, razvrščenih v ω_i, $\delta(\vec{x}) = \omega_i|_{\vec{x} \in C_i}$ med vsemi vzorci ki res pripadajo ω_i (ang. true positive, recall). Izračunamo ga tako, da ustrezen

$$\begin{array}{lc} \text{pripada:} & \begin{array}{cccc} C_1 & C_2 & \dots & C_M \end{array} \\ \begin{array}{l} C_1 \\ C_2 \\ \vdots \\ C_M \end{array} & \begin{pmatrix} \sharp\left[\delta(x)=\omega_1|_{\vec{x}\in C_1}\right] & \sharp\left[\delta(\vec{x})=\omega_1|_{\vec{x}\in C_2}\right] & \dots & \sharp\left[\delta(\vec{x})=\omega_1|_{\vec{x}\in C_M}\right] \\ \sharp\left[\delta(x)=\omega_2|_{\vec{x}\in C_1}\right] & \sharp\left[\delta(\vec{x})=\omega_2|_{\vec{x}\in C_2}\right] & \dots & \sharp\left[\delta(\vec{x})=\omega_2|_{\vec{x}\in C_M}\right] \\ \vdots & \vdots & \ddots & \vdots \\ \sharp\left[\delta(\vec{x})=\omega_M|_{\vec{x}\in C_1}\right] & \sharp\left[\delta(\vec{x})=\omega_M|_{\vec{x}\in C_2}\right] & \dots & \sharp\left[\delta(\vec{x})=\omega_M|_{\vec{x}\in C_M}\right] \end{pmatrix} \end{array}$$

Tabela 3.14: Matrika pravilnih in napačnih razvrstitev (ang. confusion matrix).

diagonalen element matrike pravilnih in napačnih razvrstitev delimo z vsoto cele vrstice, kateri pripada element.

- Delež vektorjev lastnosti, razvrščenih v razred ω_i, ki pripadajo drugemu razredu ω_k, $\delta(\vec{x} = \omega_i|_{\vec{x}\in C_k}, k \neq i$, med vsemi vzorci, ki ne pripadajo ω_i (ang. false positive). Izračunamo ga tako, da od vsote vseh elementov v matriki pravilnih in napačnih razvrstitev v obravnavanem stolpcu odštejemo diagonalen element in dobljeno delimo z vsoto elementov v vseh ostalih vrsticah.

- Delež pravilno razvrščenih vektorjev lastnosti. To so razvrstitve, ki res spadajo v $\omega_i, \delta(\vec{x}) = \omega_i|_{\vec{x}\in C_i}$, med vsemi vektorji lastnosti, ki so bili razvrščeni ω_i (ang. precission). Izračunamo ga tako, da se opazovani diagonalni element deli z vsoto vseh elementov v pripadajočem stolpcu.

Za vsako razvrščanje izračunamo sledeče parametre:

- Delež pravilno razvrščenih S. Število predstavlja delež pravilno razvrščenih vzorcev med vsemi vzorci, ki so bili razvrščeni. Izračunamo ga na sledeč način: Vsoto diagonalnih elementov matrike pravilno in napačno razvrščenih delimo z vsoto vseh elementov te matrike.

- Delež napačno razvrščenih. Število predstavlja delež napačno razvrščenih vzorcev med vsemi razvrščenimi vzorci. Izračunamo ga na sledeč način: Vsoto vseh nediagonalnih elementov matrike pravilno in napačno razvrščenih delimo z vsoto vseh elementov te matrike.

3.3.4.2 Evaluacija s testnim naborom podatkov

Preizkus učinkovitosti različnih modelov razvrščanja vektorjev lastnosti v razrede je mogoče opraviti na več načinov. Pri najpreprostejšem se model razvrščanja preizkusi

s testnim naborom podatkov. Ker je v našem primeru količina podatkov, namenjenih učenju in testiranju sistema, dokaj omejena, je uporabnejša bolj dovršena metoda navzkrižne validacije (ang. cross-validation) [4][50].

3.3.4.3 Navzkrižna validacija

Navzkrižna validacija je statistična metoda za preizkušanje učinkovitosti modelov, zgrajenih z metodami strojnega učenja [2]. Metoda se uporablja v primerih, ko ni na voljo dovolj velikega podatkovnega nabora za učenje in testiranje naučenih modelov. V postopku navzkrižne validacije se za učenje modela in njegovo validacijo uporabi iste podatke. Cena, ki jo plačamo za to, je velika računska zahtevnost postopka. V začetku postopka določimo število *pregibov* danega podatkovnega nabora m. Število pregibov pomeni število delitev podatkovnega nabora na podmnožice. V nadaljevanju postopka vzorce v podatkovnem naboru naključno premešamo in razdelimo v m enako velikih podmnožic. Po opravljeni razdelitvi se postopek učenja požene m-krat. Pri vsaki iteraciji se $m-1$ podmnožic podatkovnega nabora uporabi za učenje modela, preostanek pa za testiranje modela. Rezultat vsakega testiranja modela se zabeleži v matriko pravilnih in napačnih razvrstitev. Ob koncu postopka navzkrižne validacije rezultate posameznih testiranj zberemo in povprečimo. Dobljeni rezultat je dober približek zanesljivosti metode. Glede na število delitev dane učne množice poimenujemo metodo validacije m pregibna navzkrižna validacija [2].

Mešanje učne množice pri deljenju na podmnožice je lahko povsem naključno. Boljše rezultate dobimo, če uporabimo delno naključno mešanje. Mešanje izvedemo na tak način, da so porazdelitve števila posameznih razredov v vseh m podmnožicah,Ł uporabljenih za učenje in preizkušanje, enake porazdelitvi v celotnem podatkovnem naboru. Če je za mešanje uporabljen ravnokar opisani način, govorimo o m-pregibni *stratified* navzkrižni validaciji.

V literaturi [2] avtorji navajajo, da je standarden način statističnega ocenjevanja učinkovitosti metode strojnega učenja pri danem fiksnem naboru podatkov uporaba 10-pregibne *stratified* navzkrižne validacije. Obširni testi na različnih metodah strojnega učenja so pokazali, da je 10 pravo število pregibov za realno in pravilno oceno pravilnosti delovanja modela, naučenega z metodami strojnega učenja. 10-pregibna stratified navzkrižna validacija je *de facto* standard na področju strojnega učenja. V literaturi [2] navedeni testi so pokazali, da ni velike razlike med 10-pregibno *stratified* navzkrižno validacijo in 10-pregibno navzkrižno validacijo. *Stratified* navzkrižno validacijo je potrebno

uporabiti samo v primerih, ko je na voljo zelo omejen nabor podatkov [2].

3.3.4.4 Validacija izpusti enega

Pri validaciji izpusti enega (ang. leave one out)[2] je za m izbrano kar število vzorcev učne množice. Ker pri tem načinu validacije modela testni nabor podatkov nima enake porazdelitve razredov kot učna množica, je dobljeni podatek o zanesljivosti modela manj zanesljiv, ampak še vedno uporaben. Metodo uporabimo v primerih, ko imamo na voljo male podatkovne nabore. V tem primeru lahko za učenje modela uporabimo kar največji del učne množice.

3.3.4.5 Primerjava različnih metod

Na podlagi primerjave različnih metod strojnega učenja se odločimo, katera metoda strojnega učenja je najprimernejša za reševanje danega problema. Postopek primerjave je preprost. Za vsako od obravnavanih metod strojnega učenja izvedemo postopek validacije metode – navzkrižno validacijo. Za vsako izvedeno validacijo izračunamo delež uspešnosti S. Delež uspešnosti S je delež pravilno razvrščenih med vsemi razvrščenimi [2]. Izračunamo ga iz matrike pravilnih in napačnih razvrstitev tako, da delimo vsoto vseh diagonalnih elementov matrike pravilnih in napačnih razvrstitev z vsoto vseh elementov, kot je zapisano v enačbi 3.29:

$$S = \frac{\sharp\left[\delta(\vec{x}) = \omega_i|_{\vec{x}\in C_k},\ i = k\right]}{\sharp\left[\delta(\vec{x}) = \omega_i|_{\vec{x}\in C_k}\right]} \tag{3.29}$$

Običajno izberemo metodo strojnega učenja, pri uporabi katere dobimo model, ki za dani nabor testnih/učnih podatkov z izbranimi parametri pravilno razvrsti največji delež vzorcev.

3.4 Zajem in registracija podatkov

Zajem in registracija podatkov obravnava pridobivanje vzorcev za izdelavo učnih in testnih množic. Obravnava tudi pridobivanje vzorcev za analizo in razpoznavanje v delujočem sistemu za samodejno interpretacijo rezultatov predtransfuzijskih testiranj. Podatki, ki jih bomo obravnavali, se nahajajo v sistemu za telekonzultacije. Podatki so kompleksne podatkovne strukture, sestavljene iz datotek, ki vsebujejo slike gelskih kartic, določenih

stopenj jakosti aglutinacije za posamezne kolone, tipe testiranja izvajanega z gelsko kartico in dokončne interpretacije predtransfuzijskega testiranja.

Ker obravnavamo reševanje problema v dveh korakih, bomo podatke, iz katerih bomo tvorili nabore učnih, testnih in neznanih množic, obravnavali v dveh korakih. Tako bomo ločili pridobivanje vzorcev za prvi in drugi korak. Vzorci za prvi korak, korak določanja stopnje jakosti aglutinacije v posameznih kolonah, predstavljajo slike posameznih kolon in v primeru učne in testne množice pripadajoče stopnje jakosti aglutinacije iz množice 7 možnosti – prazno, NEG, 1+, 2+, 3+, 4+, DCP. Vzorci za drugi korak, korak razvrščanja vektorjev nabora stopnje jakosti aglutinacije eritrocitov v končni rezultat preiskave, predstavljajo v primeru učne in testne množice tip preiskave, vektor z določenimi stopnjami jakosti aglutinacije za vsako od 6 kolon, vsebovanih na obravnavani gelski kartici, in dokončno interpretacijo predtransfuzijske preiskave.

Vzorci, ki jih želimo analizirati ali razpoznavati, so pogosto slabše kakovosti [39]. Prav tako ti vzorci običajno še niso na voljo v primerni obliki za namene analize in razpoznavanja. Zato je potrebno te vzorce najprej na ustrezen način zajeti iz okolja, v katerem so nastali. Po zajemu jih je potrebno ustrezno označiti in izboljšati. Za namene izboljšave vzorcev moramo poznati modele virov popačenj, ki so pokvarili vzorce. V našem primeru je glavni vir popačenj slikovnih vzorcev rotacija in translacija slik. Odpravljanje teh dveh virov popačenja podatkov je opisano v nadaljnjem tekstu.

3.4.1 Sistem za telekonzultacije v transfuzijski medicini

Kot smo že omenili, smo sistem za telekonzultacije v transfuzijski medicini zgradili in uvedli v prakso transfuzijske službe v Sloveniji. Sistem omogoča prenos vseh podatkov, potrebnih za izvajanje transfuzijskih storitev. Sistem poleg prenosa podatkov omogoča tudi hrambo laboratorijskih rezultatov. V sistemu se zbirajo podatki o telekonzultacijskih sejah, ki zajemajo vse za dokončno interpretacijo predtransfuzijskih preiskav potrebne diagnostične podatke in s strani specialistov določene interpretacije teh testov. Podatke iz sistema zajamemo in iz njih zgradimo nabor podatkov, ki ga uporabimo za razvoj sistema samodejne interpretacije predtransfuzijskih preiskav.

Predstavljene rešitve in orodja so služile pri razvoju sistema samodejne interpretacije predtransfuzijskih preiskav. Razvoj sistema smo opisali v sledečem poglavju.

Poglavje 4

Razvoj sistema za samodejno interpretacijo

Z uporabo obstoječih rešitev in orodij, opisanih v prejšnjem poglavju in z razvojem novih, smo izdelali osnovo sistema za samodejno interpretacijo predtransfuzijskih preiskav. V okviru raziskav smo izbrali optimalno kombinacijo in parametrizacijo posameznih algoritmov, primernih za gradnjo sistema za samodejno interpretacijo predtransfuzijskih preiskav.

4.1 Razdrobitev problema na korake, ki posnemajo delo specialista transfuzijske medicine

Od postopka samodejne interpretacije rezultatov preiskav pričakujemo, da se bo na podlagi zabeleženih odločitev specialistov transfuzijske medicine naučil ekspertize teh specialistov in bo znal to ekspertizo ponoviti na neznanih primerih.

Problem samodejne interpretacije rezultatov preiskav je mogoče razdeliti na več podproblemov. Razdelitev na podprobleme je sorodna razdelitvi, ki jo opravijo specialisti transfuzijske medicine. S sistemom za samodejno interpretacijo rezultatov smo posnemali delo specialista transfuzijske medicine. Specialist interpretira predtransfuzijsko preiskavo v dveh korakih. V prvem koraku določi za vsako kolono gelske kartice uporabljene v preiskavi stopnjo jakosti aglutinacije. Glede na stopnjo jakosti aglutinacije razvrsti vsako kolono v enega od 7 razredov (Prazno, NEG, 1+, 2+, 3+, 4+, DCP). V drugem koraku na podlagi kombinacije določenih stopenj jakosti aglutinacije kolon določi dokončno inter-

pretacijo preiskave. Postopek interpretacije v dveh korakih je ilustriran na sliki 4.1. Za vsako od preiskav, ki jih opravljajo v postopku predtransfuzijskih preiskav obstaja končni nabor možnih interpretacij. V nadaljevanju smo problem samodejne interpretacije predtransfuzijskih preiskav obravnavali kot dva ločena problema. Sam postopek reševanja obeh problemov je med seboj neodvisen. Pri skupni obravnavi pa je od rezultatov reševanja prvega problema odvisno reševanje drugega, saj so vhodni podatki drugega problema rezultati prvega.

Za vsakega od problemov smo zgradili ločen model. Prvi modelira določanje stopnje jakosti aglutinacije, drugi pa določanje dokončne interpretacije preiskave.

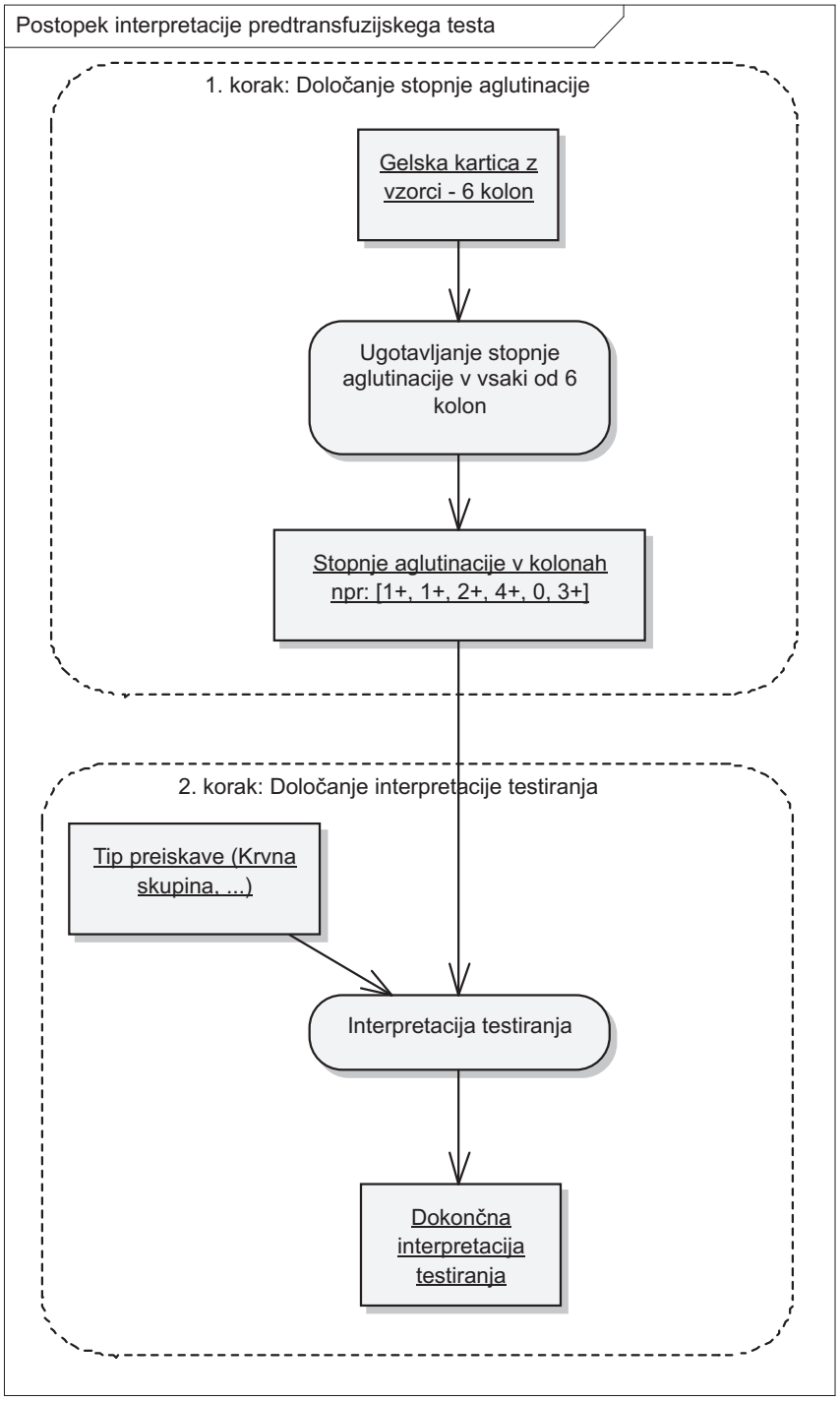
Gradnjo obeh modelov smo obravnavali ločeno s postopki strojnega učenja. Za vsakega od problemov smo izbrali optimalen algoritem strojnega učenja, z njim zgradili model interpretacije in ga preizkusili. Na koncu smo rezultate združili in sicer zaradi ugotavljanja stopnje zaupanja dobljenih končnih rezultatov evaluacije. Stopnja zaupanja zajema evaluacijo obeh korakov strojnega učenja za vsak posamezen rezultat, izračunan s sistemom.

4.2 Strojno učenje

Postopki strojnega učenja spadajo v področje umetne inteligence. Metode in tehnike strojnega učenja omogočajo strojem, da se na podlagi kombinacije znanih vzrokov (vhodov) in posledic (rezultatov) nauče napovedovanja neznanih posledic iz danih vzrokov. Cilj postopkov strojnega učenja je izdelava modela sistema, ki na podlagi danih vhodov v opazovani realni sistem napove rezultat sistema [17][39][44].

Celoten postopek obravnave in dela realnega sistema je sestavljen iz dveh faz. V prvi fazi, fazi učenja, smo z algoritmi strojnega učenja zgradili in preizkusili model obravnavanega sistema. Ta model smo v drugi fazi, fazi razpoznavanja, uporabili za napovedovanje delovanja obravnavanega sistema. Prvo fazo, fazo učenja, smo predstavili na sliki 4.2. Drugo fazo, faza razpoznavanja, pa na sliki 4.3. V sistemu za samodejno interpretacijo preiskav je obema fazama skupen način pridobivanja podatkov iz sistema. Tako smo morali podatke, ki smo jih obravnavali v našem postopku, iz obravnavanega sistema najprej zajeti in jih po zajemu obdelati, da so bili primerni za obdelavo z algoritmi strojnega učenja. To obdelavo avtorji v literaturi imenujejo predobdelava vzorcev in registracija podatkov [39].

Za fazo učenja so pridobljeni podatki kombinacija vhodnih podatkov v realnem obravnavanem sistemu ter pripadajočih rezultatov. V našem primeru so to bile slike gelskih



Slika 4.1: Interpretacija predtransfuzijske preiskave poteka v dveh korakih. V prvem koraku je določena stopnja jakosti aglutinacije za vsako od 6 kolon v obravnavani gelski kartici. V drugem koraku se na podlagi v prvem koraku določenih stopenj jakosti aglutinacije in tipa preiskave določi dokončna interpretacija preiskave.

kartic, tipi preiskav, določene stopnje jakosti aglutinacije za posamezne kolone in dokončne interpretacije preiskav. Zajemu in obdelavi podatkov je sledilo deljenje podatkov na učno in testno množico podatkov. Z učno množico smo z algoritmi strojnega učenja zgradili model, ki smo ga s testno množico podatkov preverili in ocenili njegovo učinkovitost. Podatek o učinkovitosti modela smo potrebovali iz več razlogov. Za našo obravnavo so bili najpomembnejši trije:

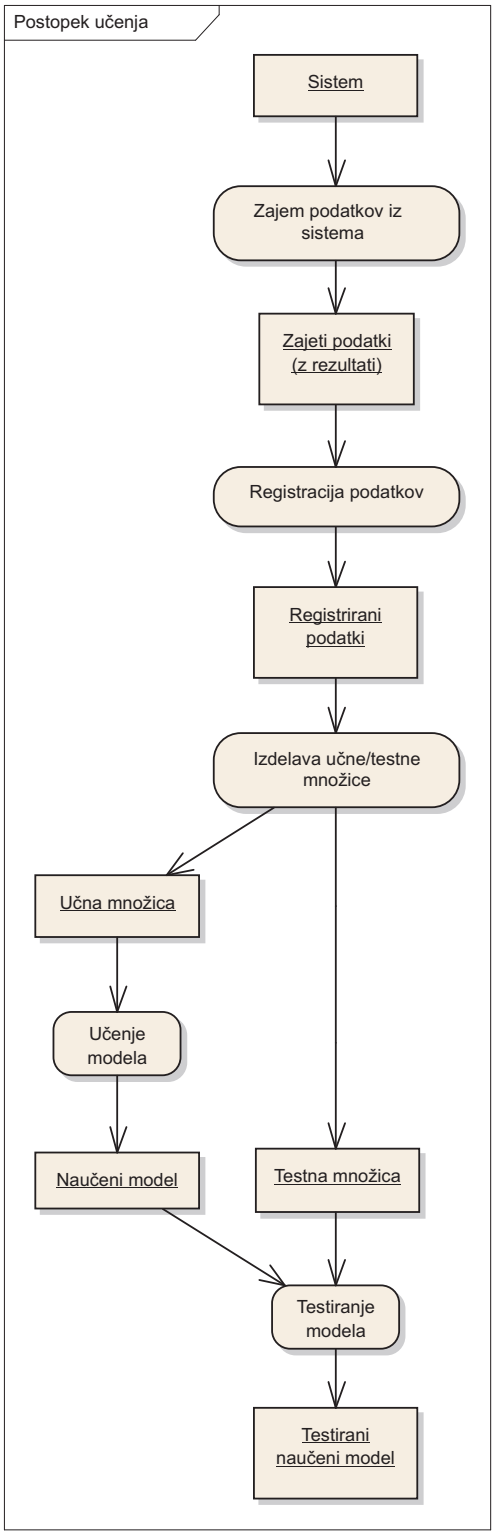
- Primerjava učinkovitosti modelov, naučenih na različne načine, in s tem izbira najboljšega.
- Ugotavljanje, ali dodatni elementi v učni množici prispevajo k izboljšanju učinkovitosti modela.
- Izračun zanesljivosti napovedi, na podlagi katere se odločimo, ali bomo rezultatu verjeli ali ga bomo iz danih podatkov ocenili sami.

Pri našem delu smo preizkusili in med sabo primerjali več metod strojnega učenja in izbrali za dani problem najboljšo. Rezultat postopka je bil testiran, z najoptimalnejšim algoritmom strojnega učenja naučeni model.

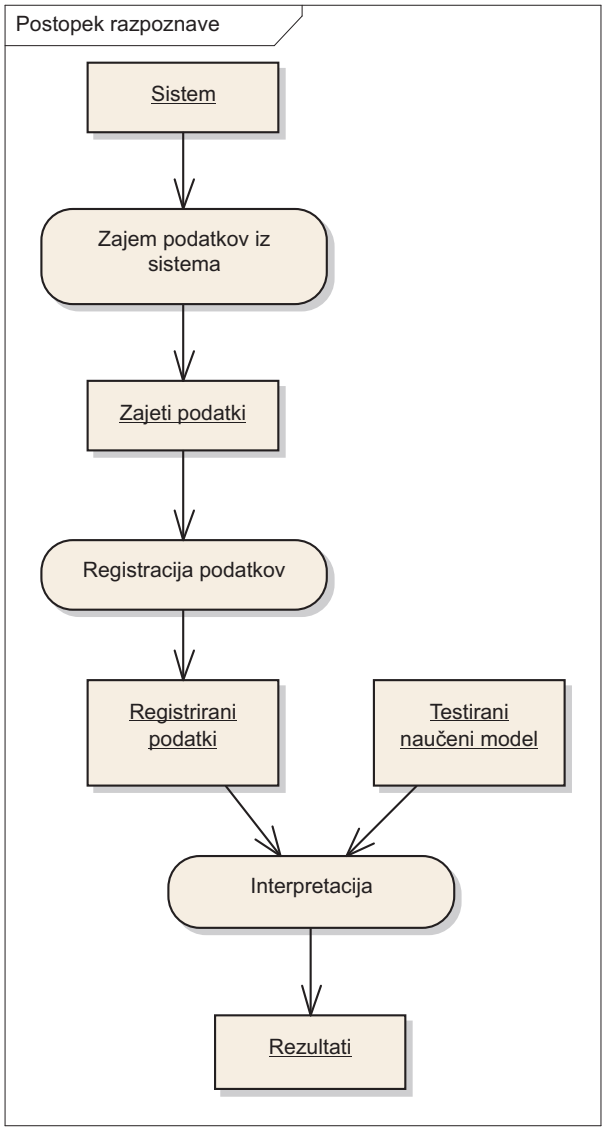
Drugo fazo, fazo razpoznavanja, smo predstavili na sliki 4.3. Prvi del faze (zajem podatkov in registracija) je enak kot pri fazi učenja. V drugem delu faze smo na zajetih registriranih podatkih uporabili v prvi fazi naučeni model interpretacije teh podatkov.

4.2.1 Uporabljeni algoritmi strojnega učenja

Pri naši raziskavi smo preizkusili algoritme strojnega učenja, podane v tabeli 4.1. Tako veliko množico algoritmov strojnega učenja smo preizkusili zato, ker smo želeli ugotoviti, kateri algoritem deluje najbolje. V poglavju 3.3.3 smo predstavili osnovne ideje in prijeme, na katerih so osnovani uporabljeni algoritmi strojnega učenja. Podrobna obravnava vseh uporabljenih algoritmov bi bila preobsežna, zato smo jo izpustili in navedli samo vire literature, kjer so posamezni algoritmi opisani. Izbira algoritmov je zajela vse glavne pristope strojnega učenja. Na podlagi primerjave s podobnimi orodji menimo, da je bila v raziskavo zajeta glavnina trenutno obstoječih uporabnih algoritmov strojnega učenja. Uporabili smo metode na osnovi statističnega modeliranja podatkov. To so metode tipa *bayes*. Uporabili smo metode, ki se modelov opazovanega sistema učijo tako, da gradijo drevesa. To so metode tipa *trees*. Uporabili smo metode, ki se modelov opazovanega



Slika 4.2: Faza učenja modela – splošno. V postopku učenja je potrebno najprej iz opazovanega sistema pridobiti testno in učno množico podatkov. Z učno množico smo z algoritmi strojnega učenja zgradili model, ki smo ga s testno množico preizkusili.



Slika 4.3: Faza razpoznave – splošno. V fazi razpoznave uporabljamo v fazi učenja pridobljeni model sistema za simuliranje delovanja realnega opazovanega sistema. Za to fazo je potrebno iz obravnavanega sistema zajeti podatke, ki so običajno brez rezultatov. S preizkušenim in naučenim modelom, pridobljenim v prvi fazi, fazi učenja, interpretiramo podatke in napovemo rezultat obravnavanega realnega sistema.

sistema naučijo z generiranjem pravil. To so metode tipa *rules*. V teoriji strojnega učenja so se pred nedavnim pojavili različni prijemi izboljšav delovanja osnovnih metod. Metode, ki uporabljajo te prijeme, so znane pod imenom *meta*.

Številka metode	Tip metode	Ime metode
1	bayes	BayesNet [2]
2	bayes	ComplementNaiveBayes [51]
3	bayes	NaiveBayes [2]
4	bayes	NaiveBayesMultinomial [52]
5	bayes	NaiveBayesUpdateable [53]
6	functions	Logistic [54]
7	functions	MultilayerPerceptron [55]
8	functions	RBFNetwork [56]
9	functions	SimpleLogistic [57]
10	functions	SMO [58]
11	lazy	IB1 [59]
12	lazy	IBk [59]
13	lazy	KStar [60]
14	lazy	LWL [61]
15	meta	AdaBoostM1 [62]
16	meta	AttributeSelectedClassifier [4][5]
17	meta	Bagging [63]
18	meta	ClassificationViaRegression [64]
19	meta	CVParameterSelection [65]
20	meta	Decorate [6]
21	meta	FilteredClassifier [4][5]
22	meta	Grading [66]
23	meta	LogitBoost [67]
24	meta	MultiBoostAB [68]
25	meta	MultiClassClassifier [4][5]
26	meta	MultiScheme [4][5]
27	meta	OrdinalClassClassifier [4][5]
28	meta	RacedIncrementalLogitBoost [4][5]
29	meta	RandomCommittee [4][5]
30	meta	Stacking [69]
31	meta	StackingC [70]
32	meta	Vote [4][5]
33	misc	HyperPipes [4][5]
34	misc	VFI [71]
35	trees	DecisionStump [72]
36	trees	J48 [7]
37	trees	LMT [8]
38	trees	NBTree [73]
39	trees	RandomForest [9]
40	trees	RandomTree [4][5]
41	trees	REPTree [4][5]
42	rules	ConjunctiveRule [4][5]
43	rules	DecisionTable [74]
44	rules	JRip [10]
45	rules	NNge [75]
46	rules	OneR [76][2]
47	rules	PART [11]
48	rules	Ridor [4][5]
49	rules	ZeroR [2][4][5]

Tabela 4.1: Preizkušeni algoritmi strojnega učenja.

4.2.2 WEKA

Za izvedbo eksperimentov, s katerimi smo ugotavljali primernost algoritmov strojnega učenja za izdelavo modelov interpretacije rezultatov predtransfuzijskih preiskav, smo uporabili programski paket WEKA [5][4][72]. Pregled uporabljenih algoritmov strojnega učenja je podan v tabeli 4.1

Algoritmi so implementirani v programskem jeziku Java in so vključeni v okolje, ki omogoča njihovo poganjanje. WEKA je opremljen z orodji, ki omogočajo nalaganje in shranjevanje podatkovnih struktur, ter z orodji za delo s podatkovnimi nabori in zgrajenimi modeli. Vsebuje tudi orodja za evaluacijo modelov. WEKA vsebuje tudi okolje, ki omogoča gradnjo in poganjanje kompleksnih in obširnih eksperimentov. Eksperiment v WEKA-i predstavlja opis preizkusa metod strojnega učenja na določenem naboru podatkov. V eksperiment je vključen tudi postopek validacije naučenih modelov. Posamezen eksperiment je sestavljen iz poti do ARFF datotek, ki vsebujejo podatke testne in učne množice, izbire v eksperiment vključenih algoritmov strojnega učenja in algoritmom pripadajoče parametrizacije. Eksperiment vsebuje tudi opis metode validacije dobljenih modelov. Opis eksperimenta se zapiše v XML datoteko. XML datoteke z opisi eksperimentov smo za vse eksperimente zgradili ročno s tekstovnim urejevalnikom.

Da smo podatke lahko uporabili v WEKA-i, smo jih zapisali v ustreznem podatkovnem formatu. Format datotek je bil ARFF format. Format je opisan v nadaljevanju teksta.

4.2.2.1 ARFF format

ARFF format je razviden iz vzorčne datoteke. Vzorčni podatki v predstavljeni datoteki so povzeti iz [2] in ustrezno popravljeni. Datoteka vsebuje odločitve, ali gremo na sprehod. Odločitev predstavlja rezultat sistema. Osnovane so na podlagi vremenskih razmer. Te vremenske razmere so posamezni atributi. Posamezni atributi govore o stanju neba, temperaturi ozračja, vlažnosti ozračja in vetrovnosti.

```
%
% ARFF datoteka, ki opisuje odločitev ali iti na sprehod
%
@relation vreme
@attribute nebo { sončno, oblačno, dež }
@attribute temperatura numeric
@attribute vlaga numeric
```

```
@attribute veter { da, ne }
@attribute sprehod { da, ne }
@data
%
% 14 vzorcev
%
sončno,  29, 85, ne, ne
sončno,  26, 90, da, ne
oblačno, 28, 86, ne, da
dež,     21, 96, ne, da
dež,     20, 80, ne, da
dež,     18, 70, da, ne
oblačno, 17, 65, da, da
sončno,  22, 95, ne, ne
sončno,  20, 70, ne, da
dež,     24, 80, ne, da
sončno,  24, 70, da, da
oblačno, 22, 90, da, da
oblačno, 27, 75, da, da
dež,     21, 91, da, ne
```

Datoteka se prične z opisom vsebine. V datoteko je zapisano ime obravnavane relacije. V našem primeru je to vreme. Opis relacije se začne z rezervirano besedo `@relation`, kateri sledi ime relacije. Sledi naštevanje vseh atributov in definicija njihovih tipov. Opis posameznih atributov se začne z rezervirano besedo `@attribute`, nadaljuje z imenom ter opisom tipa. Atributi so lahko numerični ali nominalni. V našem primeru so atributi `nebo`, `veter` in `sprehod` nominalni atributi, atributa `temperatura` in `vlaga` pa sta numerična. Če je atribut numeričen, je to označeno z rezervirano besedo `numeric`, ki sledi imenu atributa. V primeru nominalnih atributov je znotraj zavitih oklepajev podan nabor možnih vrednosti. Za naš primer so za atribut `nebo` možne vrednosti `sončno`, `oblačno` in `dež`. Opisu atributov sledijo podatki za posamezne vzorce. Začetek bloka podatkov je označen z rezervirano besedo `@data`. Vsak vzorec je zapisan v svoji vrstici. Atributi vzorca so navedeni v enakem vrstnem redu, kot so bili našteti v glavi datoteke. Med sabo so ločeni z vejico. Če so atributi nominalni, potem je na mestu atributa napisana vrednost iz zaloge vrednosti, podane v opisu atributa. Če je atribut numeričen, je podana njegova

vrednost.

4.2.2.2 Eksperiment

WEKA omogoča zapis eksperimenta v XML datoteko. Vsak ekperiment smo zapisali v XML datoteko. V datoteki, ki opisuje eksperiment, je definirana pot do ARFF datoteke, ki vsebuje nabor podatkov. Podroben opis formata datoteke za opis eksperimenta, ki je potreben za gradnjo datotek z eksperimenti, je podan v [4], [72] in [2]. V tekstu disertacije smo ta opis izpustili in povzeli le osnovne značilnosti. V opisu eksperimenta je podana pot do javanskega razreda, v katerem je implementacija algoritma strojnega učenja. Za npr. algoritem Naivni Bayes je pot podana kot *class = "weka.classifiers.bayes.NaiveBayes"*. V eksperimentu je definiran tip eksperimenta in metoda validacije. Tako npr. izberemo navzkrižno validacijo z 10 pregibi. V opisu eksperimenta sledi definicija poti do datoteke, kamor se naj shranijo rezultati posameznega eksperimenta. Opis formata datotek s podatki je podan v [4], [72] in [2]. Posamezen eksperiment lahko predpisuje testiranje več različnih algoritmov strojnega učenja. V našem primeru smo z vsakim eksperimentom preizkusili 49 algoritmov strojnega učenja.

Eksperimente smo poganjali iz ukazne vrstice s sledečim ukazom:

```
\java weka.experiment.Experiment -l c:\experiment.xml -r
```

Izvajanje posameznega eksperimenta na zmogljivem osebnem računalniku je trajalo razmeroma dolgo časa. Eksperiment, s katerim smo preverili 49 metod strojnega učenja na osebnem računalniku s procesorjem Intel Core DUO 6420 s frekvenco delovanja 2,13 GHz, s 3,25 GB RAM, in operacijskim sistemom Windows XP SP2, je trajal v povprečju 20 minut. Pri tem velja poudariti, da je bilo potrebnih 20 minut za gradnjo modelov z 49 algoritmi strojnega učenja in preverjanje delovanja teh modelov. Ker smo za preverjanje uporabili metodo navzkrižne validacije z desetimi pregibi, je bilo potrebno vsak model zgraditi desetkrat. Ko imamo na voljo zgrajen in preizkušen model interpretacije, interpretiramo z njim neznani vzorec v zanemarljivem času.

Ker smo generirali veliko množico eksperimentov (352 za določanje stopenj jakosti aglutinacije), s katerimi smo podrobno preučili in preizkusili delovanje kombinacije različnih metod strojnega učenja s segmentacijskimi metodami in izračunom vektorjev lastnosti, smo izvajanje eksperimentov razdelili med več računalnikov. Posamezne eksperimente smo združili v pakete tako, da smo izdelali datoteke tipa .bat, s katerimi smo po vrsti poganjali posamezne eksperimente iz množice eksperimentov.

4.2.3 Zajem podatkov iz sistema za telekonzultacije v transfuzijski medicini

Ker smo sami zasnovali, razvili in izdelali sistem za telekonzultacije, imamo nadzor nad programsko kodo sistema in strukturo ter zasnovo podatkovnih baz, v katerih so shranjeni podatki o telekonzultacijskih sejah. Te podatke smo potrebovali za gradnjo učne in testne množice, ki smo ju potrebovali za gradnjo modelov interpretacije predtransfuzijskih preiskav. Ob upoštevanju ustreznih pogojev za varstvo osebnih podatkov je bil iz delujočega sistema možen zajem delovnih podatkov v želeni obliki. Zajem podatkov smo sestavili iz:

- definicije potrebnih podatkov za izdelavo sistema za samodejno interpretacijo predtransfuzijskih preiskav,
- definicije podatkovnih struktur, ki te podatke vsebujejo,
- izdelave programskega modula v aplikaciji sistema za telekonzultacije ter zajema realnih diagnostičnih podatkov.

V končni izvedbi modula za samodejno interpretacijo se bo zajem podatkov in popravljanje modelov na podlagi teh podatkov vršil sproti med delovanjem sistema. V praksi delujoči sistem za telekonzultacije je zaradi varnosti zaprt in omogoča dostop samo iz vozlišč, ki sestavljajo transfuzijsko mrežo. Zaradi občutljivosti podatkov in zakonskih predpisov o varovanju le-teh razvijalci sistema za samodejno interpretacijo predtransfuzijskih preiskav nimamo neposrednega dostopa do delujočega sistema za telekonzultacije. Ker smo za razvoj modelov sistema za samodejno interpretacijo potrebovali realne diagnostične podatke in interpretirane rezultate, smo morali poskrbeti za zajem realnih diagnostičnih podatkov in pripadajočih interpretacij iz delujočega sistema. V ta namen smo razvili in v aplikacijo sistema za telekonzultacije vgradili modul, ki skrbi za zajem potrebnih podatkov iz delujočega sistema. Modul shrani podatke v datoteko, ki jo pooblaščeni operater posreduje nam. Ker so podatki v sistemu strukturirani hierarhično, smo za zajem in hrambo uporabili sistem, ki omogoča repliciranje teh podatkovnih struktur. Za izvedbo modula za hrambo podatkov smo uporabili odprto-kodni Java projekt Castor [77]. Castor omogoča preprosto izvedbo preslikave med Java objekti in XML [78] dokumenti. Izvedbo modula za zajem podatkov z uporabo sistema Castor smo izvedli v več korakih. V prvem koraku je bilo potrebno načrtovati podatkovno strukturo, ki vsebuje na pravi način strukturirane podatke. Podatkovno strukturo smo predstavili na sliki 4.4. Njena

definicija je podana z uporabo XML sheme. V posebni datoteki so definirani posamezni podatkovni objekti, njihovi tipi in medsebojne povezave. Nabor podatkovnih objektov osnovnih tipov je povezan v kompleksnejše objekte. Kompleksni objekti lahko vsebujejo tudi sezname objektov. Na tak način zgrajena hierarhična struktura je sposobna hraniti vse potrebne podatke. Rezultat načrtovanja podatkovne strukture je bila XML shema, shranjena v datoteki XSD. Vsebino datoteke smo predstavili na sliki 4.6. Načrtovanje podatkovne strukture in generacijo datoteke XSD smo izvedli s programskim orodjem Altova XMLSpy [79].

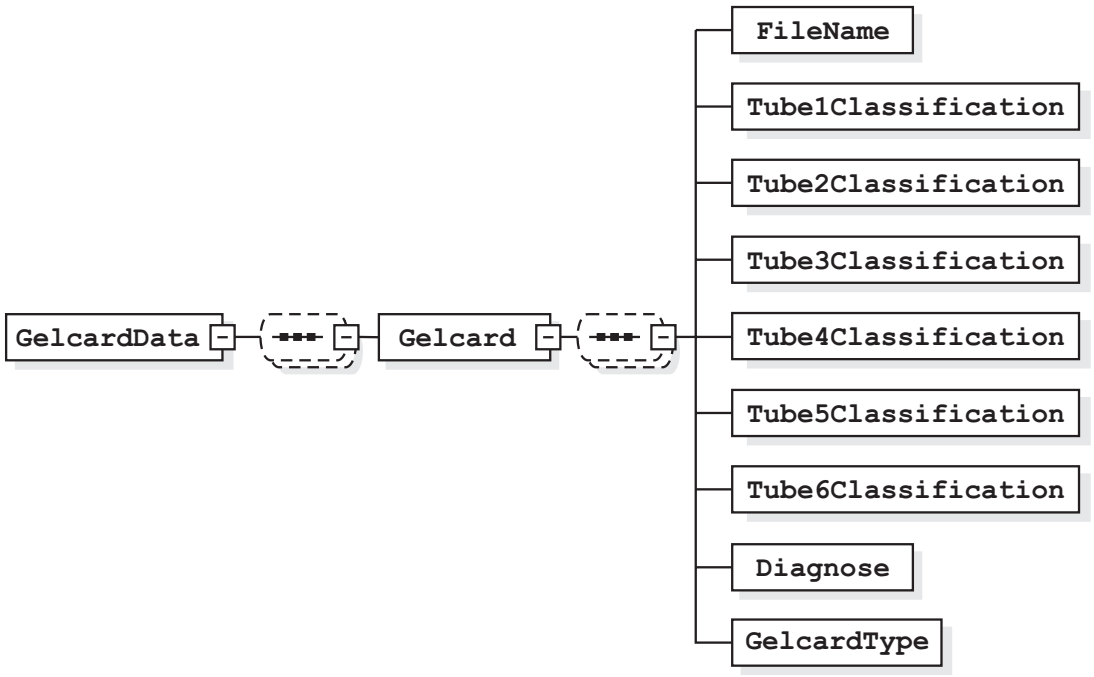
V našem primeru je osnovni podatkovni element gelska kartica `gelcard`. Ta podatkovni element vsebuje ime datoteke, ki vsebuje sliko obravnavane gelske kartice, podatke o stopnji jakosti aglutinacije vsake od šestih kolon te gelske kartice, podatke o interpretaciji preiskave in podatke o tipu gelske kartice, uporabljene za preiskavo.

XML dokument, ki ga je generiral modul za zajem podatkov iz sistema za telekonzultacije na podlagi te sheme, vsebuje podatke, potrebne za gradnjo učne in testne množice sistema. Osnovni element tega dokumenta je seznam elementov `gelcard`. Vsak od teh elementov vsebuje podatke o eni zaključeni preiskavi z uporabo ene gelske kartice. Poleg tega dokumenta smo potrebovali še vse datoteke, ki vsebujejo slike gelskih kartic. Oris podatkov, zajetih iz sistema, je predstavljen na sliki 4.5.

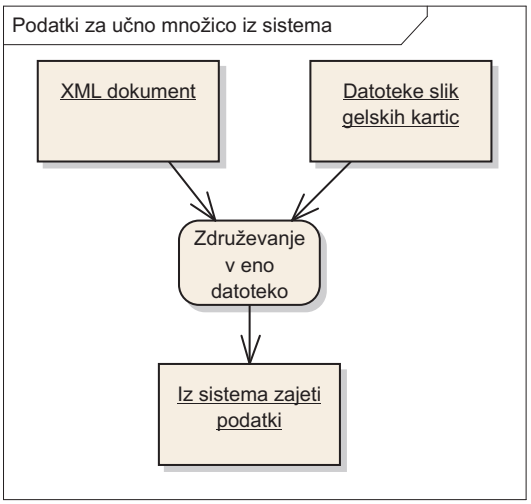
Ko smo končali proces načrtovanja podatkovne strukture, smo na podlagi datoteke XSD z orodjem Castor [77] zgenerirali javanske razrede, ki podatke hranijo, omogočajo njihovo dodajanje, brisanje in iskanje. Ti razredi omogočajo tudi izvoz podatkovne strukture v XML dokument.

V nadaljevanju je bilo potrebno javanskim razredom dodati logiko, ki podatke pridobi in jih na koncu postopka pridobivanja tudi shrani. Opisana celota predstavlja modul za zajem podatkov. Ta modul smo integrirali v aplikacijo za telekonzultacije na odjemalcu.

Ob zagonu modula na odjemalcu, povezanem v sistem za telekonzultacije, je modul od strežnika zahteval pregled vseh zaključenih telekonzultacijskih sej. Strežnik je odgovoril z vsemi znanimi podatki o sejah. Za razvoj sistema za samodejno interpretacijo rezultatov so zanimivi podatki slike gelskih kartic, tip preiskave, klasifikacija posameznih kolon glede na stopnjo jakosti aglutinacije eritrocitov in končna interpretacija preiskave. Slike gelskih kartic je modul shranil v posamezne datoteke. Imena teh datotek je modul dodal v podatkovno strukturo k ostalim podatkom seje. Ko je bil postopek zajema podatkov iz strežnika končan, je modul iz celotne strukture ustvaril XML dokument, ki ga je prav tako shranil v datoteko. Vse datoteke slik in XML dokument je modul za lažje manipuliranje



Slika 4.4: Z orodjem XMLSPY načrtovana podatkovna struktura za izvoz podatkov.



Slika 4.5: Iz sistema za telekonzultacije zajeti skupini podatkov. Podatki shranjeni v XML datoteki in JPEG slike gelskih kartic. Vsi podatki so z namenom preprostega prenašanja zgoščeni v eno zip datoteko.

```
<?xml version="1.0" encoding="UTF-8"?> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="Glecard_data">
        <xs:complexType>
            <xs:sequence minOccurs="0" maxOccurs="unbounded">
                <xs:element name="gelcard">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="file_name"/>
                            <xs:element name="tube1_classification"/>
                            <xs:element name="tube2_classification"/>
                            <xs:element name="tube3_classification"/>
                            <xs:element name="tube4_classification"/>
                            <xs:element name="tube5_classification"/>
                            <xs:element name="tube6_classification"/>
                            <xs:element name="diganose"/>
                            <xs:element name="gelcard_type"/>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

Slika 4.6: Vsebina datoteke XSD: z orodjem XMLSPY načrtovana podatkovna struktura za izvoz podatkov.

stisnil v eno samo datoteko .zip. Glejte sliko 4.5. To datoteko je operater shranil na prenosni medij in nam jo posredoval. To datoteko smo posneli na računalnik, kjer je potekal razvoj sistema za samodejno interpretacijo rezultatov predtransfuzijskih preiskav.

Javanske razrede, generirane na osnovi XSD sheme, ki opisuje podatkovno strukturo, smo uporabili za pretvarjanje podatkov v obliko, ki je primerna za obravnavo z algoritmi strojnega učenja na računalniku pri razvoju sistema za samodejno interpretacijo. Za pretvarjanje smo dodali logiko, ki je te podatke ustrezno pretvorila in shranila.

Postopkom zajema podatkov iz sistema je sledila izboljšava in registracija zajetih podatkov. Ker poteka samodejna interpretacija v dveh korakih – določanje jakosti aglutinacije (razvrščanje kolon) in določanje dokončne intepretacije, smo registracijo podatkov obravnavali ločeno za vsak korak.

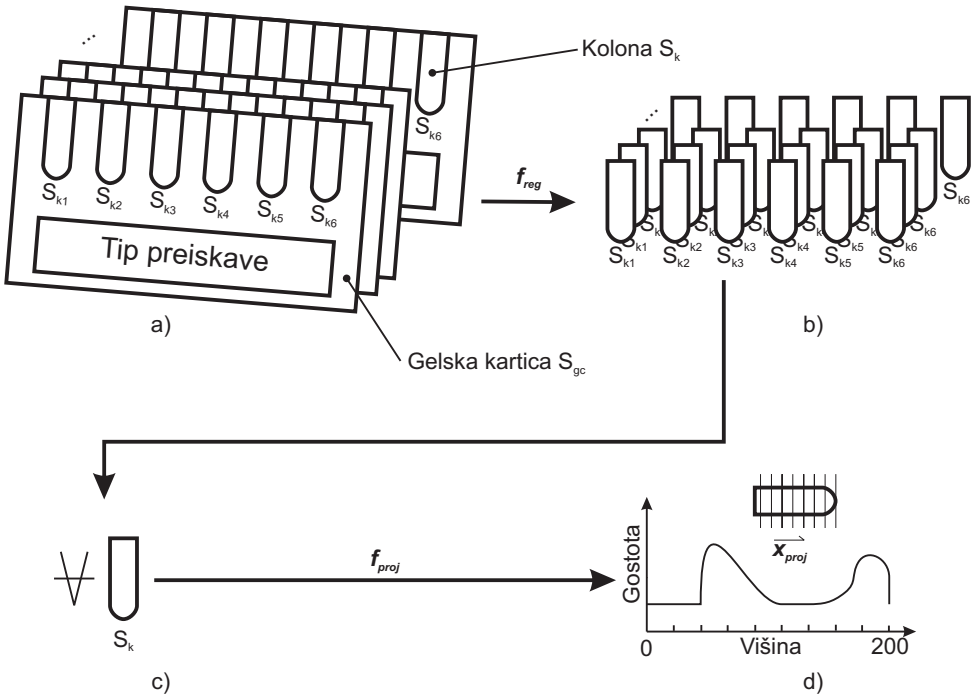
4.3 Določanje stopnje jakosti aglutinacije kolon

Določanje stopnje jakosti aglutinacije kolon je prvi korak v posnemanju interpretacije predtransfuzijskih preiskav, ki ga izvajajo specialisti transfuzijske medicine. Oba koraka smo ilustrirali na sliki 4.1. V tem koraku smo za vsako kolono obravnavane gelske kartice določili stopnjo jakosti aglutinacije eritrocitov. Vektor, sestavljen iz določenih stopenj jakosti aglutinacije v vsaki od šestih kolon gelske kartice, smo uporabili v drugem koraku, koraku dokončne interpretacije predtransfuzijske preiskave.

Za določanje stopenj jakosti aglutinacije v kolonah z metodami strojnega učenja smo zgradili model, ki modelira določanje stopnje jakosti aglutinacije. Kot smo že povedali, smo potrebovali za gradnjo modela z metodami strojnega učenja nabor podatkov, ki smo ga razdelili v učno in testno množico. Nabor podatkov je bil sestavljen iz slik kolon gelskih kartic z vzorci krvi in pripadajočimi stopnjami jakosti aglutinacije, ki so jih določili specialisti transfuzijske medicine. Podatke smo zajeli iz sistema za telekonzultacije v transfuzijski medicini, kot je opisano v podpoglavju 4.2.3.

Za uspešno delovanje algoritmov strojnega učenja smo morali podatke, zajete iz sistema primerno obdelati. Ker so podatki, namenjeni določanju stopnje jakosti aglutinacije kolon, slikovni podatki, je bilo potrebno te slikovne podatke registrirati. Postopek registracije je vse slikovne podatke transformiral v isti koordinatni sistem. Na ta način smo lahko iz slik učinkovito izluščili podatke, potrebne za nadaljnjo obdelavo [80].

V nadaljevanju postopka predobdelave smo iz podatkov izluščili lastnosti. To smo storili tako, da smo podatke ustrezno transformirali, izčistili in diskretizirali [39]. Slika

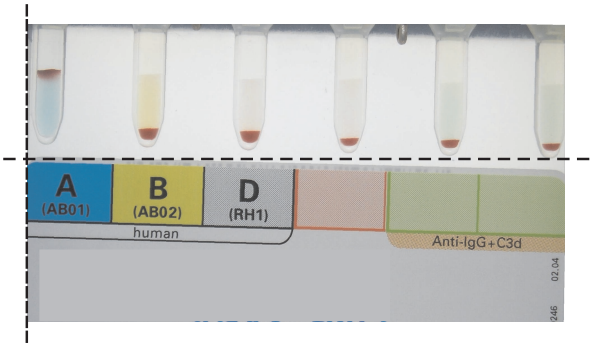


Slika 4.7: Postopek registracije slik gelskih kartic in transformacije le-teh v vektorje projekcije.

4.7 (a – d) prikazuje celoten postopek registracije vhodnih podatkov in njihovo preslikavo v vektor projekcij. Ko smo imeli na voljo obdelane podatke, zapisane v obliki vektorjev lastnosti, smo te podatke obdelali z metodami strojnega učenja. V nadaljnjem besedilu smo opisali postopke registracije podatkov, pridobivanje vektorjev lastnosti in njihovo obdelavo z metodami strojnega učenja.

4.3.1 Registracija slikovnih podatkov

Registracija vhodnih slikovnih podatkov je v našem primeru predstavljala razpoznavanje področij posameznih kolon na slikah gelskih kartic in pripravo le-teh v obliko, primerno za nadaljnjo obdelavo. Pri registraciji smo reševali dva problema. Prvi je bila identifikacija rotacije slik gelskih kartic, drugi pa je bilo določanje področij posameznih kolon na slikah gelskih kartic.



Slika 4.8: Ilustracija napake pri zajemu slike gelske kartice. Umetno pretirano rotirana fotografija gelske kartice.

Sliko vsake gelske kartice smo označili z S_{gc} (Slika 4.7 – a). Postopek določanja področij posameznih kolon pa kot $f_{reg}(S_{gc})$ (Slika 4.7). Preslikava $f_{reg}(S_{gc})$ je iz slik gelskih kartic izločila slike posameznih kolon (Slika 4.7 – b). Slike posameznih kolon smo označili z S_k (Slika 4.7 – c).

4.3.1.1 Identifikacija rotacije slik gelskih kartic z maksimiranjem dinamičnosti projekcije robov

Slike gelskih kartic se zajemajo s posebej za ta namen razvito strojno opremo, z napravo Gelscope32. Naprava je predstavljena na sliki 3.7.

Gelske kartice je mogoče v napravo Gelscope32 vstaviti postrani. Zato so zajete slike gelskih kartic lahko rotirane, kar predstavlja napako. Območje napake kota rotacije gelske kartice je nekaj kotnih stopinj v pozitivno in negativno smer. Slika 4.8 prikazuje primer napačno zajete slike gelske kartice. Napaka je vidna kot rotacija okoli osi, ki prebada največjo površino gelske kartice. Idealna slika gelske kartice je slika, na kateri je gelska kartica povsem vodoravna – stranice gelske kartice so vzporedne z osmi slike. Za odpravo rotacijske napake je potrebno za vsako obravnavano sliko ugotoviti kot rotacije glede na vodoravno lego in sliko ustrezno popraviti. Za ugotavljanje kota rotacije smo razvili metodo z maksimiranjem dinamičnosti projekcije robov. Metodo smo preizkusili in ugotovili, da dani problem rešuje zadovoljivo.

Metoda, ki smo jo prvič predstavili v objavi [81], temelji na dejstvu, da je večina elementov na slikah gelskih kartic preprosta in omejena z ostro definirani horizontalnimi in vertikalnimi robovi. Objekti na obravnavanih slikah so kolone in nalepke, na katerih

je precej horizontalnih in vertikalnih črt. Večina robov objektov na slikah je vzporedna z robovi gelske kartice. Z metodo iskanja kota odmika od idelanega položaja kartice smo vrteli sliko po vnaprej definiranih korakih znotraj pričakovanega intervala napake kota. Za vsako vrtenje smo izračunali parametre, ki so se spreminjali v odvisnosti od vzporednosti elementov na sliki z robovi slike. Te parametre smo reducirali v enoštevilsko vrednost. Ko je bila slika rotirana za tak kot, da je bila večina robov elementov slike vzporednih z robovi slike, je bil opazovani parameter največji. Z opazovanjem tega parametra znotraj pričakovanega intervala napake kota smo lahko ugotovili kot rotacije, s katerim je bila popačena slika gelske kartice. V nadaljevanju smo podrobneje opisali podani povzetek delovanja metode. Podani so posamezni koraki: iskanje robov v sliki, rotacija slike, izračun parametrov, ki govore o vzporednosti elementov na sliki z robovi slike, in iskanje najbolj vzporedne rotacije.

Sivinska slika gelske kartice vsebuje dovolj informacije za iskanje robov v tej sliki. Zato smo v postopku najprej pretvorili sliko gelske gartice v sivinsko sliko. V nadaljevanju pa smo sivinsko sliko zavrteli po definiranih korakih kota znotraj definiranega območja pričakovane maksimalne napake kota rotacije.

Za vse rotirane slike smo poiskali robove na tej sliki. Rezultat iskanja robov je bila robljena slika, na kateri so bili poudarjeni robovi. Primeri originalne in robljene slike, so predstavljeni na sliki 4.12. Postopke iskanja robov na sliki smo opisali v podpoglavju 4.3.1.1.1.

Vsako dobljeno robljeno sliko smo projicirali na navpično os in vodoravno os. Projekcijo na navpično os smo izvedli kot vsoto vrednosti vseh slikovnih elementov robljene slike po posameznih stolpcih slike. Projekcija na vodoravno os pa smo izvedli kot vsoto vrednosti vseh slikovnih elementov po posameznih vrsticah slike. Rezultat združenih projekcij na navpično in vodoravno os je vektor projekcije $v_{\vec{proj}}$. Komponente $v_{\vec{proj}}$ predstavljajo vsote po posameznih vrsticah in stolpcih.

Če opazujemo rob enega objekta na sliki in je ta rob vzporeden z navpično osjo, bo projekcija tega roba na vodoravno os v vektorju projekcije $v_{\vec{proj}}$ povzročila veliko konico. Če rob ni povsem vzporeden z osjo, bo njegova projekcija razmazana čez večje področje in bo posledično manj prispevala k razgibanosti vektorja projekcije.

Za ugotavljanje kota rotacije smo morali najprej ugotoviti, pri katerem kotu rotacije je vektor projekcije najbolj razgiban. Za vsak opazovani kot rotacije smo v ta namen izračunali skalarno vrednost $T_v = f_r(v_{\vec{proj}})$, ki je povezana z razgibanostjo tega vektorja. Za izračun razgibanosti vektorja projekcije smo uporabili preslikavo, imenovano *totalna*

variacija vektorja projekcije [82]. Preslikava je zapisana v enačbi 4.1. Za izračun preslikave totalne variacije smo najprej izračunali odvod vektorja $\vec{v_{proj}}$ in sešteli absolutne vrednosti posameznih komponent odvoda.

$$\begin{aligned} \vec{v_{proj}} &= (v_1, v_2, \ldots, v_n) \\ T_v &= \sum_{i=1}^{n-1} |v_{i+1} - v_i| \end{aligned} \tag{4.1}$$

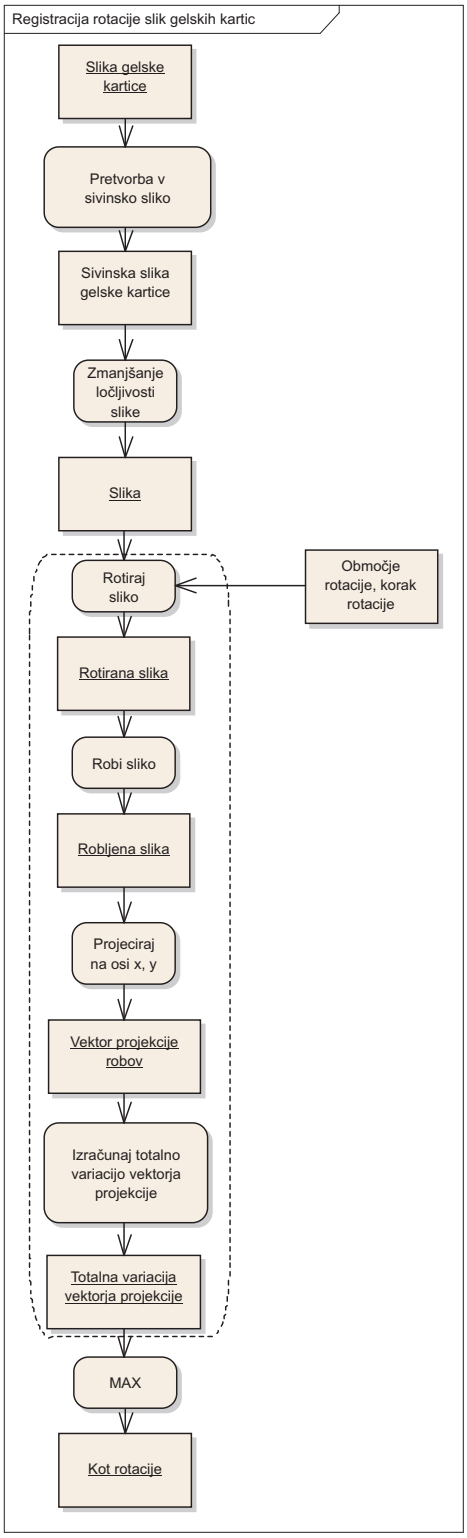
Dobljene rezultate T_v za vsak kot rotacije smo primerjali med sabo in poiskali največjega. Na sliki 4.10 smo predstavili velikost totalne variacije vektorja projekcije v odvisnosti od kota rotacije slike gelske kartice za eno od obravnavanih slik gelskih kartic.

Celoten algoritem registracije rotacije slik gelskih kartic je ilustriran na sliki 4.9.

4.3.1.1.1 Robljenje slik Za delovanje metode z maksimiranjem dinamičnosti projekcije robov smo potrebovali algoritem za robljenje slik. Robovi v slikah so področja z velikimi kontrasti v intenziteti – skoki v intenziteti med posameznimi sosednjimi slikovnimi elementi. S postopki zaznavanja robov v slikah lahko zelo zmanjšamo količino potrebnih podatkov in ohranimo pomembne strukturne lastnosti na slikah. Osnovne metode za robljenje se v grobem delijo v dve veliki skupini: gradientne in Laplaceove [83]. V nadaljevanju teksta smo predstavili obe metodi.

Gradientne metode zaznavajo robove na osnovi opazovanja vrednosti maksimumov in minimumov odvodov prvega reda obravnavane slike. Tipičen primer zaznavanja robov z gradientno metodo je zaznavanje robov z uporabo Sobelovega operatorja [83]. Z uporabo Sobelovega operatorja izračunamo dvodimenzionalni prostorski gradient obravnavane slike. Za izračun gradienta se uporabi par matrik velikosti 3×3, navedenih v enačbi 4.2. Ti matriki predstavljata konvulucijsko masko. Za oceno gradienta v smeri x se uporabi matriko Gr_x, za oceno gradienta v smeri y pa Gr_y. Velikost gradienta se izračuna z uporabo enačbe 4.3. Približna ocena se lahko izračuna tudi z uporabo enačbe 4.4.

$$\underset{Gr_x}{\begin{pmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{pmatrix}} \quad \underset{Gr_y}{\begin{pmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}} \tag{4.2}$$



Slika 4.9: Algoritem registracije rotacije slik gelskih kartic.

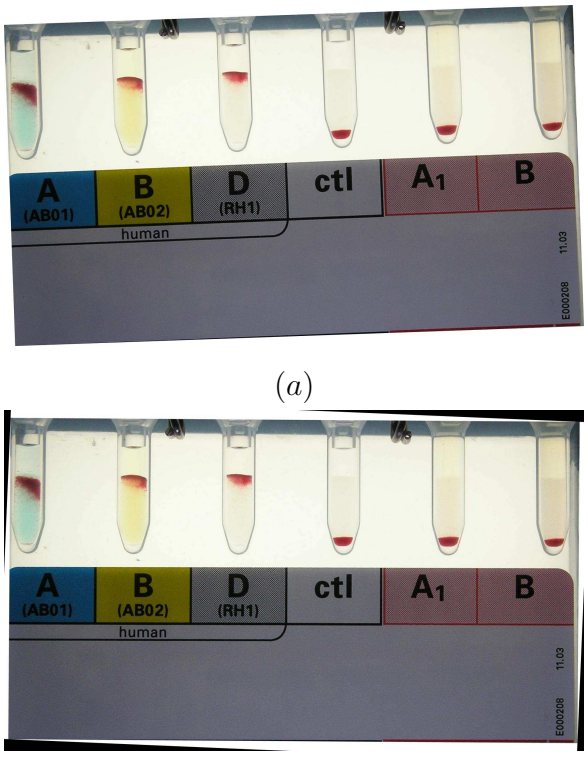
Slika 4.10: Seštevki absolutnih vrednosti odvodov projekcij – totalne variacije za posamezne kote rotacije od -5° do +5°. Maksimalna totalna variacija projekcij za obravnavano sliko je pri kotu -2,4°.

$$|Gr| = \sqrt{{Gr_x}^2 + {Gr_y}^2} \tag{4.3}$$

$$|Gr| = |Gr_x| + |Gr_y| \tag{4.4}$$

Laplaceova metoda zaznava robove z opazovanjem prehajanja vrednosti drugega reda odvodov slike preko vrednosti 0. Konvulucijska matrika, uporabljena pri metodi, je dimenzije 5×5 in je navedena v enačbi 4.5. Algoritem, ki uporablja Laplaceovo metodo, je zelo občutljiv na šum v sliki.

$$\begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix} \tag{4.5}$$



(a)

(b)

Slika 4.11: Rotirana slika (a) in njena popravljena verzija (b).

Cannyeva metoda zaznavanja robov velja za optimalen način zaznavanja robov. [84][85]. Zasnovana je bila na podlagi sledečih zahtev:

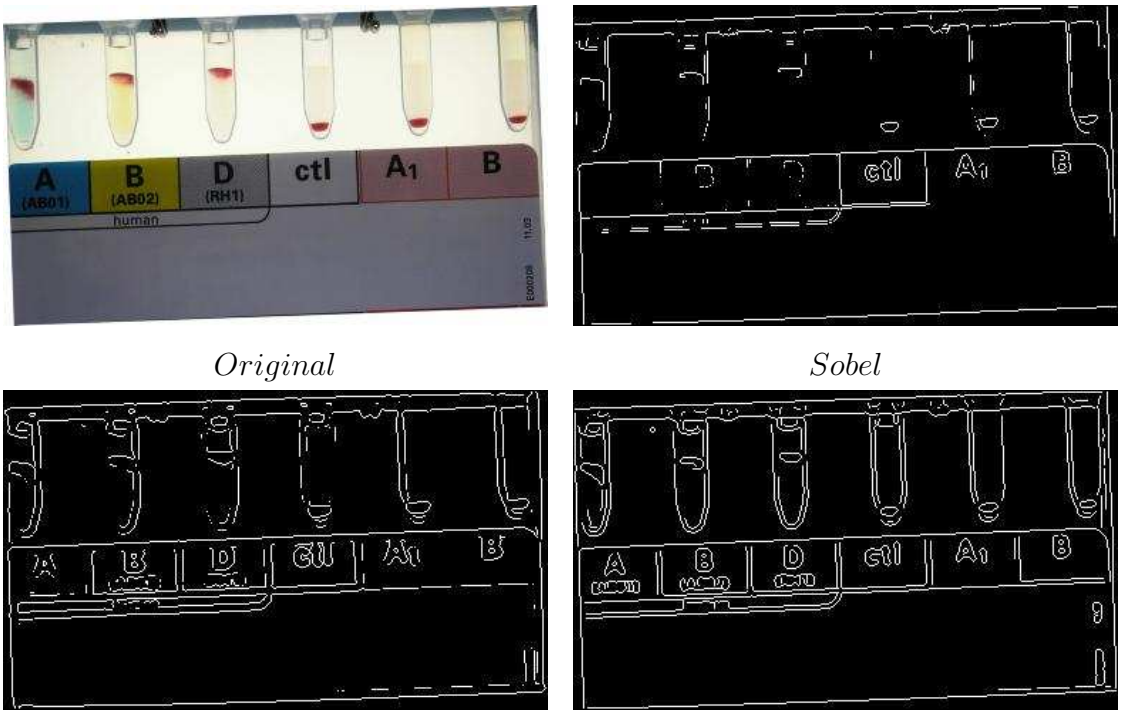
- Od algoritma za zaznavanje robov je zahtevan nizek nivo napak. Nizek nivo napak pomeni, da algoritem najde vse robove v sliki in da se ne odziva na objekte, ki niso robovi.
- Od algoritma je zahtevana dobra definicija lokacije robov. To pomeni, da je razdalja med slikovnim elementom, ki ga je predlagal za rob algoritem, in med resničnim robom, minimalna.
- Od algoritma je zahtevan le enkraten odziv za vsak rob na sliki.

Algoritem Cannyeve metode ugodi vsem tem zahtevam. Deluje na sledeč način: Najprej zgladi sliko in na ta način izloči šum. Glajenje slike je izvedeno z uporabo Gausovega filtriranja. Algoritem v nadaljevanju izračuna gradient slike. Na podlagi izračunanega gradienta označi področja slike, ki vsebujejo robove. Gradient slike izračuna algoritem z uporabo Sobelovega operatorja za obe smeri na sliki. Sobelov operator je opisan v enačbah 4.2 in 4.3. V nadaljevanju pregleda dobljena področja in potlači slikovne elemente, ki nimajo maksimalne vrednosti. Nad dobljenimi podatki izvede operacijo histereze. Operacija histereze uporablja dve mejni vrednosti. Če je vrednost obravnavanega slikovnega elementa pod spodnjo vrednostjo T_1, mu priredi vrednost 0. Če je nad zgornjo vrednostjo T_2, potem vrednost prepozna kot rob. Če pa je vrednost med T_1 in T_2, potem slikovnemu elementu algoritem priredi vrednost nič, če je le-ta osamljen. Če pa se kje v okolici slikovnega elementa nahaja slikovni element z vrednostjo večjo od T_2, potem algoritem tudi ta slikovni element prepozna kot rob.

Na sliki 4.12 je prikazana primerjava posameznih opisanih metod zaznavanja robov na sliki gelske kartice. Opazimo, da za naš problem najbolje deluje Cannyeva metoda.

4.3.1.2 Iskanje področij posameznih kolon na gelskih karticah

Za nadaljevanje postopka samodejne interpretacije (prvega koraka – ugotavljanje stopnje jakosti aglutinacije v kolonah) smo potrebovali slike posameznih kolon. Vsaka slika gelske kartice vsebuje 6 kolon. Na slikah so poleg za nas zanimivih kolon tudi nezanimivi objekti. Zato je bilo potrebno v nadaljevanju postopka registracije slik gelskih kartic natančno določiti področja posameznih kolon, jih izrezati in shraniti. Izkazalo se je, da je med posameznimi zajemi slik gelskih kartic z napravo Gelscope32 možno odstopanje položaja

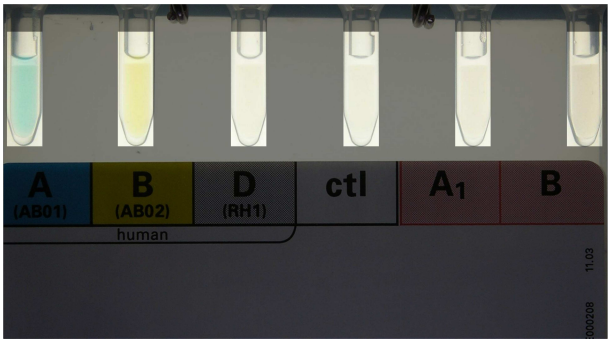


Slika 4.12: Primerjava metod za robljenje.

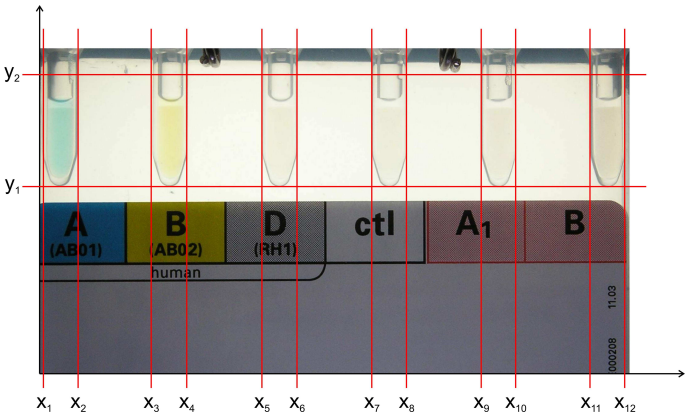
gelske kartice za nekaj milimetrov po oseh x in y. Vzrok za te razlike je uporaba različnih naprav Gelscope32. Naprave Gelscope32 smo namreč izdelali v mali seriji. Zaradi uporabljene prototipne tehnologije in ročnega načina izdelave naprave Gelscope32 mehansko niso povsem enake. Ker so posamezne kolone široke po 3 milimetre, predstavljajo omenjena odstopanja oviro za uporabo izločanja kolon z metodo fiksno določenih področij. Torej je bilo potrebno področja kolon določiti za vsako sliko gelske kartice posebej.

Ker smo na slikah gelskih kartic že odpravili rotacijo, je bilo potrebno določiti pravokotnike, ki vsebujejo slike posameznih kolon. Ti pravokotniki imajo stranice vzporedne z robovi slike. Na sliki 4.13 smo ilustrirali področje na sliki gelske kartice, ki nas je zanimalo. Pravokotnike, ki opisujejo to področje, smo definirali z dvema vektorjema. V prvem so koordinate stranic pravokotnikov na osi x, v drugem pa koordinate stranic na osi y. Ker se vse kolone začnejo na isti višini, sta v vektorju za os y le dve vrednosti.

Postopek iskanja pravokotnikov, ki opisujejo področje zanimanja, smo sestavili iz dveh korakov. V prvem smo določili lokacije kolon na osi x, v drugem lokacije na osi y. Kolone je bilo razmeroma preprosto najti, ker so preprostih oblik na skoraj povsem uniformnem ozadju. V idelanem primeru bi bilo za določanje lokacij kolon dovolj, da bi na sliki izbrali



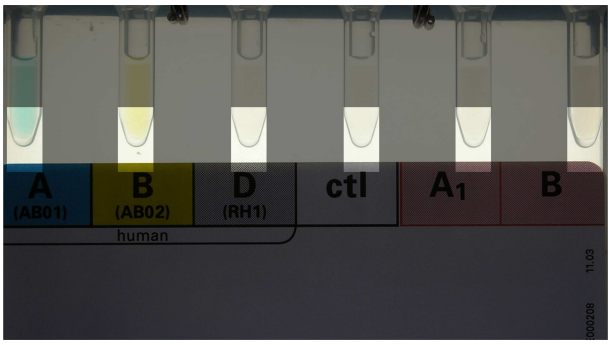
Slika 4.13: Področje zanimanja na gelski kartici – področje kolon.



Slika 4.14: Določanje področij kolon.

Slika 4.15: Opazovani pas slike za določanje lokacije kolon na osi x je izbran tako, da zagotovo vsebuje slike kolon.

eno horizontalno vrstico in v njej poiskali skoke, ki predstavljajo robove kolon. Zaradi šuma in morebitnih madežev smo postopek določanja lokacije kolon izboljšali tako, da smo za izračun lokacije robov izbrali več vrstic, ki smo jih povprečili. Ker smo poznali približno lokacijo kolon na sliki, smo najprej iz te izrezali pas, ki je zagotovo vseboval kolone. Pas smo izbrali tako, da je prerezal epruvete približno na sredini. Pas smo določili empirično z opazovanjem večjega nabora slik gelskih kartic. Glejte sliko 4.15. Vse stolpce izrezanega pasu smo povprečili in dobili en vektor. V nadaljevanju smo iskali robove kolon iz tega vektorja. Robove smo poiskali tako, da smo dani vektor najprej odvajali. Vsak rob na sliki se je v odvodu manifestiral kot konica. Ker je stena kolon na sliki debelejša kot en slikovni element, sta se za vsako kolono na vsaki strani kolone pojavila dva robova. Za ugotavljanje področja kolone je potrebno izbrati zunanjega. To smo storili tako, da smo se s kazalcem postavili na sredino med dve koloni. Lokacijo kazalca smo premikali v levo ali v desno in se približevali posameznim kolonam z leve ali desne strani. Ko smo naleteli na konico (vrednost v izbrani točki je za izbrani faktor večja od povprečne vrednosti celega vektorja), smo si njeno lokacijo zabeležili – našli smo zunanji rob kolone. Ta postopek ni našel levega roba skrajno leve kolone in desnega roba skrajno desne kolone. Položaj teh dveh robov smo določili na sledeči način: Iz predhodno določenih robov 4 kolon na sredini smo izračunali povprečno širino kolone na opazovani gelski kartici. Na gelskih karticah so namreč vse kolone enako široke. Ker je algoritem poznal notranje robove skrajnih kolon (desni rob leve kolone in levi rob desne kolone), smo lahko zunanja robova (levi rob leve kolone in desni rob desne kolone) določili tako, da smo lokaciji desnega roba leve kolone odšteli širino kolone, levemu robu desne kolone pa to širino prišteli. Iz lokacij zunanjih robov kolon smo sestavili vektor, ki je opisoval lokacije kolon na osi x.



Slika 4.16: Opazovana področja slike za določanje roba y so izbrana tako, da zagotovo vsebujejo spodnje robove kolon. Obravnavana so le področja predhodno določenih položajev kolon na osi x.

Lokacije kolon na osi y smo določili na podoben način kot lokacije na osi y. Iskali smo le spodnji del kolon. Iz slike smo izrezali pas, ki je zagotovo vseboval spodnji rob kolon. Pas smo izrezali le iz tistih področij slike, ki so bila določena kot kolone in ne iz celega področja slike. Pas smo določili empirično z opazovanjem večjega nabora slik gelskih kartic. Glejte sliko 4.16. V dobljenem pasu smo sešteli posamezne vrstice in dobili vektor. Področje med spodnjimi robovi kolon in med robom nalepke na gelski kartici ne vsebuje nobenega objekta in je zato bilo najsvetlejše. Potrebno je bilo poiskati zgornji rob tega področja. Rob področja smo določili z upragovljanjem. Prag smo določili empirično z opazovanjem večjega nabora slik gelskih kartic. Zgornji rob kolone smo določili empirično s konstanto.

Vse vrednosti, določene iz slike gelske kartice, smo primerjali z mejnimi vrednostmi, ki smo jih določili empirično. Če so lokacije kolon odstopale od mejnih vrednosti, smo javili napako.

Ko smo določili področja kolon na sliki gelske kartice, smo iz slike izrezali slike kolon in vsako shranili v svojo datoteko.

4.3.2 Preslikava slik kolon v vektor porazdelitve eritrocitov

Postopku registracije slik je sledil postopek luščenja informacije v slikah kolon gelskih kartic, ki je bila potrebna za določanje stopnje jakosti aglutinacije. Luščenje smo sestavili iz dveh korakov. V prvem koraku smo segmentirali slike glede na slikovne elemente, ki predstavljajo aglutinate in na elemente, ki aglutinatov ne predstavljajo. V drugem

koraku luščenja smo iz segmentiranih slik izluščili informacijo o porazdelitvi eritrocitov po višini kolone. Porazdelitev eritrocitov po višini kolone namreč neposredno govori o stopnji jakosti aglutinacije v koloni. Glejte sliko 4.7 d. V nadaljevanju smo predstavili oba koraka luščenja podatkov.

4.3.2.1 Segmentacija eritrocitov v slikah kolon

Za preslikavo slike posamezne kolone gelske kartice v vektor gostote eritrocitov po višini smo morali iz slike izločiti in poudariti slikovne elemente, ki predstavljajo eritrocite. Le-ti so rdeče barve, ki pa je žal od primera do primera različna.

Vzroki za različno rdečo barvo so različne barve gela: prozorna, rumena, modra. Različne barve gela v kolonah smo predstavili na sliki 4.17. Na različno barvo vplivajo tudi različne gostote skupkov aglutinatov in s tem povezana prosojnost za svetlobo z zadnje strani gelske kartice. Na različno barvo vpliva tudi spremenljiva svetilnost uporabljenih osvetljevalnih belih LED v napravah Gelscope32. Po približno pol leta uporabe naprave Gelscope32 smo ugotovili, da se LED starajo in se jim spreminja spekter izsevane svetlobe. Težavo s staranjem LED smo odpravili v naslednji verziji naprave Gelscope80. V napravi Gelscope80 LED ne svetijo več ves čas, ko je naprava vključena, marveč samo po potrebi, in sicer takrat, ko naprava zajema sliko. S tem ukrepom je drastično skrajšan čas obratovanja LED, ki se je s 24 ur/dan skrajšal na nekaj minut/dan.

4.3.2.1.1 Segmentacijski algoritmi Potreben je bil razvoj metode, ki kar naj učinkoviteje loči slikovne elemente z iskano vsebino od preostanka slike. Za izločanje slikovnih elementov, ki predstavljajo eritrocite, smo razvili, uporabili in primerjali več segmentacijskih algoritmov. Vsi algoritmi delujejo v osnovnem prostoru slike [86] in so bili osnovani ne podlagi kombinacije posameznih komponent slikovnih elementov slik v različnih barvnih prostorih.

Slike gelskih kartic so predstavljene v barvnem prostoru RGB. Barvni prostor je abstrakten matematični model, ki opisuje način, na katerega so predstavljene posamezne barve kot kombinacija različnega števila številčnih vrednosti – komponent [87]. Tipično število komponent za posamezne prostore je tri ali štiri.

Slike so sestavljene iz množice slikovnih elementov, urejenih v matrike. Posamezno matriko, ki predstavlja sliko, smo označili z $\mathbf{S}$. Vsak element matrike predstavlja en slikovni element te slike. Število komponent elementa je določeno s številom kanalov, s katerimi je podana slika. Posamezni kanali slike so predstavljeni kot matrike skalarjev. Tako

npr. rdeči kanal slike $\mathbf{S}_{RGB}$ označimo z $\mathbf{S}^R$. Slika $\mathbf{S}_{RGB}$ v prostoru RGB je sestavljena iz rdečega, zelenega in modrega kanala, kar smo zapisali kot $\mathbf{S}_{RGB} = (\mathbf{S}^R,\ \mathbf{S}^G,\ \mathbf{S}^B)$.

Predstavljene segmentacijske algoritme smo razvili na empiričen način. Med sabo smo jih primerjali tako, da smo rezultate segmentacije eritrocitov primerjali vizualno in glede na učinkovitost razvrščanja. Segmentirali smo slike iz nabora učne množice. Vizualna metoda primerjave učinkovitosti je zajemala primerjavo originalnih slik kolon s segmentiraninimi. Kot kriterij za izbor metode smo uporabili najučinkovitejše ločevanje področja z eritrociti od področij brez eritrocitov. Glejte sliko 4.17. Pri primerjavi na osnovi učinkovitosti razvrščanja smo z vsakim opazovanim segmentacijskim algoritmom izvedli segmentacijo, rezultate segmentacije pa smo uporabili kot vhod v postopke strojnega učenja. Kriterij za izbor je bilo najučinkovitejše razvrščanje. Kot kriterij za učinkovitost razvrščanja smo pri vsakem obravnavanem algoritmu opazovali delež pravilno razvrščenih vzorcev. Deleže smo izračunali iz matrik pravilno in napačno razvrščenih.

V nadaljevanju smo opisali posamezne razvite segmentacijske algoritme. Razvili smo 11 metod. Povzetek metod smo podali v tabeli 5.3. Rezultate delovanja algoritmov smo predstavili v podpoglavju 5.3.

R. Od vsake slike kolon gelskih kartic smo za izločanje in poudarjanje iskanih slikovnih elementov uporabili le rdeči kanal.

$$\mathbf{S} \longmapsto \mathbf{S}^R. \tag{4.6}$$

Sivinska slika. Iz osnovne slike $\mathbf{S}_{RGB}$ smo za izločanje in poudarjanje iskanih slikovnih elementov uporabili izračunano sivinsko sliko – vse kanale slike smo združili v enega.

$$\mathbf{S} \longmapsto \frac{1}{3}\left(\mathbf{S}^R + \mathbf{S}^G + \mathbf{S}^B\right). \tag{4.7}$$

Rdeča krominančna komponenta prostora YCbCr. Za izločanje in poudarjanje iskanih slikovnih elementov smo uporabili rdečo krominančno komponento $\mathbf{S}^{Cr}$ slike $\mathbf{S}_{YCbCr}$. Sliko $\mathbf{S}_{YCbCr}$ dobimo iz slike $\mathbf{S}_{RGB}$ s preslikavo iz prostora RGB v prostor YCbCr.

$$\mathbf{S} \longmapsto \mathbf{S}^{Cr}. \tag{4.8}$$

Iz prostora RGB v YCbCr smo uporabili preiskavo povzeto po viru [88]:

$$\mathbf{S}_{RGB} = \left(\mathbf{S}^R,\ \mathbf{S}^G,\ \mathbf{S}^B\right) \longmapsto \mathbf{S}_{YCbCr} = \left(\mathbf{S}^Y,\ \mathbf{S}^{Cb},\ \mathbf{S}^{Cr}\right). \tag{4.9}$$

Posamezne komponente preslikave 4.9 so podane kot:

$$\begin{aligned}
\mathbf{S}^{Y} &\longmapsto K_r\mathbf{S}^{R} + (1 - K_r - K_b)\mathbf{S}^{G} + K_b\mathbf{S}^{B}, \\
\mathbf{S}^{Cb} &\longmapsto \frac{1}{2(1 - K_b)}\left(\mathbf{S}^{B} - \mathbf{S}^{Y}\right), \\
\mathbf{S}^{Cr} &\longmapsto \frac{1}{2(1 - K_r)}\left(\mathbf{S}^{R} - \mathbf{S}^{Y}\right), \\
K_b &= 0,114, \\
K_r &= 0,299.
\end{aligned} \tag{4.10}$$

Cr komponenta preslikave je predstavljena na sliki 4.17 (c). Cr kompomenta slike kolone gelske kartice ima na področju, ki predstavlja eritrocite dokaj visoke, od nič veliko večje vrednosti. Ostala področja opazovane komponente imajo vrednost skoraj nič v primeru, ko opazovana kolona vsebuje brezbarvni gel in reagente. Glejte sliko 4.17 (c), *Prozoren*. V primeru, ko je opazovana kolona vsebovala modro obarvan gel, so področja, ki niso predstavljala eritrocitov, ampak gel, bila večja od nič. Glejte sliko 4.17 (c), *Moder*.

R-G-B. Za izločanje iskanih slikovnih elementov smo uporabili sledeči postopek: Najprej smo področja poizkusili izločiti z metodo odštevanja barvnih komponent v RGB barvnem prostoru. Od rdeče komponente smo odšteli zeleni in modri kanal.

$$\mathbf{S} \longmapsto \left(\mathbf{S}^{R} - \mathbf{S}^{G} - \mathbf{S}^{B}\right). \tag{4.11}$$

Rezultat operacije smo kot sivinsko sliko predstavili na sliki 4.17. Slika 4.17 (a) predstavlja originalno sliko, slika 4.17 (b) pa rezultat preslikave. Rezultat operacije ima na področjih, ki predstavljajo področja eritrocitov dokaj visoke od nič različne vrednosti. Vmesna področja imajo vrednosti skoraj enake nič. S to metodo smo zanesljivo določili področja, ki vsebujejo eritrocite, vendar je metoda poleg teh področij vključila v rezultat tudi nezaželena področja, predvsem robove kolon.

1-R. Od vsake slike kolon gelskih kartic smo za izločanje in poudarjanje iskanih slikovnih elementov uporabili le inverzni rdeči kanal. Vrednosti vseh kanalov slike se nahajajo v intervalu [0..1]

$$\mathbf{S} \longmapsto 1 - \mathbf{S}^{R}. \tag{4.12}$$

Kombinacija upragovljenih metod R-G-B in Cr. Uporaba samo ene metode od dosedaj opisanih ni dala zadovoljivih rezultatov, ker se med različnimi slikami precej spreminja tako barva slikovnih elementov, ki predstavljajo kri, kot tudi slikovnih elementov, ki eritrocitov ne predstavljajo.

Na rezultatih z obema metodama segmentiranih slik so dobro določena področja, ki vsebujejo eritrocite. Na obeh rezultatih pa so prisotne tudi napake. Napake so od nič različna področja rezultatov, na delih, ki ne vsebujejo eritrocitov. Ker so področja napak različna, področja iskanega pa ista, lahko iskano področje določimo tako, da izračunamo presek obeh rezultatov. Presek smo izračunali tako, da smo rezultata med sabo množili. Pred množenjem smo oba rezultata še upragovili in se s tem znebili vrednosti, ki so se od nič malo razlikovale. Rezultat smo predstavili na sliki 4.17 (d).

$$\begin{aligned}
\mathbf{S}_a &\longmapsto \begin{cases} \mathbf{S}^R - \mathbf{S}^G - \mathbf{S}^B, & \mathbf{S}^R - \mathbf{S}^G - \mathbf{S}^B > 0,1 \max\left(\mathbf{S}^R - \mathbf{S}^G - \mathbf{S}^B\right); \\ 0, & \mathbf{S}^R - \mathbf{S}^G - \mathbf{S}^B < 0,1 \max\left(\mathbf{S}^R - \mathbf{S}^G - \mathbf{S}^B\right), \end{cases} \\
\mathbf{S}_b &\longmapsto \begin{cases} \mathbf{S}^{Cr}, & \mathbf{S}^{Cr} > 0,1 \max\left(\mathbf{S}^{Cr}\right); \\ 0, & \mathbf{S}^{Cr} < 0,1 \max\left(\mathbf{S}^{Cr}\right), \end{cases} \\
\mathbf{S} &\longmapsto \mathbf{S}_a\, \mathbf{S}_b.
\end{aligned} \tag{4.13}$$

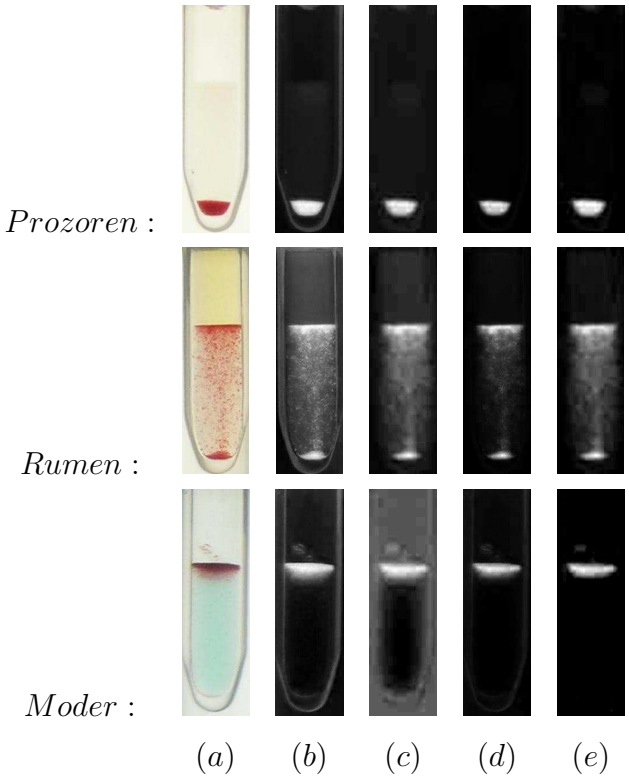
Nelinearno filtriranje 1. Z metodami, ki smo jih poimenovali s skupnim imenom nelinearno filtriranje, smo poizkušali z linearnim kombiniranjem absolutnih vrednosti razlik med posameznimi kanali izločiti slikovne elemente, ki vsebujejo krvne celice. Metode smo razvili empirično. Metode so se v postopku analize izkazale kot najučinkovitejše in so zadovoljivo reševale dani problem.

$$\mathbf{S} \longmapsto \left|\mathbf{S}^R - \mathbf{S}^B\right| + \left|\mathbf{S}^R - \mathbf{S}^G\right|. \tag{4.14}$$

Nelinearno filtriranje 2. Metodo nelinearnega filtriranja 1 smo izboljšali z dodatnim členom.

$$\mathbf{S} \longmapsto \left|\mathbf{S}^R - \mathbf{S}^G\right| + \left|\mathbf{S}^R - \mathbf{S}^B\right| - \left|\mathbf{S}^G - \mathbf{S}^B\right|. \tag{4.15}$$

Nelinearno filtriranje 3. Opazili smo, da metoda nelinearnega filtriranja 2 ne deluje zadovoljivo v primerih, ko so na sliki kolone eritrociti skoraj črne barve. Zato smo metodo razširili z zaznavanjem eritrocitov črne barve. Slike kolon s črnimi eritrociti



Slika 4.17: Vizualna primerjava obetavnih metod segmentacije slik kolon. (a) – Slika kolone; (b) – Rezultat odštevanja komponent $\mathbf{S}^R - \mathbf{S}^G - \mathbf{S}^B$; (c) – $\mathbf{S}^{Cr}$ komponenta slike, preslikane v prostor $\mathbf{S}_{YCbCr}$; (d) – Kombinacija upragovljenih slik $\mathbf{S}^B$ in $\mathbf{S}^{Cr}$; (e) – $\left|\mathbf{S}^R - \mathbf{S}^G\right| + \left|\mathbf{S}^R - \mathbf{S}^B\right| - \left|\mathbf{S}^B - \mathbf{S}^G\right|$. Prikazani so primeri za različne barve gela – prozorna, rumena in modra.

smo zaznali s primerjavo razlike vrednosti posameznih barvnih komponent posameznih slikovnih elementov. Če se večina vrednosti slikovnih elementov ni bistveno razlikovala, je algoritem domneval, da obravnava sliko kolone s črnimi eritrociti. V tem primeru je algoritem rezultat segmentacije izračunal s preslikavo na kanalu $1 - \mathbf{S}^R$. Preslikava je slikovne elemente z upragovljanjem preslikala v vrednosti 1 ali 0. Prag smo določili eksperimentalno. Če algoritem ni zaznal črnih eritrocitov, je za segmentacijo uporabil metodo nelinearnega filtriranja 2.

$$\mathbf{S} \longmapsto \begin{cases} \left|\mathbf{S}^R - \mathbf{S}^G\right| + \left|\mathbf{S}^R - \mathbf{S}^B\right| - \left|\mathbf{S}^G - \mathbf{S}^B\right|, & \text{ni črna,} \\ 1 - \mathbf{S}^R, & \text{je črna.} \end{cases} \tag{4.16}$$

Nelinearno filtriranje 4. Najprej smo na slikah izvedli zaznavanje črne barve eritrocitov na isti način, kot je opisano v metodi nelinearnega filtriranja 3. Če smo zaznali črne eritrocite, smo sliko obravnavali na enak način, kot je opisano v metodi nelinearnega filtriranja 3 za primer črnih eritrocitov. V nasprotnem primeru pa smo za vsak slikovni element poiskali razliko $\left|\mathbf{S}^R - \mathbf{S}^G\right|$ in $\left|\mathbf{S}^R - \mathbf{S}^B\right|$ ter od večje odšteli $\left|\mathbf{S}^G - \mathbf{S}^B\right|$.

$$\mathbf{S} \longmapsto \begin{cases} max\left(\left|\mathbf{S}^R - \mathbf{S}^G\right|, \left|\mathbf{S}^R - \mathbf{S}^B\right|\right) - \left|\mathbf{S}^G - \mathbf{S}^B\right| - \left|\mathbf{S}^G - \mathbf{S}^B\right|, & \text{ni črna,} \\ 1 - \mathbf{S}^R, & \text{je črna.} \end{cases} \tag{4.17}$$

LAB a*. Za izločanje in poudarjanje iskanih slikovnih elementov smo uporabili $\mathbf{S}^{a*}$ komponento slike $\mathbf{S}_{RGB}$, preslikane v barvni prostor CIE L*a*b* [89]. Barvni prostor CIE L*a*b* je barvni prostor, s katerim najučinkoviteje opišemo barve, ki jih zaznava zdravo človeško oko.

$$\mathbf{S} \longmapsto \mathbf{S}^{a*}. \tag{4.18}$$

Prostor LAB je osnovan na komplementarnih barvnih komponentah [90] z dimenzijami $\mathbf{S}^L$ za luminanco, in $\mathbf{S}^a$ in $\mathbf{S}^b$ za komplementarni barvi. Komplementarne barve so definirane glede na odziv človeškega očesa. Komponenta $\mathbf{S}^{a*}$ predstavlja razliko med zeleno in škrlatno, pri čemer negativna vrednost predstavlja zeleno, $\mathbf{S}^{b*}$ pa razliko med modro in rumeno, pri čemer negativna predstavlja modro.

Navedbe razvitih in uporabljenih postopkov segnemtacije so zbrane v tabeli 5.3.

4.3.2.2 Preslikava porazdelitve aglutinatov v vektor porazdelitve

Ker poznamo postopek določanja stopnje jakosti aglutinacije, ki ga izvajajo specialisti, smo lahko podatke ustrezno obdelali in zadržali le informacijo, ki je bila pomembna za določanje stopnje jakosti aglutinacije [39]. Za določanje stopnje jakosti aglutinacije je pomembna le porazdelitev eritrocitov po višini gela v koloni [12]. Ob upoštevanju tega dejstva, je projekcija vseh vrednosti slikovnih elementov posamezne kolone na os y primerna metoda luščenja informacije, ki govori o stopnji jakosti aglutinacije v posamezni koloni. Rezultat projekcije je vektor dimenzije n. Dimenzija n je enaka višini slike kolone izraženi v slikovnih elementih – tipično 200 (Slika 4.7 – d). Ta vektor smo imenovali vektor projekcije na y in ga označili z $x_{\vec{proj}}$. Preslikavo iz slike S_k v vektor projekcije $v_{\vec{proj}}$ smo označili kot $f_{proj}(S_k)$. Vsaka gelska kartica vsebuje šest kolon, zato smo iz vsake gelske kartice dobili šest vektorjev projekcij $x_{\vec{proj}}$.

4.3.3 Izračun vektorja lastnosti

Pri izbiri lastnosti je bilo potrebno izbrati le tiste podatke, ki so pomembni za razvrščanje. Ker smo vedeli, da je za določanje stopnje jakosti aglutinacije pomembna porazdelitev aglutinatov po višini kolone, smo v našem primeru to storili v določeni meri s preslikavo projekcije slik kolon gelskih kartic in s projekcijo vsebine po višini v en vektor. Dobljeni vektor projekcij za vsako kolono je dolg tipično 200 elementov.

Pri nadaljnji obdelavi smo iz vektorja projekcij izračunali nov vektor, ki smo ga imenovali vektor lastnosti $\vec{x} \in \mathcal{X}$. Vektor lastnosti naj bi s čim manj elementi čim bolje ločeval kolone glede na njihovo različno stopnjo jakosti aglutinacije, obenem pa naj bi čim bolje združeval kolone glede na isto stopnjo jakosti aglutinacije.

V literaturi smo zasledili več pristopov za izračun vektorjev lastnosti [2]. Ugotovitev, kateri od pristopov je za dani problem najučinkovitejši, ni bila trivialna. Do tega spoznanja smo se dokopali na empiričen način. V naši raziskavi smo za izračun vektorjev lastnosti iz vektorjev projekcij uporabili dva pristopa – metodo PCA in metodo zrnjenja. Za obe metodi smo izbrali število komponent vektorjev lastnosti tako, da smo dobili najboljše rezultate. Kot mero za ugotavljanje najboljših rezultatov smo uporabili učinkovitost delovanja modela, zgrajenega z metodami strojnega učenja z uporabo podatkov, ki so smo jih dobili z metodami za izračun vektorjev lastnosti.

4.3.3.1 Izračun vektorja lastnosti z metodo PCA

Za izvedbo preslikave vektorjev projekcij $x_{\vec{proj}}$ v vektorje lastnosti $\vec{x} \in \mathcal{X}$, ki smo jo označili z $f(x_{\vec{proj}})$, je primerna metoda, imenovana analiza glavnih komponent (ang. principle component analysis) – metoda PCA. Metoda PCA je primerna za iskanje vzorcev v naboru podatkov in za prikaz teh podatkov v taki obliki, da so poudarjene podobnosti in razlike med temi nabori podatkov [14]. Metoda PCA je uporabna pri analizi naborov podatkov velikih dimenzij, saj je možno z njeno uporabo izračunati preslikavo podatkov v prostor z manj dimenzijami na tak način, da se ohrani večina informacije vhodnih podatkov. Pogosto se jo uporablja v analizi biomedicinskih signalov in podatkov [91][92][50]. Rezultat obdelave podatkov z metodo PCA je linearna transformacija koordinatnega sistema, v katerem so predstavljeni obravnavani nabori podatkov v novem koordinatnem sistemu. V njem smeri posameznih osi sovpadajo s smermi, v katerih se obravnavani podatki med sabo najbolj razlikujejo. Smeri novega koordinatnega sistema so med sabo nekorelirane, dobljene komponente pa imajo maksimalno varianco med vsemi

nekoreliranimi linearnimi kombinacijam vhodnih podatkov [93].

Metodo PCA smo izvajali nad naborom obravnavanih vhodnih podatkov urejenimi v matrično obliko. Vhodni podatki metode PCA so bili v matriko urejeni vektorji $[\vec{x_{proj1}}, \vec{x_{proj2}}, \ldots, \vec{x_{projL}}]$, ki so vsebovali posamezne vzorce podatkov. L je število vzorcev učne množice. Rezultat metode PCA je bil sistem lastnih vektorjev te matrike. Z linearno transformacijo smo te podatke preslikali v nov prostor, ki je bil definiran z izračunanimi lastnimi vektorji. To je bila naša iskana preslikava $f(\vec{x_{proj}})$ za metodo PCA. Količina informacije, vsebovana v posamezni komponenti preslikave, je povezana z velikostjo lastne vrednosti komponenti pripadajočega lastnega vektorja. Ker je večina informacije tipično zajeta v prvih n, po velikosti lastnih vrednosti urejenih komponentah celotnega sistema lastnih vektorjev, smo pri nadaljnji obravnavi uporabili le-te komponente. Odločili smo se za uporabo n lastnih vektorjev, ki nosijo največ informacije, ostale pa smo zavrgli. Na ta način smo zmanjšali dimenzijo vhodnih podatkov in poenostavili nadaljnjo obravnavo. S postopkom optimizacije smo poiskali optimalno število uporabljenih lastnih vektorjev n. Postopek in rezultati optimizacije parametra so opisani v podpoglavju 5.4.2.

4.3.3.2 Izračun vektorja lastnosti z zrnjenjem – ZRNI

Ker je kriterij za določanje stopnje jakosti aglutinacije kolon odvisen od vertikalne porazdelitve eritrocitov v kolonah, stopenj jakosti aglutinacije pa je razmeroma malo, lahko informacijo, potrebno za določanje stopnje jakosti aglutinacije, obdržimo tudi v vektorjih, ki imajo močno zmanjšano število dimenzij. S postopkom ZRNI smo število dimenzij vektorja zmanjšali na n. Vektor porazdelitve smo razdelili na n enako dolgih odsekov. Za vsakega od odsekov smo izračunali povprečno vrednost komponent tega odseka. Iz teh povprečnih vrednosti smo sestavili nov vektor lastnosti. To je bila naša iskana preiskava $f(\vec{x_{proj}})$ za metodo ZRNI. Med sabo smo primerjali učinkovitost delovanja algoritmov strojnega učenja, ki so uporabljali te vektorje lastnosti v odvisnosti od na različne dolžine skrajšanih vektorjev lastnosti. Postopek izbire parametra n za metodo ZRNI smo opisali v podpoglavju 5.4.1.

4.3.4 Strojno učenje

Vektorje lastnosti, ki smo jih izračunali iz vektorjev projekcij segmentiranih slik kolon gelskih kartic, smo uporabili za gradnjo in testiranje modelov z metodami strojnega učenja. Metode smo izvajali z okoljem WEKA. Vektorje lastnosti smo zapisali v ARFF datoteke.

V ARFF datoteke smo dodali pripadajoče stopnje jakosti aglutinacije. Na ta način smo generirali učne in testne nabore podatkov. Generirali smo obširno množico naborov podatkov. Opis generirane množice naborov podatkov je opisan v poglavju 5. V nadaljevanju smo generirali eksperimente za okolje WEKA. Eksperimenti so zajemali test metod strojnega učenja, navedenih v tabeli 4.1. Za metodo testiranja smo izbrali postopke 10-pregibne navzkrižne validacije. Avtor literature [2] navaja, da s takim pristopom realno ocenimo delovanje modela, naučenega s sistemom strojnega učenja.

Grobo primerjavo delovanja metod strojnega učenja nad različnimi nabori podatkov iz množice naborov podatkov smo izvedli tako, da smo med sabo primerjali skupne deleže pravilno razvrščenih, doseženih v eksperimentih. Na podlagi grobe primerjave smo izbrali najboljše kandidate in jim namenili podrobnejšo obravnavo. Podrobnejša obravnava je zajemala primerjavo deležev pravilno razvrščenih posameznih razredov doseženih.

Dokončna odločitev za najprimernejšo kombinacijo metod je sledila na podlagi izračuna povprečnega kombiniranega deleža uspešnosti kombinacije kolon za določanje rezultata določene predtransfuzijske preiskave z gelsko metodo. Za izračun povprečnega deleža so bile izbrane vse kombinacije stopenj jakosti aglutinacije, ki dajo dokončen rezultat določene preiskave. Kombinacije, ki dajo dokončen rezultat preiskave, smo odčitali iz pravilnostne tabele, podane v literaturi [3].

4.4 Določanje dokončne interpretacije predtransfuzijske preiskave

Za vsako gelsko kartico določeni nabor stopenj jakosti aglutinacije kolon predstavlja vmesni rezultat interpretacije predtransfuzijskega testa. Za določitev končne interpretacije je potrebno ta vmesni rezultat dokončno interpretirati. Za vsako predtransfuzijsko preiskavo obstaja končen nabor možnih končnih interpretacij. Podrobnosti so opisane v podpoglavju 3.1.2.2.

Vsak vmesni rezultat interpretacije predtransfuzijske preiskave predstavimo kot n dimenzionalni vektor stopenj jakosti aglutinacije kolon. Vrednosti posameznih elementov tega vektorja so diskretne vrednosti. Vse možne vrednosti, ki jih lahko zavzamejo elementi tega vektorja, so predstavljene v podpoglavju 3.1.2.1. Ker v disertaciji obravnavamo le preiskave, ki zahtevajo za izvedbo le eno gelsko kartico, in ker ima gelska kartica 6 kolon, so ti vektorji največ 6–dimenzionalni.

Posplošitev metode določanja dokončnega rezultata preiskave je možna tudi na kompleksnejše predtransfuzijske preiskave. Te preiskave se izvajajo z več kot le eno gelsko kartico. Kolone teh gelskih kartic razvrstimo v vektor stopnje jakosti aglutinacije, ki ima dimenzijo $M \times 6$, pri čemer je M število v preiskavi uporabljenih gelskih kartic. Pri interpretaciji predtransfuzijske preiskave, ki zajema več kot eno gelsko kartico, se je potrebno dogovoriti za vrstni red posameznih gelskih kartic, uporabljenih v preiskavi, in ga upoštevati pri generaciji učne množice in pri strojni interpretaciji preiskav.

4.4.1 Zajem podatkov

Zajem podatkov, ki jih potrebujemo za razvoj modela za določanje dokončne interpretacije predtransfuzijskih preiskav, smo opisali v podpoglavju 4.2.3. Na sliki 4.6 vidimo, da smo imeli na voljo podatkovno strukturo, ki je vsebovala določene stopnje jakosti aglutinacije za v obravnavani gelski kartici vsebovane kolone, tip gelske kartice in rezultat preiskave. Določne stopnje jakosti aglutinacije za kolone vsebujejo elementi XML dokumenta z imeni *tube1_classification* do *tube6_classification*. Tip gelske kartice vsebuje element z imenom *gelcard_type*. Iz tipa gelske kartice smo ugotovili tip preiskave. Element z imenom *diganose* vsebuje dokončno interpretacijo predtransfuzijske preiskave. V programskem jeziku Java smo napisali program, ki iz dokumentov z opisano vsebino generira podatkovne nabore in jih zapiše v ARFF datoteke, ki jih uporabimo v okolju WEKA.

Ker pa je nabor podatkov, ki smo ga zajeli iz sistema za telekonzultacije, vseboval premalo podatkov za učinkovito gradnjo modela intepretacije za katerokoli od obravnavanih preiskav, smo se odločili, da nabor generiramo sami iz pravilnostne tabele. Za osnovne predtransfuzijske preiskave namreč obstajajo pravilnostne tabele. V pravilnostnih tabelah so navedene kombinacije stopenj jakosti aglutinacije za posamezne kolone in pripadajoče interpretacije preiskave.

Poleg osnovnih predtransfuzijskih preiskav obstajajo tudi kompleksnejše predtransfuzijske preiskave, katerih interpretacije ne obstajajo v pravilnostnih tabelah. Z eksperimentom, pri katerem smo podatke generirali iz pravilnostnih tabel, smo dokazali, da je mogoče za gradnjo modela intepretacije teh preiskav uporabiti metode strojnega učenja.

Odločili smo se za obravnavo preiskave: "Določanje krvne skupine na gelski kartici *humana*". Gelska kartica *humana* je posebna gelska kartica za določanje krvne skupine.

4.4.2 Strojno učenje

Za postopek gradnje modelov določanja dokončne interpretacije preiskav smo izvedli enak postopek kot pri gradnji modelov za določanje stopnje jakosti aglutinacije kolon. V okolju WEKA smo zgradili eksperiment, s katerim smo preizkusili učinkovitost modelov interpretacije, zgrajenih z algoritmi strojnega učenja, navedenimi v tabeli 4.1.

Grobo primerjavo delovanja metod strojnega učenja nad različnimi nabori podatkov iz množice naborov podatkov smo izvedli tako, da smo med sabo primerjali skupne deleže pravilno razvrščenih, doseženih v posameznih eksperimentih. Na podlagi grobe primerjave smo izbrali najboljše kandidate in jim namenili podrobnejšo obravnavo. Podrobnejša obravnava je zajemala primerjavo posameznih interpretacij in njihove dosežene deleže pravilno določenih.

Dokončno odločitev za najprimernejšo kombinacijo metod smo podali na podlagi izračuna povprečnega kombiniranega deleža uspešnosti, doseženega s kombinacijo deleža uspešnosti določanja vektorjev z določenimi stopnjami jakosti aglutinacije ter pripadajočo dokončno intepretacijo rezultata preiskave.

4.4.3 Ocenjevanje učinkovitosti modela dokončnega napovedovanja rezultatov

Validacijo modela interpretacije vektorjev z določenimi stopnjami jakosti aglutinacije v kolonah v končne rezultate posameznih preiskav smo izvedli na podoben način kot validacijo modela določanja stopnje jakosti aglutinacije v kolonah, ki smo jo predstavili v poglavju 3.3.4. Rezultate navzkrižne validacije posameznih modelov smo zapisali v matriko pravilnih in napačnih razvrstitev. Metodo navzkrižne validacije smo opisali v podpoglavju 3.3.4.3. Metodo navzkrižne validacije smo uporabili zato, ker je nabor podatkov, ki smo ga zbrali za gradnjo in testiranje modelov vseboval premalo vzorcev.

Iz matrike pravilnih in napačnih razvrstitev smo izračunali skupni delež pravilno razvrščenih, kot tudi delež pravilno razvrščenih za vsakega od posameznih razredov. Delež pravilno razvrščenih za vsakega od posameznih razredov smo v nadaljevanju kombinirali z deležem uspešnosti, ki smo ga izračunali za uporabljeni vektor z določenimi stopnjami jakosti aglutinacije.

4.5 Učinkovitost interpretacije predtransfuzijskih preiskav – kombinacija modela določanja stopnje jakosti aglutinacije in modela dokončne interpretacije

Rezultate meritev učinkovitosti samodejne interpretacije predtransfuzijskih preiskav smo izračunali iz združenih rezultatov meritev učinkovitosti modela za določanje stopnje jakosti aglutinacije in modela za dokončno interpretacijo preiskav. Rezultat smo predstavili kot delež uspešnosti za vsak posamezen možen rezultat, ki ga je napovedala kombinirana uporaba modela za določanje stopnje jakosti aglutinacije in modela za dokončno interpretacijo preiskav. Na podlagi najvišjega povprečnega rezultata deleža uspešnosti izbrane kombinacije postopkov smo se odločili za kombinacijo postopkov, ki je pripeljala do tega rezultata.

4.5.1 Delež uspešnosti

Za vsak dokončen strojno predlagan rezultat preiskave želimo poznati delež uspešnosti. Delež uspešnosti je normirana vrednost, ki pove, v kolikšni meri lahko zaupamo rezultatu modela strojne interpretacije. Z deležem uspešnosti smo opremili vsak samodejno interpretirani rezultat. Vrednost deleža uspešnosti se nahaja v intervalu med 0 in 1. Vrednost 0 pomeni, da je rezultat zagotovo napačen, vrednost 1 pa pomeni, da je rezultat zagotovo pravilen. Realno je pričakovati, da vrednost deleža uspešnosti ne bo nikoli zavzela vrednosti natančno 1 ali 0. Ob izračunu deleža uspešnosti smo privzeli, da je bila učna množica pravilna in smiselna. Izračun deleža uspešnosti za posamezno interpretacijo rezultata preiskave smo izvedli za vsak opravljen in končan postopek interpretacije. Izračunali smo ga iz podatkov o deležu pravilno določenih rezultatov (ang. precission) posameznih, v seriji uporabljenih modelov za vsak možen rezultat. Z uporabo deleža uspešnosti smo lahko med sabo primerjali posamezne rezultate interpretacije preiskav. Ker je pravilnost delovanja sistema odvisna od posameznih modelov in učnih množic, ki smo jih uporabili za gradnjo teh modelov, smo lahko s spremljanjem pravilnosti delovanja sistema identificirali slabo delujoče modele sistema. Modeli so lahko slabi zato, ker so bile za njihovo gradnjo uporabljene neprimerne metode strojnega učenja ali pa zato, ker je bila za njihovo gradnjo uporabljena neprimerna učna množica. Slabe modele lahko popravimo tako, da popravimo učno množico za gradnjo teh modelov in jih z metodami strojnega učenja ponovno zgradimo ali pa izberemo druge, učinkovitejše metode strojnega učenja. Na ta način

smo lahko s spreminjanjem posameznih parametrov sistema eksperimentalno izboljševali celotno delovanje sistema.

Za oceno deleža uspešnosti smo za prvi korak – določanje stopnje jakosti aglutinacije z modelom, dobljenim s postopkom strojnega učenja v posameznih kolonah za vsako od določenih stopenj jakosti aglutinacije, zapisali verjetnost, da je le-ta pravilna. Model je namreč ugotavljal različne stopnje jakosti aglutinacije različno dobro.

Model je stopnjo jakosti aglutinacije v koloni določil tako, da je kolono razvrstil v enega od razredov. Verjetnost, da je bila stopnja jakosti aglutinacije, ki jo je predlagal model, pravilna, smo izračunali iz matrike pravilnih in napačnih razvrstitev modela. Posamezne razrede so predstavljale posamezne stopnje jakosti aglutinacije. Verjetnost, da je bila določena stopnja jakosti aglutinacije prava, je bila enaka deležu pravilno razvrščenih vzorcev v določen razred med vsemi vzorci, razvrščenimi v ta razred. Ta delež smo izračunamo iz matrike pravilno in napačno razvrščenih, ki smo jo ocenili v postopku evaluacije modela sistema.

Ker se vsaka preiskava opravi z uporabo ene gelske kartice, na kateri je 6 kolon, smo morali v prvem koraku za vsako preiskavo šestkrat uporabiti model določanja stopnje jakosti aglutinacije. Rezultat prvega koraka, določanja stopnje jakosti aglutinacije, je bil vektor, ki smo ga označili z $\vec{ag}$, s šestimi komponentami, ki smo jih označili z a_n. Vsaka komponenta predstavlja določeno stopnjo jakosti aglutinacije za posamezno kolono gelske kartice. Vektor je pravilen, če so pravilne vse komponente tega vektorja. Privzeli smo, da so dogodki, da so stopnje jakosti aglutinacije v posameznih kolonah določene pravilno, neodvisni. Zato je ocenjena verjetnost, da je pravilen cel vektor, enaka produktu verjetnosti $p([a_n \in \mathrm{OK}])$, da so posamezne kolone razvrščene pravilno. Verjetnost, da je celoten vektor pravilen, smo označili s $p([\vec{ag} \in \mathrm{OK}])$.

Komponenta $\vec{ag}$	Verjetnost, da je pravilna
a_1	$p([a_1 \in \mathrm{OK}])$
a_2	$p([a_2 \in \mathrm{OK}])$
a_3	$p([a_3 \in \mathrm{OK}])$
a_4	$p([a_4 \in \mathrm{OK}])$
a_5	$p([a_5 \in \mathrm{OK}])$
a_6	$p([a_6 \in \mathrm{OK}])$

Tabela 4.2: Verjetnosti, da je posamezna komponenta vektorja pravilna.

$$p([\vec{ag} \in \mathrm{OK}]) = \Pi_{n=1}^{6} p([a_n \in \mathrm{OK}]) \tag{4.19}$$

V drugem koraku smo na podlagi vektorja stopenj jakosti aglutinacije, določenega v prvem koraku, z modelom dokončne interpretacije določili dokončno interpretacijo preiskave, ki smo jo označili z r. Ta model ni deloval idealno in je v določenih primerih naredil napako. Za vsako posamezno interpretacijo, ki jo je napovedal, smo iz matrike pravilno in napačno razvrščenih, ki smo jo izmerili v postopku validacije modela, ocenili verjetnost, da je rezultat r pravilen. Verjetnost ocene pravilnosti delovanja drugega koraka smo označili z $p([r_2 \in \mathrm{OK}])$. Ocenili smo jo kot delež pravilno razvrščenih vzorcev v določen razred.

Ker je končna interpretacija odvisna od pravilnosti prvega in drugega koraka, smo najslabšo oceno verjetnosti $p([r \in \mathrm{OK}])$, da je končna interpretacija pravilna, izračunali tako, da smo množili verjetnosti $p([\vec{ag} \in \mathrm{OK}])$ in $p([r_2 \in \mathrm{OK}])$

$$p([r \in \mathrm{OK}]) = p([\vec{ag} \in \mathrm{OK}])\; p([r_2 \in \mathrm{OK}]). \tag{4.20}$$

Pričakujemo, da je bila verjetnost, da je interpretacija pravilna, višja od ocenjene, saj smo upoštevali najslabši možni primer: že ena napačna vrednost v vektorju lastnosti $\vec{x}$ je pomenila napačnost celega vektorja lastnosti. Znano je, da razlika med določenimi stopnjami jakosti aglutinacije ni velika. Zato lahko v mnogih primerih različne stopnje jakosti aglutinacije v določenih kolonah pripeljejo do iste končne interpretacije preiskav.

V nadaljevanju smo predstavili eksperimentalno ugotovljene rezultate kombinacije posameznih opisanih pristopov za reševanje posameznih problemov.

Poglavje 5

Rezultati eksperimentov

V pričujočem poglavju smo predstavili rezultate eksperimentov uporabe različnih metod za reševanje posameznih problemov pri gradnji sistema za samodejno interpretacijo predtransfuzijskih preiskav. Opazovali smo vpliv izbire metod in parametrizacijo teh metod na končno učinkovitost sistema. Mera za končno učinkovitost je bila delež uspešnosti pri interpretaciji predtransfuzijskih preiskav. Ločeno smo obravnavali vpliv segmentacijskih metod, vpliv izbire in parametrizacije metod izračuna vektorjev lastnosti, izbire algoritma strojnega učenja za gradnjo modela določanja stopnje jakosti aglutinacije in izbire algoritma strojnega učenja za določanje dokončne interpretacije preiskave.

5.1 Označevanje kombinacije uporabljenih algoritmov

Za lažjo in preglednejšo predstavitev rezultatov smo za označevanje kombinacije uporabljenih algoritmov za izračun stopnje jakosti aglutinacije izbrali sledeči način:

$$[\text{PCA/ZRNI}]_{\{parameter\}}\ \text{S}_{\{način\ segmentacije\}}\ \text{M}_{\{metoda\ strojnega\ učenja\}}$$

Primer imenovanja metode, pri kateri smo vektorje lastnosti računali z zrnjenjem, pri kateri izračunali vektorje lastnosti s šestimi komponentami iz vektorjev projekcij, katere smo izračunali iz slik, segmentiranih z metodo nelinearnega filtriranja 1 in smo nabor teh vektorjev lastnosti uporabili za gradnjo modela z metodo strojnega učenja drevesa, J84 je "$\text{ZRNI}_6\ \text{S}_7\ \text{M}_{36}$".

5.2 Sestava učne/testne množice

5.2.1 Stopnje jakosti aglutinacije uporabljenih kolon učne/testne množice

V množici slik kolon slik gelskih kartic smo imeli na voljo 182 slik kolon. Vse slike so bile opremljene s pripadajočo stopnjo jakosti aglutinacije. Povzetek vsebine učne/testne množice kolon s pripadajočimi stopnjami jakosti aglutinacije smo predstavili v tabeli 5.1. Obstaja še stopnja jakosti aglutinacije DCP – dvojna celična populacija, ki se zaradi redkosti pojavljanja v času trajanja zbiranja podatkov v okviru naše raziskave v sistemu za telekonzultacije ni pojavila. Zato je manjkala v našem naboru testnih/učnih podatkov in je v raziskavi nismo obravnavali.

Številčna oznaka stopnje	Stopnja jakosti aglutinacije	Število kolon
1	Prazno	21
2	NEG	74
3	1+	9
4	2+	14
5	3+	11
6	4+	53

Tabela 5.1: Specifikacija porazdelitve stopnje jakosti aglutinacije 182, v postopek strojnega učenja zajetih kolon.

5.2.2 Dokončna interpretacija – KS

V splošni transfuzijski praksi se izvaja več različnih tipov preiskav z gelsko metodo. Vsem preiskavam je skupno, da uporabljajo isti diagnostični pripomoček: gelske kartice. Posamezne preiskave se med sabo razlikujejo po protokolu ravnanja z vzorci, vzorcih in reagentih, uporabljenimi za obdelavo vzorcev, in reagentih v posameznih kolonah gelskih kartic. Glede na v gelski kartici uporabljene reagente se le-te razlikujejo med sabo in so specifične za posamezne preiskave.

Začetni del postopka interpretacije rezultatov preiskav je za vse preiskave enak. Najprej se v kolone gelske kartice s pipeto nakaplja vzorce krvi. Vzorci krvi v kolonah različno reagirajo – aglutinirajo. Gelske kartice se po končani reakciji centrifugira. V nadaljevanju postopka sledi odčitavanje stopnje jakosti aglutinacije za vsako od šestih kolon.

Določanju stopnje jakosti aglutinacije v vsaki od šestih kolon sledi za vsak tip preiskave specifična interpretacija kombinacije določenih stopenj jakosti aglutinacije. Najpogosteje uporabljene preiskave so naštete v podpoglavju 1.1.

Ker smo imeli v času nastajanja tega dela na voljo premajhno učno množico za ustrezno testiranje, smo množico za simulacijo in preizkus delovanja ročno generirali iz pravilnostne tabele. Odločili smo se za preiskavo: določanje krvne skupine. Ostale, pogosto uporabljene preiskave so podane v podpoglavju 3.1.2.2.

Postopek določanja krvne skupine poteka z gelsko kartico *humana* [3]. Z gelsko kartico *humana* se določi krvna skupina krvi. Rezultati določanja krvne skupine z gelsko kartico *humana*, povzeti po literaturi [3], so: A, B, AB, AB ot ali 0.

Generiran podatkovni nabor je vseboval 1.296 vzorcev, ki so bili sestavljeni iz stopenj jakosti aglutinacije v kolonah 1:$AntiA$, 2:$AntiB$, 5:A_1 in 6:B in pripadajoče interpretacije rezultatov. Specifikacijo smo podali v tabeli 5.2.

Rezultat	Število rezultatov
A	20
B	16
AB	6
AB ot	10
0	23
/	1221

Tabela 5.2: Specifikacija porazdelitve rezultatov krvne skupine v podatkovnem naboru, generiranem na podlagi literature [3].

5.3 Rezultati analize izbire za dani problem optimalnih segmentacijskih algoritmov

Med sabo smo primerjali učinkovitost uporabe različnih segmentacijskih algoritmov na razvrščanje slik kolon gelskih kartic v enega od šestih razredov. Za gradnjo učne/testne množice smo imeli na voljo 182 slik kolon gelskih kartic z določeno stopnjo jakosti aglutinacije. Specifikacijo podatkov učne množice smo podali v tabeli 5.1.

5.3.1 Opis eksperimenta

Z obravnavanimi segmentacijskimi algoritmi smo dani nabor slik kolon gelskih kartic segmentirali in iz segmentiranih slik izračunali vektorje lastnosti. Obravnavane segmentacij-

ske algoritme smo podali v tabeli 5.3.

Za izračun vektorjev lastnosti smo uporabili dve metodi. To sta metoda zrnjenja – ZRNI in metoda analize glavnih komponent – PCA. Metodi sta opisani v podpoglavju 4.3.3. Pri obeh metodah smo spreminjali število komponent vektorja lastnosti, v katerega sta ti metodi preslikali vektor projekcij.

Dobljene vektorje lastnosti smo opremili s pripadajočimi stopnjami jakosti aglutinacije in na ta način dobljene podatkovne nabore zapisali v datoteko v podatkovnem formatu ARFF. Te podatkovne nabore smo uporabili kot učno in testno množico v metodah strojnega učenja. Z vsakim od podatkovnih naborov smo zgradili in preizkusili model določanja stopnje jakosti aglutinacije z 49 metodami strojnega učenja. Seznam uporabljenih metod smo predstavili v tabeli 4.1. Ker so bili posamezni nabori podatkov omejeni, smo za preizkus delovanja dobljenih modelov uporabili metodo navzkrižne validacije in na ta način učinkovito preizkusili obnašanje posameznih algoritmov na danem naboru podatkov. Gradnja in preizkušanje 49 modelov na vsakem od podatkovnih naborov so predstavljali en eksperiment, ki smo ga pognali v okolju WEKA. Eksperiment je vseboval zagonske parametre za posamezne algoritme strojnega učenja, zajete v raziskavo. Za celoten postopek smo generirali zbirko 352 eksperimentov. Z načinom poganjanja eksperimentov, opisanem v podpoglavju 4.2.2.2, smo za izvajanje eksperimentov porabili 11 dni.

Segmentacijske algoritme za segmentacijo slik kolon gelskih kartic smo napisali v programskem okolju Matlab. Predhodno smo pripravili datoteke, ki so vsebovale posamezne slike kolon gelskih kartic in datoteke z meta-podatki, ki so zajemali imena datotek s slikami in pripadajoče oznake razredov. Z algoritmom v Matlabu smo prebrali meta-podatke iz datotek, naložili slike, izvedli segmentacijske algoritme in postopke za izračun vektorjev lastnosti. V nadaljevanje smo podatke za vsako kombinacijo segmentacijskega algoritma in postopka za izračun vektorjev lastnosti zapisali v ARFF datoteko. Imena ARFF datotek smo izbrali tako, da so opisovala izbrani segmentacijski algoritem in postopek za izračun vektorjev lastnosti.

5.3.2 Uporabljeni segmentacijski algoritmi

Segmentacijske algoritme, uporabljene za segmentacijo slikovnih elementov, ki predstavljajo eritrocite v slikah kolon gelskih kartic, smo predstavili v tabeli 5.3. Obravnavali smo 11 različnih segmentacijskih algoritmov. Algoritmi so opisani v podpoglavju 4.3.2.1. Ker

v tem trenutku raziskave še ni bilo znano, kako bo izbira posamezne segmentacijskega algoritma vplivala na nadaljnje postopke, smo opazovali rezultate pri uporabi vseh segmentacijskih algoritmov v kombinaciji z vsemi metodami za izračun vektorjev lastnosti in algoritmi strojnega učenja. V nadaljevanju smo predstavili način kombiniranja teh metod.

Številčna oznaka algoritma	Ime algoritma
1	R
2	Sivinska
3	Cr
4	R-G-B
5	1 - R
6	prag(R-G-B)*prag(Cr))
7	Nelinearno filtriranje 1
8	Nelinearno filtriranje 2
9	Nelinearno filtriranje 3
10	Nelinearno filtriranje 4
11	LAB a*

Tabela 5.3: Specifikacija segmentacijskih algoritmov, s katerimi smo segmentirali v postopek strojnega učenja zajete slike kolon. Podroben opis se nahaja v podpoglavju 4.3.2.1.

5.3.2.1 Metode za izračun vektorjev lastnosti

V raziskavi smo uporabili več različnih metod za izračun vektorjev lastnosti kolon. Uporabili smo metodo zrnjenja in metodo PCA. Vektorje lastnosti s pripadajočimi rezultati za posamezen nabor podatkov smo zapisali v ARFF datoteke. Te datoteke smo uporabili v WEKA-i.

5.3.2.1.1 Zrnjenje – ZRNI Spreminjali smo število komponent vektorja lastnosti, ki smo ga izračunali z metodo zrnjenja iz vektorjev projekcije slik kolon gelske kartice na os y. Število komponent smo spreminjali v intervalu [1..15]. Na ta način smo za vsak tip segmentacije izdelali 15 naborov podatkov. Za testiranje učinkovitosti razvrščanja kolon glede na uporabljen segmentacijski algoritem pri izračunu vektorja lastnosti z zrnjenjem smo pripravili skupaj $11 \times 15 = 165$ naborov podatkov.

5.3.2.1.2 PCA Spreminjali smo število komponent, ki smo jih obdržali in iz njih tvorili vektorje lastnosti. Število obdržanih komponent smo označili z n. Raziskavo

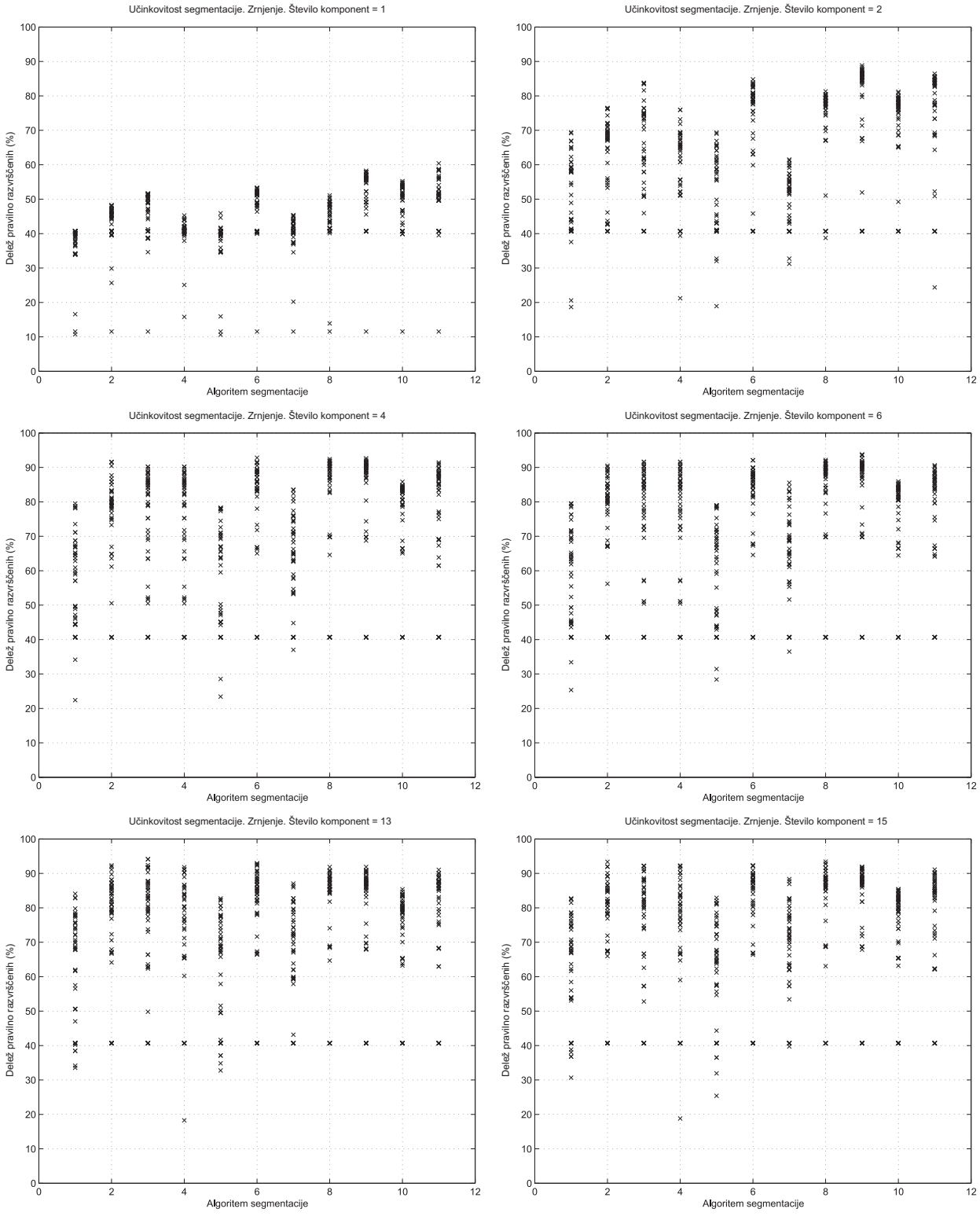
smo izvedli za obdržano število komponent v intervalu [1..17]. Za vsak tip segmentacije smo izdelali 17 naborov podatkov. Za testiranje učinkovitosti razvrščanja kolon glede na uporabljen segmentacijski algoritem pri izračunu vektorja lastnosti s PCA smo pripravili skupaj $11 \times 17 = 187$ naborov podatkov.

5.3.3 Primerjava metod učinkovitosti segmentacije

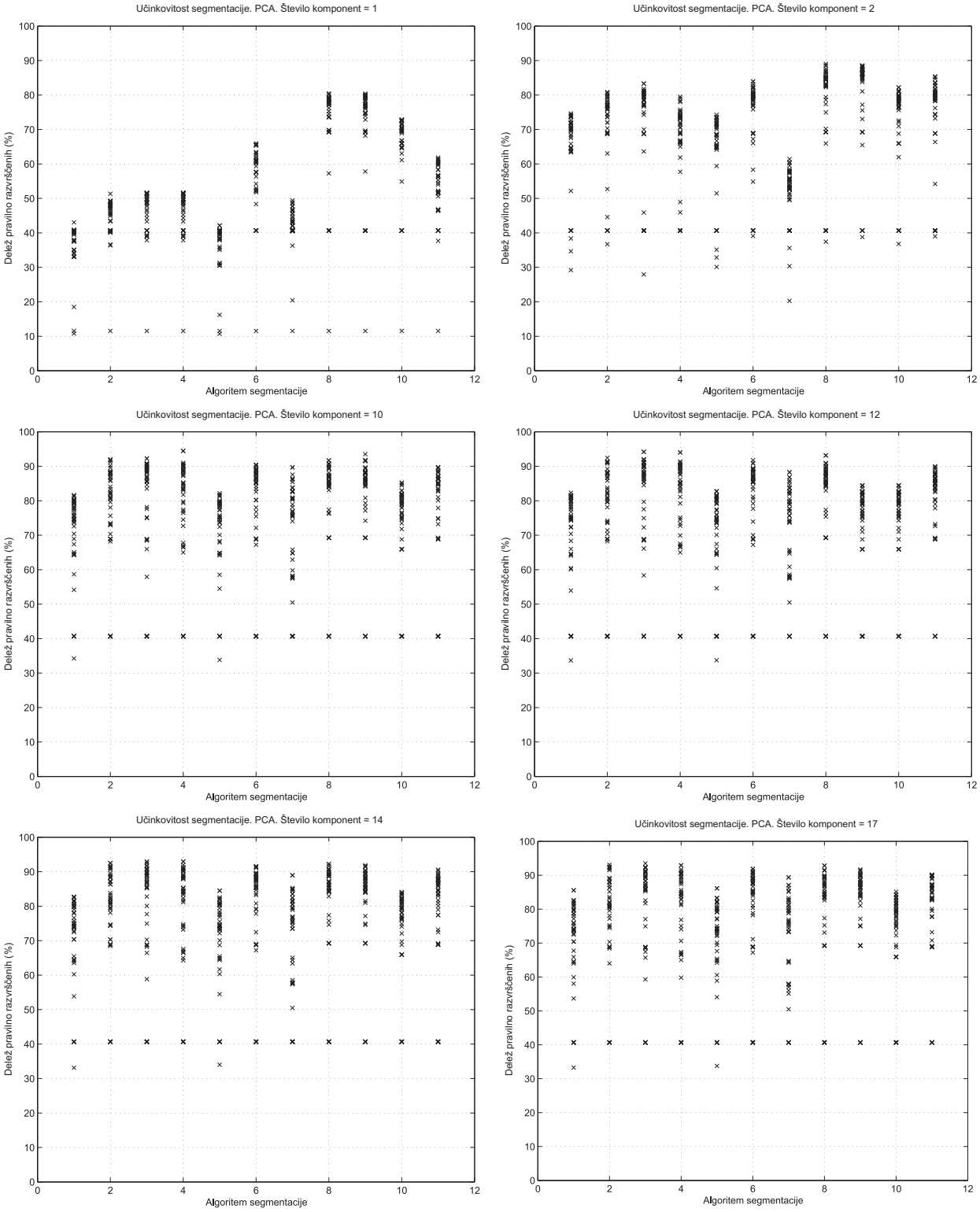
Ko smo zaključili z izvajanjem 352 eksperimentov za preizkus in parametrizacijo metod ZRNI in PCA, smo imeli na voljo podrobne rezultate delovanja vseh 49 metod strojnega učenja za vsak podatkovni nabor. Iz datotek z rezultati eksperimentov smo izbrali parameter deleža pravilno določenih stopenj jakosti aglutinacije za vsak algoritem strojnega učenja in te rezultate primerjali med sabo. Za grobo izbiro najbolje delujočih algoritmov segmentacije smo narisali potek deleža pravilno razvrščenih za posamezne algoritme strojnega učenja za posamezne eksperimente. Empirično smo izbrali tiste eksperimente, pri katerih je bil dosežen najvišji delež pravilno določenih stopenj jakosti aglutinacije.

Iz slike 5.1 lahko razberemo, kako uporaba posameznih segmentacijskih algoritmov pri uporabi metode izračuna vektorjev lastnosti ZRNI vpliva na učinkovitost delovanja algoritmov strojnega učenja. Ker smo se v tej točki raziskave ukvarjali z izbiro za dani problem najboljšega segmentacijskega algoritma, so nas zanimali le najboljši rezultati deleža pravilno razvrščenih. Najboljši rezultati so deleži pravilno razvrščenih, ki so najbližje 100 %. Na sliki 5.1 smo narisali poteke deleža uspešnosti določanja stopnje jakosti aglutinacije slik kolon s posameznimi algoritmi strojnega učenja, uporabljenimi v raziskavi. Na osi x smo navedli številčne oznake segmentacijskih algoritmov, na osi y pa deleže uspešnosti razvrščanja. Za vsakega od enajstih segmentacijskih algoritmov smo dobili 49 rezultatov deležev uspešnosti 49 modelov določanja stopnje jakosti aglutinacije, ki smo jih kot točke vrisali v graf. Narisali smo več grafov – vsak predstavlja uporabo določenega, v glavi grafa navedenega števila komponent vektorja lastnosti, izračunanega z metodo zrnjenja. Na sliki 5.2 smo na enak način predstavili podatke pri uporabi metode PCA za izračun vektorjev lastnosti. Narisanih je več grafov, vsak za svoje število obdržanih komponent.

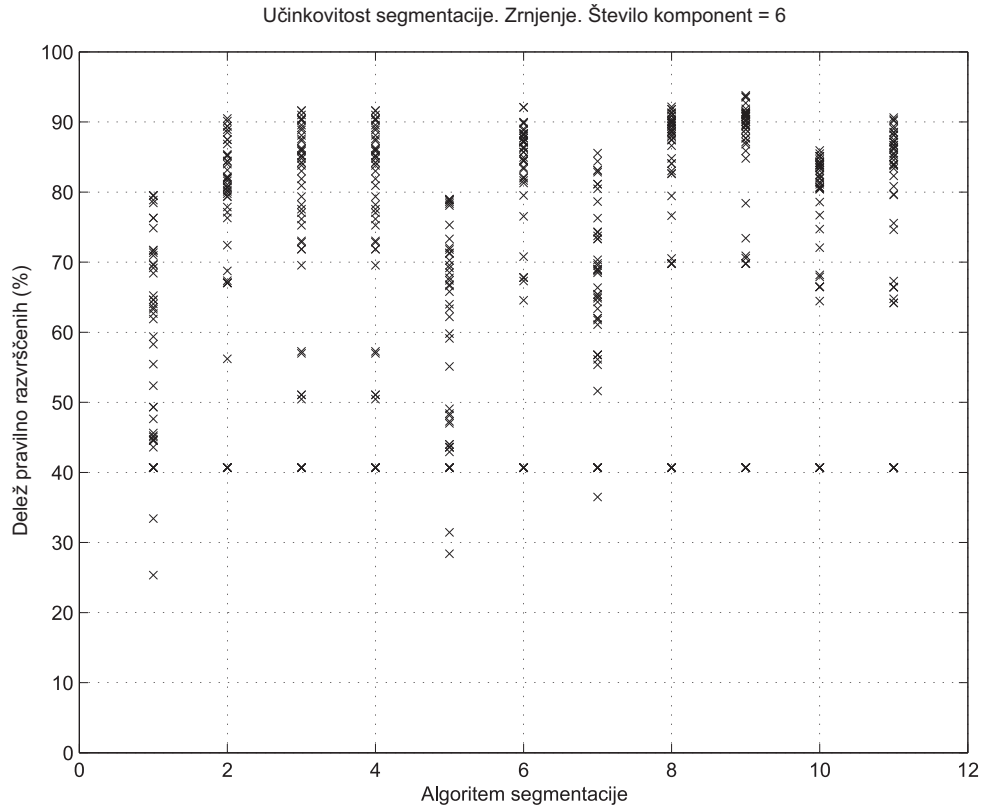
Z opazovanjem slik 5.1 in 5.2 smo ugotovili, da je vzorec uspešnosti algoritmov strojnega učenja v odvisnosti od izbranega segmentacijskega algoritma podoben za različne načine pridobivanja vektorjev lastnosti. To pomeni, da uporaba segmentacijskega algoritma daje podobne rezultate pri uporabi različno parametriziranih algoritmov za izračun vektorjev lastnosti. Če npr. segmentacijski algoritem 9 deluje dobro pri ZRNI_2, deluje



Slika 5.1: Učinkovitost različnih algoritmov strojnega učenja, uporabljenih v raziskavi, ob izbiri različnih algoritmov segmentacije. Vektorji lastnosti so bili izračunani z metodo zrnjenja. V napisih nad slikami je podano število komponent, na katere je bila razdeljena projekcija slike posameznih kolon. V tem trenutku obravnave so pomembne točke z najvišjim deležem uspešnosti.



Slika 5.2: Učinkovitost različnih algoritmov strojnega učenja, uporabljenih v raziskavi, ob izbiri različnih algoritmov segmentacije. Vektroji lastnosti so bili izračunani z metodo PCA. V napisih nad slikami je podano število lastnih vektorjev, ki smo jih obdržali. V tem trenutku obravnave so pomembne točke z najvišjim deležem uspešnosti.

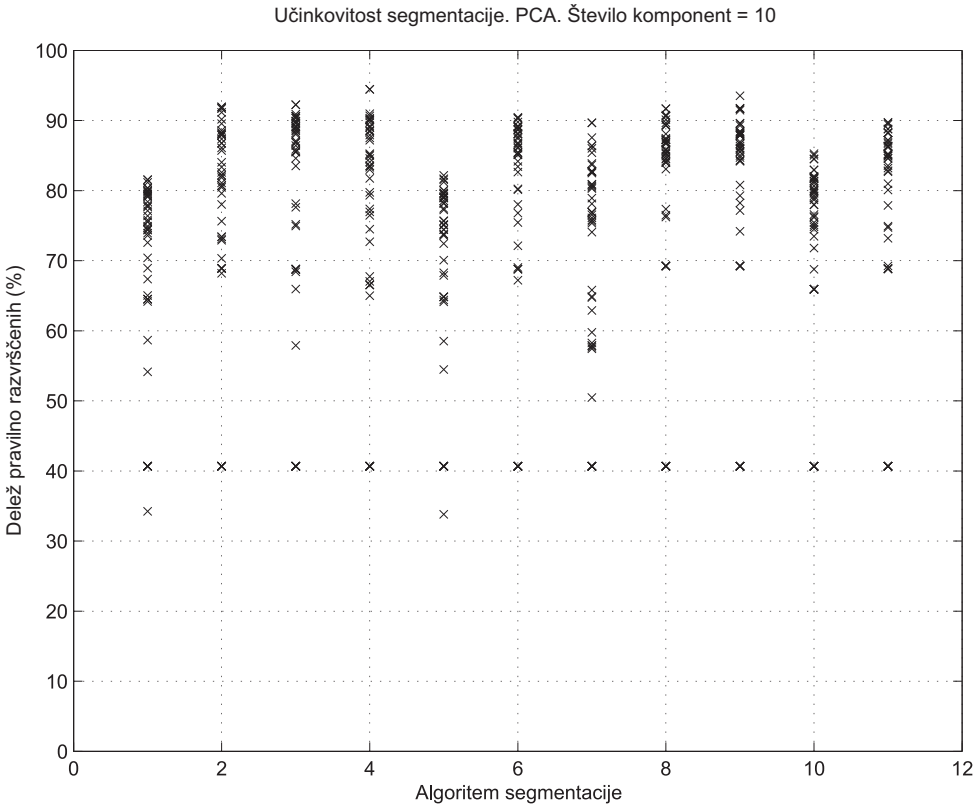


Slika 5.3: Učinkovitost različnih algoritmov strojnega učenja, uporabljenih v raziskavi, ob izbiri različnih segmentacijskih algoritmov. Za izračun vektorja lastnosti je izbran algoritem zrnjenja ZRNI_6.

dobro tudi pri ZRNI_{15}.

Ob podrobnejšem ogledu slike 5.3 smo ugotovili, da smo najboljše rezultate dosegli pri uporabi segmentacijskega algoritma številka 9. V vrh najuspešnejših pa so se se uvrstili tudi segmentacijski algoritmi 2, 3, 4, 6 in 8. Zrnjenje ZRNI_6 je bilo izbrano zato, ker smo pri analizi, opisani v podpoglavju 5.4, ugotovili, da smo z na ta način pridobljenimi vektorji lastnosti dosegli najvišje deleže pravilno določenih stopenj jakosti aglutinacije.

Če smo za izračun vektorja lastnosti izbrali metodo PCA, smo ugotovili, da so algoritmi strojnega učenja najuspešnejši, če smo obdržali prvih 10 komponent. Natančnejšo analiza števila obdržanih komponent smo podali v podpoglavju 5.4. Uspešnost določanja stopnje jakosti aglutinacije smo predstavili na sliki 5.4. Ugotovili smo, da najboljše rezultate dosežemo pri uporabi segmentacijskega algoritma številka 9. V vrh uspešnosti pa so se uvrstili tudi segmentacijski algoritmi 2, 3, 4, 6 in 8.



Slika 5.4: Učinkovitost različnih algoritmov strojnega učenja, uporabljenih v raziskavi, ob izbiri različnih segmentacijskih algoritmov. Za izračun vektorja lastnosti je izbran algoritem PCA, ki za preslikavo podatkov v novi prostor uporabi prvih 10 lastnih vektorjev.

5.4 Rezultati analize optimalne metode za izračun vektorjev lastnosti

V predstavljeni analizi smo izbirali optimalno metodo za izračun vektorjev lastnosti. Optimalnost metode smo merili glede na učinkovitost določanja stopnje jakosti aglutinacije z modeli, zgrajenimi z algoritmi strojnega učenja. Algoritmi strojnega učenja so za gradnjo teh modelov uporabili vektorje lastnosti, ki smo jih izračunali z obravnavanimi metodami. Za merilo učinkovitosti razvrščanja smo izbrali delež pravilno razvrščenih vzorcev za posamezni podatkovni nabor. Eksperiment je podoben eksperimentu, ki smo ga opisali v podpoglavju 5.3.1. Obravnavali smo dva algoritma. Prvi je algoritem zrnjenja – ZRNI, drugi pa je algoritem analize glavnih komponent – PCA. Za oba algoritma smo spreminjali število komponent izračunanih vektorjev lastnosti in opazovali delovanje modelov. V nadaljevanju smo za oba algoritma predstavili ločeno obravnavo izbire števila komponent.

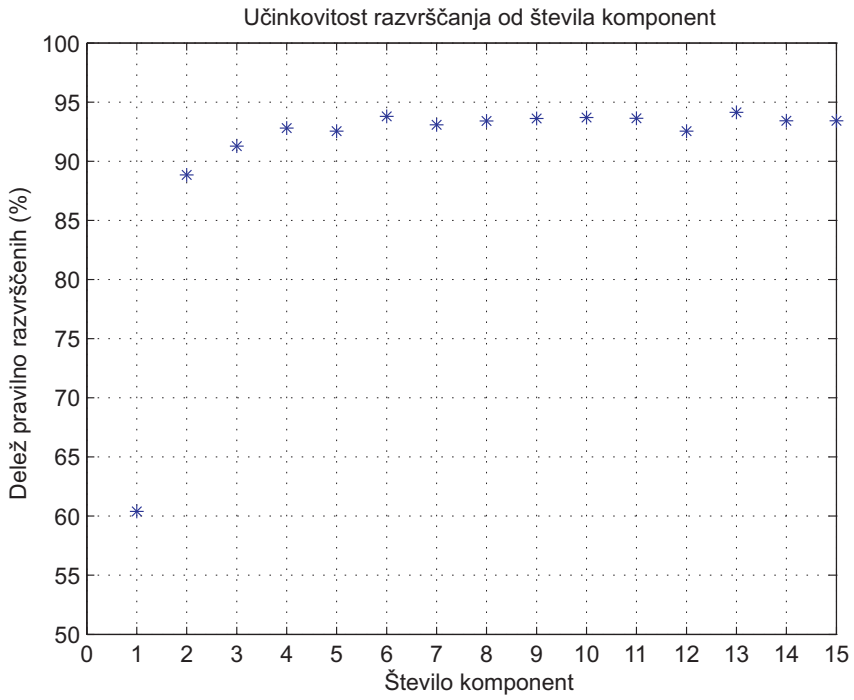
5.4.1 Algoritem za izračun vektorjev lastnosti z zrnjenjem

V pričujočem podpoglavju smo predstavili obravnavo rezultatov algoritma za izračun vektorjev lastnosti z zrnjenjem. Podroben opis algoritma smo podali v podpoglavju 4.3.3.2.

5.4.1.1 ZRNI: Vpliv izbranega števila komponent na uspešnost algoritmov strojnega učenja

Raziskali smo vpliv algoritma za izračun vektorja lastnosti z zrnjenjem na učinkovitost razvrščanja. Algoritem vektor projekcije razdeli na enako dolge odseke in za vsakega izračuna njegovo srednjo vrednost. Na ta način smo opazovano kolono razdelili na segmente in ugotavljali, v katerih segmentih se nahajajo eritrociti. Dobljene srednje vrednosti smo zapisali v vektor lastnosti. V eksperimentu smo spreminjali število komponent, na katere je algoritem razdelil vektor projekcije segmentirane slike gelske kartice in opazovali učinkovitost delovanja algoritmov strojnega učenja pri uporabi podatkovnih naborov, sestavljenih iz teh vektorjev lastnosti.

Iz nabora slik registriranih slik kolon gelskih kartic, opremljenih z določitvami stopenj jakosti aglutinacije smo z različnimi segmentacijskimi algoritmi in projekcijo segmentiranih slik na os y generirali vektorje projekcij, iz katerih smo z algoritmom zrnjenja generirali posamezne vektorje lastnosti. Iz vektorjev lastnosti s pripadajočimi klasifikacijami smo generirali posamezne podatkovne nabore. Za vsako opazovano število kompo-



Slika 5.5: Vpliv izbranega števila komponent vektorja lastnosti izračunanega z algoritmom za izračun vektorjev lastnosti z zrnjenjem na učinkovitost razvrščanja z algoritmi strojnega učenja. Analiza je izvedena za število komponent v intervalu [1..15].

nent vektorjev lastnosti smo generirali svoj podatkovni nabor. Analizo smo izvedli za število komponent v intervalu [1..15]. Podatkovne nabore smo zapisali v ARFF datoteko in generirali eksperiment, kot je opisano v podpoglavju 4.2.2. Za vsak nabor podatkov smo izvedli eksperiment, ki je zajemal preizkus delovanja postopka posameznih algoritmov strojnega učenja z metodo navzkrižne validacije. V rezultatih eksperimentov smo poiskali maksimalno doseženo uspešnost strojnega učenja, doseženo nad podatkovnimi nabori, generiranimi z vsemi kombinacijami enajstih segmentacijskih algoritmov in 49 algoritmov strojnega učenja. Maksimalno uspešnost smo ocenili iz deleža pravilno razvrščenih vzorcev posameznega podatkovnega nabora. Rezultate smo predstavili na sliki 5.5. Ugotovili smo, da se delež pravilno razvrščenih strmo povečuje do števila komponent $n = 4$, po tem pa delež pravilno razvrščenih ne narašča več bistveno. Kot optimalno izbiro števila komponent smo izbrali $n = 6$.

5.4.2 Algoritem za izračun vektorjev lastnosti z metodo PCA

Raziskali smo vpliv algoritma za izračun vektorja lastnosti z metodo PCA na učinkovitost razvrščanja. Podrobnejši opis algoritma smo podali v podpoglavju 4.3.3.1. Metoda PCA je primerna za iskanje vzorcev v naboru podatkov in za prikaz teh podatkov v taki obliki, da so poudarjene podobnosti in razlike med temi nabori podatkov [14]. Metoda PCA je posebej uporabna pri analizi naborov podatkov velikih dimenzij, saj je možno z njeno uporabo izračunati preslikavo podatkov v prostor z manj dimenzijami na tak način, da se ohrani večina informacije vhodnih podatkov.

5.4.2.1 Izbira števila komponent vektorja lastnosti s PCA

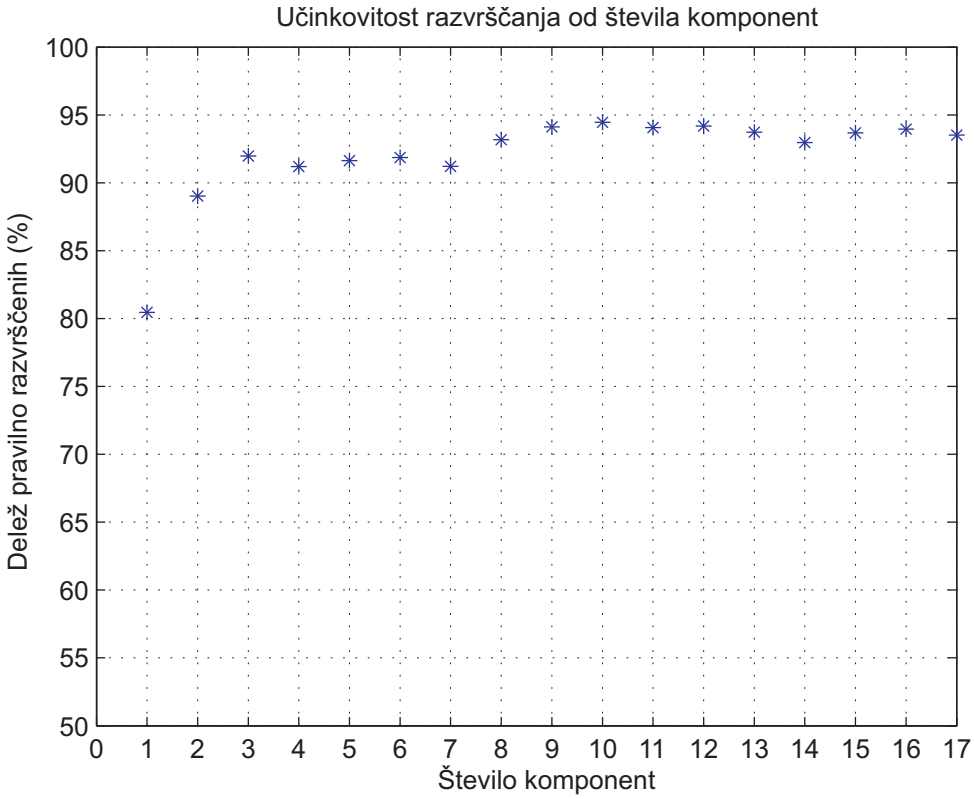
V naši raziskavi smo spreminjali število uporabljenih komponent, dobljenih z metodo PCA, za izračun vektorjev lastnosti. Pri raziskavi smo preizkusili, kako dodajanje posameznih komponent vpliva na učinkovitost algoritmov strojnega učenja. Eksperiment smo začeli z uporabo le prve komponente za vektor lastnosti in nadaljevali tako, da smo dodajali naslednje. To smo ponavljali do sedemnajste komponente.

Ob pregledu slike 5.6 ugotovimo, da dokaj visok delež pravilno razvrščenih dosežemo že z uporabo samo prve komponente. Uporaba nadaljnjih dveh (skupaj 3) še bistveno doprinese k deležu pravilno razvrščenih, po tem pa se doseženi maksimalni delež uspešnosti ne spreminja več bistveno. Opazimo, da je dosežen najvišji delež uspešnosti pri desetih uporabljenih komponentah. Ker je graf deleža pravilno razvrščenih v odvisnosti od izbranega števila komponent naraščajoča funkcija, predvidevamo, da z uporabo v literaturi pogosto predlagane metode, ki svetuje izpuščanje prvih nekaj komponent, ne bi izboljšali uspešnosti.

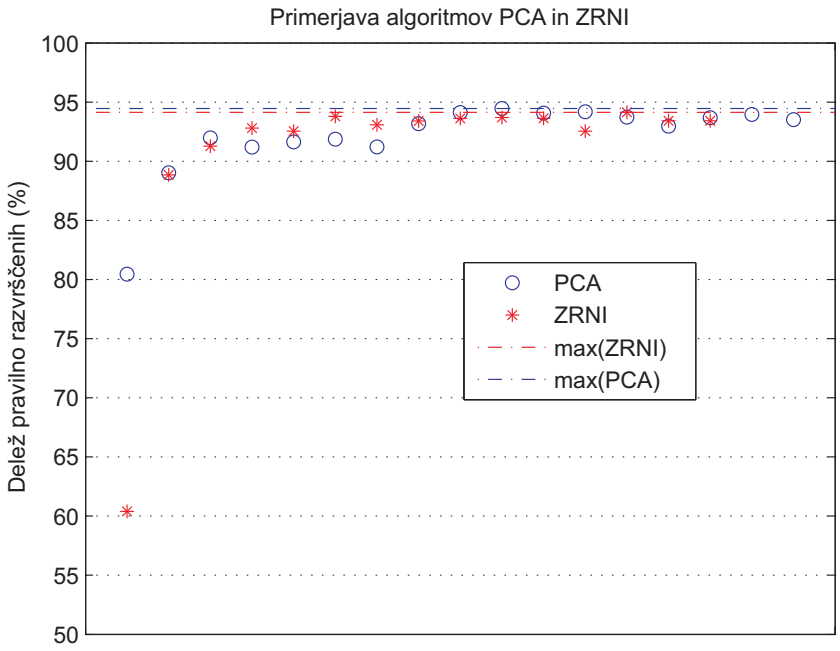
5.4.3 Primerjava učinkovitosti določanja stopnje jakosti aglutinacije pri uporabi algoritma za izračuna vektorjev lastnosti z metodo zrnjenja in PCA

Med sabo smo primerjali učinkovitost uporabe algoritmov za izračun vektorjev lastnosti z metodo zrnjenja in PCA na uspešnost razvrščanja vzorcev z algoritmi strojnega učenja.

Analizirali smo učinkovitost razvrščanja podatkovnih naborov, katerih vektorji lastnosti so bili generirani z metodo zrnjenja in metodo PCA. Eksperimenta smo opisali v podpoglavjih 5.4.1 in 5.4.2.



Slika 5.6: Vpliv izbranega števila komponent algoritma PCA za izračun vektorjev lastnosti na učinkovitost razvrščanja z algoritmi strojnega učenja. Analiza je izvedena za število komponent v intervalu [1..17].



Slika 5.7: Primerjava doseženega deleža uspešnosti pri uporabi različno parametriziranih metod ZRNI in PCA za izračun vektorjev lastnosti iz vektorjev projekcij segmentiranih slik kolon. Upoštevani so najvišji doseženi deleži pravilno razvrščenih med rezultati uporabljenih 49 metod strojnega učenja.

Rezultat primerjave smo predstavili na sliki 5.7. Na sliki smo predstavili dosežene uspešnosti določanja stopnje jakosti aglutinacije z modeli, zgrajenimi z algoritmi strojnega učenja. S črtkanimi črtami smo označili dosežena maksimuma za vsakega od uporabljenih algoritmov. Ugotovili smo, da smo z uporabo metod PCA in ZRNI dosegli primerljive rezultate. V poglavju 5.4.2 smo ugotovili, da je metoda PCA najučinkovitejša ko se odločimo za 10 lastnih vektorjev, v podpoglavju 5.4.1 pa smo ugotovili, da je metoda ZRNI najučinkovitejša, ko se odločimo za 6 komponent.

5.5 Rezultati analize primernosti metod strojnega učenja za določanje stopnje jakosti aglutinacije

Raziskali smo učinkovitost metod strojnega učenja za določanje stopnje jakosti aglutinacije slik kolon gelskih kartic. Preizkusili in med sabo smo primerjali učinkovitost 49 različnih metod. V raziskavo vključene metode smo navedli v tabeli 4.1.

5.5.1 Izbira načina izračuna vektorjev lastnosti

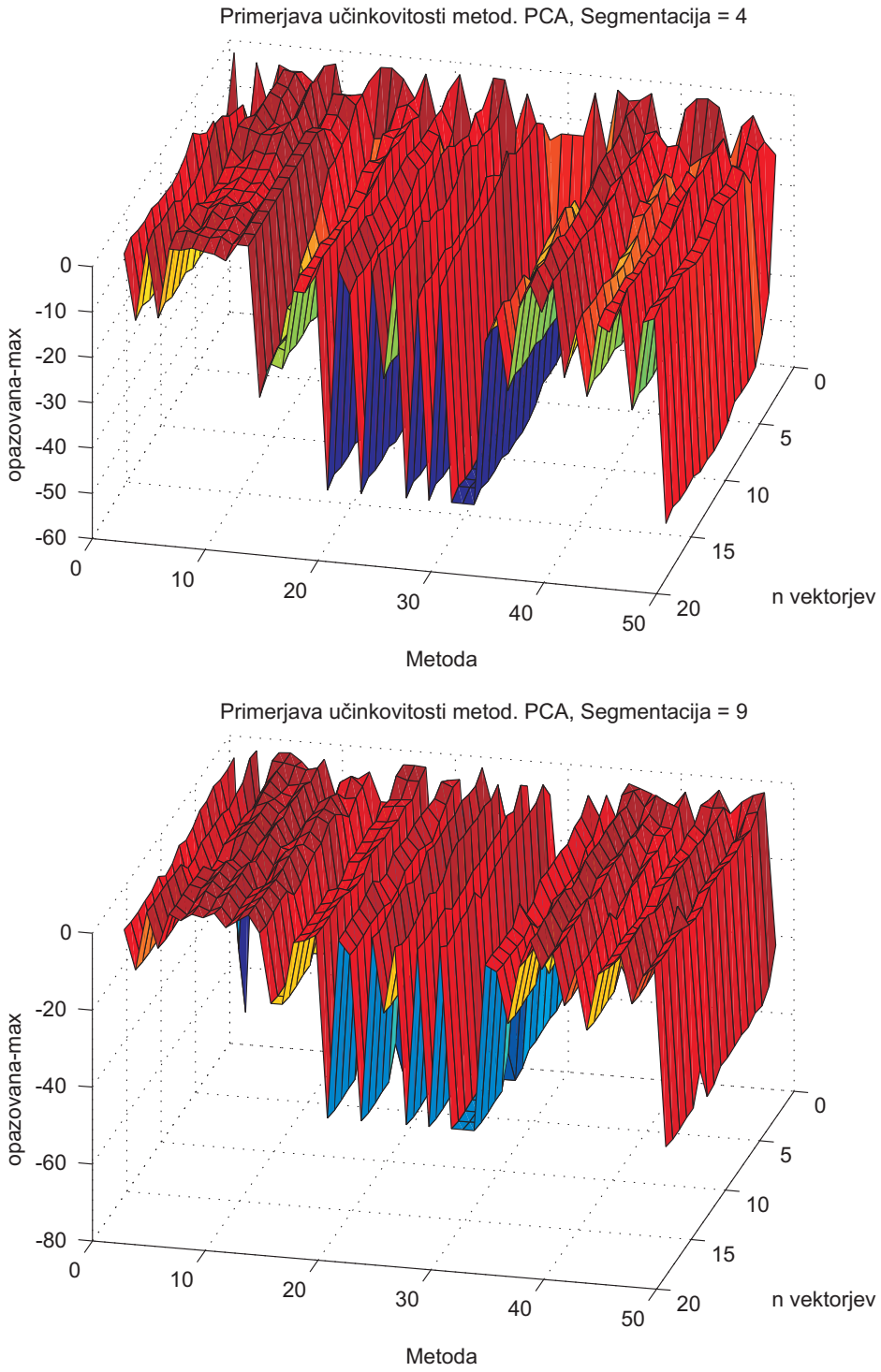
Podrobno analizo načinov izračuna vektorjev lastnosti smo podali v podpoglavju 5.4. Na slikah 5.8 in 5.9 smo predstavili relativno učinkovitost delovanja algoritmov strojnega učenja pri posameznem načinu izračuna vektorjev lastnosti. Učinkovitost je predstavljena relativno glede na najuspešnejši algoritem za dani način izračuna vektorjev lastnosti. Prikazali smo po dva primera uporabljenih, najbolje delujočih segmentacijskih algoritmov s kompletnim, v eksperiment zajetim področjem izračuna. Primera sta prikazana za oba algoritma izračuna vektorjev lastnosti (PCA in ZRNI).

Posamezne točke v grafu smo izračunali tako, da smo od deleža uspešnosti za opazovani algoritem odšteli delež uspešnosti najuspešnejšega algoritma za določen način izračuna vektorjev lastnosti.

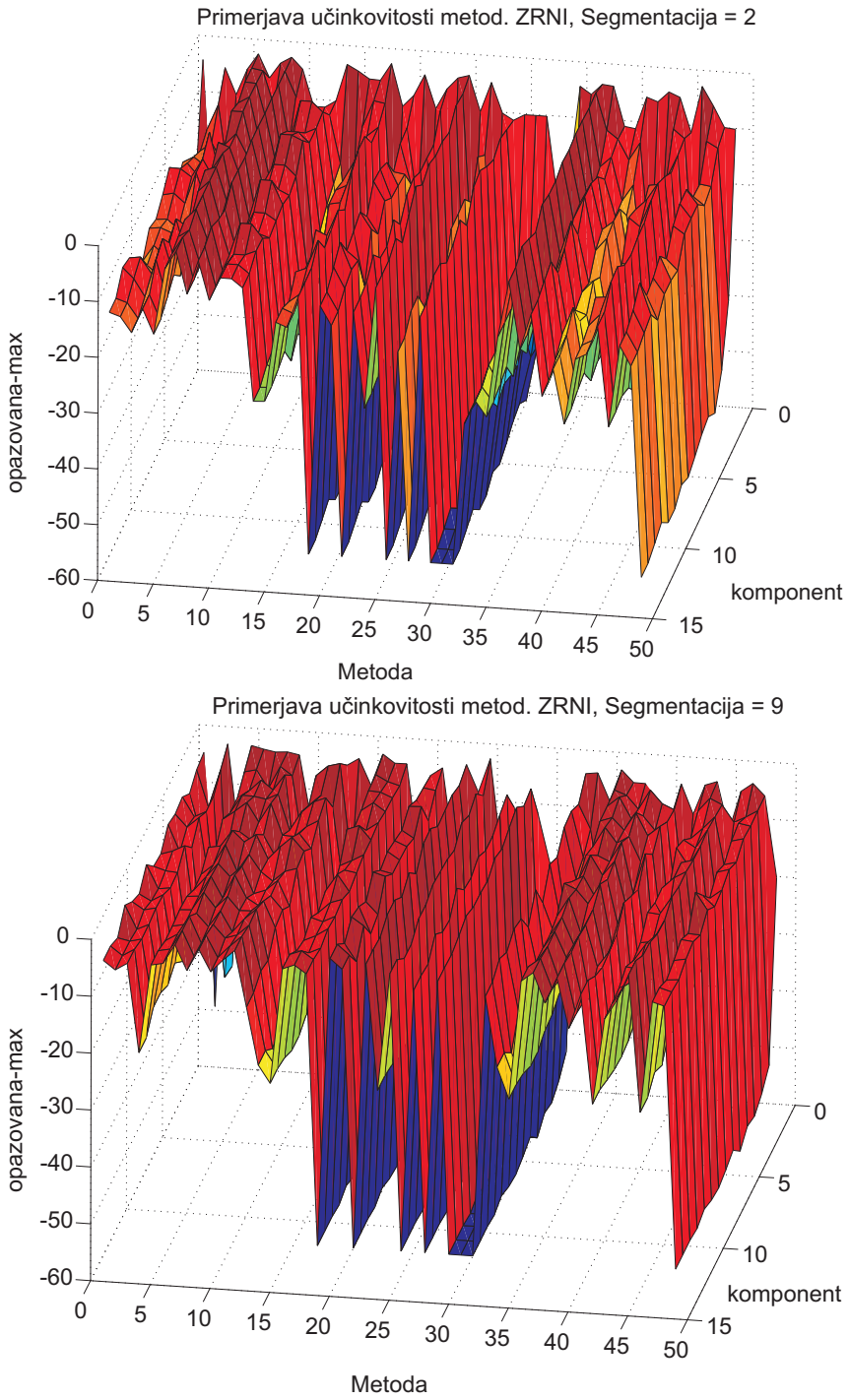
Za najuspešnejši algoritem je bil rezultat odštevanja 0, za vse ostale pa manj kot 0. Na slikah 5.8 in 5.9 opazimo, da vsi uporabljeni algoritmi strojnega učenja po celotnem področju načina izračuna vektorjev lastnosti dosegajo primerljive deleže uspešnosti. Zato lahko na slikah opazimo konstantne doline in grebene v smeri področja izračuna vektorjev lastnosti. Če je npr. algoritem strojnega učenja 12 uspešenejši od algoritma 37 pri izračunu vektorjev lastnosti z npr. ZRNI_6 S_9, bo algoritem 12 verjetno uspešnejši od algoritma 37 tudi pri izračunu vektorjev lastnosti z npr. ZRNI_{10} S_4. Če bi se relativna učinkovitost metod med sabo zelo spreminjala v odvisnosti od izbranega načina izračuna vektorjev lastnosti, na slikah ne bi opazili posameznih grebenov in dolin, marveč bi opazili le naključno razporejene koničaste vrhove.

Zato smo lahko izbrali najboljši segmentacijski algoritem in ustrezno parametrizirali metodo izračuna vektorjev lastnosti in pri izbranem med sabo primerjali in podrobno analizirali učinkovitost posameznih algoritmov strojnega učenja.

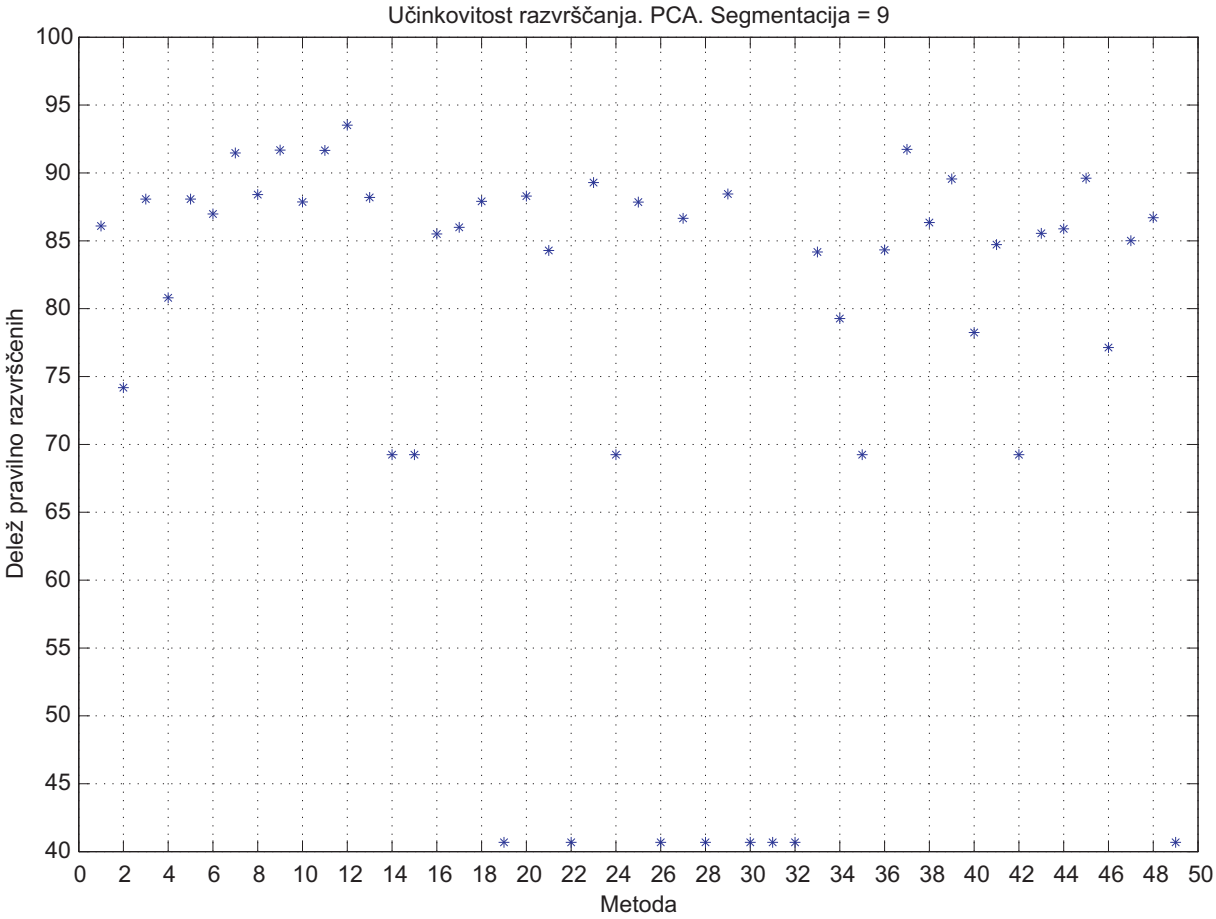
S predhodno opisanimi eksperimenti smo ugotovili, da smo najboljše rezultate določanja stopnje jakosti aglutinacije slik gelskih kartic dosegli v primeru, ko smo vektorje lastnosti



Slika 5.8: Primerjava posameznih metod strojnega učenja pri vektorjih lastnosti izračunanih z metodo PCA glede na maksimalno uspešnost razvrščanja, doseženo z uporabljenimi parametri. Vidimo, da se posamezne metode strojnega učenja v področju izračuna vektorjev lastnosti obnašajo podobno učinkovito.



Slika 5.9: Primerjava posameznih metod strojnega učenja pri vektorjih lastnosti izračunanih z metodo ZRNI glede na maksimalno uspešnost razvrščanja, doseženo z uporabljenimi parametri. Vidimo, da se posamezne metode strojnega učenja v področju izračuna vektorjev lastnosti obnašajo podobno učinkovito.



Slika 5.10: Učinkovitost algoritmov strojnega učenja na razvrščanje (delež pravilno razvrščenih) pri PCA_{10} S_9.

izračunali z metodo PCA, pri kateri smo obdržali prvih 10 lastnih vektorjev. Najboljše rezultati smo dosegli, ko smo za izračun vektorjev lastnosti uporabili vektorje projekcij segmentiranih slik, segmentiranih z metodo 9 (PCA_{10} S_9). Primerljive rezultate smo dosegli z uporabo metode ZRNI, pri kateri smo vektorje projekcije razdelili na 6 delov in za segmentacijo uporabili segmentacijski algoritem 9 ($ZRNI_6$ S_9).

5.5.2 Izbira kandidatov za optimalen algoritem strojnega učenja

Analizirali smo izbiro najbolje delujočih algoritmov strojnega učenja. Analizo smo izvedli na modelih določanja stopnje jakosti aglutinacije, zgrajenimi z 49 različnimi algoritmi strojnega učenja. Za analizo smo pripravili dva podatkovna nabora. Za izračun vektorjev lastnosti za uporabljena nabora smo uporabili najbolje delujoči kombinaciji metod PCA_{10}

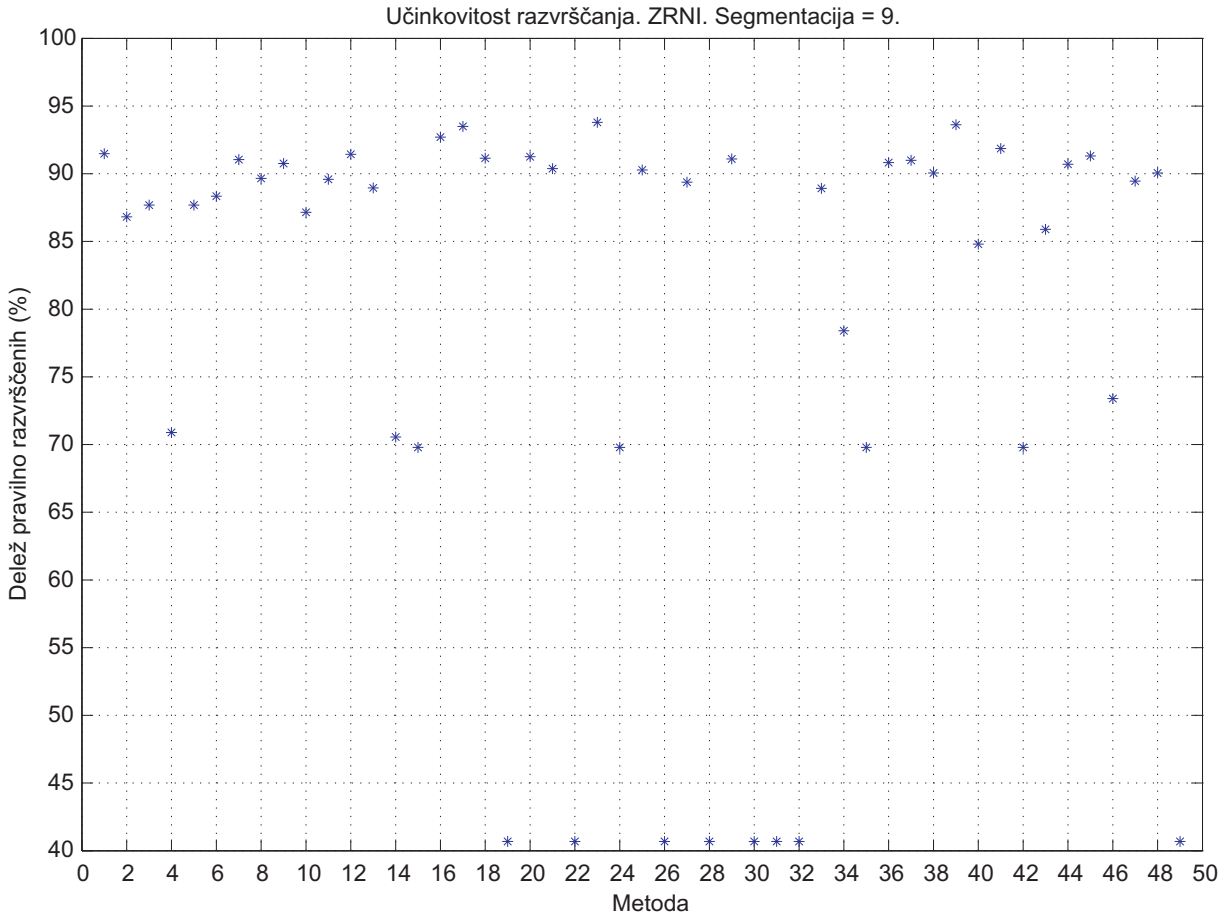
S_9 in $ZRNI_6$ S_9.

V analizi delovanja modelov zgrajenih, z algoritmi strojnega učenja z uporabo podatkovnega nabora, pridobljenega z metodo PCA_{10} S_9 smo ugotovili, da se je najbolje obnesel algoritem strojnega učenja 12: IBk [59] z deležem pravilno razvrščenih 93.52 %. Po uspešnosti za njim ni veliko zaostajal algoritem 37: LMT [8]. Rezultate vseh 49 algoritmov smo predstavili v tabeli 5.4 in ilustrirali na sliki 5.10.

V analizi delovanja modelov, zgrajenih z algoritmi strojnega učenja z uporabo podatkovnega nabora, pridobljenega z metodo $ZRNI_6$ S_9 smo ugotovili, da so se najbolje obnesli algoritmi 39: RandomForest [9] (93,62 %), 23: LogitBoost [67] (93,79 %), 12: IBk [59] (91,44 %) in 16: AttributeSelectedClassifier [4][5] (92,7 %). Rezultate vseh 49 algoritmov smo predstavili v tabeli 5.5 in ilustrirali na sliki 5.11.

Oznaka	Tip metode	Ime metode	Delež pravilno razvrščenih
1	bayes	BayesNet [2]	86,09 %
2	bayes	ComplementNaiveBayes [51]	74,18 %
3	bayes	NaiveBayes [2]	88,07 %
4	bayes	NaiveBayesMultinomial [52]	80,8 %
5	bayes	NaiveBayesUpdateable [53]	88,07 %
6	functions	Logistic [54]	86,98 %
7	functions	MultilayerPerceptron [55]	91,47 %
8	functions	RBFNetwork [56]	88,4 %
9	functions	SimpleLogistic [57]	91,68 %
10	functions	SMO [58]	87,86 %
11	lazy	IB1 [59]	91,66 %
12	lazy	IBk [59]	93,52 %
13	lazy	KStar [60]	88,19 %
14	lazy	LWL [61]	69,24 %
15	meta	AdaBoostM1 [62]	69,24 %
16	meta	AttributeSelectedClassifier [4][5]	85,5 %
17	meta	Bagging [63]	85,99 %
18	meta	ClassificationViaRegression [64]	87,91 %
19	meta	CVParameterSelection [65]	40,68 %
20	meta	Decorate [6]	88,29 %
21	meta	FilteredClassifier [4][5]	84,29 %
22	meta	Grading [66]	40,68 %
23	meta	LogitBoost [67]	89,29 %
24	meta	MultiBoostAB [68]	69,24 %
25	meta	MultiClassClassifier [4][5]	87,85 %
26	meta	MultiScheme [4][5]	40,68 %
27	meta	OrdinalClassClassifier [4][5]	86,65 %
28	meta	RacedIncrementalLogitBoost [4][5]	40,68 %
29	meta	RandomCommittee [4][5]	88,45 %
30	meta	Stacking [69]	40,68 %
31	meta	StackingC [70]	40,68 %
32	meta	Vote [4][5]	40,68 %
33	misc	HyperPipes [4][5]	84,17 %
34	misc	VFI [71]	79,27 %
35	trees	DecisionStump [72]	69,24 %
36	trees	J48 [7]	84,33 %
37	trees	LMT [8]	91,74 %
38	trees	NBTree [73]	86,35 %
39	trees	RandomForest [9]	89,55 %
40	trees	RandomTree [4][5]	78,24 %
41	trees	REPTree [4][5]	84,72 %
42	rules	ConjunctiveRule [4][5]	69,24 %
43	rules	DecisionTable [74]	85,55 %
44	rules	JRip [10]	85,88 %
45	rules	NNge [75]	89,61 %
46	rules	OneR [76][2]	77,14 %
47	rules	PART [11]	85,01 %
48	rules	Ridor [4][5]	86,7 %
49	rules	ZeroR [2][4][5]	40,68 %

Tabela 5.4: Delež pravilno razvrščenih z modeli zgrajenimi s posameznimi algoritmi strojnega učenja. Za generiranje vektorja lastnosti smo uporabili prvih 10 komponent, izračunanih z metodo PCA. Uporabili smo vektorje projekcije, izračunane iz slik, segmentiranih z metodo 9. (PCA_{10} S_9)



Slika 5.11: Učinkovitost algoritmov strojnega učenja na razvrščanje (delež pravilno razvrščenih) pri ZRNI_6 S_9.

Oznaka	Tip metode	Ime metode	Delež pravilno razvrščenih
1	bayes	BayesNet [2]	91,48 %
2	bayes	ComplementNaiveBayes [51]	86,82 %
3	bayes	NaiveBayes [2]	87,68 %
4	bayes	NaiveBayesMultinomial [52]	70,89 %
5	bayes	NaiveBayesUpdateable [53]	87,68 %
6	functions	Logistic [54]	88,34 %
7	functions	MultilayerPerceptron [55]	91,05 %
8	functions	RBFNetwork [56]	89,66 %
9	functions	SimpleLogistic [57]	90,76 %
10	functions	SMO [58]	87,15 %
11	lazy	IB1 [59]	89,59 %
12	lazy	IBk [59]	91,44 %
13	lazy	KStar [60]	88,95 %
14	lazy	LWL [61]	70,56 %
15	meta	AdaBoostM1 [62]	69,79 %
16	meta	AttributeSelectedClassifier [4][5]	92,7 %
17	meta	Bagging [63]	93,5 %
18	meta	ClassificationViaRegression [64]	91,15 %
19	meta	CVParameterSelection [65]	40,68 %
20	meta	Decorate [6]	91,26 %
21	meta	FilteredClassifier [4][5]	90,38 %
22	meta	Grading [66]	40,68 %
23	meta	LogitBoost [67]	93,79 %
24	meta	MultiBoostAB [68]	69,79 %
25	meta	MultiClassClassifier [4][5]	90,28 %
26	meta	MultiScheme [4][5]	40,68 %
27	meta	OrdinalClassClassifier [4][5]	89,38 %
28	meta	RacedIncrementalLogitBoost [4][5]	40,68 %
29	meta	RandomCommittee [4][5]	91,09 %
30	meta	Stacking [69]	40,68 %
31	meta	StackingC [70]	40,68 %
32	meta	Vote [4][5]	40,68 %
33	misc	HyperPipes [4][5]	88,92 %
34	misc	VFI [71]	78,4 %
35	trees	DecisionStump [72]	69,79 %
36	trees	J48 [7]	90,82 %
37	trees	LMT [8]	90,99 %
38	trees	NBTree [73]	90,06 %
39	trees	RandomForest [9]	93,62 %
40	trees	RandomTree [4][5]	84,8 %
41	trees	REPTree [4][5]	91,86 %
42	rules	ConjunctiveRule [4][5]	69,79 %
43	rules	DecisionTable [74]	85,89 %
44	rules	JRip [10]	90,7 %
45	rules	NNge [75]	91,31 %
46	rules	OneR [76][2]	73,4 %
47	rules	PART [11]	89,45 %
48	rules	Ridor [4][5]	90,06 %
49	rules	ZeroR [2][4][5]	40,68 %

Tabela 5.5: Delež pravilno razvrščenih z modeli zgrajenimi s posameznimi algoritmi strojnega učenja. Za generiranje vektorja lastnosti smo vektor projekcije z metodo ZRNI razdelili na 6 delov. Uporabili smo vektorje projekcije, izračunane iz slik segmentiranih z metodo 9. ($ZRNI_6$ S_9)

V nadaljevanju analize postopkov strojnega učenja smo podrobneje raziskali dosežene rezultate delovanja modelov določanja stopnje jakosti aglutinacije. Izračunali smo deleže pravilno razvrščenih za vsak posamezen razred – stopnjo jakosti aglutinacije, v katerega so modeli razvrščali vzorce. Deleže smo izračunali iz matrik pravilno in napačno razvrščenih, ki smo jih dobili v postopku validacije modelov. Analizo smo opravili za kombinacijo dveh metod izračuna vektorjev lastnosti PCA_{10} S_9 in $ZRNI_6$ S_9 z algoritmi strojnega učenja 12: IBk [59], 17: Bagging [63], 23: LogitBoost [67], 37: LMT [8] in 39: RandomForest [9].

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	74	0	0	0	0
1+	0	4	4	1	0	0
2+	0	1	0	11	1	1
3+	0	0	0	0	8	3
4+	3	0	0	0	1	49

Tabela 5.6: Matrika pravilno in napačno razvrščenih $ZRNI_6$ S_9 M_{12}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.6, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.7:

Razred	Delež pravilno razvrščenih
Prazno	87,5 %
NEG	93,7 %
1+	100 %
2+	91,7 %
3+	80 %
4+	92,5 %

Tabela 5.7: Delež pravilno razvrščenih za eksperiment $ZRNI_6$ S_9 M_{12}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	72	2	0	0	0
1+	0	3	6	0	0	0
2+	0	0	0	14	0	0
3+	0	0	0	0	10	1
4+	4	0	0	0	1	48

Tabela 5.8: Matrika pravilno in napačno razvrščenih $ZRNI_6$ S_9 M_{17}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.8, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.9:

Razred	Delež pravilno razvrščenih
Prazno	84 %
NEG	96 %
1+	75 %
2+	100 %
3+	90,9 %
4+	98 %

Tabela 5.9: Delež pravilno razvrščenih za eksperiment ZRNI_6 S_9 M_{17}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	73	1	0	0	0
1+	0	4	5	0	0	0
2+	0	0	1	13	0	0
3+	0	0	0	1	10	0
4+	3	0	0	0	1	49

Tabela 5.10: Matrika pravilno in napačno razvrščenih ZRNI_6, S_9, M_{23}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.10, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.11:

Razred	Delež pravilno razvrščenih
Prazno	87,5 %
NEG	94,8 %
1+	71,4 %
2+	92,9 %
3+	90,9 %
4+	100 %

Tabela 5.11: Delež pravilno razvrščenih za eksperiment ZRNI_6, S_9, M_{23}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	71	3	0	0	0
1+	0	3	6	0	0	0
2+	0	0	0	13	1	0
3+	0	0	0	1	8	2
4+	3	0	0	0	2	48

Tabela 5.12: Matrika pravilno in napačno razvrščenih ZRNI_6 S_9 M_{37}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.12, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.13:

Razred	Delež pravilno razvrščenih
Prazno	87,5 %
NEG	95,9 %
1+	66,7 %
2+	92,9 %
3+	72,7 %
4+	96 %

Tabela 5.13: Delež pravilno razvrščenih za eksperiment ZRNI_6 S_9 M_{37}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	73	0	1	0	0
1+	0	3	5	1	0	0
2+	0	0	0	14	0	0
3+	0	0	0	0	9	2
4+	3	0	0	0	1	49

Tabela 5.14: Matrika pravilno in napačno razvrščenih ZRNI_6 S_9 M_{39}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.14, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.15:

Razred	Delež pravilno razvrščenih
Prazno	87,5 %
NEG	96,1 %
1+	100 %
2+	87,5 %
3+	90 %
4+	96,1 %

Tabela 5.15: Delež pravilno razvrščenih za eksperiment ZRNI_6 S_9 M_{39}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	74	0	0	0	0
1+	0	1	6	2	0	0
2+	0	0	0	13	1	0
3+	0	0	0	0	9	2
4+	4	0	0	0	2	47

Tabela 5.16: Matrika pravilno in napačno razvrščenih PCA_{10} S_9 M_{12}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.16, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.17:

Razred	Delež pravilno razvrščenih
Prazno	84 %
NEG	98,7 %
1+	100 %
2+	86,7 %
3+	75 %
4+	95,9 %

Tabela 5.17: Delež pravilno razvrščenih za eksperiment PCA_{10} S_9 M_{12}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	1	72	1	0	0	0
1+	0	5	1	3	0	0
2+	0	0	1	12	0	1
3+	0	0	0	0	4	7
4+	3	0	0	0	1	49

Tabela 5.18: Matrika pravilno in napačno razvrščenih PCA_{10} S_9 M_{17}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.18, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.19:

Razred	Delež pravilno razvrščenih
Prazno	84 %
NEG	93,5 %
1+	33,3 %
2+	80 %
3+	80 %
4+	86 %

Tabela 5.19: Delež pravilno razvrščenih za eksperiment PCA_{10} S_9 M_{17}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	1	71	2	0	0	0
1+	0	4	5	0	0	0
2+	0	1	0	11	2	0
3+	0	0	0	1	6	4
4+	3	0	0	0	3	47

Tabela 5.20: Matrika pravilno in napačno razvrščenih PCA_{10} S_9 M_{23}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.20, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.21:

Razred	Delež pravilno razvrščenih
Prazno	84 %
NEG	93,4 %
1+	71,4 %
2+	91,7 %
3+	54,5 %
4+	92,2 %

Tabela 5.21: Delež pravilno razvrščenih za eksperiment PCA_{10} S_9 M_{23}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	0	74	0	0	0	0
1+	0	2	6	1	0	0
2+	0	0	0	13	1	0
3+	0	1	0	0	6	4
4+	3	0	0	0	2	48

Tabela 5.22: Matrika pravilno in napačno razvrščenih PCA_{10} S_9 M_{37}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.22, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.23:

Razred	Delež pravilno razvrščenih
Prazno	87,5 %
NEG	96,1 %
1+	100 %
2+	92,9 %
3+	66,7 %
4+	92,3 %

Tabela 5.23: Delež pravilno razvrščenih za eksperiment PCA_{10} S_9 M_{37}.

Razvrščeno:	*Prazno*	*NEG*	1+	2+	3+	4+
Prazno	21	0	0	0	0	0
NEG	1	73	0	0	0	0
1+	0	1	7	1	0	0
2+	0	0	1	12	1	0
3+	0	0	0	0	8	3
4+	3	0	0	0	3	47

Tabela 5.24: Matrika pravilno in napačno razvrščenih PCA_{10} S_9 M_{39}.

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.24, smo izračunali sledeče deleže pravilno razvrščenih in jih podali v tabeli 5.25:

Razred	Delež pravilno razvrščenih
Prazno	84 %
NEG	98,6 %
1+	87,5 %
2+	92,3 %
3+	66,7 %
4+	94 %

Tabela 5.25: Delež pravilno razvrščenih za eksperiment PCA_{10} S_9 M_{39}.

Podatke, ki smo jih predstavili v tem podpoglavju, smo uporabili v analizi najboljše kombinacije algoritmov za celoten sistem za samodejno interpretacijo rezultatov predtransfuzijskih preiskav. Analizo smo podali v podpoglavju 5.7. V sledečem poglavju pa smo podali analizo izbire algoritmov strojnega učenja za izvedbo drugega koraka samodejne interpretacije predtransfuzijskih preiskav: določanja dokončne interpretacije rezultatov preiskav.

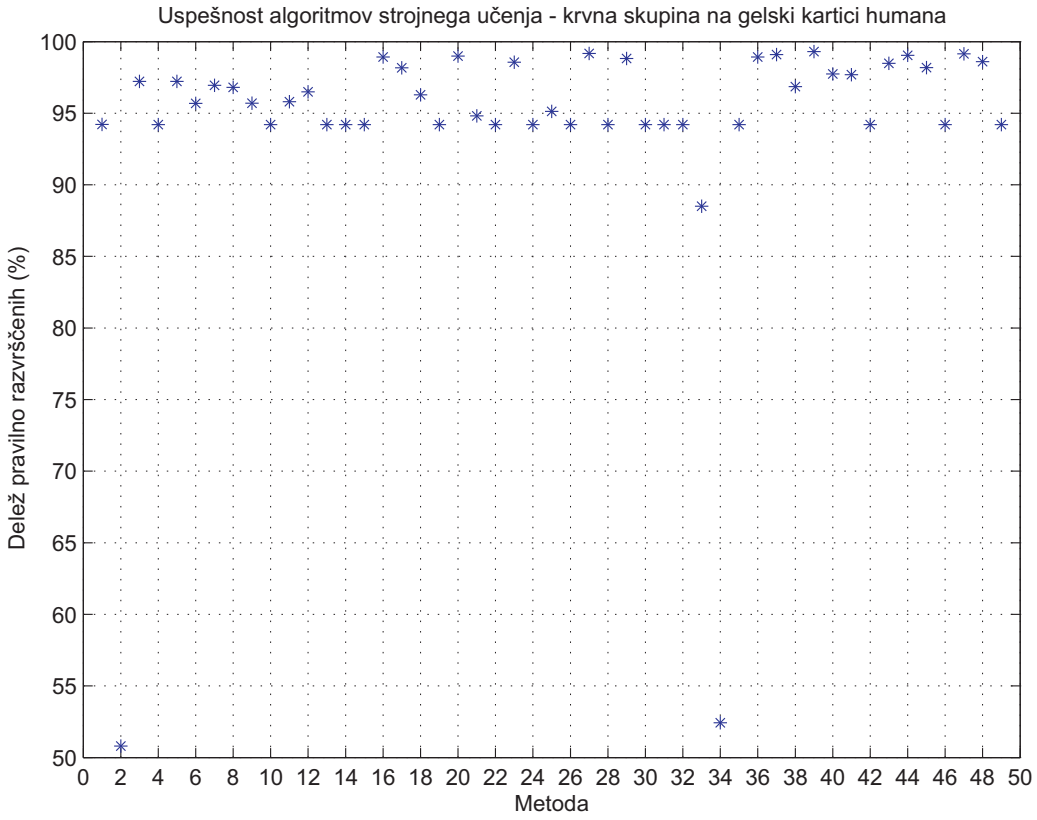
5.6 Rezultati modelov dokončne interpretacije preiskav

V drugem koraku samodejne interpretacije predtransfuzijskih preiskav je potrebno na podlagi določenih stopenj jakosti aglutinacije v posameznih kolonah gelskih kartic, pridobljenih v prvem koraku, določiti dokončno interpretacijo preiskave.

5.6.1 Samodejna interpretacija preiskave KS

Izvedeli smo obširen eksperiment, v katerem smo preizkusili učinkovitost modelov določanja dokončne interpretacije preiskave "Določanje krvne skupine z gelsko kartico *humana*". V sklopu interpretacije preiskave se na podlagi klasifikacije kolon 1:$AntiA$, 2:$AntiB$, 5:A_1 in 6:B določi krvno skupino.

Za gradnjo modela določanja krvne skupine smo preizkusili v tabeli 4.1 naštete algoritme strojnega učenja in med sabo primerjali njihovo uspešnost. Ker nismo imeli na voljo dovolj obširne množice testnih/učnih podatkov, smo podatkovni nabor generirali iz pravilnostnih tabel. Pravilnostne tabele so izdelali specialisti transfuzijske medicine na ZTM. Povzeli smo jih po literaturi [3].



Slika 5.12: Uspešnost algoritmov strojnega učenja za določanje krvne skupine na gelski kartici *humana*

Rezultate uspešnosti modelov, zgrajenih z algoritmi strojnega učenja, smo navedli v tabeli 5.26. Rezultate smo ilustrirali tudi na grafu na sliki 5.12.

Oznaka	Tip metode	Ime metode	Delež pravilno razvrščenih
1	bayes	BayesNet [2]	94,22 %
2	bayes	ComplementNaiveBayes [51]	50,8 %
3	bayes	NaiveBayes [2]	97,22 %
4	bayes	NaiveBayesMultinomial [52]	94,21 %
5	bayes	NaiveBayesUpdateable [53]	97,22 %
6	functions	Logistic [54]	95,69 %
7	functions	MultilayerPerceptron [55]	96,95 %
8	functions	RBFNetwork [56]	96,8 %
9	functions	SimpleLogistic [57]	95,7 %
10	functions	SMO [58]	94,21 %
11	lazy	IB1 [59]	95,81 %
12	lazy	IBk [59]	96,49 %
13	lazy	KStar [60]	94,21 %
14	lazy	LWL [61]	94,21 %
15	meta	AdaBoostM1 [62]	94,21 %
16	meta	AttributeSelectedClassifier [4][5]	98,93 %
17	meta	Bagging [63]	98,17 %
18	meta	ClassificationViaRegression [64]	96,28 %
19	meta	CVParameterSelection [65]	94,21 %
20	meta	Decorate [6]	98,99 %
21	meta	FilteredClassifier [4][5]	94,81 %
22	meta	Grading [66]	94,21 %
23	meta	LogitBoost [67]	98,56 %
24	meta	MultiBoostAB [68]	94,21 %
25	meta	MultiClassClassifier [4][5]	95,13 %
26	meta	MultiScheme [4][5]	94,21 %
27	meta	OrdinalClassClassifier [4][5]	99,17 %
28	meta	RacedIncrementalLogitBoost [4][5]	94,21 %
29	meta	RandomCommittee [4][5]	98,82 %
30	meta	Stacking [69]	94,21 %
31	meta	StackingC [70]	94,21 %
32	meta	Vote [4][5]	94,21 %
33	misc	HyperPipes [4][5]	88,5 %
34	misc	VFI [71]	52,42 %
35	trees	DecisionStump [72]	94,21 %
36	trees	J48 [7]	98,93 %
37	trees	LMT [8]	99,09 %
38	trees	NBTree [73]	96,86 %
39	trees	RandomForest [9]	99,3 %
40	trees	RandomTree [4][5]	97,74 %
41	trees	REPTree [4][5]	97,69 %
42	rules	ConjunctiveRule [4][5]	94,21 %
43	rules	DecisionTable [74]	98,48 %
44	rules	JRip [10]	99,04 %
45	rules	NNge [75]	98,19 %
46	rules	OneR [76][2]	94,21 %
47	rules	PART [11]	99,15 %
48	rules	Ridor [4][5]	98,6 %
49	rules	ZeroR [2][4][5]	94,21 %

Tabela 5.26: Uspešnost delovanja modelov dokončne interpretacije preiskave "Določanje krvne skupine z gelsko kartico *humana*". Uporabili smo kolone 1:$AntiA$, 2:$AntiB$, 5:A_1 in 6:B

Ugotovili smo, da smo učinkovite modele za dokončno interpretacijo predtransfuzijske preiskave: "Določanje krvne skupine z gelsko kartico *humana*" zgradili s sledečimi algoritmi strojnega učenja 16: AttributeSelectedClassifier [4] (94,21 %), 20: Decorate [6] (98,99 %), 27: OrdinalClassClassifier [4][5] (99,17 %), 36: J48 [7] (98,93 %), 37: LMT [8] (99,09 %), 39: RandomForest [9] (99,3 %), 44: JRip [10] (99,04 %) in 47: PART [11] (99,15 %). V nadaljnjem tekstu smo za naštete algoritme strojnega učenja podali matrike pravilnih in napačnih razvrstitev in deleže pravilno razvrščenih za vsako posamezno dokončno interpretacijo preiskave.

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	3	0	0	0	0
/	1	1214	0	0	3	3
A	0	4	16	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	6	0
ABot	0	0	0	0	2	8

Tabela 5.27: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 16: AttributeSelectedClassifier [4][5].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.27, smo izračunali sledeče deleže pravilno razvrščenih 5.28:

Razred	Delež pravilno razvrščenih
0	95,2 %
/	99,4 %
A	100 %
B	100 %
AB	54,5 %
AB ot	72,7 %

Tabela 5.28: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 16: AttributeSelectedClassifier [4][5].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	3	0	0	0	0
/	1	1215	0	0	3	2
A	0	4	16	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	5	1
ABot	0	0	0	0	1	9

Tabela 5.29: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 20: Decorate [6].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.29, smo izračunali sledeče deleže pravilno razvrščenih 5.30:

Razred	Delež pravilno razvrščenih
0	95,2 %
/	99,4 %
A	100 %
B	100 %
AB	83,3 %
AB ot	90 %

Tabela 5.30: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 20: Decorate [6].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	3	0	0	0	0
/	1	1220	0	0	0	0
A	0	4	16	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	6	0
ABot	0	0	0	0	2	8

Tabela 5.31: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 27: OrdinalClassClassifier [4][5].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.31, smo izračunali sledeče deleže pravilno razvrščenih 5.32:

Razred	Delež pravilno razvrščenih
0	95,2 %
/	99,4 %
A	100 %
B	100 %
AB	75 %
AB ot	100 %

Tabela 5.32: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 27: OrdinalClassClassifier [4][5].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	3	0	0	0	0
/	1	1214	0	0	3	3
A	0	4	16	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	6	0
ABot	0	0	0	0	2	8

Tabela 5.33: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 36: J48 [7].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.33, smo izračunali sledeče deleže pravilno razvrščenih 5.34:

Razred	Delež pravilno razvrščenih
0	95,2 %
/	99,4 %
A	100 %
B	100 %
AB	54,5 %
AB ot	72,7 %

Tabela 5.34: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 36: J48 [7].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	2	1	0	0	0
/	3	1216	0	0	1	1
A	0	0	20	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	5	1
ABot	0	0	0	0	0	10

Tabela 5.35: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 37: LMT [8].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.35, smo izračunali sledeče deleže pravilno razvrščenih 5.36:

Razred	Delež pravilno razvrščenih
0	87 %
/	99,8 %
A	95,2 %
B	100 %
AB	83,3 %
AB ot	83,3 %

Tabela 5.36: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 37: LMT [8].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	2	1	0	0	0
/	4	1213	0	0	3	1
A	0	0	20	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	6	0
ABot	0	1	0	0	0	9

Tabela 5.37: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 39: RandomForest [9].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.37, smo izračunali sledeče deleže pravilno razvrščenih 5.38:

Razred	Delež pravilno razvrščenih
0	83,3 %
/	99,8 %
A	95,2 %
B	100 %
AB	66,7 %
AB ot	90 %

Tabela 5.38: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 39: RandomForest [9].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	20	1	2	0	0	0
/	0	1217	0	0	2	2
A	0	0	18	0	0	2
B	0	0	0	16	0	0
AB	0	0	0	0	6	0
ABot	0	2	0	0	0	8

Tabela 5.39: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 44: JRip [10].

Iz dane matrike pravilnih in napačnih razvrstitev dane v tabeli 5.39 smo izračunali sledeče deleže pravilno razvrščenih 5.40:

Razred	Delež pravilno razvrščenih
0	100 %
/	99,8 %
A	90 %
B	100 %
AB	75 %
AB ot	66,7 %

Tabela 5.40: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 44: JRip [10].

Razvrščeno:	0	/	*A*	*B*	*AB*	*ABot*
0	21	1	1	0	0	0
/	1	1217	0	0	1	2
A	0	0	20	0	0	0
B	0	0	0	16	0	0
AB	0	0	0	0	6	0
ABot	0	1	0	0	1	8

Tabela 5.41: Matrika pravilno in napačno razvrščenih: strojno učenje interpretacije na kartici *humana*. Algoritem strojnega učenja 47: PART [11].

Iz dane matrike pravilnih in napačnih razvrstitev, podane v tabeli 5.41, smo izračunali sledeče deleže pravilno razvrščenih 5.42:

Razred	Delež pravilno razvrščenih
0	95,5 %
/	99,8 %
A	95,2 %
B	100 %
AB	75 %
AB ot	80 %

Tabela 5.42: Delež pravilno razvrščenih za eksperiment strojnega učenja na kartici *humana*. Algoritem strojnega učenja 47: PART [11].

Predstavljene deleže pravilno razvrščenih modelov dokončne interpretacije preiskav smo v nadaljevanju postopka izbire najboljše kombinacije algoritmov sistema za samodejno interpretacijo predtransfuzijskih preiskav združili z deleži pravilno razvrščenih z modeli za določanje stopnje jakosti aglutinacije in izbrali najboljšo kombinacijo algoritmov. Rezultate te obravnave smo predstavili v sledečem podpoglavju.

5.7 Ocena deleža uspešnosti in izbira najboljše kombinacije algoritmov

Delež uspešnosti je podatek, s katerim smo opremili vsako odločitev sistema za samodejno interpretacijo rezultatov predtransfuzijskih preiskav. Vrednost predstavlja ocenjeno verjetnost, da je rezultat, ki ga predlaga sistem za samodejno interpretacijo predtransfuzijskih preiskav, pravilen.

5.7.1 Ocena deleža uspešnosti za vektorje stopnje jakosti aglutinacije za določanje krvne skupine

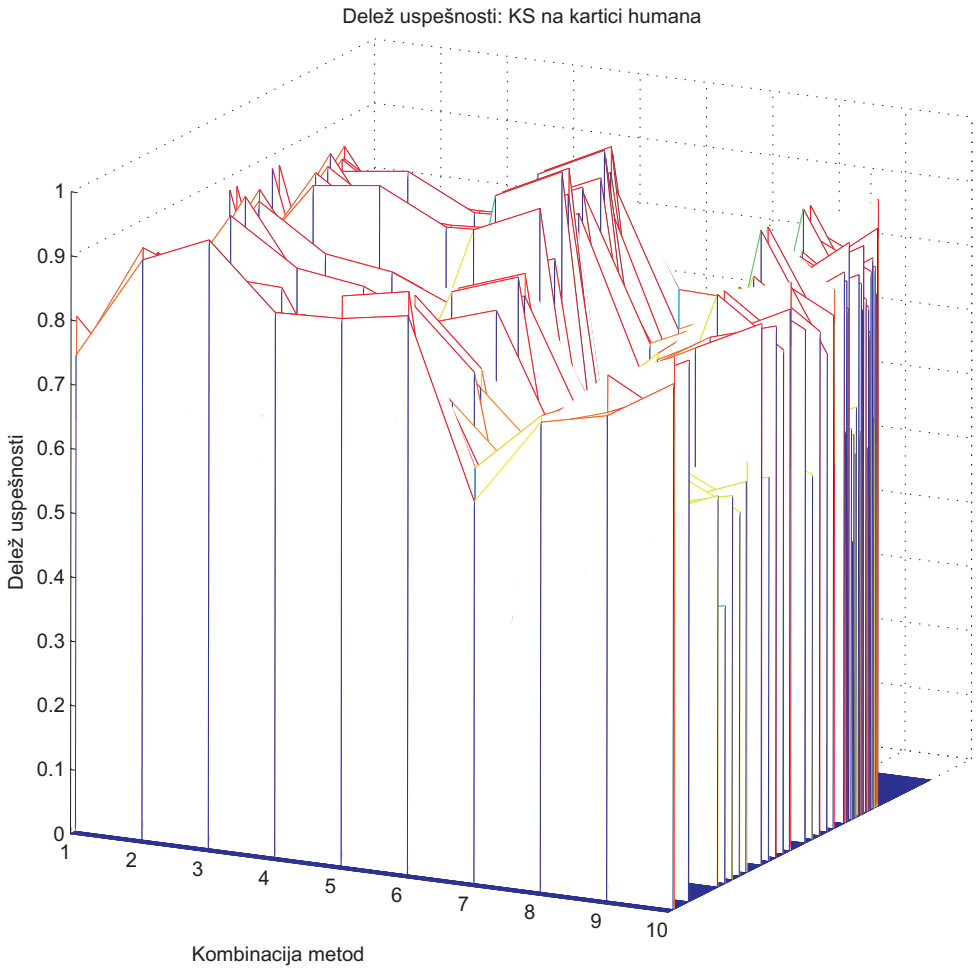
Na podlagi popolne pravilnostne tabele za določanje krvne skupine z uporabo kartice *humana* smo za vsak rezultat zapisali ustrezne vektorje stopenj jakosti aglutinacije. V analizo smo vključili samo kolone na lokacijah, ki so relevantne za določanje krvne skupine. To so kolone na lokacijah *AntiA*, *AntiB*, A_1 in *B*.

Za vsak posamezen vektor smo delež uspešnosti določili tako, da smo med sabo zmnožili posamezne deleže uspešnosti za stopnjo jakosti aglutinacije kolone. Potem smo izračunali povprečno vrednost dobljenih deležev uspešnosti. Izračun smo opravili za posamezne, v ožji izbor uvrščene kombinacije segmentacije slik kolon, metod izračuna vektorjev lastnosti in metod strojnega učenja. Seznam obravnavanih metod z rezultati je v podan tabeli 5.43 in na sliki 5.13.

Ugotovimo, da najboljše rezultate dosežemo z metodo 5 – ZRNI_6 S_9 M_{39} kar pomeni: segmentacija z uporabo metode nelinearnega filtriranja 3, izračun vektorja lastnosti z metodo ZRNI, ki razdeli vektor projekcij na 6 delov, in gradnja modela z algoritmom strojnega učenja RandomForest [9]. Do ugotovitve pridemo tako, da poiščemo maksimalno vrednost povprečnega kombiniranega deleža uspešnosti za nabore stopenj jakosti aglutinacije, ki dajo za preiskavo določanje krvne skupine na kartici *humana* smiselen rezultat.

5.7.2 Ocena deleža uspešnosti za dokončen rezultat za določanje krvne skupine

Izbrali smo najbolj obetavne algoritme strojnega učenja, ki smo jih preizkusili za izdelavo modela interpretacije in deleže pravilno razvrščenih za vsako dokončno interpretacijo, ter



Slika 5.13: Delež uspešnosti vektorjev stopenj jakosti aglutinacije za posamezne rezultate preiskave določanje krvne skupine na gelski kartici *humana*. Rezultati preiskav so določeni s pravilnostno tabelo, podano v [3]. Seznam kombinacije metod je naveden v tabeli 5.43. Na sliki so narisani samo deleži vektorjev, ki po pravilnostni tabeli pomenijo določitev krvne skupine.

	Metoda	Povprečna vrednost	Standardna deviacija
1	$ZRNI_6$ S_9 M_{12}	73 %	7,83 %
2	$ZRNI_6$ S_9 M_{17}	77,1 %	11,49 %
3	$ZRNI_6$ S_9 M_{23}	71,9 %	12,02 %
4	$ZRNI_6$ S_9 M_{37}	63,8 %	13,98 %
5	**$ZRNI_6$ S_9 M_{39}**	**80,8 %**	**5,87 %**
6	PCA_{10} S_9 M_{12}	78,9 %	11,87 %
7	PCA_{10} S_9 M_{17}	45 %	19,37 %
8	PCA_{10} S_9 M_{23}	54,3 %	15,52 %
9	PCA_{10} S_9 M_{37}	72,4 %	14,34 %
10	PCA_{10} S_9 M_{39}	72,1 %	13,21 %

Tabela 5.43: Srednje vrednosti in standardna deviacija deležev uspešnosti vektorjev določene stopnje jakosti aglutinacije v kolonah, relevantnih za določitev krvne skupine s kartico *humana*. Upoštevani so le rezultati, ki pomenijo določitev krvne skupine.

jih množili s pripadajočim deležem uspešnosti vektorja določenih stopenj jakosti aglutinacije, ki je pripeljal do opazovane dokončne interpretacije. Izbrali smo nabor vektorjev, ki smo jih dobili z uporabo $ZRNI_6$ S_9 M_{39}. Povprečne vrednosti deležev uspešnosti za posamezne modele smo predstavili v tabeli 5.44. Najvišji rezultat deleža uspešnosti smo dosegli s kombinacijo sledečih metod: za gradnjo modela določanja stopnje jakosti aglutinacije uporabimo segmentacijo nelinearnega filtriranja 3, za izračun vektorja lastnosti metodo ZRNI, ki razdeli vektor projekcij na 6 delov, te vektorje uporabimo za gradnjo modela z algoritmom strojnega učenja RandomForest [9]; za gradnjo modela dokončne interpretacije pa uporabimo metodo strojnega učenja OrdinalClassClassifier [4][5].

Algoritem strojnega učenja	Povprečna vrednost	Standardna deviacija
M_{20}	77,5 %	7,13 %
M_{27}	**78 %**	**8,28 %**
M_{36}	73,8 %	12,74 %
M_{37}	73,7 %	7,41 %
M_{39}	72,4 %	9,46 %
M_{44}	73,4 %	11,04 %
M_{47}	74,9 %	8,51 %

Tabela 5.44: Srednje vrednosti in standardna deviacija deležev uspešnosti dokončnega določanja krvne skupine s preiskavo za določitev krvne skupine s kartico *humana*. Upoštevani so le rezultati, ki pomenijo določitev krvne skupine. Za gradnjo modela stopnje jakosti aglutinacije je uporabljena kombinacija $ZRNI_6$ S_9 M_{39}.

Poglavje 6

Zaključek

V disertaciji smo predstavili razvoj sistema za samodejno interpretacijo rezultatov predtransfuzijskih preiskav. Obdelali smo celotno pot razvoja, ki zajema spoznavanje s problemom, zajem in registracijo podatkov, izdelavo modelov sistema ter njihovo testiranje.

Obravnavane predtransfuzijske preiskave se izvajajo z gelsko metodo, ki je osnovana na zaznavanju aglutinacije eritrocitov. Gelska metoda se izvaja z gelskimi karticami. Interpretacijo specialisti transfuzijske medicine opravljajo z vizualnim pregledom gelskih kartic. Pri gradnji modelov sistema za samodejno interpretacijo smo posnemali delo specialistov transfuzijske medicine. Zato smo model samodejne interpretacije izdelali v obliki dveh v serijo povezanih modelov. Prvi model določi stopnjo jakosti aglutinacije v vsaki od kolon gelske kartice. Drugi model na podlagi teh določenih stopenj jakosti aglutinacije določi dokončno interpretacijo preiskave.

Podatke za gradnjo sistema za samodejno interpretacijo preiskav smo pridobili iz sistema za telekonzultacije v transfuzijski medicini. Ta sistem smo vpeljali v transfuzijsko prakso v oddelke za transfuzijski medicino po Sloveniji. Z uporabo sistema specialisti transfuzijske medicine na daljavo interpretirajo predtransfuzijske preiskave. Podatki, potrebni za interpretacijo preiskav in pripadajoče interpretacije, se beležijo v sistemu. Iz sistema zajeti podatki predstavljajo učno in testno množico za razvoj modelov interpretacije preiskav.

Za gradnjo posameznih modelov interpretacije preiskav smo uporabili metode strojnega učenja. Med sabo smo primerjali delovanje 49 različnih metod strojnega učenja. Primerjavo smo izvedli z okoljem WEKA.

Primerjali smo tudi vpliv različnih načinov predobdelave podatkov na delovanje algoritmov strojnega učenja. Tako smo raziskovali vpliv različnih segmentacijskih postopkov,

s katerimi smo iz slik gelskih kartic izločili zanimiva področja. V ta namen smo razvili in preizkusili vpliv enajst različnih postopkov segmentacije. Raziskovali smo tudi vpliv različnih metod za izračun vektorjev lastnosti. V ta namen smo implementirali dve različni metodi izračuna vektorjev lastnosti. Prva je bila na področju koncentriranja informacije v čim manj podatkov splošno znana metoda PCA, druga, ki smo jo razvili sami, pa metoda ZRNI. Za metode izračuna vektorjev lastnosti smo na empiričen način določili parametre, pri katerih smo dobili najboljše rezultate. Metode smo izbirali v več korakih. Najprej smo naredili zelo obširen eksperiment, v katerem smo preizkusili vse kombinacije postopkov segmentacije, različno prametriziranih postopkov izračuna vektorjev lastnosti in metod strojnega učenja. Eksperiment je zajemal izdelavo in testiranje modelov strojnega učenja s podatki, izračunanimi na opisani način. Zaradi razmeroma male množice podatkov, ki so bili na voljo, smo za izdelavo in testiranje modelov uporabili metodo navzkrižne validacije, ki je opisana v [2]. Med sabo smo primerjali skupne deleže pravilno razvrščenih vzorcev z naučenim modelom. V nadaljevanju smo kandidate, ki so se obnesli najbolje, med sabo primerjali glede na vpliv na dokončno interpretacijo preiskav. Vpliv smo preverili z uporabo deleža uspešnosti za pravilno celotno kombinacijo vektorjev določenih stopenj jakosti aglutinacije. Postopek je opisan v podpoglavju 4.5.1.

Ugotovili smo, da najboljše rezultate modela za določanje stopnje jakosti aglutinacije dobimo, če uporabimo metodo, ki smo jo poimenovali metoda nelinearnega filtriranja. Metoda je opisana v podpoglavju 4.3.2.1.1. Ugotovili smo, da je rezultate vektorjev porazdelitve eritrocitov po višini kolone, ki jih izračunamo tako, da posamezne segmentirane slike kolon projeciramo na os y, z vidika učinkovitosti grajenja modela z algoritmi strojnega učenja najbolje preračunati v vektor lastnosti z metodo ZRNI z uporabo 6 razdelkov. Metoda je opisana v podpoglavju 4.3.3.2. Ko imamo na voljo na opisani način izračunane vektorje lastnosti, je za izgradnjo modela določanja stopnje jakosti aglutinacije v kolonah najbolje uporabiti metodo strojnega učenja iz skupine gradnje dreves – RandomForest [9]. Za uporabo omenjene metode smo za določanje rezultata krvne skupine na kartici *humana* ocenili povprečni delež kombinirane uspešnosti za preiskavo relevantnih stopenj jakosti aglutinacije vektorja na 80,8 %, s standardno deviacijo $\sigma = 5,87$ %.

V nadaljevanju smo za model interpretacije stopenj jakosti aglutinacije v dokončno interpretacijo preiskave izbrali algoritme strojnega učenja, ki so primerni za reševanje tega problema. Za eksperimentiranje smo si izbrali določanje krvne skupine z gelsko kartico *humana*. Testno in učno množico smo generirali sami iz pravilnostne tabele, ki je podana v [3]. Na podatkih smo izvedli obširen eksperiment, v katerega smo vključili 49 različnih

algoritmov strojnega učenja. Ugotovili smo, da najboljše rezultate dobimo z uporabo metode OrdinalClassClassifier [94]. Z metodo razviti model je pravilno razvrščal v 99,17 %. Kombinirani delež uspešnosti za napovedovanje interpretacij preiskave krvne skupine na gelski kartici *humana* z deležem kombinirane uspešnosti za preiskavo relevantnimi stopnjami jakosti aglutinacije vektorja v povprečju znaša 78 %, s standardno deviacijo $\sigma = 8,28$ %.

Rezultati kažejo, da je izdelava sistema za samodejno interpretacijo z uporabo izbranih algoritmov smiselna. Sistem za samodejno interpretacijo bo predstavljal pomembno podporno orodje osebju, ki dela na področju transfuzije krvi. S predlaganjem interpretacij bo olajšal delo specialistom, služil pa bo lahko tudi kot sistem za kontrolo napak. Dokončno interpretacijo pa mora še vedno potrditi specialist transfuzijske medicine.

6.1 Nadaljnje delo

Ob razvoju sistema se nam je porodila kopica idej, s katerimi bi lahko izboljšali delovanje sistema. Idej zaradi pomanjkanja različnih virov (časa, podatkov) še nismo izpeljali. V nadaljevanju smo podali oris teh idej.

6.1.1 Razširjen preizkus

Sistem za samodejno interpretacijo rezultatov predtransfuzijskih preiskav je potrebno preizkusiti tudi z ostalimi preiskavami in rezultati ostalih preiskav. Te preiskave zajemajo indirektni Coombsov test, direktni Coombsov test, navzkrižni preizkus in specifikacijo protiteles. Za te preizkuse je potrebno zbrati dovolj rezultatov, določenih s strani specialistov transfuzijske medicine. Rezultati se zbirajo v sistemu za telekonzultacije v transfuzijski medicini.

6.1.2 Dinamično izbiranje modela za interpretacijo

Ker poznamo delež uspešnosti za posamezen rezultat, napovedan s kombinacijo posameznih metod, lahko izdelamo kompleksnejši sistem, ki bo združeval rezultate, napovedane z uporabo različnih metod. Verjetno je, da bo ta kompleksnejši sistem izboljšal pravilnost napovedovanja rezultatov.

V predstavljenem delu je uporabljen pristop, pri katerem se odločimo za eno kombinacijo algoritma za segmentacijo, izračun vektorjev lastnosti in algoritma strojnega

učenja. Izbrali smo tisto kombinacijo, ki je bila v povprečju najuspešnejša. To kombinacijo algoritmov uporabljamo za napovedovaje vseh rezultatov.

V predlaganem sistemu z dinamičnim izbiranjem modela interpretacije naj se rezultati preiskav izračunavajo ločeno z vsemi razvitimi metodami. Na koncu se rezultate metod združi z upoštevanjem deleža uspešnosti posameznega rezultata. Predlagamo več načinov za upoštevanje deleža uspešnosti:

Lahko se odločimo za preprost algoritem, ki bo izbral tisti rezultat, pri katerem bo ocenjeni delež uspešnosti največji.

Lahko pa rezultate kombiniramo na tak način, da jih otežimo z utežmi, ki predstavljajo pri razvoju modelov ocenjene deleže uspešnosti.

6.1.3 Vpeljava sistema v realno prakso

Rezultat doktorske disertacije so izbrani parametrizirani algoritmi za izdelavo sistema za samodjeno interpretacijo rezultatov. Del teh algoritmov je napisan v programskem okolju Matlab, del je napisan v Javi, del algoritmov pa obstaja kot paket v okolju WEKA. Ker je WEKA odprto-kodni projekt, imamo dostop do izvorne kode vseh algoritmov strojnega učenja. Celotno okolje WEKA je napisano v programskem jeziku Java.

Za vpeljavo sistema v realno prakso je potrebno vso kodo združiti v programski modul in ga vključiti v sistem za telekonzultacije v transfuzijski medicini. Del kode, ki predstavlja algoritme, napisane v programskem okolju Matlab, je potrebno prevesti v programski jezik Java.

6.2 Prispevki znanosti

1. **Modeliranje postopka interpretacije predtransfuzijskih testiranj z gelsko metodo v dveh korakih.** Z modelom smo posnemali postopek interpretacije predtransfuzijskih testiranj, kot ga opravljajo specialisti transfuzijske medicine. V prvem koraku smo določili stopnjo jakosti aglutinacije posameznih kolon slike gelske kartice. Določene stopnje jakosti aglutinacije predstavljajo vmesni rezultat postopka. V drugem koraku smo na podlagi določenih stopenj jakosti aglutinacije in tipa preiskave določili dokončno interpretacijo predtransfuzijskega testiranja.

2. **Izbira najprimernejših algoritmov strojnega učenja za modeliranje obeh korakov interpretacije predtransfuzijskih testiranj z gelsko metodo.** Z

metodami strojnega učenja smo zgradili modele, ki modelirajo posamezna koraka interpretacije predtransfuzijskih testiranj. Za gradnjo modelov smo za vsak korak preizkusili 49 različnih algoritmov strojnega učenja. Modele z algoritmi strojnega učenja smo zgradili na osnovi podatkovne zbirke, pridobljene iz sistema za telekonzultacije, v katerem se beležijo interpretacije predtransfuzijskih testiranj, ki so jih opravili specialisti transfuzijske medicine. Na podlagi analize deleža uspešnosti modela postopka interpretacije smo izbrali najprimernejšo kombinacijo algoritmov za gradnjo modela postopka interpretacije.

3. **Gradnja podatkovne zbirke slikovnih diagnostičnih podatkov predtransfuzijskih testiranj z gelsko metodo s pripadajočimi interpretacijami.** Podatkovno zbirko slikovnih diagnostičnih podatkov predtransfuzijskih testiranj z gelsko metodo, opremljenih z interpretacijami preiskav, ki so jih določili specialisti transfuzijske medicine, potrebujemo za gradnjo učne in testne množice. Z učno in testno množico z uporabo algoritmov strojnega učenja zgradimo in preizkusimo modele za modeliranje interpretacije predtransfuzijskih testiranj. Podatkovno zbirko smo zbrali z uvedbo sistema za telekonzultacije v transfuzijski medicini, s katerim specialisti transfuzijske medicine na daljavo interpretirajo predtransfuzijska testiranja.

4. **Razvoj in analiza uspešnosti algoritmov za segmentacijo slik kolon gelskih kartic.** Razvili in preizkusili smo 11 segmentacijskih algoritmov, ki delujejo v osnovnem prostoru slike. Algoritmi ločijo slikovne elemente na podlagi različnih lastnosti posameznih komponent barvnih prostorov, v katere smo preslikali opazovane slike kolon. Algoritmi delujejo tako, da kombinirajo posamezne komponente barvnih prostorov na tak način, da med slikovnimi elementi poudarijo razlike, na podlagi katerih lahko ločimo zanimive slikovne elemente od nezanimivih. Algoritme smo razvili na empiričen način. Kriterij uspešnosti posameznega algoritma je bil vpliv le-tega na uspešnost delovanja modela za določanje stopnje jakosti aglutinacije v kolonah, kateri je bil zgrajen z algoritmi strojnega učenja s podatki, ki smo jih obdelali z opazovanim segmentacijskim algoritmom.

Literatura

[1] Marko Breskvar, Irena Bricl, Jurij F. Tasič, Marko Meža, and Primož Rožman. Zagotavljanje kakovosti v transfuzijski službi z uporabo telekonzultacij. In *E-zdravje v e-Sloveniji : zbornik kongresa Slovenskega društva za medicinsko informatiko*, pages 241–250, Bled, Dec 2004. Slovensko društvo za medicinsko informatiko.

[2] Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition.* Morgan Kaufmann Publishers Elsevier, 2005.

[3] Polonca Stopar. Avtomatizacija aglutinacijskih imunohematoloških preiskav, Maj 2006. Diplomsko delo na visokošolskem strokovnem programu laboratorijske biomedicine na Fakulteti za farmacijo.

[4] WekaDoc. The documentation project for weka. Dostopno na http://weka.sourceforge.net/wekadoc/index.php/en:Primer. Zadnji dostop 6. april 2006.

[5] Weka machine learning project. Dostopno na http://www.cs.waikato.ac.nz/ ml/, 2007. Zadnji dostop 23. 3. 2007.

[6] Prem Melville and Ray Mooney. Constructing diverse classifier ensembles using artificial training examples. In *In Proc. of 18th Intl. Joint Conf. on Artificial Intelligence IJCAI 2003*, pages 505–510, Acapulco, Mexico, Avg 2003.

[7] Ross Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo, CA., 1993.

[8] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. In Ljupco Todorovski Hendrik Blockeel Nada Lavrac, Dragan Gamberger, editor, *Machine Learning: ECML 2003, 14th European Conference on Machine Learning*, pages 241–252, Cavtat-Dubrovnik, Croatia, September 2003.

[9] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Okt 2001.

[10] William W. Cohen. Fast effective rule induction. In Stuart Russell Armand Prieditis, editor, *Proc. of the 12th International Conference on Machine Learning*, pages 115–123, Tahoe City, CA, Jul 1995. Morgan Kaufmann.

[11] Eibe Frank and Ian H. Witten. Generating accurate rule sets without global optimization. In Shavlik J., editor, *Machine Learning: Proceedings of the Fifteenth International Conference*. Morgan Kaufmann Publishers, 1998.

[12] Y. Lapierre, D. Rigal, J. Adam, D. Josef, F. Meyer, S. Greber, and C. Drot. The gel test: a new way to detect red cell antigen-antibody reactions. *Transfusion*, 30(2):109–13, Feb 1990.

[13] M. M. Langston, J. L. Procter, K. M. Cipolone, and D. F. Stroncek. Evaluation of the gel system for abo grouping and d typing. *Transfusion*, 39(3):300–5, Mar 1999.

[14] Lindsay I. Smith. A tutorial on principal components analysis. Dostopno na http:// csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf, Feb 2002. Zadnji dostop 24. april 2006.

[15] Marko Meža, Marko Breskvar, Andrej Košir, Irena Bricl, Jurij F. Tasič, and Primož Rožman. Telemedicine in the blood transfusion laboratory - remote interpretation of pre-transfusion tests. *Journal of telemedicine and telecare*, 13(7):357–362, Okt 2007.

[16] Primož Rožman and Dragoslav Domanović. Transfusion medicine in slovenia - current status and future challenges. *Transfusion medicine and haemotherapy*, 33:420–426, 2006.

[17] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning - Data Mining, Inference and Prediction*. Springer, 2001.

[18] Marko Meža, Marko Breskvar, and Jurij F. Tasič. Arhitektura sistema za telekonzultacije v transfuzijski medicini. *Elektrotehniški vestnik*, 72(2/3):145–151, 2005.

[19] Bret Harnett. Telemedicine systems and telecommunications. *Journal of telemedicina and telecare*, 12(1):4–15, Jan 2006.

[20] Marko Breskvar, Irena Bricl, Jurij F. Tasič, Marko Meža, and Primož Rožman. Telekonzultacije v transfuzijski službi. *Zdravniški vestnik*, 73:105–108, 2004.

[21] Marko Breskvar, Irena Bricl, Jurij F. Tasič, Marko Meža, and Primož Rožman. Telemedicine in the blood transfusion service. *Vox Sanguinus*, 87(3):142, Jul 2004.

[22] Marko Meža, Matevž Pogačnik, Marko Tkalčič, Andraž Jere, Marko Breskvar, Primož Rožman, Irena Bricl, Jurij F. Tasič, and Marijan Leban. Description of pilot implementation of telemedicine system in blood transfusion practice. In Mezaris Vasileios Strintzis Michael, Kompatsiaris Ioannis, editor, *Proceedings of the COST. Thessaloniki: Informatics and Telematics Institute, Centre for Research and Technology*, pages 61–65, Thessaloniki, 2004.

[23] Marko Meža. Support of the blood transfusion diagnostic process with telemedicine. In Milić Ljiljana, editor, *Proceedings EUROCON 2005 - The International Conference on Computer as a Tool*, pages 195–198. University of Belgrade, School of Electrical Engineering: Institute of Electrical and Electronics Engineers, Nov 2005.

[24] Primož Rožman, Marko Meža, Marko Breskvar, Irena Bricl, Božidar Voljč, and Jurij F. Tasič. Closing the information loops in the algorhitms of transfusion medicine. part 1: teleconsultation. *Maked. med. pregl.*, 58(63):33–37, 2004.

[25] Primož Rožman, Irena Bricl, Matjaž Urbajs, Marjeta Maček, Marko Breskvar, Marko Meža, and Jurij F. Tasič. A new teleconsultation system for pretransfusion testing. *Vox Sanguinus*, 91(3):311–312, 2006.

[26] Marko Meža, Jurij F. Tasič, Tomaž Finkšt, Marko Breskvar, Primož Rožman, and Irena Bricl. Pilotni sistem telemedicine v transfuzijski službi republike slovenije. In Trost Andrej Zajc Baldomir, editor, *Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK 2005*, volume B, pages 322–325, Portorož, Slovenija, Sept 2005. IEEE Region 8, Slovenska sekcija IEEE.

[27] Marko Breskvar, Irena Bricl, Polonca Stopar, Jurij F. Tasič, Marko Meža, and Primož Rožman. Pilotna uvedba telekonzultacij v transfuzijsko službo. In *Zdravje na informacijski poti: zbornik kongresa Slovenskega društva za medicinsko informatiko*, Zreče, Slovenija, Apr 2006. Slovensko društvo za medicinsko informatiko.

[28] Apache tomcat. Dostopno na http://tomcat.apache.org/, 2007. Zadnji dostop 15. 6. 2007.

[29] Java media framework api (jmf). Dostopno na http://java.sun.com/products/java-media/jmf/, 2007. Zadnji dostop 15. 6. 2007.

[30] RFC. Rfc 768 user datagram protocol. Dostopno na http://www.faqs.org/rfcs/rfc768.html, 2007. Zadnji dostop 15. 6. 2007.

[31] H.323 standard. Dostopno na http://www.packetizer.com/voip/h323/standards.html, 2007. Zadnji dostop 15. 6. 2007.

[32] Mysql manual. Dostopno na http://dev.mysql.com/doc/mysql/en/index.html. Zadnji dostop 6. 3. 2007.

[33] Java technology. Dostopno na http://java.sun.com/, 2007. Zadnji dostop 15. 6. 2007.

[34] Java se - java db and java database connectivity (jdbc). Dostopno na http://java.sun.com/javase/technologies/database/, 2007. Zadnji dostop 15. 6. 2007.

[35] Wikipedia. Virtual private network. Dostopno na http://en.wikipedia.org/wiki/Vpn, 2007. Zadnji dostop 15. 6. 2007.

[36] Marko Breskvar and Ljubiša Lukić. Datec - informacijski sistem v transfuziologiji. *Bilt.-ekon. organ. inform. zdrav.*, 11(2):39–42, Feb 1995.

[37] Marko Breskvar and Ljubiša Lukić. Deset let informacijskega sistema v slovenski trnasfuziologiji = [ten years of information systems in slovenian transfusiology]. *Bilt.-ekon. organ. inform. zdrav.*, 16(4):100–103, 2000.

[38] Wikipedia. Vt100 video terminal. Dostopno na http://en.wikipedia.org/wiki/VT100, Mar 2007. Zadnji dostop 5. 3. 2007.

[39] Nikola Pavešić. *Razpoznavanje vzorcev, Uvod v analizo in razumevanje vidnih in slušnih signalov 2. razširjena izdaja*. Univerza v Ljubljani, Fakulteta za elektrotehniko, 2. razširjena izdaja edition, 2000.

[40] Wikipedia. Data clustering. Dostopno na http://en.wikipedia.org/wiki/Data_clustering, Jan 2007. Zadnji dostop 8.1.2007.

[41] Matevž Pogačnik. *Uporabniku prilagojeno iskanje multimedijskih vsebin*. PhD thesis, Univerza v Ljubljani, Fakulteta za elektrotehniko, 2004.

[42] I.N. Bronštejn, K.A. Semendjajev, G. Musiol, and H. Muhlig. *Matematični priročnik*. Tehniška založba Slovenije, 1977.

[43] Wikipedia. Information entropy. Dostopno na http://en.wikipedia.org/wiki/Information_entropy. Zadnji dostop 15. 4. 2007.

[44] Margaret H. Dunham. *Data Mining - Introductory and Advanced Topics*. Prentice Hall, 2003.

[45] Wikipedia. Unsupervised learning. Dostopno na http://en.wikipedia.org/wiki/Unsupervised_learning, Jan 2007. Zadnji dostop 8.1.2007.

[46] A tutorial on clustering algorithms. Dostopno na http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/. Zadnji dostop 8.1.2007.

[47] Brian T. Luke. Divisive clustering. Dostopno na http://ncisgi.ncifcrf.gov/ lukeb/diclust.html, Jan 2007. Zadnji dostop 15. 1. 2007.

[48] Wikipedia. K-means algorithm. Dostopno na http://en.wikipedia.org/wiki/K-means, Jan 2007. Zadnji dostop 8.1.2007.

[49] Andrej Košir and Tilen Mlakar. Optimal image and video feature selection procedure. 2006.

[50] Gregory M. Palmer, Changfang Zhu, Tara M. Breslin, Fushen Xu, Kennedy W. Gilchrist, and Nirmala Ramanjujam. Comparison of multiexcitation fluorescence and diffuse reflectance spectroscopy for the diagnosis of breast cancer. *IEEE Transactions on bieomedical engineering*, 50(11):1233–1242, Nov 2003.

[51] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In Mishra N. In Fawcett, T., editor, *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, pages 616–623, Washington, D.C., 2003. Artificial Intelligence Laboratory; Massachusetts Institute of Technology; Cambridge, MA 02139, AAAI Press (2003).

[52] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[53] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In San Mateo Morgan Kaufmann, editor, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[54] S. le Cessie and J. C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[55] Wikipedia. Perceptron. Dostopno na http://en.wikipedia.org/wiki/Perceptron, Avg 2006. Zadnji dostop 24. avgust 2006.

[56] E. P. Maillard and D. Gueriot. Rbf neural network, basis functions and genetic algorithm. In *International Conference on Neural Networks*, volume 4, pages 2187 – 2192. GESMA, Brest-Naval, Jun 1997.

[57] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59(1-2):161 – 205, May 2005.

[58] B. Schoelkopf, C. Burges, and A. Smola. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[59] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[60] John, G. Cleary, Leonard, and E. Trigg. Kstar: An instance-based learner using an entropic distance measure. In *Proceedings of the 12th International Conference on Machine learning*, pages 108–114, 1995.

[61] Eibe Frank, Mark Hall, and Bernhard Pfahringer. Locally weighted naive bayes. In *Conference on Uncertainty in AI*, 2003.

[62] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In San Francisco Morgan Kaufmann, editor, *Proc International Conference on Machine Learning*, pages 148–156, 1996.

[63] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[64] E. Frank, Y. Wang, S. Inglis, G. Holmes, , and I.H. Witten. Using model trees for classification. *Machine Learning*, 32(1):63–76, 1998.

[65] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Department of Computer Science, Stanford University, 1995.

[66] A.K. Seewald and J. Fuernkranz. An evaluation of grading classifiers, in hoffmann f. et al. (eds.). In *Advances in Intelligent Data Analysis, 4th International Conference, IDA 2001, Proceedings*, pages 115–124. Springer Berlin/Heidelberg/New York/Tokyo, 2001.

[67] J. Friedman, T. Hastie, and R. Tibshiran. Additive logistic regression: a statistical view of boosting. Technical report, Stanford University, 1998.

[68] Geoffrey I. Webb. Multiboosting: A technique for combining boosting and wagging. *Machine Learning*, 40(2):159–196, 2000.

[69] David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

[70] A.K. Seewald. How to make stacking better and faster while also taking care of an unknown weakness. In Hoffmann A. Sammut C., editor, *roceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, pages 554–561. Morgan Kaufmann Publishers, 2002.

[71] Gulsen Demiroz and H. Altay Guvenir. Classification by voting feature intervals. In *European Conference on Machine Learning ECML-97*, pages 85–92, 1997.

[72] Len Trigg Mark Hall Geoffrey Holmes Ian H. Witten, Eibe Frank and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations. Dostopno na http://www.cs.waikato.ac.nz/ eibe/pubs/99IHW-EF-LT-MH-GH-SJC-Tools-Java.ps.gz. Zadnji dostop 18.6.2007.

[73] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision tree hybrid. In *Procedings of the Second Internaltional Conference on Knoledge Discovery and Data Mining*, 1996.

[74] Ron Kohavi. The power of decision tables. In Wrobel Stefan Lavrac Nada, editor, *Proceedings of the European Conference on Machine Learning*, Lecture Notes in Artificial Intelligence 914, pages 174–189, Berlin, Heidelberg, New York, 1995. Springer Verlag.

[75] Martin Brent. Instance-based learning : Nearest neighbor with generalization. Master's thesis, University of Waikato, Hamilton, New Zealand, 1995.

[76] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[77] The castor project homepage. Dostopno na http://www.castor.org/, Jan 2007. Zadnji dostop 25. 1. 2007.

[78] Xml homepage. Dostopno na http://www.xml.com/. Zadnji dostop 25. 1. 2007.

[79] Altova xmlspy product homepage. Dostopno na http://www.altova.com/products/xmlspy/xml_editor.html, Jan 2007. Zadnji dostop 25. 1. 2007.

[80] Wikipedia. Image registration. Dostopno na http://en.wikipedia.org/wiki/Image registration, 2007. Zadnji dostop 15. 6. 2007.

[81] Marko Meža, Marko Tkalčič, Marko Breskvar, Irena Bricl, Primož Rožman, and Jurij F. Tasič. Registracija rotacije slik gelskih kartic. In Zajc Baldomir and Trost Andrej, editors, *Zbornik petnajste mednarodne Elektrotehniške in računalniškekonference ERK 2006*, volume B, pages 209–212, Portorož, Slovenija, Sept 2006. IEEE Region 8, Slovenska sekcija IEEE.

[82] Wikipedia. Total variation. Dostopno na http://en.wikipedia.org/wiki/Statistical distance, 2007. Zadnji dostop 15. 6. 2007.

[83] Bill Green. Edge detection tutorial. Dostopno na http://www.pages.drexel.edu/ weg22/edge.html, 2002. Zadnji dostop 20.10.2006.

[84] Bill Green. Canny edge detection tutorial. Dostopno na http://www.pages.drexel.edu/ weg22/can_tut.html, 2002. Zadnji dostop 20.10.2006.

[85] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679 – 698, Nov 1986.

[86] Tomaž Finkšt, Marko Meža, and Jurij F. Tasič. Razčlenjevanje barvnih slik z upoštevanjem percepcije vizualne informacije. In Trost Andrej Zajc Baldomir, editor, *Zbornik štirinajste mednarodne Elektrotehniške in računalniške konference ERK*

2005, volume B, pages 183–186, Portorož, Slovenija, Sept 2005. IEEE Region 8, Slovenska sekcija IEEE.

[87] Wikipedia. Color space. Dostopno na http://en.wikipedia.org/wiki/Color_space. Zadnji dostop 20. september 2006.

[88] Wikipedia. Ycbcr. Dostopno na http://en.wikipedia.org/wiki/YCbCr. Zadnji dostop 29. avgust 2006.

[89] Wikipedia. Lab color space. Dostopno na http://en.wikipedia.org/wiki/Lab_color_space, 2007. Zadnji dostop 4. jun. 2007.

[90] Wikipedia. Opponent process. Dostopno na http://en.wikipedia.org/wiki/Opponent_process, 2007. Zadnji dostop 15. 6. 2007.

[91] Ronney B. Panerai, Ana Luisa A. S. Ferreira, and Orlando F. Brum. Principal component analysis of multiple blood flow derived signals. *IEEE Transactions on biomedical engineering*, 35(7):533–538, Jul 1988.

[92] Paolo Ravazzani, Gabriella Tognola, Marta Parazzini, and Ferdinando Grandori. Principal component analysis as a method to facilitate fast detection of transient-evoked octoacoustic emissions. *IEEE Transactions on bieomedical engineering*, 50(2):249–252, Feb 2003.

[93] M. Petrou and P. Bosdogianni. *Image Processing: The Fundamentals*. John Wiley and Sons Ltd, West Sussex, West Sussex, 1999.

[94] Eibe Frank and Mark Hall. A simple approach to ordinal prediction. In *12th European Conference on Machine Learning, Freiburg, Germany*.

[95] Branko Kavšek. *Odkrivanje podskupin z uporabo algoritmov za učenje pravil*. PhD thesis, Univerza v Ljubljani, Fakulteta za racunalništvo in informatiko, 2004.

Poglavje 7

Izjava

Izjavljam, da sem doktorsko disertacijo izdelal samostojno v Laboratoriju za digitalno obdelavo signalov slik in videa na Fakulteti za elektrotehniko Univerza v Ljubljani pod vodstvom mentorja prof. dr. Jurija Tasiča. Izkazano pomoč drugih sodelavcev sem v celoti izrekel v zahvali.

Marko Meža

Dodatek A

Priloge

A.1 Terminološki slovarček

Agglutinate	Aglutinat, strdek
Agglutination	Aglutinacija
Antigen-antibody reactions	Reakcije protiteles
Association	Asociiranje
Association rule	Asociacijsko pravilo
Attribute	Značilka, lastnost, atribut
Classification	Klasifikacija, razvrščanje
Classification rule	Klasifikacijsko pravilo
Classifier	Razrvščevalnik
Clustering	Rojenje
Concept description	Opis koncepta
Confussion matrix	Matrika pravilno in napačno razvrščenih; matrika pravilnih in napačnih razvrstitev
Covering algorithms	Algoritmi s pokrivanjem; konstruiranje pravil
Cross validation	Navzkrižna validacija; prečno preverjanje [95]
Decision tree	Odločitveno drevo
Error Rate	Delež napačnih
Euclidian distance	Evklidova razdalja
Example	Vzorec

False negative = type II error	Delež zgrešenih
False positive = type I error	Delež napačno razvrščenih
Fold	Pregib
Feature vector	Vektor lastnosti (značilk)
Gel card	Gelska kartica
Information gain	Doprinos informacije
Instance	Vzorec
K-Means	Metoda K-tih povprečij
Leaf	List v odločitvenem drevesu
Leave one out validation	Validacija izpusti enega
N-fold cross validation	N-pregibna navzkrižna validacija
Measure of purity	Mera čistosti
Naive Bayes	Naivni Bayes
Node	Vozlišče v odločitvenem drevesu
Numeric prediction	Numerično napovedovanje
Polycation	Polikation
Potentiator	Potenciator
Precision	Delež pravilno razvrščenih
Principle component analysis	Analiza glavnih komponent
Recall, true positive	Delež pravilno najdenih
Root node	Korensko vozlišče
Success rate	Delež uspešnosti
Total variance	Totalna variacija
Tresholding	Upragovljanje
Truth table	Pravilnostna tabela