Web aplikacija za procjenu rizika obolijevanja i praćenje zdravstvenog stanja pacijenata s bolestima bubrega klasifikacijskim postupcima analize podataka

Varnica, Marin

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:095520

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-24**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

WEB APLIKACIJA ZA PROCJENU RIZIKA OBOLIJEVANJA I PRAĆENJE ZDRAVSTVENOG STANJA PACIJENATA S BOLESTIMA BUBREGA KLASIFIKACIJSKIM POSTUPCIMA ANALIZE PODATAKA

Diplomski rad

Marin Varnica

Osijek, 2020.

SADRŽAJ

1. UVOD 1

2. BOLESTI BUBREGA I POTPORA LIJEČENJU 2

2.1. Rad bubrega 2

2.2. Bolesti bubrega 2

2.2.1. Kronično zatajenje bubrega 2

2.2.2. Akutno zatajenje bubrega 3

2.2.3. Ostale bolesti bubrega 4

2.3. Čimbenici rizika obolijevanja od bolesti bubrega 5

2.3.1. Arterijska hipertenzija 5

2.3.2. Hematološki čimbenici 6

2.3.3. Biokemijski čimbenici 7

2.3.4. Čimbenici iz urina 9

3. TRENUTNO STANJE U PODRUČJU LIJEČENJA, PRAĆENJA I PREVENCIJE BOLESTI BUBREGA 11

3.1. Postojeća rješenja općenito u medicini 11

3.2. Postojeća rješenja s gledišta tehnologije 12

4. PRIJEDLOG MODELA WEB APLIKACIJE I ANALIZA PODATAKA 14

4.1. Funkcionalni i nefunkcionalni zahtjevi na web aplikaciju 14

4.2. Građa i cjeline web aplikacije 16

4.2.1. Obrazac programske arhitekture 17

4.3. Prikladni postupci analize podataka 19

4.3.1. Logistička regresija 20

4.3.2. Neuronska mreža 21

4.3.3. Slučajna šuma 21

4.4. Baza podataka korištena za analizu podataka 22

5. PROGRAMSKO RJEŠENJE WEB APLIKACIJE 24

5.1. Razvojna okolina u oblaku računala 24

5.1.1. Implementacija postupka analize podataka 27

5.2. Potrebne programske tehnologije i jezici 29

5.2.1. LAMP 29

5.2.2. HTML i CSS 30

5.3. Programsko rješenje na strani poslužitelja 32

5.3.1. Baza podataka web aplikacije 32

5.3.2. Izrada korisničkog računa 35

5.3.3. Prijava korisnika 37

5.3.4. Unos parametara rizika 39

5.4. Programsko rješenje na strani korisnika 42

5.4.1. Obrasci unosa podataka 42

5.4.2. Prikaz pogrešaka 45

5.4.3. Tablice rezultata 45

6. PRIKAZ RADA I NAČIN KORIŠTENJA WEB APLIKACIJE 48

6.1. Način korištenja web aplikacije 48

6.2. Ispitivanje rada web aplikacije 49

6.2.1. Rezultati arterijske hipertenzije 51

6.2.2. Rezultati hematoloških čimbenika 51

6.2.3. Rezultati biokemijskih čimbenika 52

6.2.4. Rezultati čimbenika iz urina 53

6.2.5. Rezultati akutnog zatajenja bubrega 53

6.2.6. Rezultati kroničnog zatajenja bubrega 54

6.3. Analiza korištenih postupaka strojnog učenja u aplikaciji 55

6.3.1. Vrednovanje modela 55

6.3.2. Analiza logističke regresije 56

6.3.3. Analiza neuronske mreže 57

6.3.4. Analiza slučajne šume 57

6.3.5. Usporedba korištenih postupaka strojnog učenja 58

7. ZAKLJUČAK 60

LITERATURA 61

SAŽETAK 64

ŽIVOTOPIS 65

PRILOZI 66

1. UVOD

U ovom diplomskom radu prikazano je ostvarenje web aplikacije koja omogućuje procjenu rizika oboljenja od bubrežnih bolesti na temelju određenih klasifikacijskih postupaka strojnog učenja te praćenje zdravstvenog stanja pacijenta. Prilikom ostvarenja aplikacije ispitani su klasifikaciji postupci logističke regresije, neuronske mreže i slučajne šume. Njihovom se implementacijom mogu bolje i jednostavnije definirati i razvrstati dobiveni rezultati. Zbog zaštite osobnih podataka, svaki je pacijent prije korištenja aplikacije dužan izraditi korisnički račun nakon čega se može prijaviti i procijeniti rizik oboljenja od bubrežne bolesti.

Funkcionalnost web aplikacije potrebno je ostvariti programskim jezikom *PHP*, dok se za izgled sučelja koriste *HTML* i *CSS*. Interakcija korisnika i web aplikacije svodi se na izvršavanju pet temeljnih funkcija. Prva je funkcija već spomenuta izrada korisničkog računa koja mu omogućuje korištenje aplikacije. Druga funkcija označava prijavu elektroničkom poštom ili korisničkim imenom i lozinkom koja stvara informaciju u obliku sesije. Nakon toga, prijavljeni korisnici mogu popuniti obrazac s vlastitim vrijednostima čimbenika rizika obolijevanja. Web aplikacija treba unesene podatke proslijediti web usluzi gdje će se procijeniti rizik. Kao odgovor, web usluga će aplikaciji vratiti rezultat i vjerojatnost pojavljivanja rizika. Četvrta funkcija koju korisnici mogu izvršiti u interakciji s web aplikacijom je promjena lozinke unesene pri registriranju korisničkog računa. Na samome kraju, svaki se korisnik može odjaviti iz web aplikacije pritiskanjem određene tipke kojom će se obrisati informacija o sesiji.

Drugo poglavlje opisuje funkciju bubrega, pojedine bolesti bubrega, te čimbenike rizika obolijevanja od tih bolesti. U trećem poglavlju spominju se postojeća rješenja s gledišta medicine i tehnologije. Četvrto poglavlje prikazuje prijedlog modela web aplikacije, odnosno funkcionalne i nefunkcionalne zahtjeve na web aplikaciju, te građu i cjeline web aplikacije. Peto poglavlje opisuje razvojnu okolinu, potrebne programske tehnologije i programska rješenja na korisničkoj i poslužiteljskoj strani. U šestom poglavlju prikazuje se način rada web aplikacije, provodi njeno ispitivanje na primjerima procjene rizika obolijevanja od bubrežnih bolesti, te se analiziraju korišteni postupci strojnog učenja.

2. BOLESTI BUBREGA I POTPORA LIJEČENJU

Unutar ovog poglavlja opisana je funkcija bubrega i obolijevanje od kroničnog i akutnog zatajenja. Također je objašnjena arterijska hipertenzija, hematološki čimbenici, biokemijski čimbenici i čimbenici iz urina pomoću kojih se ostvarivala procjena obolijevanja od bolesti.

2.1. Rad bubrega

Bubreg (lat. *Ren*) parni je tubularni žljezdani organ čija je glavna zadaća izlučivanje urina. U čovjeku se bubrezi nalaze kod 12. rebra, gdje je desni smješten nešto niže od lijevog. Postoje dva ruba, lateralni koji je konveksan i medijalni koji je konkavan. Na medijalnom se rubu nalazi hilus, gdje izlaze mokraćovod i bubrežne vene, a ulaze bubrežne arterije. Težina bubrega zdrave i odrasle osobe je 120 – 200 g, dužina oko 11 cm, širina oko 5 cm, a debljina 3 cm. Histološka građa karakterizira bubreg cjevčicama gdje je nefron osnovna jedinica svakog bubrega. Njihova je zadaća izlučivanje oko 60 g otopljenih tvari metaboličke razgradnje dnevno i viška vode u organizmu. Zbog izlučivanja natrija, kalija i klora mogu pomoći u regulaciji acido-bazične ravnoteže. Bubrezi stvaraju i amonijak kojeg koriste za uklanjanje kiselina u urinu. Također reguliraju i krvni tlak proizvodnjom proteolitičkog enzima renina [1].

2.2. Bolesti bubrega

Obolijevanje od bilo kakvih tipova bolesti, ponajviše ovisi o samom stilu života i životnim navikama. Primjerice, konzumiranje cigareta i alkohola mogu povećati krvni tlak, smanjiti protok krvi u bubrezima što može uzrokovati bolest. Promjena takvih navika, stila života i redovita fizička aktivnost može smanjiti rizik pojavljivanja bolesti. Postoje brojni problemi bubrežnih funkcija, a neki od njih hematurija i mikroalbuminurija te dva znatna svjetska problema, akutno i kronično zatajenje bubrega.

2.2.1. Kronično zatajenje bubrega

Kronično zatajenje bubrega jedan je od vodećih svjetskih zdravstvenih problema. To je posljedica gubitka bubrežne funkcije zbog smanjenog pokazatelja same funkcionalnosti, glomerularne filtracije. Ovaj se indeks u praksi može dobiti koncentracijom kreatinina u serumu, te on varira ovisno o spolu i životnoj dobi pacijenta, a procjenjuje se sljedećom jednadžbom [2]:

$$eGFR = 186 \times \left(\frac{serumski\ kreatinin}{88.4}\right)^{-1.154} \times (\check{z}ivotna\ dob)^{-0.203} \quad (2\text{-}1)$$

Kod žena je potrebno dobivenu vrijednost pomnožiti s 0.742

Jednadžba (2-1) prikazuje modifikaciju prehrane kod bubrežnih bolesti (engl. *Modification of Diet in Renal Disease - MDRD*). Procjena brzine glomerularne filtracije na ovakav način najbolji je pokazatelj bubrežne funkcije te on može razvrstati oboljele osobe po skupinama [2].

Prema tablici 2.2, kronično zatajenje bubrega odnosi se na brzinu glomerularne filtracije manje od 60 mL/min/1.73m^2 kroz više od tri mjeseca.

Tablica 2.2: Stadiji kroničnog zatajenja bubrega

Stadij	*Komentar*	*eGFR*
1	Oštećeni bubrezi s povećanom glomerularnom filtracijom	≥ 90
2	Oštećeni bubrezi s blago smanjenom glomerularnom filtracijom	$60 - 89$
3	Umjereno smanjena glomerularna filtracija	$30 - 59$
4	Izrazito smanjena glomerularna filtracija	$15 - 29$
5	Zatajenje bubrega	< 15

Osim koncentracije kreatinina u serumu, procjena indeksa težine kroničnog zatajenja bubrega također se može dobiti pomoću Cockcroft-Gaultove jednadžbe [3]:

$$Klirens\ kreatinina = \frac{(140 - \check{z}ivotna\ dob) \times tjelesena\ težina}{72 \times serumski\ kreatinin} \quad (2\text{-}3)$$

Kod žena je potrebno dobivenu vrijednost pomnožiti s 0.85

Pomoću jednadžbe (2-3) može se izračunati klirens kreatinina koji ovisi o životnoj dobi pacijenta i tjelesnoj težini mjerenoj u kilogramima. Ako dobivena vrijednost padne ispod 30 mL/min, postoji mogućnost pojave umjerenog stupnja zatajenja bubrega. Jednadžba za modifikaciju prehrane kod bubrežnih bolesti je kvalitetnija i preciznija za razliku od Cockcroft-Gaultove jednadžbe koja kao dodatni parametar traži i tjelesnu težinu [3] .

2.2.2. Akutno zatajenje bubrega

Obolijevanje od akutnog zatajenja bubrega danas je sve češća. Bolest se razvije u oko 5% hospitaliziranih pacijenata. Ne postoji nikakva specifična definicija akutnoga zatajenja bubrega,

već se očituje stupnjem smanjenja bubrežne funkcije pomoću tri skupine. Prva je prerenalna azotemija, koja je također i najčešći oblik akutnog zatajenja bubrega. Odnosi se na povećani omjer ureje i kreatinina zbog smanjenog protoka krvi u bubrezima (hipoperfuzija). Druga je postrenalna azotemija koja se definira kao akutno zatajenje nastalo zbog funkcijskih i strukturnih poremećaja protoka urina. Zahvaća bilo koji dio mokraćnog sustava od tubula do uretre (mokraćne cijevi). Kako bi se takvo zatajenje uopće razvilo, ometanje mora biti barem na razini mokraćne cijevi. Ako ometanje zahvati samo jedan ureter (mokraćovod), neće doći do zatajenja jer čovjekov organizam može normalno funkcionirati s jednim bubregom, odnosno jedan bubreg može nadoknaditi drugi. Posljednja skupina je renalna azotemija koja nastaje zbog nepravilne funkcije bubrežnih krvnih žila, glomerula, tubula ili intersticija [3].

Prerenalna azotemija može su utvrditi omjerom ureje i kreatinina u serumu ako je veći od 20:1. Takav se omjer može postići ako pacijent ima povećano stvaranje ureje (ima povećan unos proteina, uzimanje kortikosteroida). Nasuprot tome, kod renalne azotemije omjer je između 10 i 15. Ureja i kreatinin se slobodno filtriraju u glomerulu, ali za razliku od kreatinina, ureja se reapsorbira u tubulima. Budući da bubrezi registriraju manjak tekućine, prilikom ukazivanja na prerenalnu azotemiju, postoji reapsorpcija soli i vode. Rezultat toga jest reapsorpcija ureje što dovodi do povećanog omjera [3].

Omjer ureje i kreatinina jedan je od načina razlikovanja akutnog od kroničnog zatajenja bubrega. Kod drugog načina, koji je ujedno bolji i prihvatljiviji, koristi se ultrazvuk. Normalna veličina bubrega je između 9 i 12 centimetara po dužini, dok je bubreg kod pacijenata oboljelih od kroničnog zatajenja nešto manji. Također, pojava anemije je češća kod bolesnika s kroničnim zatajenjem nego s akutnim [3].

2.2.3. Ostale bolesti bubrega

Hematurija se odnosi na krv u urinu. Postoji mikroskopska (mikrohematurija) i makroskopska (makrohematurija). Ako je u pitanju makroskopska, zabrinutost je značajnija jer može dovesti do zloćudne bolesti. Normalan nalaz urina sadrži jako mali broj eritrocita. Ako se mikroskopskim pregledom urina otkrije ukupan broj eritrocita preko broja 3, javlja se hematurija. Uzroci hematurije mogu biti bubrežni, glomerularni i neglomerularni, i izvanbubrežni [3].

Testiranjem mikroalbuminurije vrlo je važan za dijabetičare jer postoji rizik od obolijevanja od dijabetičke nefropatije. Ako je dijabetičaru test na proteine pozitivan, odnosno ako mu je izlučivanje albumina u urinu preko 300 mg/danu, tada ima dijabetičku nefropatiju. Ako takav test pokazuje izlučivanje albumina između 30 i 300 mg/danu, pojavljuje se albuminurija [3].

2.3. Čimbenici rizika obolijevanja od bolesti bubrega

Čimbenik rizika pokazatelj je kolika je opasnost obolijevanja pacijenta od određene bolesti. Kako bi se takva potencijalna opasnost mogla utvrditi, pacijenti moraju raditi kontrolu nalaza krvi i urina. U izradi web aplikacije za procjenu rizika obolijevanja od bolesti bubrega koristilo se trinaest čimbenika rizika podijeljenih u skupine. Prva je skupina arterijska hipertenzija u koju spada krvni tlak. U drugoj skupini nalaze se hematološki čimbenici, eritrociti i hemoglobin. Trećom se skupinom podrazumijevaju biokemijski čimbenici rizika, koncentracija glukoze, ureja, serumski kreatinin, natrij i kalij, dok se u zadnjoj skupini nalaze čimbenici koji se mogu pronaći analizom urina, specifična težina, albumin, eritrociti i leukociti. Svaki od tih čimbenika mogu pratiti zdravstveno stanje pacijenta i pokazati opasnost obolijevanja od bubrežne bolesti.

2.3.1. Arterijska hipertenzija

Arterijska hipertenzija jedan je od najvećih medicinskih problema današnjice te je ona vodeći čimbenik obolijevanja od bubrežnih bolesti. Ukupan broj oboljele odrasle populacije od arterijske hipertenzije u Hrvatskoj je oko 40%, dok je kontrola te bolesti niska [4]. Postoje dvije vrijednosti pomoću kojih se može mjeriti krvni tlak. Prva je sistolička vrijednost koja pokazuje krvni tlak dok srce pumpa krv u arterije, a druga je dijastolička vrijednost koja se odnosi na krvni tlak između otkucaja srca. Te dvije vrijednosti se najčešće izražavaju u milimetrima po živi (mmHg).

Krvni tlak može biti optimalan, odnosno kada su mu izmjerene vrijednosti manje od 120/80 mmHg. Također, krvni tlak može biti normalan i visoko normalan kojeg je Američko društvo za hipertenziju uvelo kao pojam „prehipertenzija“. Takav se krvni tlak odnosi na osobe sa sistoličkim vrijednostima između 120 i 139 mmHg i s dijastoličkim vrijednostima krvnog tlaka između 80 i 89 mmHg [4].

Tablicom 2.4 prikazuje se klasifikacija oboljenja pomoću sistoličkih i dijastoličkih vrijednosti krvnog tlaka. Oboljenja nema ako pacijent ima optimalan, normalan ili visoko normalan krvni tlak,

dok je pojava arterijske hipertenzije definirana vrijednostima krvnih tlakova iznad 140 ili 90 mmHg te se dijeli prema tri stupnja.

Tablica 2.4: Rizici oboljenja zbog krvnog tlaka

Sistolički (mmHg)	*Dijastolički (mmHg)*	*Oboljenje*
< 120	< 80	*Nema*
120 – 129	80 – 84	*Nema*
130 – 139	85 – 89	*Nema*
140 – 159	90 – 99	*Hipertenzija (1. stupanj)*
160 – 179	100 – 109	*Hipertenzija (2. stupanj)*
≥ 180	≥ 110	*Hipertenzija (3. stupanj)*

Krvni tlak naglo raste ako se u organizmu nalazi veća količina izvanstanične tekućine. Nakon što bubrezi izluče višak takve tekućine, krvni se tlak vraća na normalne vrijednosti. Osim pretjeranog uzimanja vode, na porast krvnog tlaka također može utjecati i sol. Bubrezi vodu apsorbiraju normalno, a apsorpcija soli nije toliko jednostavna. Sol u organizmu može povećati volumen izvanstanične tekućine. Primjerice, ako u organizmu ima višak soli, čovjek postane žedan. Unošenjem tekućine sol se razrjeđuje na normalnu koncentraciju. Arterijska hipertenzija, odnosno povišeni krvni tlak, zbog povećanog volumena tekućine i veće koncentracije soli može znatno oštetiti bubrežnu funkciju i dovesti do kronične bubrežne bolesti [5].

2.3.2. Hematološki čimbenici

Svrha eritrocita u organizmu je prijenos hemoglobina. Njihov se oblik može mijenjati prolaskom kroz kapilare. Kod zdravih muških osoba prosječan broj eritrocita je oko 5 milijuna po kubnom milimetru ($10^6/mm^3$), dok je kod zdrave žene nešto manji [5].

U tijelu čovjeka dobrog zdravstvenog stanja ima 3,5 – 4,5 g željeza. Oko 3 g nalazi se u hemoglobinu, dok se ostatak nalazi u sustavu stanica vezivnog tkiva, jetri, bubrezima i koštanoj srži. Organizam čovjeka jako dobro štedi željezo. Primjerice, urinom se izlučuje oko 50 μg željeza, a dnevni gubitak je oko 1 mg. Sami hemoglobin pokazuje koncentraciju hemoglobina u krvi. Ako je ta razina niža od referentnog raspona (kod muškaraca je 14 – 18 g/dL, a kod žena između 12 i 16 g/dL), postoji anemija. Vrijednosti normalne koncentracije ovisi o dobi, spolu i trudnoći [3, 6].

Anemija kod kroničnog zatajenja bubrega česta je pojava, dok je rijetka kod akutnog zatajenja. Pretežito se pojavljuje padom klirensa kreatinina ispod 40 ml/min ili zbog smanjenog stvaranja eritropetina u bubrezima. Postojanja anemije kod oboljelog pacijenta od kroničnog zatajenja, ne mora nužno biti argument postojanja anemije zbog te bolesti [3].

Tablica 2.5 prikazuje rizik oboljenja utjecajem pada, odnosno porasta broja eritrocita ili hemoglobina u organizmu.

Tablica 2.5: Rizici oboljenja zbog hematoloških čimbenika

Eritrociti ($10^6/mm^3$)		*Oboljenje*
~ 5		*Nema*
$\neq 5$		*Dodatne pretrage*
Hemoglobin (g/dL)		
Muškarci	*Žene*	
< 14	< 12	*Anemija*
14 – 18	12 – 16	*Nema*
> 18	> 16	*Dodatne pretrage*

2.3.3. Biokemijski čimbenici

Koncentracija glukoze (šećera) u krvi osobe mijenja se prilikom iscrpljujućih situacija, straha ili napora zbog pojačanog lučenja adrenalina. Nasuprot tome, prilikom trudnoće koncentracija glukoze se smanjuje. Gubitak normalne regulacije glukoze u krvi mogu uzrokovati hipoglikemija i hiperglikemija. Hipoglikemije je smanjena koncentracija glukoze. Simptomi ove bolesti se dijele na adreneričke u koje spadaju dijaforeza, glad, osjećaj trnjenja, drhtavica, anksioznost, i na neuroglikopeničke u koje spadaju osjećaj vrućine, slabost, zamor, vrtoglavica. Koncentracija glukoze koja upućuje na hipoglikemiju mora biti manja od 2,8 mmol/L. Relativni ili apsolutni manjak inzulina (šećerna bolest) najčešći je uzrok hiperglikemije, odnosno povećane koncentracije glukoze u krvi. Obolijevanje od hiperglikemije uglavnom upućuje na dijabetes. Neki čimbenici rizika razvoja dijabetesa su: dob veća od 45 godina, debljina, obiteljska anamneza, hipertenzija [3, 6].

Ureja je tvar koja se nalazi u urinu, limfi, krvi i ostalim tjelesnim tekućinama. Glavni je metabolički produkt bjelančevina i nukleinskih kiselina, a širi se staničnim membranama. Ureja se reapsorbira i filtrira u bubrezima, a najviše se izlučuje urinom. Ostatak se izlučuje putem

gastrointestinalnoga trakta i kože. Određivanje koncentracije ureje služi kao čimbenik same bubrežne funkcije. Nije dobar način za procjenu brzine glomerularne filtracije jer njezino stvaranje ovisi o faktorima poput količine bjelančevina u prehrani, katabolizmom, količinom dušikovih spojeva u krvi (engl. *blood urea nitrogen*). Kod zdrave odrasle osobe ima od 2,8 do 8,3 mmol/L ureje, dok se velike koncentracije ureje nalaze u akutnom i kroničnom zatajenju bubrega [3, 6].

Serumski kreatinin nastaje iz kreatin fosfata, spoja koji se pohranjuje u mišićima. Njegova koncentracija u serumu uvećana je kod atrofije i distrofije mišića kao i kod amputacija. Također, koncentracija ovisi o glomerularnoj filtraciji, pa se utvrđuje kao dobar pokazatelj bubrežnih funkcija. Pri oboljenju od bubrežne bolesti, postoji veća koncentracija kreatinina u serumu, a izlučivanje u urinu je smanjeno. Kod osoba normalnog zdravstvenog stanja, kreatinin u urinu je uglavnom konstantan. Referenti interval kreatinina za odrasle osobe iznosi 71 i 106 μmol/L, dok je nešto niži kod žena [3, 6].

Jedan od glavnih kationa izvanstanične tekućine je natrij. Najvećim se dijelom izlučuje urinom, dok se manjim izlučuje isparavanjem. Referentna vrijednost natrija u serumu ili plazmi je 136-145 mmol/L. Hiponatrijemija označava smanjenu koncentraciju natrija gdje je razina manja od 136 mmol/L. Kod osobe oboljele od hiponatrijemije, postoji blagi oblik (126 – 135 mmol/L) i teži oblik gdje je razina manja od 125 mmol/L. Ako razina natrija pređe 145 mmol/L, javlja se hipernatrijemija, odnosno povećana koncentracija natrija u plazmi ili serumu. Javlja se pri padu koncentracije tekućine (vode) u odnosu na natrij. Postoje brojni uzroci hipernatrijemije, a neki od njih su i renalni gubici poput uremije [3, 6].

Razgradnja kalija odvija se u tankome crijevu. Urinom se izlučuje 90%, dok je 10% stolicom. Protokom krvi u bubrezima, kalij se pretvara u glomerularni filtrat. Izlučivanje samog kalija je kontrolirano u odnosu na natrij kod kojeg regulaciju obavlja aldosteron. Kalij je jedan od glavnih kationa stanične tekućine. Bubrezi oštećene funkcije mogu loše eliminirati kalij. Kada koncentracija kalija u serumu padne ispod 3,5 mmol/l, radi se o hipokalijemiji. Hiperkalijemija je također česti poremećaj elektrolita u organizmu, dogodi se kada koncentracija kalija pacijenata prekorači 5,3 mmol/L. Jedan od uzroka može biti smanjeno izlučivanje urinom, gdje može doći do zatajenja bubrega [3, 6].

Za svaki opisani biokemijski čimbenik pomoću tablice 2.6 moguće je ustanoviti rizik oboljenja na temelju dobivenih vrijednosti analizom.

Tablica 2.6: Rizici oboljenja zbog biokemijskih čimbenika

Koncentracija glukoze (mmol/L)		*Oboljenje*
< 2.8		*Hipoglikemija*
2.8 – 6		*Nema*
6.1 – 11		*Hiperglikemija*
≥ 11.1		*Dijabetes*
Ureja (mmol/L)		
< 2.8		*Dotok krvi u bubreg*
2.8 - 8.3		*Nema*
> 8.3		*Dodatne pretrage*
Kreatinin u serumu (μmol/L)		
Muškarci	*Žene*	
< 71	< 53	*Dodatne pretrage*
71 – 106	53 – 97	*Nema*
> 106	> 97	*Smanjena bubrežna funkcija*
Natrij (mmol/L)		
< 136		*Hiponatrijemija*
136 – 145		*Nema*
> 145		*Hipernatrijemija*
Kalij (mmol/L)		
< 3.5		*Hipokalijemija*
3.5 - 5.3		*Nema*
> 5.3		*Hiperkalijemija*

2.3.4. Čimbenici iz urina

Specifična težina urina pokazuje omjer volumena urina s volumenom vode. Pokazatelj regulira stupanj hidracije, a referentna vrijednost je između 1.001 i 1.035. Urin, čija je specifična težina manja od 1.008 je razrijeđen, dok je onaj s težinom većom od 1.025 koncentriran. U slučaju pada specifične težine urina ispod 1.001 ili porasta iznad 1.035, potrebne su daljnje obrade. Postoje različiti uzroci promjene specifične težine urina. Primjerice, uzroci smanjenja mogu biti pretjerano uzimanje tekućine, diuretici, dijabetes, smanjena bubrežna sposobnost, a uzroci povećanja specifične težine mogu biti smanjeno uzimanje tekućine, dehidracija, dijabetes [3].

Proteini u krvnoj plazmi imaju različite zadaće: zaštita od infekcija (antitijela) i utjecaj na raspodjelu vode između vaskularnog i međustaničnog prostora. Također imaju i transportnu funkciju jer se razni lijekovi, hormoni, vitamini vežu na pojedine proteine poput albumina. U krvnoj plazmi postoje albumini, globulini i fibrinogeni. Proteini koji se sintetiziraju u tijelu iz aminokiselina, dok se albumini i fibrinogeni stvaraju prvenstveno u jetri. Albumini su topljivi u vodi i u razrijeđenim otopinama soli, a nije topljiv u koncentriranim otopinama soli. Zbog toga se može istaložiti zasićenom otopinom amonijevog sulfata. Ovi proteini imaju molekularnu masu od 66 kDa te su jako važni za održavanje onkotičkog tlaka unutar vaskularnog prostora [6].

U urinu osobe normalnog zdravstvenog stanja, nalazi se do 10 leukocita po mikrolitri. Veći broj od referentne vrijednosti označava upalne bolesti bubrega i urinarnog sustava. Ako se radi o proteinuriji, potrebno je posumnjati na piuriju bubrežnog podrijetla [3, 6].

Svi mogući rizici obolijevanja zbog poremećaja čimbenika iz urina, odnosno zbog odstupanja od referentnih vrijednosti, moguće je vidjeti u tablici 2.7.

Tablica 2.7: Rizici oboljenja zbog čimbenika iz urina

Specifična težina urina	*Oboljenje*
< 1.001	*Dodatne pretrage*
1.001 - 1.035	*Ne*
> 1.035	*Dodatne pretrage*
Albumin (mg/danu)	
$x < 30$	*Ne*
$30 \leq x \leq 300$	*Mikroalbuminurija*
$x > 300$	*Dijabetička nefropatija*
Eritrociti	
$x \leq 3$	*Ne*
$x > 3$	*Hematurija*
Leukociti	
$x \leq 10$	*Ne*
$x > 10$	*Infekcije, piurija*

3. TRENUTNO STANJE U PODRUČJU LIJEČENJA, PRAĆENJA I PREVENCIJE BOLESTI BUBREGA

Ovim se poglavljem opisuju trenutno aktualna rješenja za liječenje i prevenciju bubrežnih bolesti općenito u medicinskim svrhama te se opisuju primjeri znanstvenih dosega gdje su se koristili određeni postupci strojnog učenja. Također se i uspoređuju točnosti realiziranog modela u ovom diplomskom radu s drugim primjerima. Na kraju se spominju slične aplikacije koje su vezane za bubreg.

3.1. Postojeća rješenja općenito u medicini

Zdravlje bubrega ima ključnu ulogu u medicini. To se odnosi na održavanje normalne funkcije bubrega kao što je filtriranje krvi i ravnotežu tekućina. Također, razumijevanje i održavanje referentnih vrijednosti čimbenika rizika obolijevanja od različitih bolesti bubrega, smanjuje rizik od pojave takvih bolesti. Osnovna zadaća praćenja bubrežnih bolesti ima važnu značajku i u smanjivanju rizika od kardiovaskularnih bolesti. Manji broj oboljelih osoba s blažim simptomima kroničnog bubrežnog zatajenja preći će u završnu fazu liječenja. Najvažniji pokazatelj pojavljivanja kroničnog bubrežnog zatajenja je procjena brzine glomerularne filtracije koja se dobije koncentracijom kreatinina [7].

Iz razloga što se simptomi bubrežnih bolesti razvijaju jako sporo, čimbenici rizika se mogu otkriti laboratorijskim testiranjem [8]. Kako bi se zaštitilo od oboljenja bolesti bubrega, potrebno je ograničiti dnevni unos soli i tekućine jer pretjeranim unosom može doći do hipertenzije, a kasnije do zatajenja. Pravilnije prehrana i fizička pokretljivost ključne su aktivnosti boljeg zdravstvenog stanja. Postoje i bubrežne operacije u slučaju bolesti ili ozljeda poput nefrektomije gdje se uklanja čitavi bubreg [1]. Jedno od rješenja, odnosno metoda uklanjanja bubrežnog kroničnog zatajenja, je presađivanje bubrega. Za uspješno izvođenje ovog procesa, operacijskim se zahvatom ukloni čitavi bubreg iz darivatelja organa i presadi se u oboljelog pacijenta.

Obavljaju se i brojne pretrage pomoću kojih se može pratiti zdravstveno stanje pacijenta. U hematologiji se krv može pretražiti kompletnom krvnom slikom gdje se može odrediti broj eritrocita i hemoglobin. U nefrologiji, kemijskom analizom urina može se obaviti pomoću testa trakicom. Tim se pregledom može otkriti specifična težina, bjelančevine, pH-vrijednost, krv u urinu i slično [3].

Budući da su čimbenici rizika jedni od važnijih pokazatelja ukazivanja svih oblika bubrežnih bolesti, njihovo razumijevanje ima izuzetno važnu ulogu jer se na taj način bolest može brže otkriti laboratorijskim testiranjem pacijenata. Upravo zbog toga, prije same realizacije web aplikacije za procjenu rizika bubrežnog oboljenja, potrebno je dodatno istražiti pojedini čimbenik kako bi se mogle provjeravati vrijednosti čimbenika s referentnim vrijednostima.

3.2. Postojeća rješenja s gledišta tehnologije

Budući da internet društvu nudi veliki broj informacija gotovo na svim digitalnim uređajima, može se zaključiti da postoji i puno obrazovnih materijala o bolestima bubrega. Također, na internetu se mogu pronaći svakakve teme i odgovori na pitanja u vezi oboljenja [9].

U tehnološkom smislu postoje brojne metode koje koriste različite postupke strojnog učenja kao pomoć za liječenje, praćenje i prevenciju bolesti bubrega. U članku [10] autori su prikupili bazu podataka jedne bolnice u Američkoj saveznoj državi Wisconsin. Baza sadrži različite laboratorijske nalaze pacijenata pomoću kojih su predviđali rizike modelom strojnog učenja. Implementacijom algoritma slučajne šume (engl. *random forest*) omogućena ima je binarna klasifikacija s točnošću od 54.6%. Osim tog algoritma, koristili su i algoritam logističke regresije *(*engl. *logistical regression)* čija je točnost bila za 0.8% lošija. Skupljenim podacima iz nekoliko različitih laboratorija i bolnica, autori u članku [11] procjenjuju točnost modela algoritmima potpornih vektora (engl. *support vector machines)* i k – najbližih susjeda (engl. *k-nearest neighbor)*. Koristeći *MATLAB*, uspjeli su dobiti točnost od 78.75% metodom k – najbližih susjeda, a 73.75% metodom potpornih vektora. Trećim člankom [12] autori opisuju predviđanja kroničnog zatajenja bubrega korištenjem algoritma višeslojnog perceptrona (engl. *multilayer perceptron),* radijalnih baznih funkcija (engl. *radial basic functions*) i logističke regresije. Metoda višeslojnog perceptrona je bila najtočnija s točnošću od 99.75%, dok su ostali imali točnost za 1 – 2% manje. Četvrtim člankom [13] uspoređuju se tri načina procjene obolijevanja od bubrežnih bolesti. Kao prvi način, koristili su se umjetnom neuronskom mrežom (engl. *artificial neural network*) pomoću koje su dobili točnost modela od 93.85%. S metodom logističke regresije uspjeli su postići točnost klasifikacije od 74.74%, dok su kao treći način koristili metodu stabla odluke te su dobili točnost modela od 78.48%. Petim se člankom [14] opisuju procjene obolijevanja pomoću algoritama neuronske mreže i potpornih vektora. Metoda neuronske mreže davala je bolje rezultate s točnošću od 87.7%, dok je metoda potpornih vektora bila lošija za 11.38%.

Tablicom 3.1 prikazuje se točnost korištenih postupaka strojnog učenje web aplikacije u odnosu s istim algoritmima iz navedenih članaka. Može se zaključiti da je algoritam logističke regresije najkorištenija metoda u svim klasifikacijama. Budući da se ne radi o istim bazama podataka nad kojima se izvršavali postupci strojnog učenja, ne može se s preciznošću reći koji je model davao najbolje rezultate.

Tablica 3.1: Usporedba točnosti algoritama strojnog učenja postojećih rješenja

Autor	*Logistička regresija*	*Slučajna šuma*	*Neuronska mreža*
Članak [7]	*53.8*	-	-
Članak [9]	*97.5*	-	-
Članak [10]	*74.74*	*78.48*	*93.85*
Članak [11]	-	-	*87.7*
Marin Varnica	*95.8*	*97.5*	*96.7*

Također postoje i različite mobilne aplikacije dostupne na pametnim uređajima. Aplikacijom [15] pacijenti dobivaju pristup informacijama o lijekovima koji su potrebni za prevenciju oboljenja. Također omogućavaju pacijentima mjerenje klirensa kreatinina Cockcroft-Gaultovom jednadžbom i izračunavanje brzine glomerularne filtracije pomoću jednadžbe za modifikaciju prehrane kod bubrežnih bolesti. Aplikacijom [16] pomaže se u kontroliranju dijeta pacijenata oboljelih od kroničnog bubrežnog zatajenja. Sadrži i sve korake dnevnog unosa kalija, natrija i proteina. Pomoću aplikacije [17] korisnici mogu izmjeriti brzinu glomerularne filtracije te na taj način može utvrditi stadij kroničnog bubrežnog zatajenja.

4. PRIJEDLOG MODELA WEB APLIKACIJE I ANALIZA PODATAKA

Ovim se poglavljem opisuju funkcionalni i nefunkcionalni zahtjevi web aplikacije. Prikazuje se građa, cjeline same aplikacije i obrazac programske arhitekture *Model-View-Controller*. Također, opisuju se korišteni postupci strojnog učenja u analizi podataka te baza podataka uz pomoć koje su se procjenjivali rizici obolijevanja.

4.1. Funkcionalni i nefunkcionalni zahtjevi na web aplikaciju

Glavna je ideja web aplikacije procjena rizika obolijevanja pacijenta od bubrežne bolesti na temelju unesenih podataka, odnosno već opisanih čimbenika. Postoji i povijest rezultata gdje pacijenti mogu vidjeti prijašnje rezultate i detaljniji prikaz svakog od njih. Tim se prikazom opisuje zdravstveno stanje pacijenta pomoću bilješki, dok je tablicom prikazana usporedba unesenih i referentnih vrijednosti čimbenika.

Iz sigurnosnih razloga i same privatnosti pacijenta, svaki je od njih dužan izraditi korisnički račun gdje upisuje svoje osobne podatke i podatke o korisničkom računu. Nakon izrade računa, omogućena im je prijava u aplikaciju pomoću elektroničke pošte ili korisničkog imena i lozinke. Za vrijeme prijave mogu i označiti na potvrdni okvir putem kojeg ih aplikacija pomoću kolačića može identificirati sljedećom posjetom. U aplikaciji korisnici mogu otkriti i rizik oboljenja od kroničnog bubrežnog zatajenja. Također mogu vidjeti i podatke korisničkog računa u postavkama te promijeniti lozinku.

Svaka se navedena funkcija, osim odjave, temelji na bazi podataka. Izradom korisničkog računa, prijavom korisnika ili procjenom rizika obolijevanja, model komunicira s bazom podataka. Primjerice, procjena rizika funkcionira na način da se uneseni podaci korisnika prvo provjeravaju te se na temelju toga iz baze uzimaju bilješke koje opisuju njegovo zdravstveno stanje. Nakon toga se ti podaci proslijede web usluzi. Nakon što web usluga vrati rezultat, podaci i bilješke, zajedno s tim rezultatom, spremaju se u bazu podataka koja mu omogućava detaljan prikaz zdravstvenog stanja pritiskom određene tipke na sučelju.

Za razliku od funkcionalnog zahtjeva koji prikazuje ono što web aplikacija radi, pojam nefunkcionalnosti prikazuje način na koji radi. Općenito, aplikaciju bi trebali podržavati svi operacijski sustavi i internetski preglednici. Također bi zbog svog prilagodljivog dizajna, ovisno o rezoluciji, trebali imati podršku na svim mobilnim uređajima.

Prije nego što nastavi svoju daljnju funkciju, aplikacija prvo mora provjeriti točnost unesenih podataka pomoću različitih validacijskih postupaka. Između ostalog, provodi se jedna bitna karakteristika nefunkcionalnih zahtjeva, sigurnost. Određenim algoritmom za sažimanje poruka, lozinka se šifrira u sažetak i tako se sprema u bazu. Na taj način ni vlasnik aplikacije pregledavajući bazu ne može saznati korisnikovu lozinku. Prijavom korisnika, unesena se lozinka provjerava s dešifriranim sažetkom i tek mu je tada omogućen ulazak u aplikaciju. Na sličan način funkcionira i promjena lozinke gdje se nova šifrirana lozinka u bazi zamjenjuje sa starom. Treba napomenuti da prijava stvara informaciju u obliku sesije koja pruža korisniku korištenje aplikacije.

Velike baze podataka u web aplikacijama predstavljaju veliki problem jer može doći do prigušivanja performansi. Zbog toga bi se programeri razvojem aplikacija trebali usredotočiti na brzine izvođenja određenih funkcija. *PHP* programski jezik sadrži proširenje *Xdebug* koje može pomoći u otklanjanju grešaka i profiliranju performansi web stranica.

Na sučelju se može izvršiti pet temeljnih funkcija. Tablicom 4.1 prikazuju se ukupna vremena trajanja tih temeljnih funkcija korištenjem tri različita uređaja. Vremena su dobivena preko *Xdebug* pomoćnog sučelja za profiliranje performansi, *Webgrind,* a mjereni su u milisekundama. Može se primijetiti da izrada korisničkog računa traje nešto duže od prijave. To je i razumno jer sama funkcija izrade računa ima više zadataka. Činjenica je da šifriranje i dešifriranje poruka uvijek traju nešto duže. Razlog tome je što koristi *bcrypt* algoritam za sažimanje poruka koji je iz sigurnosnih razloga napisan da bude spor. Sažimanje poruke mu može potrajati do sekunde, ovisno o podešavanju parametara funkcije [18]. To je i razlog zbog kojeg promjena lozinke traje dvostruko duže nego izrada računa i prijava korisnika. Funkcija koja procjenjuje rizik oboljenja od bubrežne bolesti traje najduže jer ona zapravo obavlja najviše zadataka. Odjavom se briše kolačić i informacija o sesiji, ali da korisnik ne označi potvrdni okvir, funkcija bi se mogla izvršiti i prije.

Ovim se prikazom mogu primijetiti i veće razlike između tri korištena uređaja, ali to nije uvijek istina jer performanse web aplikacija i općenito internetskih stranica ovise o brzini i komponentama računala i ostalih uređaja.

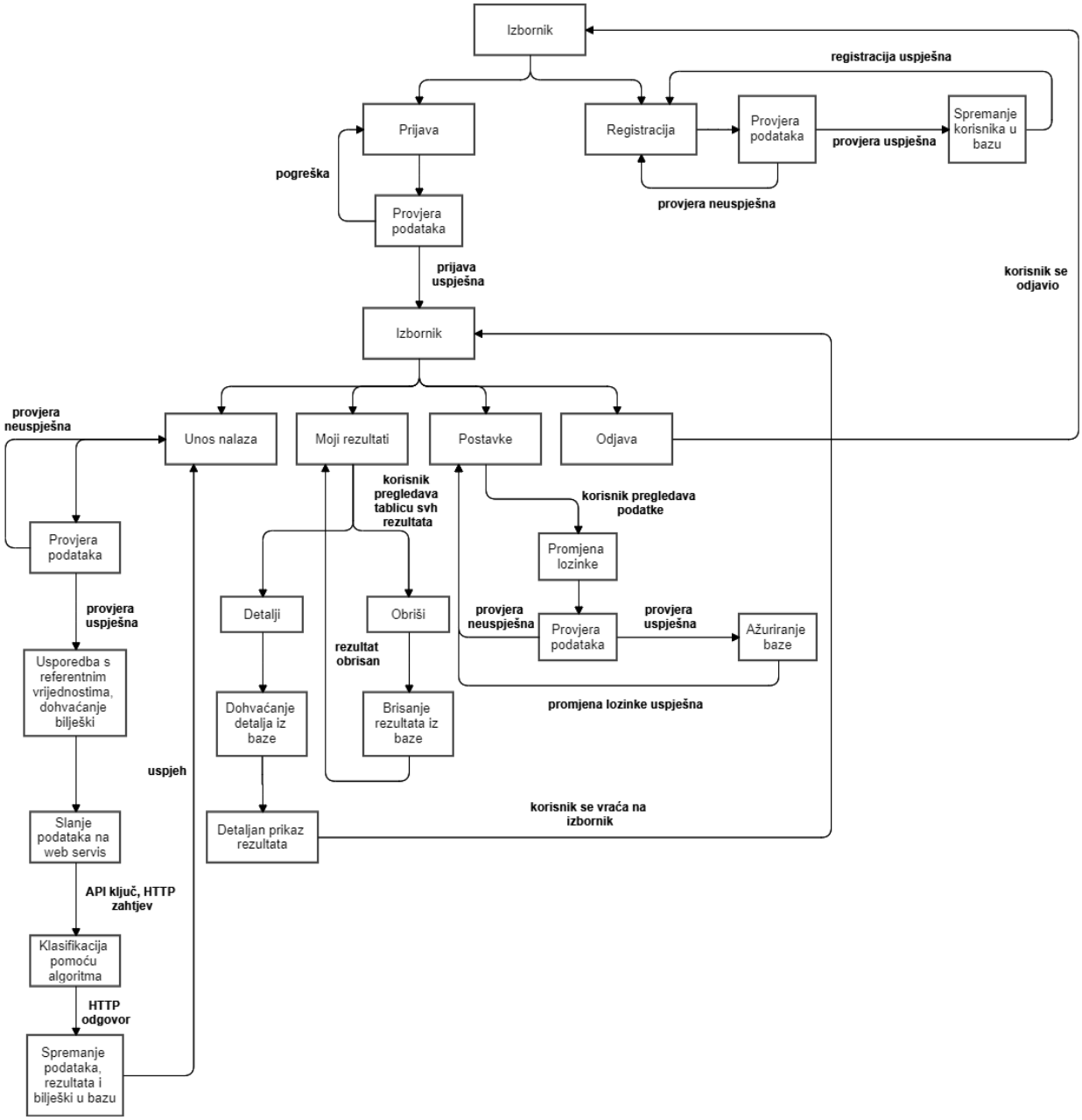
Tablica 4.1: Pregled trajanja funkcija korištenjem tri različita uređaja

Opis funkcije	Vrijeme trajanja *(prijenosno računalo)*	Vrijeme trajanja *(mobilni uređaj)*	Vrijeme trajanja *(stolno računalo)*
Izrada korisničkog računa	*65.095*	*82.745*	*76.254*
Provjera podataka	*3.818*	*1.960*	*2.025*
Šifriranje lozinke	*59.326*	*76.344*	*58.291*
Zapisivanje u bazu	*1.822*	*1.307*	*12.390*
Ostale funkcije	*0.129*	*3.134*	*3.548*
Prijava korisnika	*61.666*	*71.098*	*57.913*
Učitavanje korisnika iz baze	*1.347*	*1.585*	*1.739*
Spremanje kolačića u bazu	*1.251*	*10.915*	*1.391*
Dešifriranje i provjera lozinke	*57.447*	*56.997*	*53.395*
Ostale funkcije	*1.621*	*1.601*	*1.388*
Procjena rizika obolijevanja	*1096.614*	*1583.688*	*929.454*
Provjera podataka	*7.500*	*9.000*	*8.111*
HTTP zahtjev/odgovor	*1075.656*	*1523.161*	*906.862*
Zapisivanje u bazu	*13.000*	*48.860*	*11754*
Ostale funkcije	*0.453*	*2.667*	*2.727*
Promjena lozinke	*160.222*	*209.601*	*173.434*
Provjera podataka	*0.250*	*0.128*	*0.120*
Učitavanje lozinke iz baze	*1.763*	*1.467*	*1.477*
Dešifriranje i provjera lozinke	*57.482*	*82.306*	*83.231*
Šifriranje lozinke	*51.086*	*75.945*	*76.122*
Ažuriranje baze	*48.011*	*48.094*	*10.859*
Ostale funkcije	*1.630*	*1.661*	*1.625*
Odjava korisnika	*4.107*	*13.156*	*2.695*
Brisanje kolačića	*2.609*	*12.036*	*1.457*
Brisanje sesije	*0.153*	*0.103*	*0.120*
Ostale funkcije	*1.345*	*1.017*	*1.118*

4.2. Građa i cjeline web aplikacije

Korisnik interakcijom s aplikacijom putem sučelja može svojevoljno odraditi pet funkcija. Nakon izrade i prijave korisničkog računa dobiva mogućnost unosa vrijednosti čimbenika rizika koje će aplikacija proslijediti na web uslugu putem kojeg će se procijeniti vjerojatnost oboljenja od

bubrežne bolesti. Nakon toga može vidjeti te rezultate ili ih obrisati. Također može u postavkama promijeniti svoju lozinku. Slikom 4.2 prikazuje se dijagram tijeka interakcije korisnika, odnosno pacijenta s aplikacijom.



Slika 4.2: Dijagram tijeka rada aplikacije

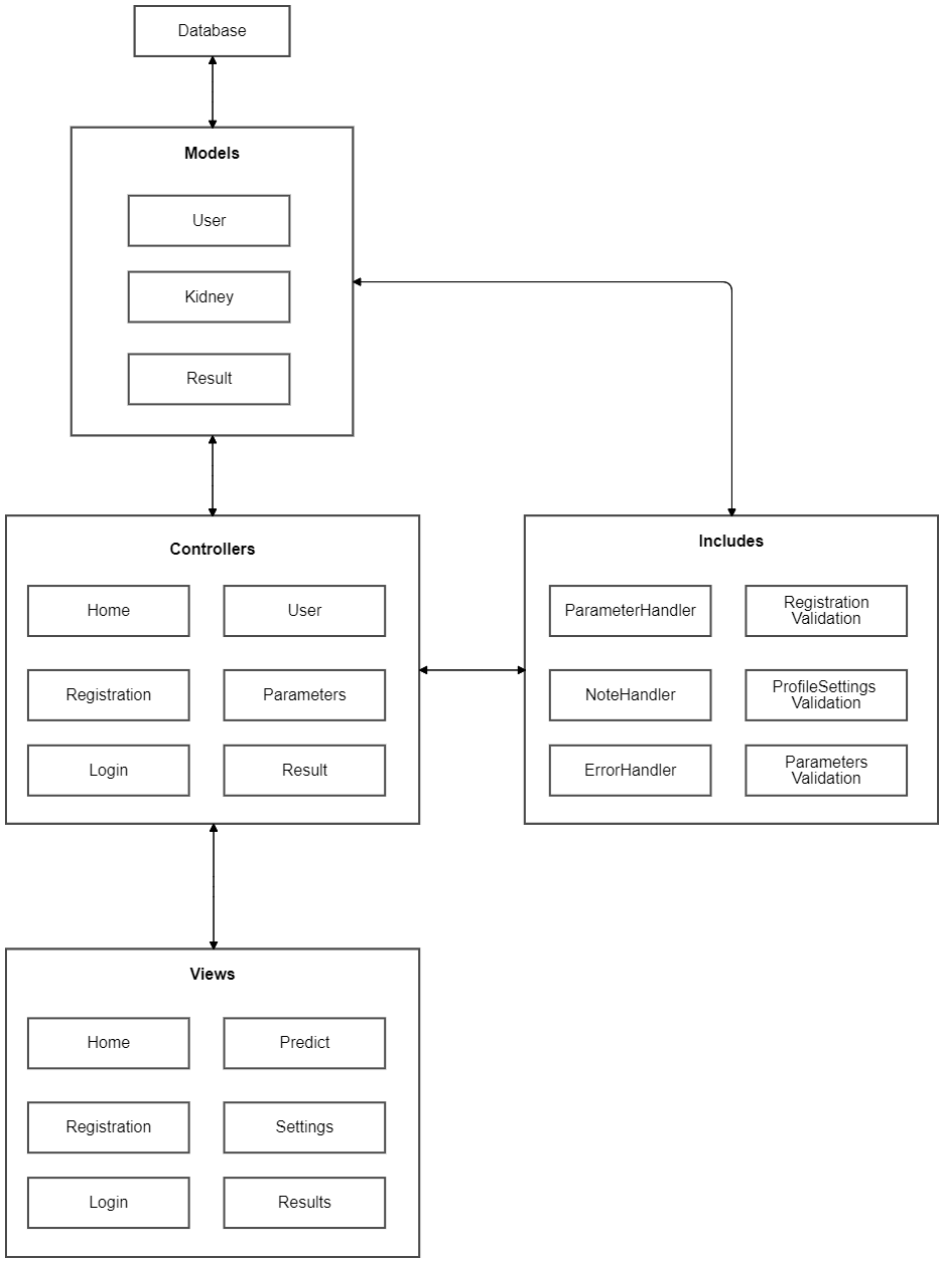
4.2.1. Obrazac programske arhitekture

Osnovni element međusobne povezanosti svih komponenata web aplikacije je obrazac softverske arhitekture *Model-View-Controller*. Taj se obrazac uglavnom veže za programske jezike objektno orijentiranog koncepta, a omogućuje komunikaciju između modela (engl. *model*), pogleda (engl.

view) i upravljačke komponente (engl. *controller*). Postoji dvosmjerna komunikacije između modela i baze podataka te je na taj način omogućena cjelokupna logika aplikacije. Pomoću pogleda korisnik ima vizualizaciju aplikacije. On zapravo uključuje sve *HTML*, *CSS* i *JavaScript* datoteke te korisniku omogućuje sučelje. Kako bi aplikacija mogla prikazati rješenje korisniku, potrebna je komunikacija između pogleda i modela. Takve usluge pružaju upravljačke komponente koje modelu prosljeđuju ulazne podatke sa sučelja.

Jedan od kontrolnih mehanizama obrasca softverske arhitekture *Model-View-Controller* je usmjerivač (engl. *router*). Njegova se funkcija bazira na usmjeravanju upravljačke komponente, odnosno njezinih metoda (akcija) i parametara temeljenim na unesenoj web adresi te se na taj način vrši komunikacija između komponenata aplikacije.

Slikom 4.3 prikazuje se povezanost i međudjelovanje svih komponenti aplikacije za procjenu rizika obolijevanja od bubrežne bolesti. Postoji *MySQL* baza podataka koja sadrži nekoliko tablica. Tablica o korisnicima pomoću koje se može izraditi račun, prijaviti se u aplikaciju ili promijeniti lozinku, tablica s bilješkama, tablicu sa sažetkom koji omogućuje korištenje kolačića i tablicu s detaljnim prikazom zdravstvenog stanja korisnika. Modeli *Kidney*, *Result* i *User* komuniciraju s bazom podataka i prosljeđuju sve informacije upravljačkim komponentama. Koriste svojstva klase (engl. *trait*) koja im olakšavaju obavljanje određenih zadataka. Primjerice, model *Kidney* uspoređuje unesene podatke s referentnim vrijednostima čimbenika rizika pomoću svojstva *ParamaterHandler*. Na sličan način funkcioniraju i upravljačke komponente koje koriste pomoćne klase kako bi mogle provjeriti valjanost unesenih podataka. Budući da su upravljačke komponente zadužene za razmjenu informacija između pogleda i modela, s provjerom valjanosti krivo unesenih podataka pomoću klase za provjeru, upravljačke komponente pogledu mogu vratiti informaciju o greški. Prikaz takve informacije korisnik može vidjeti preko sučelja web aplikacije.



Slika 4.3: Prikaz obrasca programske arhitekture MVC

4.3. Prikladni postupci analize podataka

Kako je već spomenuto, svaka se funkcija interakcije korisnika i web aplikacije, osim odjave, temelji na komunikaciji s bazom podataka. Prije same komunikacije, podaci se najprije moraju provjeriti i analizirati određenim postupcima. Ovisno o korisnikovoj želji, pokreće se određeni postupak analize tih podataka.

Izradom novog korisničkog računa, validacijskim postupcima može se provjeriti odgovara li duljina korisničkog imena, ispravnost elektroničke pošte, duljina i točnost lozinke. Prijavom se

podaci analiziraju na način da se prvo uspoređuju lozinke implementiranom *PHP* funkcijom. Zatim se uneseno korisničko ime ili elektronička pošta uspoređuju s onima iz baze podataka te ako su točni, korisniku je omogućena prijava u aplikaciju. Provjera čimbenika rizika obolijevanja od bubrežne bolest vrši se na način da se prvo unesene vrijednosti analiziraju, odnosno uspoređuju s referentnim vrijednostima opisanih u drugom poglavlju. Nakon toga se ti podaci u *JSON* (engl. *JavaScript Object Notation*) formatu preko Postman platforme pomoću *API* ključa proslijede kao *HTTP* zahtjev web usluzi. Na web usluzi se određenim postupkom strojnog učenja podaci klasificiraju te se kao odgovor vrati rezultat predviđanja i njegova vjerojatnost.

Zbog brojnih mogućnosti poput raspoznavanja uzoraka, robotskog vida, različitih analiza podataka, strojno je učenje znanstvena disciplina koja se bavi oblikovanjem algoritama čija se učinkovitost temelji pomoću skupa podataka. Performanse korištenih algoritama ovise o veličini danih podataka [19]. Postoje dva pristupa strojnom učenju, a to su nadzirano i nenadzirano učenje. Kod nadzirnog učenja treniranje modela se odnosi na već poznate ulazne i izlazne podatke kako bi se mogle dobiti izlazne procjene [19]. Najčešće tehnike ovakvog pristupa su klasifikacija i regresija. Nenadziranim se učenjem traže skrivene strukture na temelju danih podataka [19]. Tehnike koje se koriste u ovom pristupu su grupiranje (engl. *clustering*) i smanjivanje dimenzionalnosti (engl. *dimensionality reduction*).

Osim opisanih analiza podataka, web uslugom *Microsoft Azure* okruženja računarstva u oblaku podaci se analiziraju određenim klasifikacijskim postupcima. U nastavku je opisana logistička regresija, neuronska mreža i slučajna šuma.

4.3.1. Logistička regresija

Logistička regresija je često korišteni model koji koristi određene klasifikacijske postupke predviđanja vjerojatnosti. Postoje dva tipa logističke regresije, binarna, koja može imati samo dva izlaza i ona koja može imati više izlaza [20]. Za razliku od linearne regresije koja koristi linearni pristup, logistička regresija koristi sigmoidnu funkciju čija je zadaća kategoriziranje ulaznih podataka.

$$S(z) = \frac{1}{1 + e^{-z}} \tag{6-5}$$

Jednadžba (6-5) predstavlja sigmoidnu funkciju čiji je izlaz procjena vjerojatnosti (0 ili 1), *z* predstavlja ulaz, dok je *e* Eulerova konstanta [20]. Srednja vrijednost (0.5) se smatra pragom koji služi za utvrđivanje koji podatak pripada kojoj klasi. Općenito, ako je izlazna vrijednost veća od 0.5, tada ulazni podatak pripada klasi 1. Suprotno tome, ako je izlazna vrijednost manja od 0.5, tada ulazni podatak pripada klasi 0. Model logističke regresije često je korišteni pristup kategoričnih podataka. Takvi podaci mogu biti binarni, kao primjerice mogu pokazivati točnost i netočnost od nečega, a mogu biti i pokazatelji postojanja bolesti [21]. Upravo zbog jednostavnosti prikaza binarnih kategorija, ova je analiza podataka odabrana za ispitivanje u diplomskom radu.

4.3.2. Neuronska mreža

Umjetne neuronske mreže (engl. *Artificial neural network*) su najčešće korišten algoritam strojnog učenja koji se koristi za klasifikaciju kompleksnijih uzoraka podataka. Sastoje se od osnovnih jedinica, neurona koji u sebi sadržavaju sigmoidne funkcije i mogu sadržavati nekoliko slojeva. Ulazni sloj prosljeđuje podatke skrivenom sloju. Svaki neuron ovog sloja predstavlja jedan jedinstveni atribut baze podataka, poput čimbenika rizika obolijevanja od bubrežne bolesti. Skriveni se sloj nalazi između ulaznog i izlaznog sloja. Također može postojati i više skrivenih slojeva. Posljednji, odnosno izlazni sloj prima podatke prethodnog sloj, primjenjuje aktivacijsku funkciju te vraća izlaz koji predstavlja procijenjenu vrijednost [20].

Za neuronsku mrežu se također može reći da je to skup međusobno povezanih jedinica koje djeluju kao ljudski mozak. Definiraju živčane stanice te na takav način uče pomoću određenih podataka ulaza i izlaza. Njihovo se ponašanje može utvrditi računalnom simulacijom prilikom klasifikacije, prepoznavanje uzoraka ili predviđanjem. Modeli neuronskih mreža se koriste kako bi se mogla simulirati ljudska učinkovitost nad specifičnim zadacima. Težine (engl. *weights*) povezanosti između neurona se odabiru slučajno kako model ne bi imao nikakvo unaprijed programirano znanje. [22]

Upravo zbog njihove kompleksnosti i načina imitacije ljudskog mozga, kako bi mogle učiti nad određenim podacima, neuronske su se mreže ispitivale u analizi podataka.

4.3.3. Slučajna šuma

Prije opisa algoritma slučajne šume (engl. *random forest*), potrebno je razumjeti kako funkcionira algoritam stabla odluke iz razloga što slučajne šume sadrže veći broj stabala odluke. Stablo odluke

(engl. *Decision tree*) je algoritam koji pripada nadziranom učenju. Svaki čvor (engl. *node*) predstavlja testiranje na atributu iz tablice, dok svaka grana (engl. *branch*) predstavlja izlazni podatak tog testa. Za razliku od drugih algoritama naziranog učenje, može rješavati klasifikacijske i regresijske probleme. Također je i puno brži nego ostali algoritmi klasifikacijskih postupaka [23].

Funkcija stabala odluke na neki način djeluje kao jednostavno ljudsko donošenje odluka. Podatke klasificira na način da postavlja određeni broj pitanja o značajkama povezanim s tim podacima. Svako se pitanje nalazi u čvoru stabla, dok je svaki čvor usmjeren prema svom djetetu (engl. *child*) granom koja predstavlja odgovor na to pitanje. Na taj su način čvorovi i grane posloženi u hijerarhiju te na taj način predstavljaju stablo. Nakon što klasifikacija dođe do kraja, čvor bez djeteta, list, sadrži rezultat klasifikacije. [24]

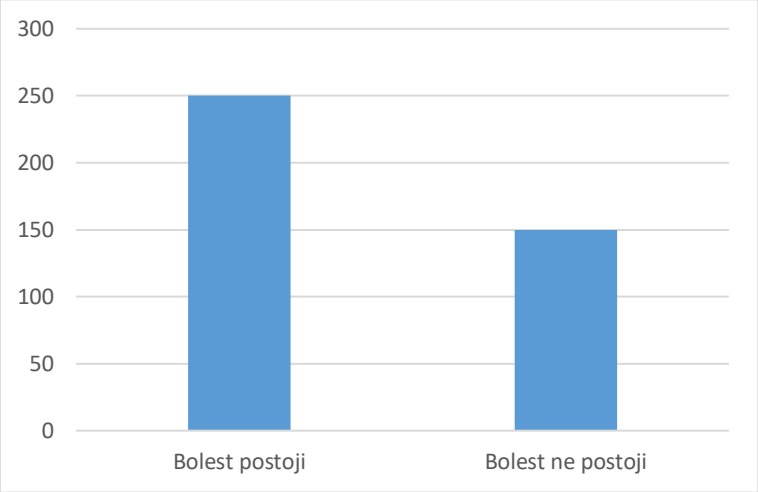
Kako je već spomenuto, slučajne šume su kombinacija većeg broja stabala odluke kako bi klasifikacijski postupci bili što točniji. Kod njega, svako stablo odluke ovisi o vrijednostima i značajkama slučajnih vektora kako bi se mogao dobiti određeni izlazni podatak. Drugim riječima, oni slučajno generiraju stabla odluke, a nakon što svako generirano stablo odluke izračuna izlaznu vrijednost, algoritam slučajne šume odabire najbolji rezultat [25]. Također, još se i nazivaju šumom nasumičnih odluka (engl. *decision forest*).

Iako se algoritam stabla odluke često koristi u različitim postupcima klasifikacije na način da postavlja niz odabranih pitanja u vezi značajki podataka te ga je jednostavno implementirati i razumjeti, u čestim slučajevima jedno stablo ne može dati najbolje rezultati. Tada se koriste algoritmi slučajnih šuma kako bi rezultati bilo što točniji. Ovi algoritmi su također često korišteni za klasifikacijske i regresijske postupke te u većini slučajeva daju jako dobre rezultate. Također, testiranjem opisanih postupaka analize podataka strojnog učenja, ovaj je algoritam davao najbolje rezultate, odnosno točnost procjene rizika oboljenja od bubrežnih bolesti te je zbog toga odabran kao klasifikacijski postupak diplomskog rada.

4.4. Baza podataka korištena za analizu podataka

Skup grupiranih podataka u tablici omogućio je modelu slučajnih šuma strojnog učenja procjenu rizika oboljenja od bubrežne bolesti pomoću odabranog postupka analize podataka. Korištena baza podataka [26] je *csv* formata, a sadrži 14 stupaca i 400 redova. Svi stupci tablice su popunjeni brojčanim vrijednostima. Naime, neke su se mjerne jedinice morale pretvarati zbog pridržavanja

Hrvatskih standarda. Jedan je stupac predstavljao kategoričnu vrijednost, pomoću koje je model mogao pravilno izvesti klasifikacijske postupke. Slikom 4.4 prikazuje se vizualizacija ukupnih brojeva te kategorične vrijednost. Može se primijetiti da ima 250 osoba kod kojih postoji bubrežna bolest te 150 osoba kod kojih bolest ne postoji, što znači da je 62.5% osoba oboljelo od bubrežne bolesti, dok je 37.5% osoba zdravo.



Slika 4.4: Vizualizacija ukupnih brojeva kategoričnih vrijednosti

Ostalih 13 stupaca tablice sadržavaju vrijednosti pojedinih čimbenika arterijske hipertenzije, hematoloških, biokemijskih i vrijednosti čimbenika iz urina. Te vrijednosti predstavljaju kolika su odstupanja od referentnih vrijednosti čimbenika pomoću kojih model može pravilno klasificirati, odnosno procijeniti rizik oboljenja od bubrene bolesti.

5. PROGRAMSKO RJEŠENJE WEB APLIKACIJE

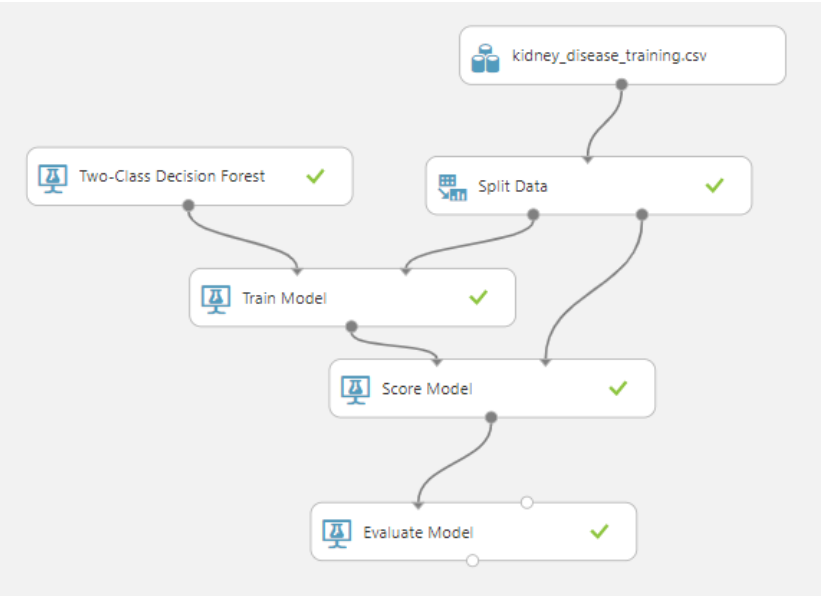
Ovo poglavlje sadrži opis razvojnog okruženja u oblaku računarstva pomoću kojeg se mogla koristiti usluga strojnog učenja. Također sadrži i opis svih korištenih tehnologija za izradu web aplikacije i njezinog sučelja.

5.1. Razvojna okolina u oblaku računala

Microsoft Azure je Microsoftovo okruženje računarstva u oblaku koja pruža niz usluga manipuliranja aplikacija. Osigurava infrastrukturu kao uslugu (*IaaS*), program kao uslugu (*SaaS*) i platformu kao uslugu (*PaaS*) te omogućuje izvršavanje složenih rješenja razvijenih različitim programskim jezicima koji nisu kompatibilni u drugim okruženjima.

Također nudi i uslugu razvojnog okruženja putem internetskih pretraživača, *Microsoft Machine Learning Studio*. Takvo okruženje omogućava spajanje različitih modula pomoću kojih se može realizirati model strojnog učenja. Takvi modeli mogu poslužiti u razvoju brojnih aplikacija koje se temelje na procjenama vrijednosti klasifikacija pomoću algoritama strojnog učenja.

Slika 5.1 prikazuje model slučajne šume pomoću kojeg su se klasificirali podaci temeljeni na čimbenicima rizika obolijevanja od bubrežne bolesti, točnije kroničkog bubrežnog zatajenja.

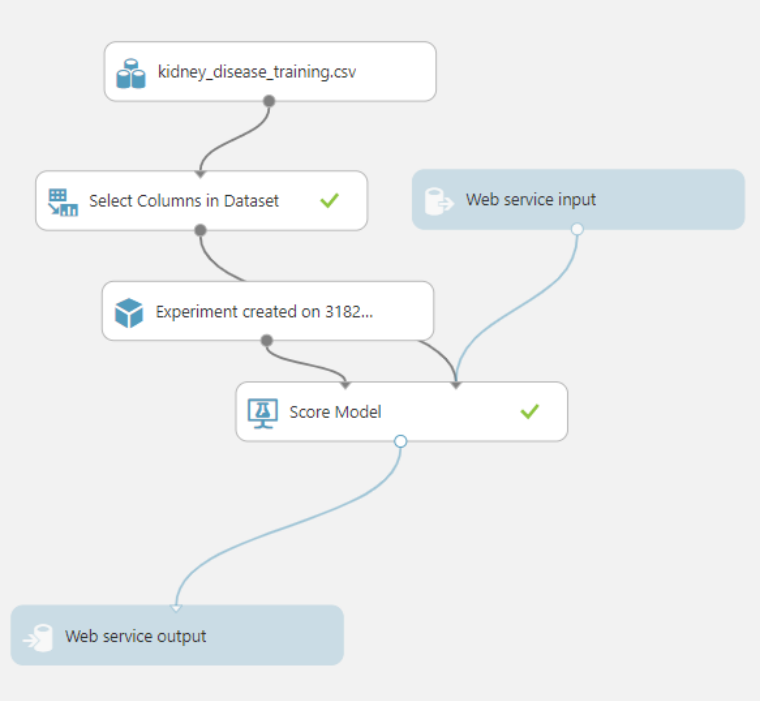


Slika 5.1: Model slučajne šume

Modulom *Split Data* baza podataka podijelila je redove omjerom 70:30 što znači da je 70% podataka korišteno kao podaci za treniranje modela, a ostatak se koristio za testiranje. Za klasifikaciju unesenih podataka koristio se binarni algoritam slučajnih šuma jer je davao najveću točnost. Modul *Train Model* omogućio je tu klasifikaciju podataka. Modulom *Score Model* stvorile su se procijenjene vrijednosti, a modul *Evaluate Model* vrednovao je model, odnosno izračunavao točnost samog modela.

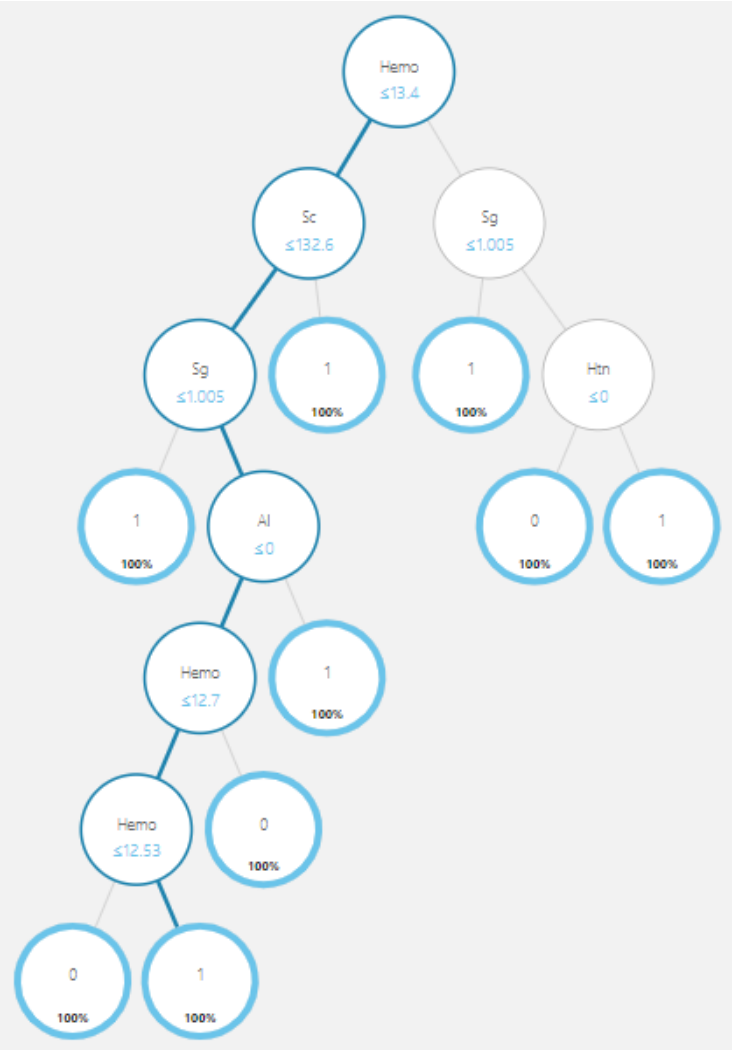
Nakon razvoja modela, koji može raditi procjene ulaznih podatka, može ga se postaviti kao web usluga kako bi se mogao upotrebljavati preko vanjskih aplikacija u stvarnom vremenu. Svako slanje zahtjeva rezultira vraćanjem odgovora koji sadrži određene vrijednosti procjene.

Slika 5.2 pokazuje web uslugu korištenog modela strojnog učenja. *API* razvojno okruženje *Postman* je omogućilo jednostavnu implementaciju web usluge u web aplikaciju te je bio zaslužan za prosljeđivanje ulaznih podataka korisnika kao zahtjev i vraćanje procjene rizika kao odgovor.



Slika 5.2: Microsoft Azure-ova web usluga modela

Spomenuta web usluga te podatke klasificira modelom slučajne šume jer su pomoću tog modela rezultati bili najtočniji. U postavkama algoritma slučajne šume postavilo se 8 stabala odluke te će na temelju toga ovaj algoritam odabrati jedno najbolje rješenje. Slika 5.3 prikazuje primjer puta od korijenskog čvora do lista jednog stabla odluke.



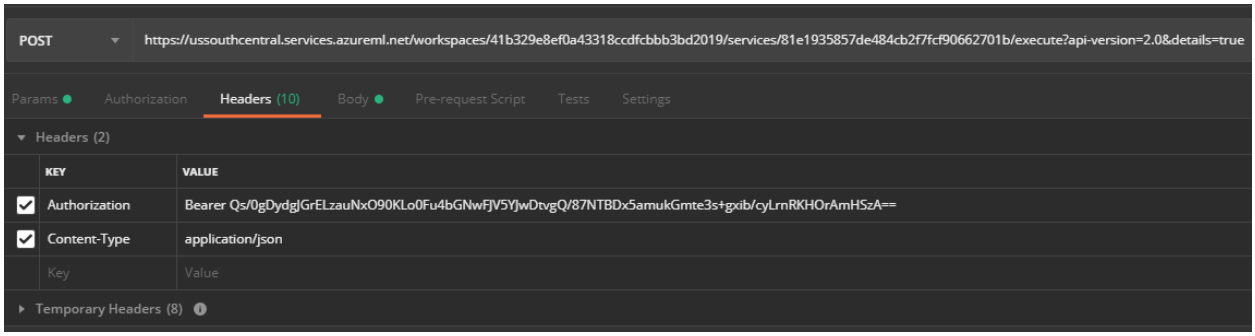
Slika 5.3: Primjer puta jednog stabla odluke

Kako je već spomenuto, algoritam stabla odluke funkcionira na način da postavi određeni broj pitanja u vezi značajki podataka iz implementirane baze podataka. Plava linija sa slike predstavlja putanju od korijenskog čvora, odnosno čvora bez roditelja, do lista, odnosno čvora bez djeteta. Budući da pitanja mogu biti bilo što vezano za značajke, ovim se slučajem prvo postavlja pitanje: „Je li vrijednost hemoglobina manja ili jednaka od 13.4?“. Zatim se postavlja pitanje: „Je li vrijednost kreatinina iz seruma manja ili jednaka 132.6?“. Nakon toga se pita: „Je li vrijednost specifične težine urina manja ili jednaka 1.005?“. Poslije toga, postavlja se pitanje: „Je li vrijednost albumina manja ili jednaka 0?“. Zatim se postavi pitanje: „Je li vrijednost hemoglobina manja ili jednaka 12.7?“ te se na samom kraju postavi pitanje: „Je li vrijednost hemoglobina manja ili jednaka 12.53?“.

5.1.1. Implementacija postupka analize podataka

Kako bi se korišteni postupak analize podataka, odnosno model strojnog učenja mogao implementirati u web aplikaciju, prvo je potrebno osigurati konzumaciju postavljene web usluge pomoću koje se stvori *API* ključ. Postman [27] je API razvojno okruženje čije značajke pojednostavljuju stvaranje bilo kakvih aplikacijskih sučelja. Zatim je ovo okruženje putem stvorenog API ključa prosljeđivalo HTTP zahtjeve između web aplikacije i konzumirane web usluge. Razvojno okruženje Postman zapravo stoji između zahtjeva (engl. *request*) i odgovora (engl. *response*) te na taj način vrši komunikacije između njih.

Kako je navedeno, konzumacijom web usluge stvori se API ključ, jedinstveni identifikator elektroničkog izvora (engl. *Uniform Resource Identifier - URI*) i podaci u JSON formatu. Slika 5.4 prikazuje korišteni zahtjev API razvojnog okruženje Postman koji je omogućio prosljeđivanje vrijednosti čimbenika rizika oboljenja od bubrežnih bolesti.



Slika 5.4: Zahtjev web aplikacije u API razvojnom okruženju Postman

POST je metoda putem koje se podaci, odnosno vrijednosti čimbenika rizika, prosljeđuju web usluzi. Također, potrebno je unijeti dva bitna zaglavlja (engl. *header*) koje treba unijeti kako bi procjena rizika obolijevanja bila moguća. Prvo je zaglavlje autorizacija (engl. *authorization*) putem API ključa, dok druga označava da se tip sadržaja (engl. *content type*) prosljeđuje u JSON formatu.

Programski kod 5.5 označava podatke u JSON formatu koje se treba unijeti u tijelo (engl. *body*) zahtjeva API razvojnog okruženja Postman. Ulazni podaci označavaju vrijednosti čimbenika rizika obolijevanja od bubrežnih bolesti koje korisnik unosi u obrazac web aplikacije.

```
{
  "Inputs": {
    "input1": {
      "ColumnNames": [
        "Bp",
        "Sg",
        "Al",
        "Su",
        "Rbc",
        "Bu",
        "Sc",
        "Sod",
        "Pot",
        "Hemo",
        "Wbcc",
        "Rbcc",
        "Htn"
      ],
      "Values": [
        [
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0",
          "0"
        ]
      ]
    }
  },
  "GlobalParameters": {}
}
```

Programski kod 5.5: Zahtjev web aplikacije u Postman API razvojnom okruženju

Zahtjev razvojnog okruženja Postman može se preoblikovati u određeni programski kod koji se može koristiti u različitim aplikacijama. Za ostvarenje prosljeđivanja HTTP zahtjeva web aplikacije koristio se PHP programski jezik i biblioteka cURL koji će biti opisani u sljedećim poglavljima.

Nakon što korisnik, odnosno pacijent, u obrazac web aplikacije unese svoje vrijednosti čimbenika rizika obolijevanja, razvojno okruženje Postman te podatke proslijedi konzumiranoj web usluzi koja ih klasificira modelom slučajne šume te kao odgovor vrati rezultat predviđanja i vjerojatnost pojavljivanja. Taj odgovor Postman proslijedi web aplikaciji te ih aplikacija spremi u bazu podataka.

5.2. Potrebne programske tehnologije i jezici

Funkcionalni se dio web aplikacije pisao *PHP* programskim jezikom, dok je izgled aplikacije, odnosno sučelja omogućio *HTML* i *CSS*. U nastavku su opisane sve korištene tehnologije i jezici.

5.2.1. LAMP

LAMP predstavlja kombinaciju tehnologija *Linux*, *Apache*, *MySQL* i *PHP*. Zbog učinkovitosti, jednostavnosti i sigurnosti, ovakva kombinacija je vodeća u razvijanju aplikacija.

Linux je besplatni operacijski sustav sličan *Unixu*. Imenovan je po njegovom autoru, Linusu Torvaldsu. Otvorenog je izvornog koda, a korištena distribucija bila je *Mint* koja je ujedno i najkorištenija i najjednostavnija za korištenje [28]. Preko njega su se sve tehnologije instalirale i koristile. Za pisanje funkcionalnog dijela koda koristilo se integrirano razvojno okruženje (engl. *Integrated Development Environment - IDE*) *PhpStorm*, koji uz sami *PHP* omogućuje manipuliranje bazom podataka i korištenje *HTML*, *CSS* i *JavaScript* jezika, dok je za verzioniranje izvornog koda koristio sustav *Git*.

Apache je odgovoran za uspješno pokretanje web aplikacije na internetskim pretraživačima jer je on web poslužitelj. Moguće je stvaranje virtualnog poslužitelja pomoću kojega bi se moglo pokrenuti veći broj stranica na jednom serveru [28]. Uz virtualnog poslužitelja koristila se njegova smjernica pseudonima, odnosno alternativno ime poslužitelja, koje je omogućilo pokretanje aplikacije na svim spojenim uređajima te mreže.

MySQL je snažan alat za upravljanje bazama podataka koje omogućuju razne funkcionalnosti poput spremanja i dohvaćanja podataka pomoću različitih programskih jezika. Korištenje baza podataka izuzetno je važno za razvoj dinamičkih web aplikacija jer se, recimo, prijavom korisnika aplikacija povezuje na bazu podataka kako bi dohvatila određene podatke.

PHP: Hypertext Preprocessor (*PHP*) programski je jezik namijenjen razvoju web aplikacija. Omogućuje i paradigmu objektno-orijentiranog programiranja čiji se koncept odnosi na upravljanje objektima i klasama. Budući da se može povezati s *HTML-om*, pojednostavljuje i poboljšava razvoj internetskih stranica [28].

Programski kod 5.6 je primjer kako se stranica web aplikacije prikazuje (engl. *render*). Ovo je ujedno i prikazivanje načina povezivanja *PHP-a* i *HTML-a*. Pozivanje metode se nalazi u upravljačkoj komponenti iz razloga što ta komponenta služi za povezivanje već objašnjenog modela i pogleda. Kao prvi parametar prihvaća putanju, odnosno direktorij, gdje se *HTML* stranica nalazi, a kao drugi parametar uzima varijable koje će se koristiti na toj *HTML* stranici. Nakon pozivanja funkcija *ob_start()*, počinju se spremati svojstva (engl. *properties*) i uključuje se stranica u međuspremnik, dok se funkcijom *ob_get_clean()* završava spremanje. Metoda kao izlazni podatak vrati taj međuspremnik, a stranica bude prikazana.

```
1.  public function render($templateFile, array $vars = [])
2.  {
3.      if (file_exists(BP . '/app/Views/' . $templateFile)) {
4.          ob_start();
5.          try {
6.              extract($vars, EXTR_SKIP);
7.              include BP . '/app/Views/' . $templateFile;
8.          } catch (\Exception $e) {
9.              ob_end_clean();
10.             throw $e;
11.         }
12.     }
13.
14.     $output = ob_get_clean();
15.
16.     return $output;
17. }
```

Programski kod 5.6: Prikazivanje stranica web aplikacije

5.2.2. HTML i CSS

U internetskim dokumentima, odnosno internetskim stranicama, nalazi se označni jezik *HyperText Markup Language* (HTML) koji služi za njihovo prezentiranje na internetskim preglednicima. Osnovna građa su mu oznake ili ključne riječi koje definiraju prikazivanje sadržaja na svim preglednicima. Kao što je navedeno prethodnim poglavljem, *PHP* programski jezik se može povezati s *HTML*-om te na taj način olakšava razvoj web aplikacija.

Programski kod 5.7 prikazuje način povezivanja *PHP*-a i *HTML*-a primjerom izrade tablice rezultata. Vrijednosti tablice su vrijednosti varijable drugog parametra metode prikazivanja stranice koja je objašnjena prethodnim poglavljem. Glavnu ulogu ima *foreach* petlja koja na temelju određenih podataka popunjava red u tablici.

```
<table>
      <thead>
      <tr>
          <th> br. </th>
          <th> Nalaz izdan </th>
          <th></th>
      </tr>
      </thead>
      <tbody>
      <?php if (isset($row)):  ?>
         <?php foreach ($row as $key => $value): ?>
             <tr>
                 <td> <?php echo $key += 1; ?> </td>
                 <td> <?php echo $value["created"]; ?> </td>
                 <td>
                   <a href="/user/details/<?php echo $value["id"]; ?>"> Pogledaj </a>
                   <a href="/user/delete/<?php echo $value["id"]; ?>"> Obriši </a>
                 </td>
             </tr>
          <?php endforeach; ?>
       <?php endif; ?>
      </tbody>
  </table>
```

Programski kod 5.7: Izrada tablice svih rezultata

Stilski jezik koji je orijentiran prema prezentaciji *HTML* oznaka i kako bi vizualno interpretirao razliku njihove strukture je *Cascading Style Sheets* (*CSS*). *CSS* se najviše koristi zbog izgleda sučelja web aplikacija, a također i omogućuje izgleda sučelja prilagodljivog dizajna koje se može, ovisno o rezoluciji uređaja, pokrenuti na različitim uređajima.

Programski kod 5.8 prikazuje prilagodljivu navigacijsku traku koja se koristila u izgledu sučelja web aplikacije. Pravilo *@media* mijenja izgled navigacijske trake ovisno o rezoluciji uređaja. U ovom slučaju ako je širina prozora manja od 768 elemenata slike (engl. *pixel*).

```
@media (max-width: 768px) {
    .toggle-button{
        display: flex;
    }

    .navbar{
        flex-direction: column;
        align-items: flex-start;
    }

    .nav-links{
        width: 100%;
    }

    .nav-links ul{
        width: 100%;
        flex-direction: column;
    }

    .nav-links li{
```

```
21.         text-align: center;
22.     }
23.
24.     .nav-links li a{
25.         padding: .5rem 1rem;
26.     }
27.
28.     .nav-links.active{
29.         display: flex;
30.     }
31. }
```

Programski kod 5.8: Prilagodljiv dizajn navigacijske trake

5.3. Programsko rješenje na strani poslužitelja

U nastavku su prikazani i objašnjeni dijelovi i primjeri programskog rješenja koji su omogućili spajanje na bazu podataka i procjenu rizika obolijevanja od određenih bubrežnih bolesti. Baza podataka je napravljena pomoću snažnog alata za upravljanje takvim bazama, *MySQL*, dok je programsko rješenje na strani poslužitelja pisano u PHP programskom jeziku.

5.3.1. Baza podataka web aplikacije

Web aplikacija se uglavnom temelji na povezivanju na bazu i manipulaciju s njom. U toj bazi postoje četiri tablice koje se koriste prilikom obavljanja određenih metoda. Glavna je tablica *users* pomoću koje se mogu izvesti metode nad korisničkim računom – izrada računa, prijava ili promjena lozinke. Stupci tablice vezani su za osobne i korisničke podatke. Druga tablica je *rememberme* koja se koristi ovisno o korisničkom zahtjevu vezanog za pamćenje prijave, odnosno kolačićima. Iz treće tablice, *notes,* web aplikacija dohvati bilješku vezanu za stupanj rizika čimbenika – nisko, srednje ili visoko. Posljednja tablica je *details_table* u koju aplikacija sprema sve podatke vezane za rizik oboljenja od bubrežne bolesti. Aplikacija podatke iz ove tablice dohvati ovisno o korisničkoj želji detaljnijeg prikaza dobivenog rezultata.

Programski kod 5.9 pokazuje način izrade tablice *users.* Izjavom (engl. *Statement*) *CREATE TABLE* se stvara tablica. Svaka korištena tablica ima svoj primarni ključ (engl. *primary key*) koji služi za jedinstvenu i jednostavniju identifikaciju svakog reda u tablici. Atribut *AUTO_INCREMENT* služi kako bi stupac s primarnim ključem imao jedinstveni broj. Tipovi *TINYTEXT* i *LONGTEXT* se razlikuju po veličini bajta. *TINYTEXT* može imati do 255 bajta, dok *LONGTEXT* može imati do oko 4 gigabajta. Tip *VARCHAR(10)* označava maksimalno 10 karaktera, dok DATETIME označava datum i vrijeme.

```
CREATE TABLE users(
                        id int(11) AUTO_INCREMENT PRIMARY KEY,
                        firstname TINYTEXT NOT NULL,
                        lastname TINYTEXT NOT NULL,
                        birthdate DATETIME NOT NULL,
                        gender VARCHAR(10) NOT NULL,
                        weight VARCHAR(10) NOT NULL,
                        username TINYTEXT NOT NULL,
                        email TINYTEXT NOT NULL,
                        password LONGTEXT NOT NULL,
                        joined DATETIME NOT NULL,
                        age VARCHAR(10) NOT NULL
);
```

Programski kod 5.9: Način izrade baze podataka i tablice

Programskim kodom 5.10 prikazuje se klasičan način povezivanja s bazom podataka korištenjem obrasca jednog slučaja objekta (engl. *singleton pattern*) pomoću kojeg se može ograničiti broj stvorenih objekata (engl. *instances*) kako ne bi utjecali na učinkovitost aplikacije. Magična metoda __conctruct(), odnosno konstruktor je privatna kako bi se onemogućilo izravno stvaranje objekata. Za povezivanje na bazu koristi se proširenje PHP programskog jezika *PHP Data Objects* (*PDO*) gdje se stvara objekt s podacima upravljačkog programa, poslužitelju i nazivu baze podataka. Kao ostale parametre treba unijeti *MySQL* korisničko ime i lozinku.

```
private function __construct()
    {
        try{
            $this->_pdo = new \PDO(
                'mysql:host='. Config::getInstance()->getConfig("mysql/host").
                ';dbname=' . Config::getInstance()->getConfig("mysql/database"),
                Config::getInstance()->getConfig("mysql/username"),
                Config::getInstance()->getConfig("mysql/password")
            );
        }catch (PDOException $e){
            die($e->getMessage());
        }

    }
    public static function getInstance()
    {
        if(!isset(self::$_instance))
        {
            self::$_instance = new self();
        }
        return self::$_instance;
    }
```

Programski kod 5.10: Način povezivanja s bazom podataka

Nakon uspješnog povezivanja web aplikacije s bazom podataka, omogućena je manipulacija njome. Mogu se odrađivati određene funkcije poput dodavanja novih redova u tablicu, odnosno

unošenje novih podataka, dohvaćanje podataka ili brisanje podataka iz tablice. Programski kod 5.11 prikazuje metodu pomoću koje se dohvaćaju podaci iz tablice. Kao parametre, metoda prima ime tablice, ime stupaca i vrijednost. Prvo se stvara objekt povezivanja pozivom metode za dohvaćanje povezivanja s bazom podataka. Zatim se, ovisno o imenu tablice ili točno definiranim stupcima, pokreće određena *if* naredba. Nakon toga se *MySQL* izjavom *SELECT* podaci mogu učitati i kao povratna izjava vratiti i poslati na daljnju obradu u web aplikaciji. Sve se nalazi unutar *try – catch* bloka kako bi se moglo upravljati iznimkama.

```php
protected function load($table, $field, $value, array $columns = [])
  {
      try {
          $this->connection = Database::getInstance()->getConnection();
          if (!empty($columns)) {
              $fields = implode(", ", $columns);
              $sql = "SELECT {$fields} FROM {$table} WHERE {$field} = '{$value}'";

              $row = $this->connection->query($sql);
              $row->setFetchMode(PDO::FETCH_ASSOC);

              return $row->fetchAll();
          }
          if ($table === "details_table") {
              $sql = "SELECT * FROM {$table} WHERE {$field} = '{$value}'";

              $row = $this->connection->query($sql);
              $row->setFetchMode(PDO::FETCH_ASSOC);

              return $row->fetchAll();
          } else {
              $sql = "SELECT * FROM {$table} WHERE {$field} = '{$value}'";

              $row = $this->connection->query($sql);
              $row->setFetchMode(PDO::FETCH_CLASS, __CLASS__);

              return $row->fetch();
          }
      } catch (\PDOException $e) {
          $e->getMessage();
      }
  }
```

Programski kod 5.11: Metoda učitavanja podataka

Programski kod 5.12 prikazuje dio programskog rješenja gdje se unose podaci u tablicu baze podataka. Implementirane funkcije *explode* i *implode* služe kako bi se stupci pripremili u određenu sintaksu kako bi se izjavom *INSERT* podaci mogli spremiti u bazu. Budući da se registracijom unosi datum rođenja, prije samog spremanja se taj datum pretvori u životu dob.

```php
if ($table == "users") {
                unset($data["signup-submit"]);
                unset($data["password-again"]);

```

```
                $fields = implode(", ", array_keys($data));

                $values = explode(",", (":" . implode(",:", array_keys($data))));
                $values = implode(", ", $values);
                $age = (int)((time()-strtotime($data["birthdate"]))/(3600*24*365));
                $sql = "INSERT INTO {$table} (" . $fields . ", joined, age)
                        VALUES (" . $values . ", :joined, :age)";

                $stmt = $this->connection->prepare($sql);

                foreach ($data as $key => $value) {
                    $stmt->bindValue(":" . $key, $value);
                }

                $stmt->bindValue(":joined", date("Y-m-d H:i:s"));
                $stmt->bindValue(":age", $age);
            }
```

Programski kod 5.12: Unos podataka u tablicu

5.3.2. Izrada korisničkog računa

Nakon što korisnik unese podatke u registracijski obrazac pozove se metoda *signup* upravljačke komponente *Registration*. Programskim kodom 5.13 prikazuje se metoda za registraciju koja modelu prosljeđuje unesene podatke. Prije nego što se novi korisnički račun izradi, mora se provjeriti točnost unesenih podataka određenim validacijskim postupcima. Ako metoda validacije *isPassed()* kao povratnu izjavu vrati neistinu (engl. *return false*), pronađena je pogreška prilikom validacije te se na sučelju prikazuje poruka o tome. Ukoliko metoda vrati istinu (engl. *return true*), stvara se novi objekt modela *User* s unesenim podacima u konstruktoru te ih na taj način prosljeđuje na daljnju obradu.

```
public function signup()
    {
        if(isset($_POST["signup-submit"]))
        {
            $validation = new RegistrationValidation($_POST);

            if(!$validation->isPassed()) {
                echo $this->view->render('Register/index.phtml', [
                    'errors' => $validation->getError()
                ]);
            }
            else{
                $user = new User($_POST);
                $user->newuser();

                echo $this->view->render('Register/index.phtml', [
                    'success' => 'Registration successfully completed.',
                    'errors' => $user->getError()
                ]);
            }
        }
    }
```

Programski kod 5.13: Metoda upravljačke komponente za izradu korisničkog računa

Validacijskim postupcima u registraciji provjerava se točnost podataka koje je unio korisnik. Nakon što su se podaci proslijedili, provjerava se ima li praznih polja, sadržavaju li ime i prezime samo slova i sl. Primjerice, programskim kodom 5.14 prikazuju se postupci validacije elektroničke pošte i lozinke. *PHP* u sebi već ima implementiranu funkciju filtracije podataka sa specifičnim podatkom koja može provjeriti pravilnu elektroničnu poštu. Ostali se podaci provjeravaju podudaranjem regularnih izraza.

```
1.  public function email()
2.      {
3.          if (!filter_var($this->_data["email"], FILTER_VALIDATE_EMAIL)) {
4.              $this->makeError("email", "Email adresa " . $this-
    >_data["email"] . " nije vazeca.");
5.          }
6.      }
7.
8.      public function password()
9.      {
10.         if (strlen($this->_data["password"]) < 5) {
11.             $this->makeError("password", "Duljina lozinke ne moze biti manja od 5.");
12.         } else if (!preg_match("#[0-9]+#", $this->_data["password"])) {
13.             $this->makeError('password', 'Lozinka mora sadrzavati broj.');
14.         } else if (!preg_match("#[A-Z]+#", $this->_data["password"])) {
15.             $this->makeError('password', 'Lozinka mora sadrzavati veliko slovo.');
16.         } else if ($this->_data["password"] !== $this->_data["password-again"]) {
17.             $this->makeError('password', 'Unesene lozinke se ne podudaraju.');
18.         }
19.     }
```

Programski kod 5.14: Primjer validacijskih postupaka izrade korisničkog računa

Metoda *newuser()* modela *User* programskog koda 5.15 prvo provjerava postoje li korisničko ime i lozinka u bazi podataka. Ako ne postoji, lozinka se šifrira pomoću *bcrypt* algoritma u sažetak poruke te se svi uneseni podaci unose u bazu podataka metodom *insert()*.

```
1.  public function newuser()
2.    {
3.        $this->usernameexists();
4.        $this->emailexists();
5.
6.
7.       if($this->isPassed()) {
8.           $this->data["password"] = password_hash(
9.                                     $this->data['password'],
10.                                    PASSWORD_DEFAULT);
11.
12.          $this->insert("users", $this->data);
13.       }
14.   }
```

Programski kod 5.15: Metoda modela za izradu računa

5.3.3. Prijava korisnika

Izradom korisničkog računa podaci se spremaju u tablicu *users* baze podataka. Prilikom prijave, odnosno kada korisnik pritisne tipku za ulazak u web aplikaciju, pozove se metoda *login()* upravljačke komponente *Login*.

Programskim kodom 5.16 je prikazana metoda za prijavu korisnika. Prema korisničkoj želji, potvrdni okvir za pamćenje prijave može, a i ne mora, biti pritisnut. To se provjerava upitom uvjetnog operatora u metodi *login()*. Ako je uvjet istinit, svojstvo *$remember* će biti istinito, a u suprotnom neistinito. Zatim se stvara novi objekt metode *User* s unesenim podacima te se poziva metoda *userlogin()* koja kao parametar prima to svojstvo.

```
1.  public function login()
2.     {
3.         if (isset($_POST["login-submit"]))
4.         {
5.             if (isset($_POST['remember']))
6.                 $remember = ($_POST['remember'] === 'on') ? true : false;
7.             else
8.                 $remember = false;
9.
10.            $user = new User($_POST);
11.
12.            if (!$user->userlogin($remember)) {
13.                echo $this->view->render('Login/index.phtml', [
14.                    'errors' => $user->getError(),
15.                ]);
16.            } else {
17.                echo $this->view->render('Home/index.phtml');
18.            }
19.        }
20.    }
```

Programski kod 5.16: Metoda upravljačke komponente za prijavu korisnika

Spomenuta metoda *userlogin()* prikazana je programskim kodom 5.17. Prvo svojstvo *$loggeduser* dohvaća podatke o korisniku na temelju unesenog korisničkog imena ili elektroničke pošte. Zatim se, ako to svojstvo nije prazno, odnosno ako sadrži dohvaćene podatke o korisniku iz baze, unesena lozinka provjerava s lozinkom iz baze podataka. Ako funkcija *password_verify()* vrati neistinu, znači da provjera lozinki nije uspjela te kao rezultat vraća pogrešku o netočno unesenoj lozinki. Nadalje, ako je provjera uspješna, stvaraju se informacije u obliku sesije *id* i *gender* pomoću globalne varijable *$_SESSION*. Informacija *id* je zapravo primarni ključ iz baze te ona korisniku pruža usluge prilikom korištenja aplikacije. Primjerice, rezultati procjene rizika obolijevanja se u bazu spremaju uz taj *id* pomoću kojeg aplikacija može prepoznati koje sve redove iz tablice mora dohvatiti. Informacija *gender* služi prilikom provjere čimbenika rizika s referentnim vrijednostima

jer neki čimbenici ovise o spolu pacijenata. S druge strane, ako je korisnik prilikom prijave pritisnuo potvrdni okvir za pamćenje prijave, svojstvo *$remember* bude istinito. Tada se generira sažetak poruke i tako se sprema u posebnu tablicu *rememberme*. Na taj način, aplikacija uz korištenje kolačića pomoću globalne varijable *$_COOKIE* stvara datoteku na korisnikovom računalu. Pomoću te datoteke, prijava ostane zapamćena prilikom sljedeće posjete aplikaciji.

```
1.  public function userlogin($remember = false)
2.  {
3.      $loggeduser = $this->loaduser(
4.                     'users',
5.                     ['username', 'email'],
6.                     $this->data['usernameoremail']);
7.
8.      if(!empty($loggeduser)) {
9.          if (!password_verify(
10.             $this->data["password"],
11.             $loggeduser->data["password"])) {
12.               $this->makeError("password", "Lozinka nije tocna.");
13.               return false;
14.         } else {
15.             Session::set('id', $loggeduser->data['id']);
16.             Session::set('gender', $loggeduser->data['gender']);
17.
18.             if($remember)
19.             {
20.                 $hash = Hash::unique();
21.                 $hashCheck = $this->checkrememberme();
22.
23.                 if(!$hashCheck)
24.                 {
25.                     $this->insert("rememberme", [
26.                         'user_id' => $loggeduser->data["id"],
27.                         'hash' => $hash
28.                     ]);
29.                 }
30.                 else{
31.                     $hash = $hashCheck->data['hash'];
32.                 }
33.                     Cookie::put(
34.                     $this->_cookiename,
35.                     $hash,
36.                     Config::getInstance()->getConfig("rememberme/cookie_expiry"));
37.             }
38.             return true;
39.         }
40.     }
41.     else{
42.         $this->makeError(
43.               "account",
44.               "Netocan korinicki racun ili email adresa.");
45.     }
46. }
```

Programski kod 5.17: Metoda modela za prijavu korisnika

5.3.4. Unos parametara rizika

Potvrdom unesenih vrijednosti čimbenika rizika obolijevanja od bubrežnih bolesti, poziva se metoda *submit()* upravljačke komponente *Parameters* koja je prikazana programskim kodom 5.18. U toj se metodi vrši provjera tih podataka na isti način kao kod izrade računa, provjerom regularnih izraza. Povodom pravilnog utvrđivanja tih podataka, stvara se objekt modela *Kidney*. Zatim se stvaraju rezultati zdravstvenog stanja korisnika pomoću metode *genetateTests()* gdje se ti svi podaci provjeravaju s referentnim vrijednostima čimbenika rizika i šalju web usluzi na provjeru pomoću postupka strojnog učenja.

```
1.  public function submit()
2.      {
3.          if (isset($_POST['check-submit'])) {
4.              $validation = new ParametersValidation($_POST);
5.
6.              if (!$validation->isPassed()) {
7.                  echo $this->view->render('Predict/index.phtml', [
8.                      'errors' => $validation->getError()
9.                  ]);
10.             } else {
11.                 $test = new Kidney($_POST);
12.                 $test->generateTests();
13.
14.                 echo $this->view->render('Predict/index.phtml', [
15.                     'success' => "Submited"
16.                 ]);
17.             }
18.         }
19.     }
```

Programski kod 5.18: Metoda upravljačke komponente za potvrdu čimbenika

Programski kod 5.19 prikazuje programsko rješenje metode *generateTests()*. Kako je već spomenuto, tom se metodom podaci uspoređuju s referentnim vrijednostima svih čimbenika rizika gdje se, ovisno o stupnju tog rizika, iz baze uzima određena bilješka o potencijalnoj bolesti. Nakon provedbe uspoređivanja, vrijednosti se šalju na web uslugu pomoću *API* ključa metodom *checkDiseaseAzure()*. Zatim se dobiveni rezultati procjene, sve vrijednosti i bilješke spremaju u bazu podataka pomoću opisane metode *insert()*.

```
1.  public function generateTests()
2.  {
3.      $systolic = (int)$this->data["bp-sys"];
4.      $diastolic = (int)$this->data["bp-dia"];
5.      $hypertension = $this->bloodPressureHandler($systolic, $diastolic);
6.
7.      $rbc = round($this->data["rbc"], 3);
8.      $hemo = round($this->data["hemo"], 3);
9.      $rbc_class = $this->hematologyFactors($rbc, $hemo);
```

```
10.
11.     $bu = round($this->data["bu"], 3);
12.     $sc = round($this->data["sc"], 3);
13.     $sod = round($this->data["sod"], 3);
14.     $pot = round($this->data["pot"], 3);
15.     $su = round($this->data["su"], 3);
16.
17.     $sugar_class = $this->biochemicalFactors($su, $bu, $sc, $sod, $pot);
18.
19.     $sg = round($this->data["sg"], 5);
20.     $al = $this->data["al"];
21.     $wbcc = $this->data["wbcc"];
22.     $rbcc = (int)$this->data["rbcc"];
23.
24.     $alscRatio = round($al / $sc, 2);
25.     $albumin_class = $this->urinaryFactors($sg, $al, $rbcc, $wbcc, $alscRatio);
26.
27.     $extraTest = $this->generateExtraTest();
28.
29.     $fromAzure = $this->checkDiseaseAzure([
30.         $diastolic, $sg, $albumin_class, $sugar_class, $rbc_class,
31.         $bu, $sc, $sod, $pot, $hemo, (int)$wbcc * 1000,
32.         $rbcc, $hypertension]);
33.
34.     $data = [];
35.     array_push($data,
36.                 $systolic, $diastolic, $sg, $al, $alscRatio,
37.                 $su, $rbc, $bu, $sc, $sod, $pot, $hemo,
38.                 $wbcc, $rbcc, $fromAzure["ckd"], $fromAzure["ckdprecision"],
39.                 $extraTest['bun_sc_ratio'], $extraTest['clearance_creatinine'],
40.                 $extraTest['gfr'], array_values($this->getNotes()));
41.
42.    $this->insert("details_table", $data);
43. }
```

Programski kod 5.19: Metoda modela za stvaranje rezultata zdravstvenog stanja

Jedan od načina usporedbe unesenih podataka prikazan je programskim kodom 5.20. Radi se o provjeri ukazivanja anemije kod kroničnog zatajenja bubrega pomoću opisane Cockcroft-Gaultove jednadžbe (2-3) u drugom poglavlju koja služi za dobivanje klirensa kreatinina koji ovisi o životnoj dobi, spolu i težini pacijenta. Vrijednost serumskog kreatinina se dijeli s 88.4 zbog pretvaranja u mjernu jedinicu koju koristi klirens kreatinina, a to je ml/min. Zatim se naredbom *if* provjerava spol korisnika definiranog kao informacija u obliku sesije prilikom prijave korisnika kako bi algoritam mogao znati treba li se klirens kreatinina pomnožiti s 0.85. Zatim se, drugom naredbom, provjerava ako je ta vrijednost manja od 40 ml/min iz razloga što se anemija pretežito javlja kod kroničnog bubrežnog zatajenja padom ispod te vrijednosti. Na kraju se na temelju te vrijednosti iz baze podataka uzima određena bilješka zdravstvenog stanja.

```
1.  $sc = $creatinine / 88.4;
2.  $clearance_creatinine = round(((140 - $age) * $weight) / (72 * $sc), 2);
3.
4.  if(Session::get("gender") === "Female") {
5.              $clearance_creatinine = $clearance_creatinine * 0.85;
6.          }
```

```
7.
8.  if($clearance_creatinine < 40)
9.    {
10.      $note = $this->getNote('crcl_anem');
11.      $this->writeNote('crcl_anem', $note["note"]);
12.
13.     }else{
14.        $note = $this->getNote('crcl_anem_normal');
15.        $this->writeNote('crcl_anem_normal', $note["note"]);
16.     }
17.   }
```

Programski kod 5.20: Primjer dijela programskog rješenja usporedbe vrijednosti

Programskim kodom 5.21 prikazuje se programsko rješenje slanja i primanja podataka web usluge. Kako bi se rezultati procjene na temelju unesenih podataka mogli dobiti, koristila se implementirana *PHP* biblioteka *cURL*. Biblioteka omogućuje komunikaciju s različitim poslužiteljima. Funkcijom *curl_init()* se stvara nova sesija biblioteke, a pomoću *curl_setopt_array()* su se postavile različite postavke za komunikaciju, poput mrežne adrese poslužitelja, unesenih podataka u obliku *JSON* formata i *API* ključ. Na kraju se funkcijom *curl_exec()* ta *cURL* sesija izvede. Drugim riječima, podaci se pošalju na web uslugu kao zahtjev i tamo pomoću postupka strojnog učenja modelom slučajne šume procijeni rizik obolijevanja od bubrežne bolesti te se kao odgovor vrati procijenjena vrijednost.

```
1.  $curl = curl_init();
2.
3.  curl_setopt_array($curl, array(
4.      CURLOPT_URL => Config::getInstance()->getConfig("azure/url"),
5.      CURLOPT_RETURNTRANSFER => true,
6.      CURLOPT_ENCODING => "",
7.      CURLOPT_MAXREDIRS => 10,
8.      CURLOPT_TIMEOUT => 0,
9.      CURLOPT_FOLLOWLOCATION => true,
10.     CURLOPT_HTTP_VERSION => CURL_HTTP_VERSION_1_1,
11.     CURLOPT_CUSTOMREQUEST => "POST",
12.     CURLOPT_POSTFIELDS => "{\n  \"Inputs\": {\n
13.     \"input1\": {\n
14.     \"ColumnNames\": [\n
15.     \"Bp\",\n
16.     \"Sg\",\n
17.     \"Al\",\n
18.     \"Su\",\n
19.     \"Rbc\",\n
20.     \"Bu\",\n
21.     \"Sc\",\n
22.     \"Sod\",\n
23.     \"Pot\",\n
24.     \"Hemo\",\n
25.     \"Wbcc\",\n
26.     \"Rbcc\",\n
27.     \"Htn\",\n      ],\n
28.     \"Values\": [\n
29.     [\n
30.     \" $data[0] \",\n
31.     \" $data[1] \",\n
```

```
32.     \" $data[2] \",\n
33.     \" $data[3] \",\n
34.     \" $data[4] \",\n
35.     \" $data[5] \",\n
36.     \" $data[6] \",\n
37.     \" $data[7] \",\n
38.     \" $data[8] \",\n
39.     \" $data[9] \",\n
40.     \" $data[10] \",\n
41.     \" $data[11] \",\n
42.     \" $data[12] \",\n
43.     ]\n      ]\n    }\n  },\n  \"GlobalParameters\": {}\n}",
44.     CURLOPT_HTTPHEADER => Config::getInstance()->getConfig("azure/header"),
45. ));
46.
47. $response = curl_exec($curl);
48.
49. curl_close($curl);
```

Programski kod 5.21: Dio programskog rješenja slanja i primanja podataka na web uslugu

5.4. Programsko rješenje na strani korisnika

U nastavku je objašnjen i prikazan izgled sučelja web aplikacije. Programska rješenja pisana su pomoću *HTML*-a i *CSS*-a. Kako je objašnjeno prethodnim poglavljem i programskim kodom 5.6, moguće je povezati *PHP* s *HTML-om*. Na taj su se način svojstva *PHP* programskog jezika proslijedila te koristila kao prikaz rješenja na strani korisnika, odnosno na sučelju web aplikacije.

5.4.1. Obrasci unosa podataka

Obrasci (engl. *Forms*) služe za prosljeđivanje podataka poslužitelju. Na taj način, podatke koje je korisnik unio u obrazac, *HTML* proslijedi upravljačkoj komponenti *PHP*-a na daljnju obradu. U web aplikaciji postoje četiri vrste obrazaca. Prvi obrazac se odnosi na izradu korisničkog računa u kojem korisnik unosi svoje osobne podatke i podatke o korisničkom računu. Drugi obrazac je obrazac prijave u kojeg se unosi korisničko ime ili elektronička pošta i lozinka. U trećem obrascu korisnici popunjavaju vrijednosti čimbenika rizika za procjenu obolijevanja kako bi mogli pratiti zdravstveno stanje, a četvrtim se obrascem potvrđuje promjena lozinke unosom trenutne.

Obrazac izrade korisničkog računa se dijeli na tri koraka. Prvi je unos osobnih podataka gdje se upisuje ime i prezime, datum rođenja, spol i tjelesna težina mjerena u kilogramima. Drugi je unos podataka o korisničkom računu gdje se unosi korisničko ime, elektronička pošta i lozinka, a treći je korak potvrda izrade računa unosom lozinke. Programski kod 5.22 prikazuje prvi korak izrade. Postoji nekoliko *div* oznaka koje dijele informacije na različite odjeljke gdje svaki ima svoj naziv

postavljen klasama i identifikacijskom oznakom. Također postoje i oznake ulaza (engl. *input*) unutar kojih korisnik unosi podatke.

```
<div id="p1">
    <div class="panel">
        <div class="panel-heading">
            <h3>Korak 1. Osobni podaci</h3>
        </div>
        <div class="page-heading">
            <h3 class="page-text">Molimo upišite ime i prezime:</h3>
        </div>

        <input type="text" name="firstname" placeholder="Ime"><br>
        <input type="text" name="lastname" placeholder="Prezime"><br>

        <div class="page-heading">
            <h3 class="page-text">Molimo upišite datum rođenja:</h3>
        </div>

        <input type="date" name="birthdate" min="1920-12-31" max="2010-12-31"><br>

        <div class="page-heading">
            <h3 class="page-text">Molimo upišite spol:</h3>
        </div>

        <input list="browsers" name="gender" placeholder="Spol">
        <datalist id="browsers">
            <option value="Male">
            <option value="Female">
        </datalist><br>

        <div class="page-heading">
            <h3 class="page-text">Molimo upišite tjelesnu težinu (kg):</h3>
        </div>

        <input step="any" type="number" name="weight" placeholder="Težina">

        <div class="btn-group" role="group" aria-label="...">
            <label for='step2' class="continue">
                <div>»</div>
            </label>
        </div>
    </div>
</div>
```

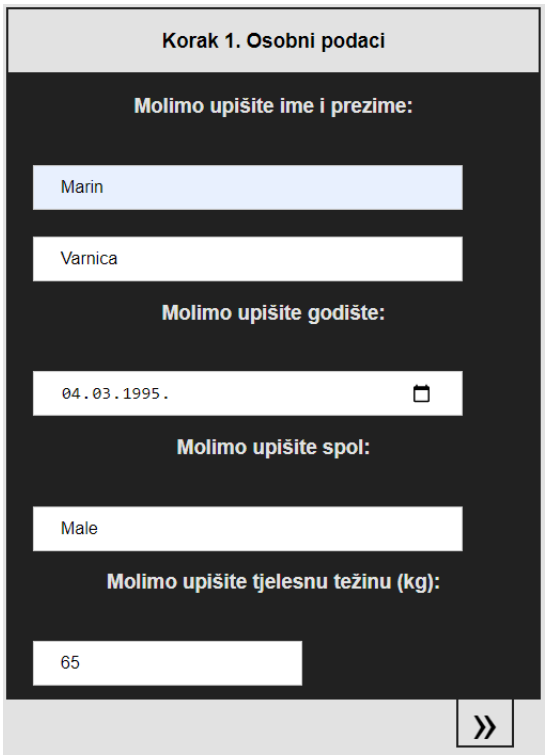
Programski kod 5.22: Dio programskog rješenja za izradu korisničkog računa

Programski kod 5.23 pokazuje primjer programskog rješenja koje označava izgled određenih *HTML* oznaka koje označavaju dijelove sučelja. Selektori *.panel* i *.panel-heading* označavaju klase iz *HTML*-a. Svaki selektor sadrži svojstva i vrijednosti koje označavaju način kako će se pojedini dio web aplikacije prikazati. Primjerice, vrijednost *center* svojstva *align-items* selectora *.page-heading* usmjerava na centriranje poravnanja za sve elemente unutar te *div* stavke. U tom slučaju oznaka *h3* će biti centrirana.

```
.panel {
    background: #212121;
}

.panel .panel-heading {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    justify-content: center;
    align-items: center;
    border: 2px solid #212121;
    background-color: #e2e2e2;
}

.page-heading {
    display: flex;
    justify-content: center;
    align-items: center;
}

.page-heading h3{
    color: #e2e2e2;
}
```

Programski kod 5.23: Dio programskog rješenja za izgled HTML oznaka

Primjer unosa osobnih podataka je prikazan slikom 5.24. Primjerice, iz upisanog datum rođenja će se izračunati životna dob prilikom spremanja u bazu podataka. Na temelju unesenog spola će se izvoditi određeni dijelovi programskog rješenja prilikom usporedbe s referentnim vrijednostima čimbenika rizika obolijevanja od bubrežnih bolesti.



Slika 5.24: Prikaz dijela obrasca izrade korisničkog računa

5.4.2. Prikaz pogrešaka

Ako u nekom slučaju, korisnik unese podatak koji ne odgovara regularnom izrazu prilikom provjere tih podataka, na sučelju će mu se prikazati poruka o toj greški. Programskim kodom 5.25 prikazuje se primjer korištenja *PHP*-a u *HTML*-u. Naredbom *if* prvo se provjerava postoji li pogreška o imenu. Ako postoji, pogreška se prikazuje na sučelju.

```
1.  <?php if(isset($errors["firstname"])):?>
2.  <label class="error-label">
3.    *<?php echo $errors["firstname"];?>
4.  </label><br>
5.  <?php endif;?>
```

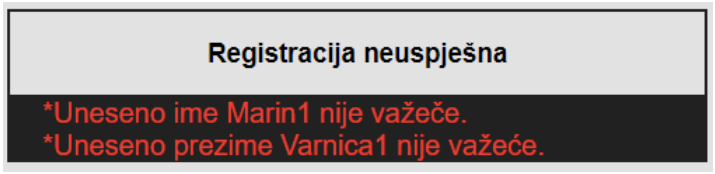
Programski kod 5.25: Primjer programskog rješenja prikaza pogrešaka

Programski kod 5.26 prikazuje izgled *HTML* oznake pogreške. Primjerice, vrijednost *#EA4335* svojstva *color* govori da će boja tekstualnog sadržaja biti crvena.

```
1.  .error-label {
2.      margin-left: 5%;
3.      color: #EA4335;
4.      font-size: 20px;
5.  }
```

Programski kod 5.26: Dio programskog rješenja za izgled HTML oznaka

Ako se upišu podaci koji ne odgovaraju regularnim izrazima, određena se poruka o pogreški proslijedi na sučelje web aplikacije kako je prikazano slikom 5.27. Na taj se način javlja korisniku da se dogodila pogreška prilikom provjere podataka. Također postoje i poruke koje pokazuju da da su se podaci uspješno obradili.



Slika 5.27: Prikaz pogrešaka na sučelju web aplikacije

5.4.3. Tablice rezultata

Tablice se na sučelju prikazuju uz pomoć povezivanja s PHP programskim jezikom. Programski kod 5.28 prikazuje programsko rješenje popunjavanja tablice rezultatima zdravstvenog stanja registriranih korisnika web aplikacije. Svaki red u tablici predstavlja jedno rješenje dobiveno

procjenom rizika obolijevanja od bubrežnih bolesti. Također, boja pojedinog reda ovisi o samom vjerojatnosti.

```
<table>
        <thead>
        <tr>
            <th class="col-name" scope="col">br.</th>
            <th class="col-name" scope="col">Nalaz izdan</th>
            <th class="col-name" scope="col"></th>
        </tr>
        </thead>
        <tbody>
        <?php
            if (isset($row)):
            foreach ($row as $key => $value):
                ?>
                <tr <?php if ($value["ckd"] == 1) {
                    if ($value["ckdprecision"] < 0.3333) {
                        echo 'id="neg"';
                    } else if ($value["ckdprecision"] <= 0.3333 ||
                               $value["ckdprecision"] < 0.6666) {
                        echo 'id="halfsure"';
                    } else {
                        echo 'id="pos"';
                    }
                } else {
                        echo 'id="neg"';
                } ?>>
                    <td data-label="br."><?php echo $key += 1; ?></td>
                    <td data-label="Nalaz izdan">
                       <?php echo $value["created"]; ?></td>
                    <td>
                        <a href="/result/details/<?php echo $value["id"]; ?>/">
                            Pogledaj</a>
                        <a href="/result/delete/<?php echo $value["id"]; ?>">
                            Obriši</a>
                    </td>
                </tr>
            <?php endforeach;
        endif; ?>
        </tbody>
</table>
```

Programski kod 5.28: Primjer programskog rješenja prikaza tablice

Slikom 5.29 prikazuje se izgled popunjene tablice s rezultatima. Također se može vidjeti i prilagodljivi dizajn te tablice jer ovisi o rezoluciji uređaja. Kako je već navedeno, postoje tri prikaza boje koje ovise o vjerojatnosti rizika oboljenja od bubrežne bolesti. Crvenom se bojom označava visoki, žutom srednji, a zelenom niski stupanj rizika obolijevanja. Postoje i dvije dodatne tipke koje omogućuju interakciju s aplikacijom. Mogu prikazati detaljniji sadržaj svakog rezultata tipkom *Pogledaj*, a mogu i obrisati pojedini rezultat iz baze podataka tipkom *Obriši*.

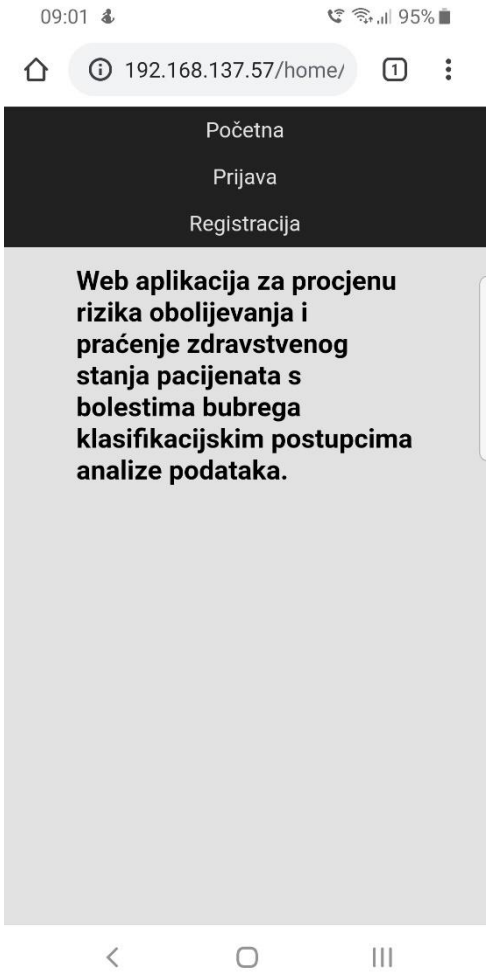
Slika 5.29: Prikaz tablice na računalu (lijevo) i mobilnom uređaju (desno)

6. PRIKAZ RADA I NAČIN KORIŠTENJA WEB APLIKACIJE

Ovim se poglavljem ukratko opisuje način korištenja web aplikacije prilikom korisnikovog korištenja. Zatim se ispituje rad web aplikacije s tri moguća scenarija. Na kraju se analiziraju i vrednuju određeni postupci strojnog učenja koji su korišteni u aplikaciji.

6.1. Način korištenja web aplikacije

Prvim ulaskom u web aplikaciju prikazuje se početna stranica kao na slici 6.1. Tada putem navigacijske trake aplikacija omogućuje interakciju s korisnicima pritiskom određenih tipki. Primjerice, pritiskom tipke *Prijava*, aplikacija će ih pomoću usmjerivača odvesti na upravljačku komponentu koja će im prikazati obrazac za prijavu u aplikaciju.



Slika 6.1: Prikaz početne stranice web aplikacije

Prije same prijave korisnik prvo mora izraditi novi korisnički račun. To radi unosom svojih osobnih podataka i podataka vezanih za korisnički račun. Nakon toga, prijava u web aplikaciju je omogućena pomoću korisničkog imena ili elektroničke pošte i lozinke. Prilikom prijave korisnici mogu i pritisnuti na potvrdni okvir kako bi im se prijava mogla zapamtiti prilikom sljedeće posjete korištenjem kolačića. Zatim registrirani korisnici u obrazac mogu unijeti vrijednosti čimbenika arterijske hipertenzije, hematoloških čimbenika, biokemijskih čimbenika i čimbenika iz urina kako bi im aplikacija mogla procijeniti rizik oboljenja od bubrežne bolesti. Nakon što korisnici potvrde taj obrazac, mogu vidjeti rezultate svog zdravstvenog stanja. Prilikom korištenja web aplikacije korisnici mogu vidjeti detaljni prikaz rezultata zdravstvenog stanja, mogu obrisati svoje rezultate, vidjeti podatke o svom računu te, ako žele, promijeniti svoju lozinku unosom trenutne i nove lozinke.

6.2. Ispitivanje rada web aplikacije

U ispitivanju rada web aplikacije prikazuju se tri vrste scenarija ispitivanja, a to su niski, srednji i visoki rizik oboljenja. U nastavku su postavljeni i prikazani rezultati scenarija.

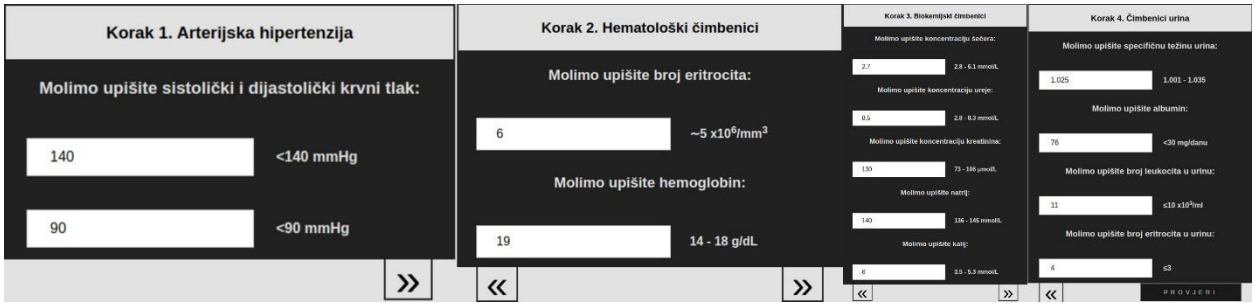
Rizici oboljenja najviše ovise o vrijednostima čimbenika samog pacijenta, odnosno što je veće odstupanje čimbenika od referentne vrijednosti, rizik oboljenja od bubrežnih bolesti je veći. Stoga je za ovo testiranje potrebno za svaki scenarij unijeti određene vrijednosti kako bi se prikazao određeni stupanj rizika.

Primjerice za niski stupanj rizika, vrijednosti je potrebno postaviti na optimalne vrijednosti. Kako arterijska hipertenzija može biti pokazatelj povišenog krvnog tlaka koji se može naglo povećati pojavom izvanstanične tekućine u organizmu, sistoličku i dijastoličku vrijednost potrebno je postaviti na normalnu vrijednost. Hematološki čimbenici, eritrociti i hemoglobin, mogu utjecati na pojavu anemije, stoga se za eritrocite unosi $5x10^6/mm^3$, a za hemoglobin 15.6 g/dL, što je normalna vrijednost kod muškaraca. Kod koncentracije glukoze u krvi potrebno je paziti da se ne unese smanjena vrijednost jer je takva vrijednost pokazatelj hipoglikemije, dok je povećana pokazatelj hiperglikemije. Kod unosa vrijednosti ureje i serumskog kreatinina treba biti oprezan jer je njihov omjer jedan od načina razlikovanja akutnog od kroničnog zatajenja bubrega. Povećana, odnosno smanjena vrijednost biokemijskih čimbenika rizika natrija i kalija može biti pokazatelj hiponatrijemije, hipokalijemije, hipernatrijemije i hiperkalijemije. Kod unosa čimbenika iz urina, specifična težina urina je najveći pokazatelj hidracije. Također njihova

vrijednost može biti pokazatelj razrijeđenosti i koncentriranosti mokraće. Povećanom vrijednosti albumina, može se pojaviti mikroalbuminurija ili pak dijabetička nefropatija. Uz samu vrijednost albumina, omjer ureje i serumskog kreatinina može biti pokazatelj ove dvije bolesti. Za broj eritrocita u urinu, smatra se da je do 3 eritrocita normalan broj, dok je za leukocite 10 po mikrolitri.

Također postoje i dodatne pretrage kroničnog i akutnog zatajenja bubrega na temelju unesenih vrijednosti čimbenika rizika. Kod kroničnog bubrežnog zatajenja, koristi se formula modifikacije prehrane kod bubrežnih bolesti pomoću serumskog kreatinina i životne dobi pacijenta Pomoću te jednadžbe se mjeri brzine glomerularne filtracije te se može otkriti stadij zatajenja. Osim takvog mjerenja stupnja zatajenja, još se izračunava klirens kreatinina pomoću Cockcroft-Gaultove jednadžbe. Na samom kraju, akutnim se zatajenjem može razotkriti prerenalna, renalna i postrenalna azotemija izračunavanjem omjera ureje i serumskog kreatinina.

Na sličan se način postavljaju niski i visoki stupanj rizika. Kod srednjeg, vrijednosti su nešto veće kako bi ukazali na neke bolesti, a kod visokog stupnja, vrijednosti su puno veće od referentnih vrijednosti. Slika 6.2 predstavlja primjer unosa parametara u četiri koraka. Svaki korak predstavlja unos jedne skupine čimbenika rizika visokog scenarija oboljenja.



Slika 6.2: Prikaz koraka unosa čimbenika rizika

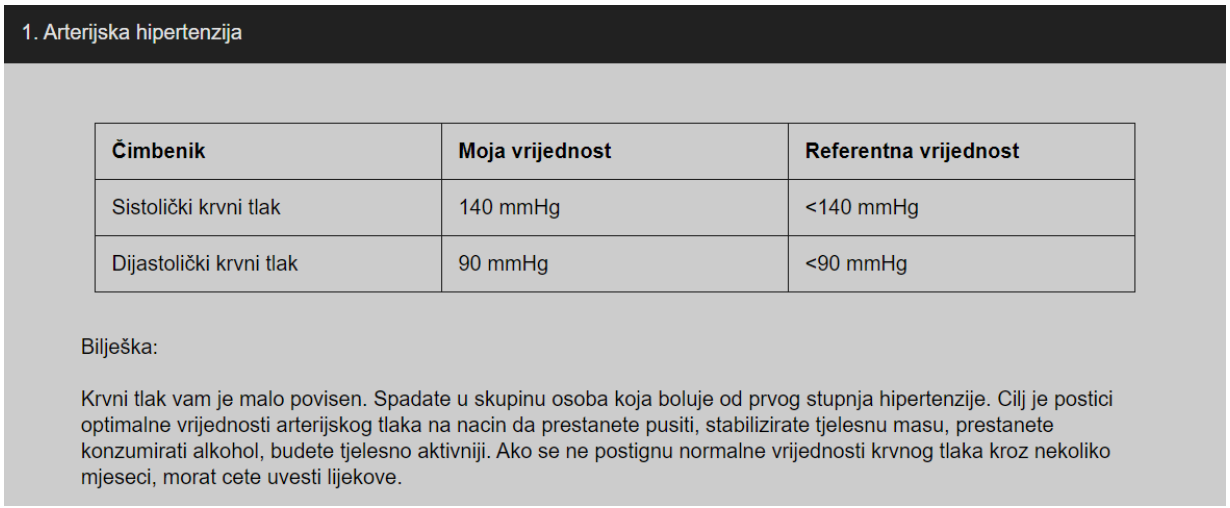
Nakon što se vrijednosti parametara unesu u obrazac, podaci se proslijede web usluzi koji na temelju određenog postupka strojnog učenja procjenjuje rizik oboljenja pacijenta od bubrežne bolesti. Podaci se spreme u bazu podataka te korisnici mogu vidjeti svoje rezultate u bilo kojem trenutku.

Svaki pacijent, odnosno korisnik web aplikacije, može vidjeti detaljan prikaz rezultata. Takav mu prikaz nudi sve informacije vezane za čimbenike rizika podijeljene u pet skupina. Svaka skupina

sadrži tablicu s nazivom čimbenika, vrijednosti koju je unio u obrazac i referentnu vrijednost tog čimbenika rizika. Također sadrži i bilješku u kojoj pišu pojedinosti njegovog zdravstvenog stanja. Na samome kraju, korisnici mogu pročitati postoji li ukazivanje bubrežne bolesti i procijenjenu vrijednost rizika oboljenja. U nastavku su prikazani i objašnjeni rezultati pojedine skupine čimbenika visokog scenarija oboljenja.

6.2.1. Rezultati arterijske hipertenzije

Slikom 6.3 prikazuju se rezultati čimbenika arterijske hipertenzije, odnosno sistoličkog i dijastoličkog krvnog tlaka. Kako se može primijetiti, pacijent ima prvi stupanj hipertenzije (povišeni krvni tlak) zbog povišenih vrijednosti tlakova. U bilješkama svaki pacijent može pročitati njegovo zdravstveno stanje te kako ga poboljšati.


1. Arterijska hipertenzija

Čimbenik	Moja vrijednost	Referentna vrijednost
Sistolički krvni tlak	140 mmHg	<140 mmHg
Dijastolički krvni tlak	90 mmHg	<90 mmHg

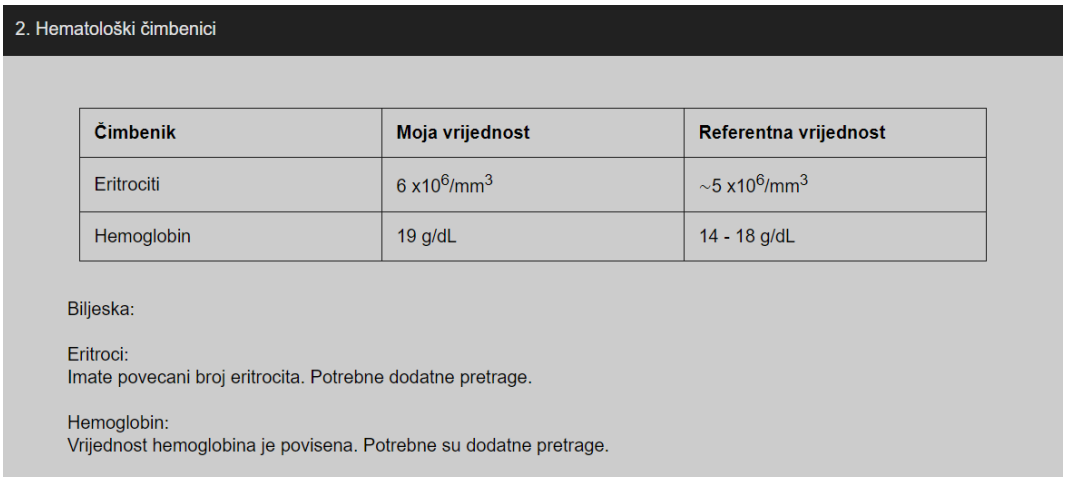
Bilješka:

Krvni tlak vam je malo povisen. Spadate u skupinu osoba koja boluje od prvog stupnja hipertenzije. Cilj je postici optimalne vrijednosti arterijskog tlaka na nacin da prestanete pusiti, stabilizirate tjelesnu masu, prestanete konzumirati alkohol, budete tjelesno aktivniji. Ako se ne postignu normalne vrijednosti krvnog tlaka kroz nekoliko mjeseci, morat cete uvesti lijekove.

Slika 6.3: Prikaz rezultata arterijske hipertenzije

6.2.2. Rezultati hematoloških čimbenika

Vrijednosti hematoloških čimbenika prikazani slikom 6.4 su povišeni. Bilješkama informiraju pacijenta kako su potrebne dodatne pretrage kompletne krvni slike.



2. Hematološki čimbenici

Čimbenik	Moja vrijednost	Referentna vrijednost
Eritrociti	6 x10^6/mm^3	~5 x10^6/mm^3
Hemoglobin	19 g/dL	14 - 18 g/dL

Biljeska:

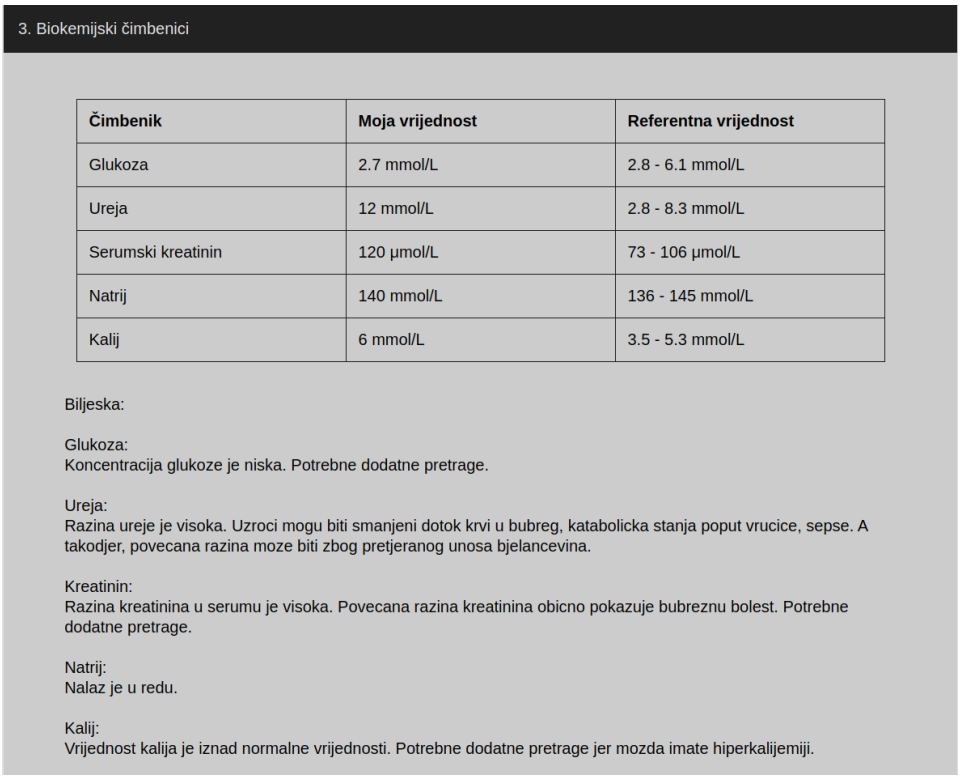
Eritroci:
Imate povecani broj eritrocita. Potrebne dodatne pretrage.

Hemoglobin:
Vrijednost hemoglobina je povisena. Potrebne su dodatne pretrage.

Slika 6.4: Prikaz rezultata hematoloških čimbenika

6.2.3. Rezultati biokemijskih čimbenika

Slika 6.5 prikazuje vrijednosti biokemijskih čimbenika na temelju kojih korisnici web aplikacije mogu vidjeti rezultate njihovog zdravstvenog stanja. Budući da mu je koncentracija glukoze u krvi relativno niska, mora obaviti dodatne pretrage jer možda ima hipoglikemiju. Razine ureje je iznad referentnih vrijednosti što može pokazuje da će omjer ureje i serumskog kreatinina biti povećan jer je on jedan od glavnih pokazatelja bubrežnih bolesti. Koncentracija kalija je iznad referentne vrijednosti što može biti pokazatelj hiperglikemije.



3. Biokemijski čimbenici

Čimbenik	Moja vrijednost	Referentna vrijednost
Glukoza	2.7 mmol/L	2.8 - 6.1 mmol/L
Ureja	12 mmol/L	2.8 - 8.3 mmol/L
Serumski kreatinin	120 µmol/L	73 - 106 µmol/L
Natrij	140 mmol/L	136 - 145 mmol/L
Kalij	6 mmol/L	3.5 - 5.3 mmol/L

Biljeska:

Glukoza:
Koncentracija glukoze je niska. Potrebne dodatne pretrage.

Ureja:
Razina ureje je visoka. Uzroci mogu biti smanjeni dotok krvi u bubreg, katabolicka stanja poput vrucice, sepse. A takodjer, povecana razina moze biti zbog pretjeranog unosa bjelancevina.

Kreatinin:
Razina kreatinina u serumu je visoka. Povecana razina kreatinina obicno pokazuje bubreznu bolest. Potrebne dodatne pretrage.

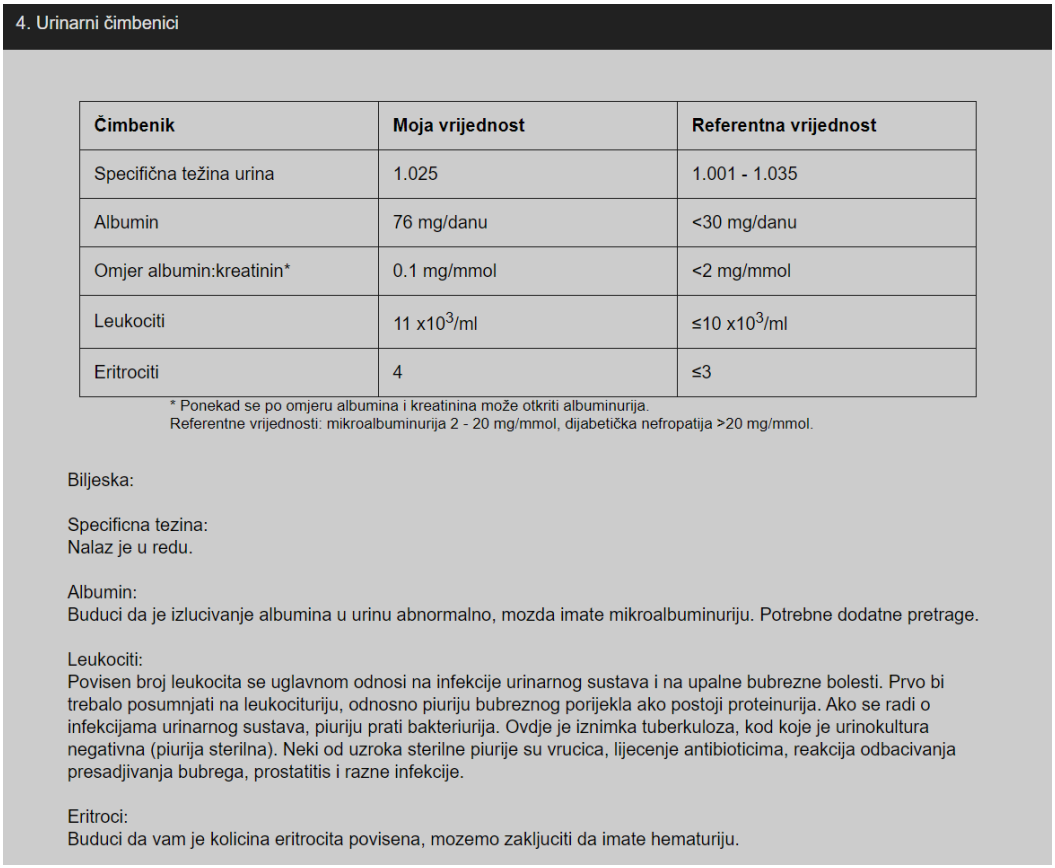
Natrij:
Nalaz je u redu.

Kalij:
Vrijednost kalija je iznad normalne vrijednosti. Potrebne dodatne pretrage jer mozda imate hiperkalijemiji.

Slika 6.5: Prikaz rezultata biokemijskih čimbenika

6.2.4. Rezultati čimbenika iz urina

Slikom 6.6 se mogu vidjeti rezultati dobiveni unosom urinarnih čimbenika u obrazac. Dobivena vrijednost albumina je preko 30 što označava pojavu mikroalbuminurije. U ovom slučaju, omjer ureje i kreatinina može biti pokazatelj bolesti albumina, no budući da se taj omjer mjeri u mg/mmol, vrijednosti su u intervalu referentne vrijednosti. Broj bijelih krvnih stanica u urinu su preko 10, što označava infekcije i upalne bubrežne bolesti, a povećani broj crvenih krvnih stanica označava hematuriju, odnosno krv u urinu.



4. Urinarni čimbenici

Čimbenik	Moja vrijednost	Referentna vrijednost
Specifična težina urina	1.025	1.001 - 1.035
Albumin	76 mg/danu	<30 mg/danu
Omjer albumin:kreatinin*	0.1 mg/mmol	<2 mg/mmol
Leukociti	11 x10^3/ml	≤10 x10^3/ml
Eritrociti	4	≤3

* Ponekad se po omjeru albumina i kreatinina može otkriti albuminurija.
Referentne vrijednosti: mikroalbuminurija 2 - 20 mg/mmol, dijabetička nefropatija >20 mg/mmol.

Biljeska:

Specificna tezina:
Nalaz je u redu.

Albumin:
Buduci da je izlucivanje albumina u urinu abnormalno, mozda imate mikroalbuminuriju. Potrebne dodatne pretrage.

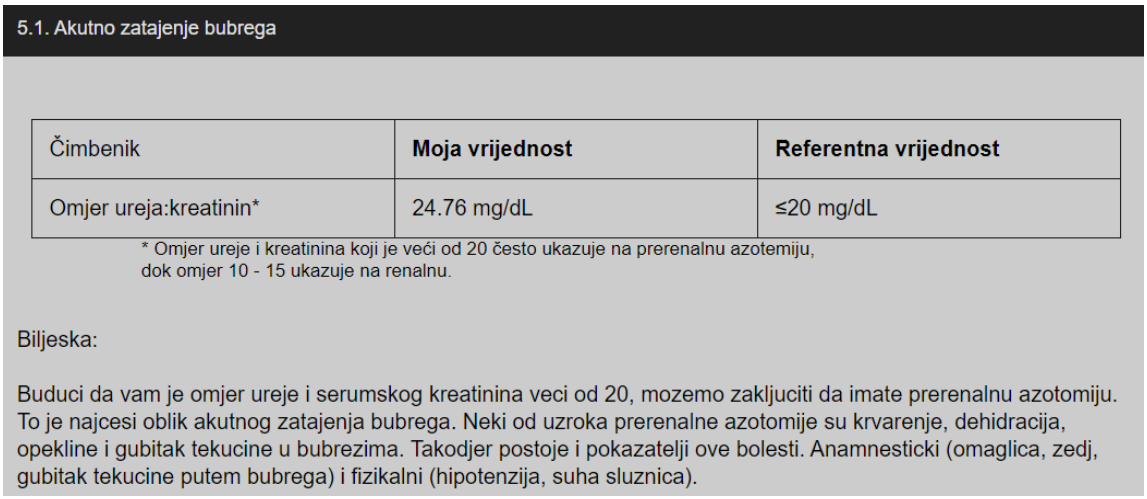
Leukociti:
Povisen broj leukocita se uglavnom odnosi na infekcije urinarnog sustava i na upalne bubrezne bolesti. Prvo bi trebalo posumnjati na leukocituriju, odnosno piuriju bubreznog porijekla ako postoji proteinurija. Ako se radi o infekcijama urinarnog sustava, piuriju prati bakteriurija. Ovdje je iznimka tuberkuloza, kod koje je urinokultura negativna (piurija sterilna). Neki od uzroka sterilne piurije su vrucica, lijecenje antibioticima, reakcija odbacivanja presadjivanja bubrega, prostatitis i razne infekcije.

Eritroci:
Buduci da vam je kolicina eritrocita povisena, mozemo zakljuciti da imate hematuriju.

Slika 6.6: Prikaz rezultata čimbenika iz urina

6.2.5. Rezultati akutnog zatajenja bubrega

Slika 6.7 pokazuje dodatni test web aplikacije što se tiče akutnog bubrežnog zatajenja. Rezultati se odnose na omjer ureje i serumskog kreatinina. U ovom se slučaju vrijednosti mjere u mg/dL iz razloga što se zapravo mjeri odnos s dušikovih spojeva u krvi (engl. *blood urea nitrogen*) jer je on izražen u mg/dL. Vrijednost tog omjera je veća od referentne vrijednosti što može značiti da pacijent boluje od prerenalne azotemije. Taj oblik azotemije je najčešći oblik akutnog kroničnog zatajenja bubrega.



5.1. Akutno zatajenje bubrega

Čimbenik	**Moja vrijednost**	**Referentna vrijednost**
Omjer ureja:kreatinin*	24.76 mg/dL	≤20 mg/dL

* Omjer ureje i kreatinina koji je veći od 20 često ukazuje na prerenalnu azotemiju, dok omjer 10 - 15 ukazuje na renalnu.

Biljeska:

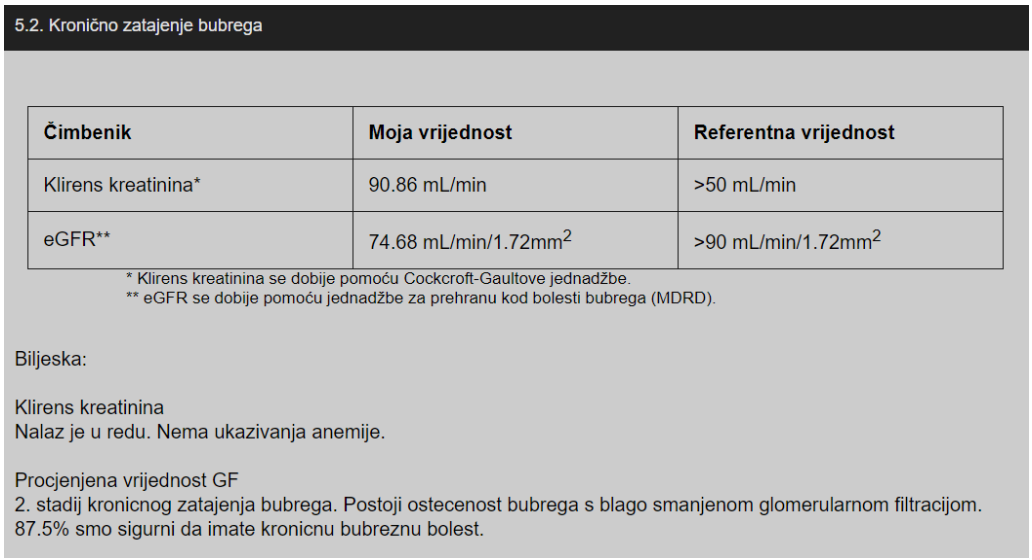
Buduci da vam je omjer ureje i serumskog kreatinina veci od 20, mozemo zakljuciti da imate prerenalnu azotomiju. To je najcesi oblik akutnog zatajenja bubrega. Neki od uzroka prerenalne azotomije su krvarenje, dehidracija, opekline i gubitak tekucine u bubrezima. Takodjer postoje i pokazatelji ove bolesti. Anamnesticki (omaglica, zedj, gubitak tekucine putem bubrega) i fizikalni (hipotenzija, suha sluznica).

Slika 6.7: Prikaz rezultata akutnog zatajenja bubrega

Dva česta čimbenika rizika od obolijevanja od kroničnog bubrežnog zatajenja su klirens kreatinina i procjenu brzine glomerularne filtracije. Dobivanje ovih vrijednosti ovise o životnoj dobi, spolu i tjelesnoj masi mjerenoj u kilogramima koje pacijenti unose prilikom izrade novog korisničkog računa. Osim ta dva parametra, ovise i o serumskom kreatininu. Budući da je vrijednost procjene brzine glomerularne filtracije između 60 i 80 mL/min/1.72mm^2, radi se o drugom stadiju kroničnog zatajenja.

6.2.6. Rezultati kroničnog zatajenja bubrega

Na samome kraju, obradom i analizom unesenih vrijednosti čimbenika rizika, postoji sigurnost od 87.5% da se radi o kroničnom bubrežnom zatajenju.



5.2. Kronično zatajenje bubrega

Čimbenik	**Moja vrijednost**	**Referentna vrijednost**
Klirens kreatinina*	90.86 mL/min	>50 mL/min
eGFR**	74.68 mL/min/1.72mm^2	>90 mL/min/1.72mm^2

* Klirens kreatinina se dobije pomoću Cockcroft-Gaultove jednadžbe.
** eGFR se dobije pomoću jednadžbe za prehranu kod bolesti bubrega (MDRD).

Biljeska:

Klirens kreatinina
Nalaz je u redu. Nema ukazivanja anemije.

Procjenjena vrijednost GF
2. stadij kronicnog zatajenja bubrega. Postoji ostecenost bubrega s blago smanjenom glomerularnom filtracijom. 87.5% smo sigurni da imate kronicnu bubreznu bolest.

Slika 6.8: Prikaz rezultata kroničnog zatajenja bubrega

6.3. Analiza korištenih postupaka strojnog učenja u aplikaciji

Kako je već spomenuto, postoje dva pristupa strojnom učenju, a to su nadzirano i nenadzirano učenje. Za izradu web aplikacija pomoću kojih se procjenjuju bilo kakvi rizici oboljenja, kao bolesti bubrega, najlakše je realizirati tehnikama klasifikacije nadzirnog učenja jer se koriste postojeće vrijednosti čimbenika rizika različitih pacijenata.

6.3.1. Vrednovanje modela

Vrednovanje modela najbolji je pokazatelj najboljeg rješenja jer se na taj način može vidjeti točnost procjenjivanja modela korištenjem određenih postupaka strojnog učenja, odnosno algoritama. Ne mora značiti da će model uvijek davati točne rezultate. Primjerice, mijenjanjem hiperparametara može znatno utjecati na učinkovitost. Jedan način vrednovanja klasifikacijskog modela je pomoću matrice zabune (engl. *confusion matrix*). Matrica uspoređuje klasificirane i stvarne podatke.

Tablica 6.9 predstavlja matricu zabune teorijskog slučaja gdje postoje četiri kategorije: *TP*, *TN*, *FN* i *FP*. *TP* označava broj pacijenata koji imaju bubrežnu bolest i algoritam ih je točno identificirao (engl. *true positive*). *TN* predstavlja broj pacijenata koji nemaju bubrežnu bolest i model ih je točno identificirao (engl. *true negative*). *FN* se odnosi na broj pacijenata koji imaju bubrežnu bolest, ali ih je algoritam identificirao da nemaju (engl. *false negative*). *FP* prikazuje broj pacijenta koji nemaju bubrežnu bolest, a algoritam ih identificira da imaju (engl. *false positive*). Pomoću matrice zabune možemo izračunati mjere poput točnosti, preciznosti, odziva i F1-mjere.

Tablica 6.9: Primjer teorijskog slučaja matrice zabune

		Stvarni podaci	
		Ima bubrežnu bolest	Nema bubrežnu bolest
Klasificirani podaci	Ima bubrežnu bolest	TP	FP
	Nema bubrežnu bolest	FN	TN

Točnost (engl. *Accuracy)* se prikazuje jednadžbom (6-10), a predstavlja omjer ukupnog broja ispravnih primjera i ukupnog broja primjera [29]. Pomoću točnosti može se ustanoviti koliko je dobro model učinkovit prilikom klasificiranja primjera iz skupa podataka.

$$Točnost = \frac{TP + TN}{TP + TN + FP + FN} \tag{6-10}$$

Jednadžbom (6-11) se prikazuje preciznost (engl. *precision*). Nju definiramo omjerom broja primjera koji su identificirani kao pozitivni i ukupnog broja pozitivnih (klasificiranih) primjera [29]. Preciznost zapravo prikazuje udio točnih primjera između onih koji su identificirani kao pozitivni.

$$Preciznost = \frac{TP}{TP + FP} \tag{6-11}$$

Odzivom (engl. *Recall*) pomoću jednadžbe (6-12) moguće je dobiti omjer broja koji su identificirani kao pozitivni i ukupnog broja pozitivnih (stvarnih) primjera [29].

$$Odziv = \frac{TP}{TP + FN} \tag{6-12}$$

F1-mjera (engl. *F1-score*) jednadžbom (6-13) prikazuje harmonijsku sredinu između odziva i preciznosti [29].

$$F1 - mjera = \frac{2}{\frac{1}{Preciznost} + \frac{1}{Odziv}} \tag{6-13}$$

6.3.2. Analiza logističke regresije

Slika 6.14 prikazuje vrednovanje modela logističke regresije dobivenog pomoću *Microsoft Machine Learning Studio* razvojnog okruženja.

TP = 75	FP = 0
FN = 5	TN = 40

$$Točnost = \frac{TP + TN}{TP + TN + FP + FN} = \frac{75 + 40}{75 + 40 + 0 + 5} = \frac{115}{120} = 0.9583$$

$$Preciznost = \frac{TP}{TP + FP} = \frac{75}{75 + 0} = \frac{75}{75} = 1$$

$$Odziv = \frac{TP}{TP + FP} = \frac{75}{75 + 5} = \frac{75}{80} = 0.9375$$

$$F1 - mjera = \frac{2}{\frac{1}{Preciznost} + \frac{1}{Odziv}} = \frac{2}{\frac{1}{1} + \frac{1}{0.9375}} = \frac{2}{1 + \frac{80}{75}} = \frac{2}{\frac{31}{15}} = \frac{30}{31} = 0.9677$$

Slika 6.14: Vrednovanje modela logističke regresije

6.3.3. Analiza neuronske mreže

Slika 6.15 prikazuje vrednovanje modela neuronske mreže dobivenog pomoću *Microsoft Machine Learning Studio* razvojnog okruženja.

TP = 76	FP = 0
FN = 4	TN = 40

$$Točnost = \frac{TP + TN}{TP + TN + FP + FN} = \frac{76 + 40}{77 + 40 + 0 + 4} = \frac{116}{120} = 0.967$$

$$Preciznost = \frac{TP}{TP + FP} = \frac{76}{76 + 0} = \frac{76}{76} = 1$$

$$Odziv = \frac{TP}{TP + FP} = \frac{76}{76 + 4} = \frac{76}{80} = 0.95$$

$$F1 - mjera = \frac{2}{\frac{1}{Preciznost} + \frac{1}{Odziv}} = \frac{2}{\frac{1}{1} + \frac{1}{0.95}} = \frac{2}{1 + \frac{80}{76}} = \frac{2}{\frac{39}{19}} = \frac{38}{39} = 0.9744$$

Slika 6.15: Vrednovanje modela neuronske mreže

6.3.4. Analiza slučajne šume

Slika 6.16 prikazuje vrednovanje modela slučajne šume dobivenog pomoću *Microsoft Machine Learning Studio* razvojnog okruženja.

TP = 77	FP = 0
FN = 3	TN = 40

$$Točnost = \frac{TP + TN}{TP + TN + FP + FN} = \frac{77 + 40}{77 + 40 + 0 + 3} = \frac{117}{120} = 0.975$$

$$Preciznost = \frac{TP}{TP + FP} = \frac{77}{77 + 0} = \frac{77}{77} = 1$$

$$Odziv = \frac{TP}{TP + FP} = \frac{77}{77 + 3} = \frac{77}{80} = 0.9625$$

$$F1 - mjera = \frac{2}{\frac{1}{Preciznost} + \frac{1}{Odziv}} = \frac{2}{\frac{1}{1} + \frac{1}{0.9625}} = \frac{2}{1 + \frac{80}{77}} = \frac{2}{\frac{157}{77}} = \frac{154}{157} = 0.981$$

Slika 6.16: Vrednovanje modela slučajne šume

6.3.5. Usporedba korištenih postupaka strojnog učenja

Kao što je već navedeno, slike 6.14, 6.15 i 6.16 prethodnih potkategorija prikazuju vrednovanje modela korištenih postupaka strojnog učenja dobivenih pomoću *Microsoft Machine Learning Studio* razvojnog okruženja. Postupci su bili omogućeni pomoću korištenog skupa podataka čimbenika rizika obolijevanja od kroničnog bubrežnog zatajenja.

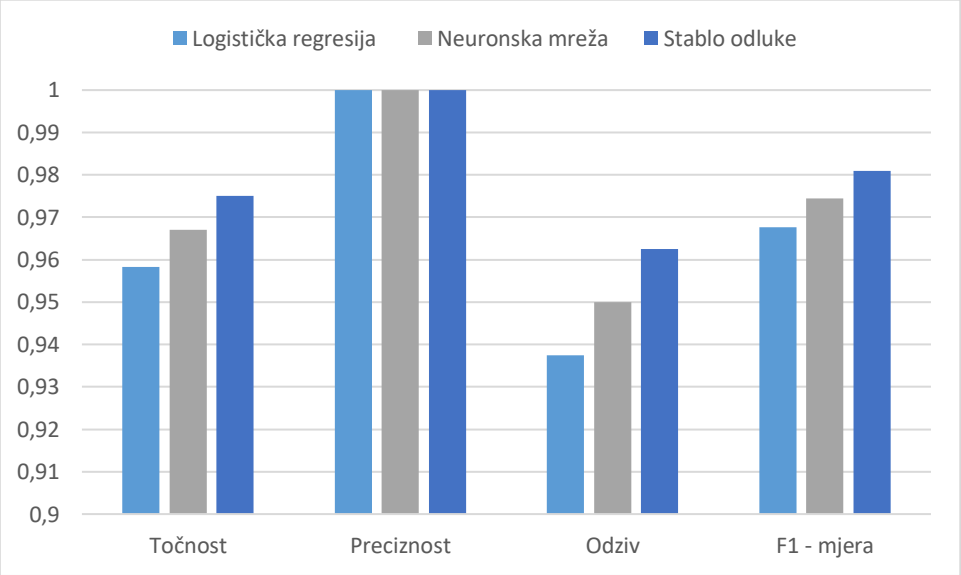
U korištenoj bazi podataka postoji 14 stupaca i 400 redova. Jedan stupac je bio popunjen s kategoričnim vrijednostima pomoću kojih je model mogao procijeniti rizike oboljenja pacijenta od bubrežne bolesti. Podaci iz te baze podataka su podijeljeni na skup za treniranje i testiranje omjerom 70:30. Drugim riječima, od ukupno 400 podataka, odnosno primjera iz korištenog skupa, 280 primjera je pomoglo prilikom treniranja modela dok je 120 primjera poslužilo modelu za procjenjivanje. Dobivenim matricama zabuna razvojnog okruženja, pomoću kategorija vrednovanja mogu se izračunati mjere točnosti, preciznosti, odziva i F1-mjere.

Tablica 6.17 prikazuje usporedbu korištenih postupaka strojnog učenja. Može se primijetiti da su preciznosti sva tri postupka jednaka. Najtočniji model je model slučajne šume s točnošću od 97.5% te je zbog toga taj postupak bio korišten u realizaciji web aplikacije za procjenu bubrežne bolesti.

Tablica 6.17: Usporedba korištenih postupaka

	Logistička regresija	*Neuronska mreža*	*Stablo odluke*
Točnost	0.9583	0.967	0.975
Preciznost	1	1	1
Odziv	0.9375	0.95	0.9625
F1 - mjera	0.9877	0.9744	0.981

Slika 6.18 prikazuje vizualizaciju korištenih postupaka strojnog učenja. Primjerice, najmanju točnost je davao algoritam logističke regresije, a iznosi 95.83%. Također, priloženom vizualizacijom se može primijetiti da je točnost neuronske bolja od točnosti logističke regresije za 0.87%.



Slika 6.18: Vizualni prikaz usporedbe korištenih postupaka

7. ZAKLJUČAK

Diplomskim radom teorijski je obrađena grana medicine koja proučava funkciju i bolest bubrega, odnosno nefrologija. Zatim je ostvarena web aplikacija za procjenu obolijevanja od bubrežnih bolesti pomoću čimbenika rizika. Sama procjena obolijevanja analizirala se klasifikacijskim postupcima modela slučajne šume koji se sastoji od više stabala odluke. Osim postupaka slučajne šume, za klasifikaciju su se koristili i postupci logističke regresije i neuronske mreže. Vrednovanjem se utvrdilo da je postupak analize podataka neuronske mreže imao bolju točnost od logističke regresije za 0.87%.

Osim same svrhe izrađene web aplikacije, implementirane su i dodatne funkcionalnosti, poput izrade novog korisničkog računa i prijave korisnika. Na taj su način korisnici anonimni i zbog toga mogu sačuvati svoje podatke od javnosti. Baza podataka je temelj rada ove aplikacije jer se obavljanje većine funkcionalnosti obavlja povezivanjem i manipuliranjem bazom podataka. Budući da se radi o manjoj web aplikaciji, baza podataka je razmjerno malog obujma dok bi kod većine većih web aplikacije i općenito internetskih stranica bila korištena veća baza podatka. U tom slučaju, bilo bi potrebno provoditi dodatne nefunkcionalne zahtjeve. Aplikacije je jednostavnog izgleda kako bi bila lakša za korištenje starijim osobama te je također i prilagodljivog dizajna, odnosno prikaz sadržaja na sučelju ovisi o rezoluciji uređaja koji pacijent koristi.

Ispitivanjem rada web aplikaciju prikazane su tri vrste ispitivanja niskog, srednjeg i visokog rizika oboljenja od bubrežne bolesti. Takvi rizici najviše ovise o odstupanju od referentnih vrijednosti čimbenika, stoga se takvom procjenom može utvrditi zdravstveno stanje pacijenta. Rezultatima ispitivanja rada web aplikacije može se uočiti da su klirens kreatinina i procjena brzine glomerulane filtracije jedni od glavnih pokazatelja obolijevanja od kroničnog bubrežnog zatajenja, dok se omjerom ureje i kreatinina može prikazati zdravstveno stanje što se tiče akutnog zatajenja bubrega.

U sklopu dorade web aplikacije mogu se dodati različite funkcije koje može obavljati korisnik aplikacije poput brisanja korisničkog računa gdje mu se obrišu svi rezultati. Mogu se dodati i vizualni prikazi rezultata procjene rizika. Također, mogli su se i koristiti drugi klasifikacijski postupci i metode strojnog učenja kako bi procjena rizika bilo učinkovitija.

LITERATURA

[1] Medicinska enciklopedija 1, Jugoslavenski leksikografski zavod, Zagreb 1967.

[2] E. Noble, D. W. Johnson, Automatizirani laboratorijski nalazi određivanja brzine glomerularne filtracije: jesu li dobri za zdravlje bolesnika i njihove liječnike?, Biochemia Medicina, br./1, sv./17, str. 16 - 28, 2007., dostupno na: https://hrcak.srce.hr/12857 [19.05.2020.]

[3] S. P. Desai, Ž. Ivančević, Laboratorijske pretrage u kliničkoj medicini, PLACEBO d.o.o., Split 2006.

[4] M. Pavletić Peršić, S. Vuksanović-Mikuličić, S. Rački, Arterijska hipertenzija, Medicina Fluminensis, br./4, sv./46, str. 376 - 389, Rijeka, 2010., dostupno na: https://hrcak.srce.hr/63061 [12.05.2020.]

[5] A. C. Guyton, Fiziologija čovjeka i mehanizmi bolesti, Medicinska naklada, Zagreb 1995.

[6] D. Čvoriščec, I. Čepelak, Štrausova Medicinska biokemija, Medicinska naklada, Zagreb 2009.

[7] S. Fraser, T. Blakeman, Chronic kidney disease: identification and management in primary care, AMIA Jt Summits Transl Sci Proc, Pragmat Obs Res, br./2016, sv./7, str. 21 - 32, kolovoz 2018., dostupno na: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5087766/ [05.07.2020.]

[8] J. Turner, C. Bauer, M. Abramowitz, M. Melamed, T. Hostetter, Treatment of chronic kidney disease, International Society of Nephrology, br./81, sv./4, str. 351 – 362, prosinac 2011., dostupno na: https://www.researchgate.net/publication/51876006_Treatment_of_chronic_kidney_disease [05.07.2020.]

[9] C. Diamantidis, S. Becker, Health information technology (IT) to improve the care of patients with chronic kidney disease (CKD), BMC Nephrology, br./7, sv./15, 2014., dostupno na: https://bmcnephrol.biomedcentral.com/articles/10.1186/1471-2369-15-7 [05.07.2020.]

[10] R.-S. Kleiman, E. R. LaRose, J. C. Badger, D. Page, M. D. Caldwell, J. A. Clay, P. L. Peissig, Using Machine Learning Algorithms to Predict Risk for Development of Calciphylaxis in Patients with Chronic Kidney Disease, AMIA Jt Summits Transl Sci Proc, br./2018, sv./2018, str. 139 – 146, svibanj 2018.,

dostupno na: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5961821/ [21.05.2020.]

[11] P. Sinha, P. Sinha, Comparative Study of Chronic Kidney Disease Prediction using KNN and SVM, International Journal of Engineering Research & Technology, br./12, sv./4, prosinac 2015., dostupno na: https://www.semanticscholar.org/paper/Comparative-Study-of-Chronic-Kidney-Disease-using-Sinha-Sinha/f850c084395d7fb1f75eb21d0ab8c2f1df6bc938 [21. 05.2020.]

[12] L.-J. Rubini, P. Eswaran, Generating comperative analysis of early stage prediction of Chronic Kidney Disease, International Journal of Modern Engineering Research, br./7, vol./5, srpanj 2015., dostupno na: https://issuu.com/ijmer/docs/g5734955 [21.05.2020.]

[13] K.-R. Lakshmi, Y. Nagesh, M.-V. Krisha, Performance Comparison of Three Dana Mining Techniques for Predicting Kidney Dialysis Survivability, International Journal of Engineering Research & Technology, br./1, vol./7, ožujak 2014., dostupno na: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=26E33F771F4135FBD2451A2768D0492D?doi=10.1.1.668.9133&rep=rep1&type=pdf [11.06.2020.]

[14] S. Vijayarani, S. Dhayanand, KIDNEY DISEASE PREDICTION USING SVM AND ANN ALGORITHMS, International Journal of Computing and Bussiness Research, br./2, vol./6, ožujak 2015., dostupno na: https://www.researchgate.net/publication/339551723_KIDNEY_DISEASE_PREDICTION_USING_SVM_AND_ANN_ALGORITHMS [11.06.2020.]

[15] Mobilna aplikacija, Renal Dose Adjustment & Creatinine Clearance, dostupno na: https://play.google.com/store/apps/details?id=info.ekamus.renal

[16] Mobilna aplikacija, Renal Diet Advisor (Kidney diet), dostupno na: https://play.google.com/store/apps/details?id=com.mahapps.renaldietadviser

[17] Mobilna aplikacija, eGFR Renal Function Calculator, dostupno na: https://play.google.com/store/apps/details?id=net.oxdb.KidneyFunc&hl=en_US

[18] Laminas Project, Password, Zend Framework, 2020., dostupno na: https://docs.laminas.dev/laminas-crypt/password/ [27.05.2020.]

[19] Mathworks, Introducing Machine Learning, The MathWorks, 2016., dostupno na: https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/i/88174_92991v00_machine_learning_section1_ebook.pdf [03.06.2

020.]

[20] ML Glossary, Machine Learning Glossary, Revision, 2017, dostupno na: https://ml-cheatsheet.readthedocs.io/en/latest/index.html# [07.06.2020.]

[21] ScienceDirect, Logistic Regression Analysis, Elsevier B.V., 2020., dostupno na: https://www.sciencedirect.com/topics/medicine-and-dentistry/logistic-regression-analysis [05.07.2020.]

[22] ScienceDirect, Neural Network, Elsevier B.V., 2020., dostupno na: https://www.sciencedirect.com/topics/social-sciences/neural-network [05.07.2020.]

[23] H. Sharma, S. Kumar, A Survey on Decision Tree Algorithm of Classification in Data Mining, International Journal of Engineering Research & Technology, br./5, sv./4, travanj 2016., dostupno na: https://www.researchgate.net/publication/324941161_A_Survey_on_Decision_Tree_Algorithms_of_Classification_in_Data_Mining [28. 05.2020.]

[24] C. Kingford, S.-L. Salzberg, What are decision trees?, Nat Biotechnol, br./26, sv./9, 2008., dostupno na: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2701298/ [05.07.2020.]

[25] L. Breiman, Random Forests, Machine Learning, br./2001, sv./45, str. 5 – 32, 2001., dostupno na: https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf [07.07.2020.]

[26] Korištena baza podataka, dostupno na: https://www.kaggle.com/abhia1999/chronic-kidney-disease

[27] Postman, dostupno na: https://www.postman.com/

[28] E. RoseBrock, E. Filson, Setting Up LAMP: Getting Linux, Apache, MySQL and PHP Working Together, Sybex, San Franscisco, 2004.

[29] T. Đuričić, A. Merćep, Strojno učenje – bilješke s predavanja, Fakultet elektrotehnike i računarstva, Zagreb 2015.

SAŽETAK

Glavni zadatak ovog diplomskog rada bio je izraditi web aplikaciju za procjenu rizika obolijevanja od bubrežnih bolesti klasifikacijskim postupcima analize podataka. Web aplikacija ostvarena je pomoću programskih jezika i pristupa kao što su *PHP, HTML* i *CSS*. Postupci su u web aplikaciju implementirani pomoću *PHP* biblioteke *cURL* gdje se ulazni podaci putem razvojnog okruženja *Postman* proslijede web usluzi koja kao rezultat vraća procjenu rizika obolijevanja. Za klasifikaciju su korišteni postupci logističke regresije, neuronske mreže i slučajne šume, a to su ujedno i najčešće korišteni algoritmi klasifikacije. Najtočnije rezultate daje algoritam slučajne šume jer odabire najtočniju izlazne vrijednost od većeg broja algoritma stabala odluke. Rezultati ispitivanja web aplikacije pokazuju da su klirens kreatinina i procjena brzine glomerularne filtracije glavni pokazatelji kroničnog bubrežnog zatajenja, dok je omjer ureje i kreatinina pokazatelj akutnog bubrežnog zatajenja.

Ključne riječi: bolesti bubrega, čimbenici rizika, klasifikacija, strojno učenje, web aplikacija

ABSTRACT

The main task of this master's thesis was to realize the web application for predicting risk of kidney failures with classification data analysis procedures. The realization is described with *PHP, HTML* and *CSS*. Procedures has been implemented in the web application with cURL PHP library where the input data is forwarded by Postman development environment to the web servis which returns the disease risk prediction. For classification, logistic regression, neural network and random forest were used as data analysis algorithms because they are most widely used for classification. The most accurate results were given by Random forest algorithm because it chooses the most accurate output value of multiple Decision tree algorithms. Results of testing the web application show that Creatinine clearance and estimated glomerular filtration rate are the main indicators of chronic kidney failure, while the urea and creatinine ratio is indicator of accute kidney failure.

Keywords: classification, kidney failures, machine learning, risk factors, web application

ŽIVOTOPIS

Marin Varnica rođen je 4. ožujka 1995. godine u Đakovu. U razdoblju od 2002. do 2010. pohađao je Osnovnu školu „Ivan Goran Kovačić“. Nakon završene Osnovne škole upisuje Srednju strukovnu školu „Braća Radić“ u Đakovu (današnja Ekonomska škola „Braća Radić“) te ju završava 2014. godine. Nakon srednje škole, 2014. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija (tadašnji Elektrotehnički fakultet u Osijeku), smjer računarstvo. Preddiplomski sveučilišni studij završava 12. rujna 2018. i dobiva kvalifikaciju sveučilišnog prvostupnika inženjera računarstva (*univ.bacc.ing.comp.*). Zatim upisuje diplomski studij, izborni blok Informacijske i podatkovne znanosti te ga pohađa u razdoblju od 2018. do danas. U razdoblju od 6. kolovoza 2019. do 30. kolovoza 2019. obavlja ljetnu developersku praksu kao back-end developer u tvrtki „Inchoo“ u Osijeku.

PRILOZI

Prilog 1. Diplomski rad u *docx* i *pdf* formatu

Prilog 2. Programsko rješenje web aplikacije za procjenu rizika obolijevanja i praćenje zdravstvenog stanja pacijenata s bolestima bubrega