

Web aplikacija za procjenu rizika obolijevanja i praćenje tijeka bubrežnih bolesti korištenjem strojnog učenja

Adžić, Patricia

Undergraduate thesis / Završni rad

2024

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:200:975282>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

Sveučilišni studij

**WEB APLIKACIJA ZA PROCJENU RIZIKA
OBOLJEVANJA I PRAĆENJE TIJEKA BUBREŽNIH
BOLESTI KORIŠTENJEM STROJNOG UČENJA**

Završni rad

Patricia Adžić

Osijek, 2024.

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	1
2. BOLESTI BUBREGA, STANJE U PODRUČJU I POSTOJEĆA RJEŠENJA	2
2.1. Bubrezi.....	2
2.2. Bolesti bubrega i simptomi	2
2.2.1. Akutna bubrežna ozljeda.....	3
2.2.2. Kronična bolest bubrega	3
2.2.3. Bubrežni kamenci	3
2.2.4. Glomerulonefritis.....	4
2.2.5. Policistična bolest bubrega	4
2.2.6. Infekcije mokraćnog sustava.....	5
2.2.7. Bubrežna tuberkuloza	6
2.3. Stanje u području praćenja bolesti bubrega	6
2.4. Postojeća slična rješenja	8
2.4.1. Renalytix AI	8
2.4.2. DeepMind Health's Streams.....	8
2.4.3. Biovitals Kidney tvrtke Biofourmis.....	8
2.4.4. PredKid: Predikcija bolesti bubrega korištenjem ML algoritama	8
2.4.5. Usporedba s KidneyCare AI	9
3. MODEL I ARHITEKTURA WEB APLIKACIJE	10
3.1. Parametri za izradu profila	10
3.2. Parametri za unos podataka	10
3.3. Funkcionalni zahtjevi za web aplikaciju	11
3.3.1. Prijava korisnika	11
3.3.2. Registracija korisnika.....	11
3.3.3. Unos podataka	12
3.3.4. Preporuke s obzirom na procjenu stanja	13
3.3.5. Pregled pacijenata	13
3.3.6. Odjava korisnika	13
3.4. Nefunkcionalni zahtjevi za web aplikaciju.....	13
3.5. Korištenje strojnog učenja.....	15
3.5.1. Skup podataka	15
3.5.2. Korišteni postupci strojnog učenja.....	15
3.5.3. Razvoj modela strojnog učenja	16
4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE.....	20

4.1. Korišteni programski alati, jezici i tehnologije	20
4.1.1. Python	20
4.1.2. Django Framework	20
4.1.3. SQLite baza podataka	20
4.1.4. HTML, CSS, JavaScript	20
4.1.5. Bootstrap Framework.....	21
4.2. Programsko rješenje na strani korisnika	21
4.2.1. Programsko rješenje registracije korisnika	21
4.2.2. Programsko rješenje prijave korisnika.....	22
4.2.3. Programsko rješenje unosa podataka	23
4.2.4. Programsko rješenje za analizu podataka	24
4.2.5. Programsko rješenje praćenja simptoma i pregleda pacijenata	25
4.2.6. Programsko rješenje odjave korisnika	27
4.3. Programsko rješenje na strani poslužitelja	27
4.3.1. Stvaranje i prikaz preporuka	27
4.3.2. Ostvarenje baze podataka	28
4.4. Implementacija modela strojnog učenja u web aplikaciju	28
4.4.1. Trening modela i serijalizacija	28
4.4.2. Integracija modela i predikcija.....	28
4.4.3. Obrada i analiza podataka u stvarnom vremenu	28
4.4.4. Sigurnost i zaštita podataka	29
4.4.5. Kontinuirana integracija i implementacija.....	29
4.4.6. Zapisivanje i praćenje	29
5. NAČIN KORIŠTENJA I ISPITIVANJE RADA WEB APLIKACIJE I MODELA STROJNOG UČENJA	30
5.1. Način korištenja web aplikacije	30
5.2. Ispitivanje rada web aplikacije	31
5.2.1. Ispitivanje postupka registracije korisnika.....	31
5.2.2. Ispitivanje postupka prijave korisnika	32
5.2.3. Ispitivanje postupka unosa podataka	32
5.2.4. Ispitivanje postupka praćenje pacijenta.	33
5.2.5. Ispitivanje i vrednovanje modela strojnog učenja.....	34
5.3. Ispitivanje web aplikacije na slučajevima korištenja i analiza	35
5.3.1. Ispitni slučaj 1	35
5.3.2. Ispitni slučaj 2	35
5.3.3. Ispitni slučaj 3	36
5.3.4. Analiza rezultata ispitivanja.....	36

6. ZAKLJUČAK.....	38
LITERATURA.....	39
POPIS SLIKA.....	42
POPIS TABLICA	43
SAŽETAK.....	44
ABSTRACT	45
ŽIVOTOPIS.....	46
PRILOZI.....	47

1. UVOD

Prema [1], u suvremenom svijetu društvo pridaje veliki značaj ljudima oboljelim od bubrežnih bolesti, čiji broj se procjenjuje na više od 850 milijuna u svijetu, a u ujedno zauzima sedmo mjesto po uzroku smrti u Hrvatskoj. Zbog velikog broja oboljelih Međunarodno društvo za nefrologiju (The International Society of Nephrology) i Međunarodno udruženje Fondacije za bubrege (International Federation of Kidney Foundations) pokrenulo je kampanju obilježavanja Svjetskog dana bubrege, koja se obilježava svake godine drugog četvrtka u ožujku.

Ovaj završni rad analizira probleme i izazove u procjeni rizika obolijevanja od bubrežnih bolesti i praćenju njihova tijeka. Cilj je razviti web aplikaciju koja koristi model strojnog učenja za analizu zdravstvenih podataka pacijenata. Aplikacija će pružati predviđanje o riziku od kronične bubrežne bolesti i preporuke za daljnje korake. Dodatno, rad definira funkcionalne i nefunkcionalne zahtjeve, model i arhitekturu web aplikacije, te analizira rezultate ispitivanja rada aplikacije na primjerima.

1.1. Zadatak završnog rada

U završnom radu potrebno je analizirati probleme i izazove procjene rizika obolijevanja i praćenja tijeka bubrežnih bolesti, te stanje u području i postojeća slična rješenja. Nadalje, treba odrediti značajke modela i prikladne postupke strojnog učenja prema vrsti problema, pripremiti odgovarajuće skupove podataka za treniranje i testiranje modela, trenirati i ispitati model strojnog učenja, pripremiti ga za ugradnju u web sustav s mogućnošću stvaranja preporuka u skladu sa stanjem pacijenta. Također, treba definirati funkcionalne i nefunkcionalne zahtjeve, model i arhitekturu web aplikacije s korisničke i poslužiteljske strane. Web aplikacija treba omogućiti registriranje i prijavu korisnika, izradu osobnog profila, unos i pohranu podataka, određivanje rizika obolijevanja, praćenje tijeka bolesti, te stvaranje i prikaz preporuka pacijentu. Web aplikaciju i model strojnog učenja treba ostvariti koristeći prikladne programske jezike, tehnologije i razvojne okvire, te ih ispitati i analizirati na odgovarajućim ulaznim podacima i slučajevima korištenja.

2. BOLESTI BUBREGA, STANJE U PODRUČJU I POSTOJEĆA RJEŠENJA

U ovome će poglavlju biti opisan rad bubrega, bolesti i simptomi vezani uz bolesti bubrega, te će biti prikazano stanje u području i postojeća rješenja za bolesti bubrega.

2.1. Bubrezi

Bubreg [2] je parni organ smješten neposredno ispod dijafragme s obje strane kralježnice, u stražnjem dijelu trbušne šupljine. Poprima oblik graha i obično je veličine skupljene šake odrasle osobe. Bubrezi su okruženi slojem masnog tkiva koje im osim sigurnosti pruža i stabilnost, dok je na gornjoj strani svakog bubrega smještena nadbubrežna žlijezda. Iznimno su važni za naš organizam jer svojom pravilnom funkcijom uklanjaju različite otrove iz tijela i reguliraju izlučivanje vode, zbog čega su smatrani predstavnicima centralnog dijela mokraćnog sustava. Također zaslužni su za održavanje ravnoteže elektrolita, od kojih kao predstavnike možemo izdvojiti natrij, kalcij i kalij. Nadalje, sudjeluju u aktivaciji vitamina D, koji ima važnu ulogu za zdravlje kostiju. Osim navedenog važno je i napomenuti da proizvode hormon renin za regulaciju krvnog tlaka i hormon eritropoetin, koji stimulira proizvodnju eritrocita.

2.2. Bolesti bubrega i simptomi

Bolesti bubrega mogu se naglo pojaviti, uzrokovati ozbiljnu disfunkciju organa, a zatim se brzo povući s potpunim ili djelomičnim oporavkom [3]. Najčešći uzročnici koji dovode do smanjenja funkcije bubrega su dijabetes tipa 1 i dijabetes tipa 2 ili visok krvni tlak. Osobe sa dijabetesom imaju sklonost obolijevanju od bubrežnom bolesti jer povišena razina šećera u krvi može izazvati oštećenje krvnih žila bubrega, zvanih kapilara, čije oštećenje onemogućuje ispravno pročišćavanje otpadnih tvari iz krvi, odnosno iz cijelog organizma. Drugi veliki uzročnik obolijevanja od bolesti bubrega je povišen krvni tlak. U slučajevima učestalog povišenja krvnog tlaka vrši se pritisak na stijenke kapilara. Kao posljedica hipertenzije, stijenke kapilara gube elastičnost što otežava prolazak krvi kroz bubrege. Glomerulonefritis se uzima kao najčešći poremećaj obolijevanja bubrega. Podjela glomerulonefritisa može biti na primarni, bolest potječe iz bubrega, ili na sekundarni, bolest obuhvaća širi spektar poremećaja organizma. Druga podjela i ona poznatija je na akutni i kronični. Također razne nesreće i traume kao primjerice automobilske nesreće ili sportske ozljede mogu dovesti do trajnih oštećenja bubrega.

2.2.1. Akutno bubrežno oštećenje

Akutno bubrežno oštećenje (eng. Acute Kidney Injury - AKI) je smanjenje normalnog rada bubrega zbog pojave raznih štetnih čimbenika, što može imati kratkoročni, ali i dugoročni učinak na oštećenje funkcije bubrega [4]. Bolest se može razviti u razdoblju od svega nekoliko sati do nekoliko dana, što nam pokazuje i činjenica da se pri primitku u bolnicu 5% pacijenata dijagnosticira oboljelim od ove bolesti. S obzirom na to da akutno bubrežno oštećenje dovodi do nakupljanja štetnih tvari i poremećene ravnoteže elektrolita u organizmu, ova se bolest smatra iznimno opasnom za ljudski život. Za liječenje je nužno otkrivanje te uklanjanje osnovnog uzročnika koji je doveo do bolesti. Što se prije prepozna, može se započeti liječenje i veći su izgledi za oporavak normalne bubrežne funkcije.

2.2.2. Kronična bolest bubrega

Kronična bolest bubrega (eng. Chronic Kidney Disease - CKD) je stanje narušavanja funkcije bubrega, što je najčešće izazvano učestalim upalnim procesima, čija posljedica može biti trajno oštećenje bubrega. Bolest koja remeti funkciju te uništava građu bubrega smatra se kroničnom ako joj je duljina trajanja duža od tri mjeseca. Kronična bolest bubrega je češća kod starijih osoba, žena, rasnih manjina i kod ljudi koji boluju od dijabetesa melitusa i hipertenzije. Kronična bolest bubrega predstavlja posebno velik teret u zemljama s niskim i srednjim prihodima, koje su najmanje opremljene za suočavanje s njezinim posljedicama[5]. Simptomi zatajenja bubrega sporo napreduju što dodatno otežava prepoznavanje bolesti. Također simptomi su vrlo nespecifični jer mogu ukazivati na neki drugi problem, dok je pravi problem zapravo u oboljenju bubrega. Razlog tome je što su bubrezi prilagodljiv organi i u slučaju bolesti pokušavaju nadomjestiti svoju funkciju pa se bolest teško dijagnosticira sve do napredovalog stadija. Nažalost, kronična bolest bubrega je neizlječiva te jedini cilj je pokušaj smanjenja progresije oštećenja bubrega, za što postoje samo dvije opcije terapije. Prva je dijaliza, odnosno umjetno filtriranje otpadnih tvari iz krvi i organizma. Druga je transplantacija bubrega u kojoj kirurškim zahvatom zamjenjuju oboljeli bubreg pacijenta sa zdravim bubregom donora.

2.2.3. Bubrežni kamenci

Bubrežni kamenci nastaju taloženjem kristala iz mokraće i nakupljanjem na unutrašnjim površinama bubrega. No, iako se nazivaju bubrežnim kamencima, ispravnije ih je zvati mokraćnim kamencem ili urolitijazom jer mogu zahvatiti bilo koji dio mokraćnog sustava. Prema [6] bolest zahvaća 19% muške populacije te 9% ženske, što ukazuje na veliku učestalost bolesti. Podjela

kamenaca se odvija prema vrsti tvari koje ga čine i one mogu biti: kalcijeve, uratne, cistinske i struvitne. Kamenci građeni kalcijem čine udio od 80% svih mokraćnih kamenaca. Liječenje ovisi o veličini kamena. Sitni kamenci predstavljaju manji problem jer uzrokuju minimalne smetnje i za njihovo otklanjanje potreban je unos 2,5-3 litre tekućine dnevno, kako bi se kamenac izmokrio. Veći kamenci uklanjaju se postupkom ekstrakorporealne litotripsije šok valovima, poznatije kao razbijanje kamena. Kamenci se tom terapijom pretvaraju u pijesak i odlaze kroz urinarni trakt mokraćom.

2.2.4. Glomerulonefritis

Glomerulonefritis [7] već je spomenuti, najčešći oblik poremećaja normalne funkcije rada bubrega. Karakterizira ga oštećenje i upala glomerula, nakupine sitnih krvnih žila u bubrezima s porama koje im omogućuju filtraciju krvi. Podjela ove bolesti je na akutnu, često poznati i kao post-streptokokni glomerulonefritis, jer je izazvan bakterijskom infekcijom, najčešće streptokoknom infekcijom grla. Kronični glomerulonefritis je pak progresivna bolest jer se polako razvija vremenom, a uzrokovana je autoimunom bolešću ili postojanjem genetske predispozicije. Bergerova bolest također je jedna od vrsta glomerulonefritisa, poznata je po nakupljanju IgA antitijela u glomerulima. Simptomi glomerulonefritisa ovise o težini i vrsti bolesti i mogu uključivati: povećanu količinu proteina u urinu, hematurija ili prisutnost krvi u mokraći, oticanje u licu, rukama, nogama i abdomenu, zbog posljedica zadržavanja tekućine te jedan od simptoma može biti i hipertenzija ili visok krvni tlak. Liječenje varira o uzroku i težini bolesti, obuhvaća različite pristupe kao što su konzumacija lijekova za smanjivanje imunološkog odgovora kako bi se spriječilo daljnje oštećenje glomerula te lijekova za smanjenje upale i regulacije krvnog tlaka. Postoji i mogućnost dijalize kako bi se spriječilo trajno oštećenje bubrega i omogućilo pacijentu kvalitetan život usprkos oboljenju glomerulonefritisom. Transplantacija je korak liječenja koji je nužan tek u terminalnom stadiju bolesti, kada dijaliza više nije od pomoći. Transplantacija je u tom slučaju jedina opcija za dugoročan život bolesnika, ali zahtjeva cjeloživotnu uporabu imunosupresivnih lijekova koji sprječavaju organizam od odbacivanja transplantiranog organa.

2.2.5. Policistična bolest bubrega (PKD)

Policistična bolest bubrega (eng. Polycystic Kidney Disease - PKD) je životno ugrožavajuća bolest uzrokovana genetskim poremećajem kojeg karakterizira tekućinom ispunjena cista u bubrezima. Smatra se genetskim poremećajem i jedna je od vodećih uzročnika kroničnog zatajenja bubrega, a prisutna je kod odraslih, ali i kod djece. Podjela se svodi na jedan od dva oblika PKD-a. Autonomno dominantna policistična bolest bubrega (eng. Autosomal Dominant Polycystic

Kidney Disease - ADPKD) je češći oblik te se naziva i auditivnim oblikom PKD-a jer se ne razvija tek do 30. ili 40. godine života. Izazvan je mutacijom PKD1 ili PKD2 gena [8]. Rjeđim oblikom se smatra autosomno recesivna policistična bolest bubrega (eng. Autosomal Recessive Polycystic Kidney Disease - ARPKD). ARPKD je relativno rijetka bolest koja uzrokuje kliničke simptome pri rođenju, s visokim mortalitetom tijekom prvog mjeseca života[9]. Uzrokovan je mutacijom u obje kopije PKHD1 gena. Simptomi PKD-a razlikuju se u broju i veličini cista te mogućnosti dodatnih komplikacija. Najčešći su bolovi u trbuhu ili u boku, zbog pritiska cista na okolna tkiva. Treba spomenuti i hipertenziju, hematuriju, infekciju mokraćnog sustava te pojavu bubrežnih kamenaca, koja nastaje pri promijeni bubrežne funkcije i strukture. Otkrivanje PKD-a i procjena težina bolesti moguća je uporabom raznih metoda kao što su: ultrazvuk, genetičko testiranje, testiranje PKD1, PKD2 i PKHD1 gena, koje pruža uvid radi li se o ADPKD-u ili ARPKD-u, te metode CT ili MRI koje omogućuju detaljniju sliku bubrega te praćenje njegove funkcije i strukture. Pravovremeno dijagnosticiranje bolesti utječe na načine liječenja pacijenta. Lijek za PKD ne postoji, ali postoje načini za smanjivanje simptoma i održavanje kvalitetnog života bolesnika. Određeni lijekovi za hipertenziju i određeni analgetici i antibiotici za liječenje infekcije mokraćnog sustava mogu se prepisati za kontrolu simptoma pacijenta. U slučaju zatajenja bubrega potrebna je transplantacija, a ako je transplantacija onemogućena provodi se dijaliza za održavanje normalne funkcije rada bubrega. Eksperimentalni tretmani mogu usporiti rast cista, ali uključuju rizik od uporabe nepotpuno ispitanih lijekova. Važno ulogu imaju i redoviti medicinski pregledi zbog kontroliranja razvoja ciste i odlučivanja o vrsti tretmana. Potrebna je i promjena svakodnevnog načina života u smislu prilagodbe prehrane, izbacivanja unosa soli i proteina, povećavanje tjelesne aktivnosti te prestanka pušenja, koje može pomoći pri usporavanju progresije bolesti i u kontroli simptoma.

2.2.6. Infekcije mokraćnog sustava (IMS)

Najčešći uzročnik infekcije mokraćnog sustava, bolesti prisutne pretežito među ženskom populacijom, je bakterija *Escherichia coli*, koja preko mokraćnih puteva te na kraju mokraćnog mjehura odlazi do bubrega gdje izaziva akutnu upalu mokraćnog mjehura ili akutnu upalu bubrega [10]. Također je češće prisutna i kod pacijenata oboljelih od policistične bolesti bubrega zbog strukturnih abnormalnosti i pojave cista, što otežava liječenje radi prodora spomenute bakterije u nastale ciste i bubrežno tkivo. Simptomi kod bolesnika oboljelih PKD-om su razni, a obuhvaćaju: osjećaj boli te učestalost prilikom mokrenja, hematuriju, bolove u stomaku ili u boku, groznicu i zimnicu te iscrpljenost i osjećaj slabosti. U slučaju prisutnosti nekih od simptoma, prisutnost bolesti može se dijagnosticirati analizom urina, krvnim testom i ultrazvukom ili CT skeniranjem,

koje detektiraju prisutnost oboljelih cista ili drugih bubrežnih abnormalnosti. Standardno liječenje zahtjeva antibiotsku terapiju, dok su kod ozbiljnijih infekcija potrebni intravenski antibiotici. Prevencija bolesti je moguća povećanim unosom tekućine te pravovremenim liječenjem i redovitim posjetima liječniku, kako ne bi došlo do komplikacija kao što su: sepsa, zatajenje bubrega te oboljenja od kronične infekcije, koja zahtjeva dugotrajnu antibiotsku terapiju te druge intervencije.

2.2.7. Bubrežna tuberkuloza

Tuberkulozni nefritis jedan je od oblika ekstrapulmonalne tuberkuloze uzrokovane bakterijom *Mycobacterium tuberculosis* koja, nakon što dospije u pluća, širi se krvotokom i taloži u organima, kao što je bubrež [11]. Unutar bubrega stvara nakupine upalnih stanica granulozne, koje štete bubrežno tkivo te ostavljaju ožiljke. Suptilnost pojave i spori razvoj simptoma bubrežne tuberkuloze otežavaju pravovremenu reakciju bi liječenje ove bolesti, a najčešći su: bol u lumbalnom dijelu leđa, učestala i bolna mokrenja, hematurija, opća slabost i osjećaj umora, groznica, gubitak apetita te gubitak težine. Bubrežna tuberkuloza izazovna je za dijagnosticiranje, jer njeni simptomi mogu zavarati i ukazati na druge bubrežne infekcije [12]. Dijagnostičke metode prepoznavanje bolesti su: analiza urina, PCR testiranje, tuberkulozni kožni test, biopsija bubrega i razne radiološke pretrage kao što su ultrazvuk, CT ili rendgenski snimci bubrega. Pravodobna dijagnoza i liječenje ključni su za sprječavanje trajnog oštećenja i očuvanje pravilne funkcije bubrega. Tretmani liječenja uključuju antituberkulozne lijekove, redovite liječničke preglede te u slučajevima uznapredovanja bolesti kirurški zahvat. Bubrežna tuberkuloza bolest je koja zahtjeva rano prepoznavanje zbog rizika od raznih komplikacija kao što su: kronično zatajenje bubrega, kronično zatajenje bubrega, stenoza mokraćovoda te širenje infekcije na druge dijelove mokraćnog sustava.

2.3. Stanje u području praćenja bolesti bubrega

Kronična bolest bubrega globalni je zdravstveni izazov koji pogađa milijune ljudi diljem svijeta. U mnogim slučajevima, ovaj se poremećaj razvija bez ikakvih simptoma, a pogođeni pojedinci mogu se naći u kritičnom stanju prije nego što uopće shvate problem. Stoga je rana detekcija uz kontinuirano praćenje stanja bitna za učinkovito upravljanje bolesti i poboljšanje ishoda za pacijente.

Prema [13], razvoj umjetne inteligencije i tehnologija strojnog učenja omogućio je stvaranje niza naprednih rješenja usmjerenih na rano otkrivanje, stratifikaciju rizika i personaliziranu njegu u upravljanju CKD-om.

U sustavu KidneyCare AI [14] primijećena je primjena programskih tehnologija koje omogućuju korisnicima jednostavan način unosa relevantnih zdravstvenih podataka, uključujući krvni tlak, urin i kreatinin, putem web-platforme. Sustav također koristi algoritam strojnog učenja u pozadini, Random Forest, za automatiziranu analizu podataka. Primjerice, Renalytix AI [15], tehnološka tvrtka u zdravstvu, koristi algoritme strojnog učenja za predviđanje rizika od napredovanja bolesti na temelju analize biomarkera u krvi i povijesti bolesti.

Algoritamski pristupi sustavno pristupaju identifikaciji rizika od CKD-a. U ovom radu je uključen razvoj nadziranih modela strojnog učenja. Model Random Forest, pomaže u poboljšanju preciznosti predviđanja razvojem šume višestrukih stabala odluke. Ovaj model trenira se na velikim skupovima povijesnih podataka o pacijentima, što mu omogućuje da prepozna obrasce povezane s ovom bolešću. Takav pristup osigurava valjane prognoze, a model se preobučava kada su dostupni novi podaci.

Obrada velikih količina podataka zahtijeva primjenu tehnika predobrade, poput normalizacije, rješavanja nedostajućih vrijednosti i odabira značajki, kako bi se podaci pripremili za obuku modela i predviđanje. Konačno, KidneyCare AI pruža izlaz koji se može koristiti u praksi, izveden iz prikaza u stvarnom vremenu koji također prikazuje rizik od bolesti koji je analiziran.

Kada sustav upozori na visoki rizik od CKD-a, preporučuje se da pacijenti slijede smjernice za daljnju dijagnostiku, što može uključivati dodatne pretrage, te da se savjetuju s liječnikom kako bi se odredili potrebni koraci za liječenje i praćenje bolesti.

Drugi primjeri sustava koji koriste AI za praćenje CKD-a uključuju DeepMind Health's Streams [16], koji prati funkciju bubrega u stvarnom vremenu i šalje upozorenja zdravstvenim djelatnicima te Biofourmis' Biovitals Kidney, koji koristi nosive uređaje za praćenje pacijenata s CKD-om u stvarnom vremenu.

Sustav PredKid [17] koristi algoritme strojnog učenja za predviđanje rizika od CKD-a na temelju podataka o biomarkerima. PredKid koristi nekoliko modela strojnog učenja, kao što su logistička regresija, stabla odluke i potporni vektorski strojevi, kako bi se pacijenti klasificirali prema riziku od CKD-a. Ovaj model se može koristiti u raznim ustanovama koje izravno rade s pacijentima.

2.4. Postojeća slična rješenja

Tehnologija u zdravstvu ubrzano napreduje, a nekoliko rješenja temeljenih na strojnim algoritmima za praćenje bolesti bubrega već je dostupno. Ta se rješenja razlikuju po prediktivnim modelima, unosima podataka, sučeljima i dostupnosti:

2.4.1. Renalytix AI

Renalytix AI je tehnološka tvrtka u zdravstvu koja se fokusira na upotrebu umjetne inteligencije za rano otkrivanje i stratifikaciju rizika od CKD-a [18]. Njihova tehnologija "Kidney IntelX" koristi algoritme strojnog učenja za predviđanje rizika od progresije bolesti obrađujući elektroničke zdravstvene zapise pacijenata. Predikcija se temelji na kombinaciji biomarkera u krvi te povijesti bolesti i pacijenata, koristeći algoritam strojnog učenja obučen na kombiniranim podacima o ekspresiji gena bubrega i krvi. Renalytix AI pomaže u određivanju pacijentovog rizika unutar različitih kategorija biomarkera, kao i na temelju demografskih i kliničkih podataka. Renalytix AI je već komercijalno dostupan i integriran u kliničku praksu u SAD-u.

2.4.2. DeepMind Health's Streams

DeepMind, dio Googlea za umjetnu inteligenciju, razvio je "Streams" – sustav za praćenje funkcije bubrega u stvarnom vremenu [19]. Streams koristi AI algoritme za detekciju akutnog oštećenja bubrega (AKI). Ova rješenja koriste se u bolnicama kako bi se olakšao rad zdravstvenim djelatnicima i omogućilo brže i učinkovitije liječenje pacijenata.

2.4.3. Biovitals Kidney tvrtke Biofourmis

Platforma Biofourmis "Biovitals Kidney" koristi nosivé uređaje za praćenje pacijenata s CKD-om u stvarnom vremenu. Platforma prikuplja podatke iz uređaja poput monitora otkucaja srca i tlakomjera kako bi se analizirali trendovi i rano otkrili problemi [20]. Rješenje omogućava praćenje pacijenata izvan bolnice, a cilj je smanjiti potrebu za hospitalizacijom i osigurati pravovremene intervencije.

2.4.4. PredKid: Predikcija bolesti bubrega korištenjem ML algoritama

PredKid koristi algoritme strojnog učenja za predviđanje rizika od CKD-a na temelju podataka o biomarkerima. PredKid koristi nekoliko modela strojnog učenja, kao što su logistička regresija, stabla odluke i potporni vektorski strojevi, kako bi se pacijenti klasificirali prema riziku od CKD-a. Ovaj model se može koristiti u raznim ustanovama koje izravno rade s pacijentima[21].

2.4.5. Usporedba s KidneyCare AI

Tablica 2.1 prikazuje usporedbu postojećih rješenja s web aplikacijom KidneyCare AI.

Tablica 2.1 Usporedba postojećih rješenja sa KidneyCareAI

Značajka	KidneyCare AI	Renalytix AI (KidneyIntelX)	DeepMind Health (Streams)	Biofourmis (Biovitals Kidney)	PredKid
Modeli strojnog učenja	✓	✓	✓	✓	✓
Izvori podataka	✓	✓	✓	✓	✓
Praćenje u stvarnom vremenu	✗	✗	✓	✓	✗
Samostalno upravljanje pacijentima	✓	✗	✗	✓	✗
Integracija s bolničkim sustavima	✗	✓	✓	✗	✗
Izlaz predikcije	✓	✓	✓	✓	✓
Korisničko sučelje	✓	✗	✓	✓	✓
Pristupačnost	✓	✗	✗	✓	✓
Fokus na CKD	✓	✓	✗	✓	✓
Preporuke za pacijente	✓	✗	✗	✓	✓

Jedan od važnih trendova u zdravstvenim tehnologijama je primjena umjetne inteligencije i strojnog učenja u praćenju bolesti bubrega. U kontekstu postojećeg stanja u industriji, sve je očitije da umjetna inteligencija i strojno učenje počinju igrati ključnu ulogu. Dostupna rješenja poput Renalytix AI ili DeepMind Health's Streams, uz Biofourmis' Biovitals Kidney i PredKid, pokazuju koliko se raznoliko pristupa rješavanju ovog rastućeg problema. Dok neka rješenja pomažu u predviđanju rizika od CKD-a na temelju povijesti bolesti pojedinca, druga uključuju proces praćenja u stvarnom vremenu korištenjem nosivih uređaja i odgovarajuće mobilne aplikacije. Stoga se budućnost praćenja bolesti bubrega zasniva na daljnjem usavršavanju AI alata kako bi bili precizniji, dostupniji i bolje integrirani u rad zdravstvene industrije. Zdravstvena industrija će moći zadovoljiti povećanu upotrebu AI alata s obzirom na prethodno iskustvo na obje strane, od strane liječnika i pacijenata.

3. MODEL I ARHITEKTURA WEB APLIKACIJE

U ovom poglavlju bliže će se opisati model i arhitektura web aplikacije. Detaljno će se razmotriti parametri za izradu profila, unos podataka te funkcionalni i nefunkcionalni zahtjevi za web aplikaciju.

3.1. Parametri za izradu profila

Prije nego što korisnik stupi u interakciju s platformom KidneyCare AI, mora kreirati novi profil. Započinje se unosom osnovnih informacija, a za personalizirano iskustvo, korisnik može dodati dodatne podatke koji će prilagoditi preporuke i poboljšati iskustvo. Tablica 3.1 prikazuje sve potrebne parametre za stvaranje profila.

Tablica 3.1 Parametri za izradu profila

Parametar	Opis
Ime i prezime	Puno ime korisnika za potrebe identifikacije.
Adresa e-pošte	Ispravna email adresa za verifikaciju računa, komunikaciju i oporavak lozinke.
Odabir uloge	Korisnik odabire svoju ulogu (pacijent ili zdravstveni djelatnik), što prilagođava korisničko sučelje i funkcionalnosti.
Lozinka	Zaštitna lozinka koja ispunjava specifične sigurnosne kriterije (npr. minimalna duljina, posebni znakovi) za zaštitu korisničkih računa.

3.2. Parametri za unos podataka

Korisnik, prilikom registracije, unosi ključne zdravstvene parametre potrebne za pokretanje modela za predviđanje bolesti bubrega. Unos se provodi putem jasnog i jednostavnog obrasca na korisničkom sučelju kako bi se osiguralo točno prikupljanje svih potrebnih podataka. Tablica 3.2 opisuje parametre za unos podataka:

Tablica 3.2 Parametri za unos podataka

Parametar	Opis
Dob	Korisnikova dob, koja je značajan faktor u procjeni rizika za funkciju bubrega.
Krvni tlak	Sistolički i dijastolički tlak, ključni za prepoznavanje oštećenja bubrega povezanih s hipertenzijom.
Specifična težina	Mjera koncentracije urina, koja ukazuje na sposobnost filtracije bubrega.
Razina albumina	Otkriva prisutnost proteina u urinu, što može ukazivati na moguće oštećenje bubrega.

Razina šećera	Mjeri šećer u urinu kako bi se provjerilo postojanje dijabetesa, koji je glavni faktor rizika za bubrežne bolesti.
Broj crvenih krvnih stanica	U slučaju abnormalnosti može ukazivati na temeljne probleme s bubrezima.
Karbamid u krvi	Krvni test koji mjeri razine otpadnih tvari za procjenu funkcije bubrega.
Serumski kreatinin	Glavni pokazatelj funkcije bubrega, povišene razine sugeriraju oštećenje.
Elektroliti (npr. Natrij, Kalij)	Neravnoteža može ukazivati na moguću disfunkciju bubrega.
Nasumična razina glukoze u krvi	Pomaže u otkrivanju dijabetesa mjerenjem razine šećera u krvi u nasumičnim trenucima.
Ostali zdravstveni pokazatelji	Uključuje broj gnojnih stanica, razinu hemoglobina, zapremninu staničnog paketa, itd., kako bi se procijenilo opće zdravlje i otkrile infekcije.

3.3. Funkcionalni zahtjevi za web aplikaciju

Funkcionalni zahtjevi detaljno opisuju operacije koje web aplikacija mora izvršavati kako bi učinkovito ispunila potrebe svojih korisnika [22, 23]. U slučaju platforme KidneyCare AI, omogućena je funkcionalna registracija za nove korisnike, prijava te unos parametara za izračun rizika od oboljenja. Platforma također provodi procjenu rizika od bubrežne bolesti i nudi preporuke na temelju stanja pacijenta. Sve ove informacije korisnici mogu pregledati putem sveobuhvatne nadzorne ploče, na kojoj su prikazani njihovi zdravstveni podaci, predviđeni ishodi i preporuke. Na kraju, omogućena je odjava koja osigurava sigurnost osobnih informacija.

3.3.1. Prijava korisnika

Aplikacija omogućuje korisnicima sigurnu prijavu na svoje račune. Korisnici unose svoje korisničko ime i lozinku, a sustav provjerava valjanost podataka i autentificira korisnika. U slučaju uspješne prijave, korisnik se preusmjerava na odgovarajuću početnu stranicu, ovisno o ulozi (pacijent ili liječnik).

3.3.2. Registracija korisnika

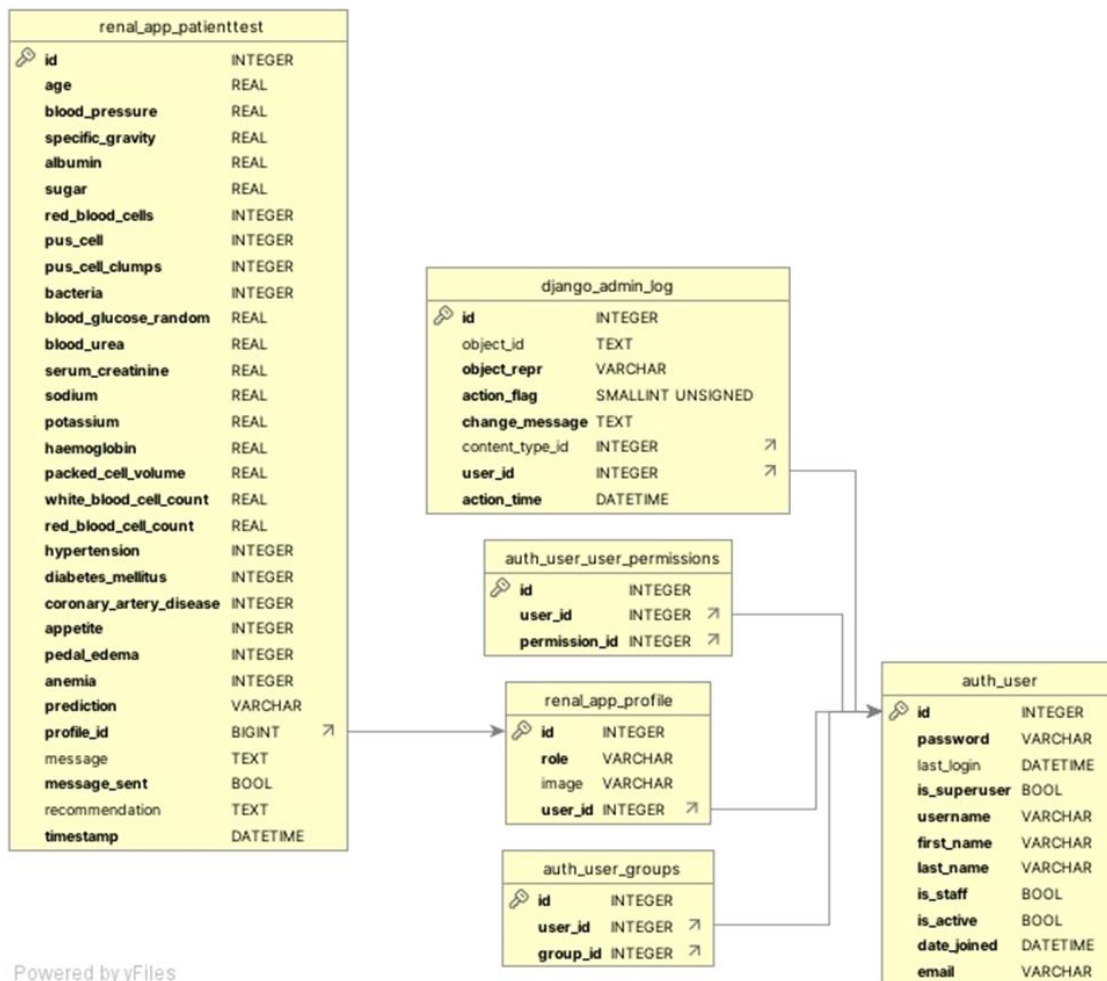
Aplikacija omogućuje jednostavan i siguran proces registracije novih korisnika. Korisnik prilikom registracije unosi osobne podatke (ime, prezime, adresa e-pošte, lozinka) i odabire svoju ulogu (pacijent ili liječnik). Ugrađena provjera valjanosti podataka na strani klijenta i poslužitelja sprječava pogreške i osigurava točnost unesenih podataka. Uspješna registracija rezultira slanjem potvrde na email adresu korisnika i mogućnošću prijave.

3.3.3. Unos podataka

Pacijenti unose svoje zdravstvene podatke kroz jasno strukturirani obrazac u web aplikaciji. Ugrađena provjera unosa u svako polje obrasca sprječava pogreške i osigurava točnost podataka. Ova funkcionalnost osigurava visoku kvalitetu podataka, što rezultira preciznijim prognozama i skraćenim vremenom obuke modela.

Korisnici unose informacije o svojem zdravstvenom stanju, poput dobi, krvnog tlaka, razine šećera, rezultata krvnog testa i drugih relevantnih parametara. Ovi podaci se koriste za izradu profila pacijenta i za generiranje predikcija.

Slika 3.1 prikazuje dijagram relacija između entiteta u bazi podataka. Ova baza podataka sadržava informacije o pacijentima, njihovim zdravstvenim podacima, predikcijama, preporukama i korisničkim profilima. Također, dijagram relacija pomaže da se razumije struktura baze podataka i kako su podaci organizirani. Ove relacije omogućuju web aplikaciji da pristupi podacima učinkovito i izvodi potrebne operacije.



Slika 3.1. Odnosi entiteta

3.3.4. Preporuke s obzirom na procjenu stanja

Nakon što korisnik unese svoje zdravstvene podatke, aplikacija koristi Random Forest model strojnog učenja za analizu podataka i predviđanje postoji li rizik od bolesti bubrega. Rezultat predikcije je binaran: aplikacija će pokazati da osoba boluje od bolesti bubrega ili ne boluje. Ova se predikcija zatim prikazuje liječniku na njegovom profilu. Liječnik ima pristup rezultatima testiranja svih pacijenata te može napisati personalizirane preporuke temeljene na rezultatu predikcije i u skladu s medicinskim stajalištima. Važno je napomenuti da će se sve preporuke temeljiti na kliničkim smjernicama i u skladu su s medicinskim stajalištima. Aplikacija će korisnicima pružiti jasne i sažete informacije, te će im omogućiti da samostalno poduzmu korake za poboljšanje svog zdravlja.

3.3.5. Pregled pacijenta

Korisnicima se prikazuje sveobuhvatna nadzorna ploča na kojoj mogu vidjeti svoje zdravstvene podatke, predviđene ishode i preporuke. To je osobito važno za praćenje promjena kroz vrijeme i kao temelj za racionalne zdravstvene odluke.

3.3.6. Odjava korisnika

Sigurna funkcija odjave omogućuje korisnicima da sigurno izađu iz aplikacije i sprječava neovlašteni pristup njihovim zdravstvenim podacima. Također uključuje automatsku odjavu nakon neaktivnosti radi veće sigurnosti.

3.4. Nefunkcionalni zahtjevi web aplikacije

Ne-funkcionalni zahtjevi fokusiraju se na ukupne kvalitete aplikacije, osiguravajući da bude korisnička, sigurna, pouzdana i učinkovita. Tablica 3.3 prikazuje nefunkcionalne zahtjeve za web aplikacije:

Tablica 3.3 Nefunkcionalni zahtjevi za web aplikaciju

Zahtjev	Opis
Performanse	Aplikacija mora brzo odgovarati na korisničke radnje, s minimalnim kašnjenjem u obradi podataka i pružanju rezultata.
Uporabljivost	Korisničko sučelje treba biti intuitivno, omogućujući korisnicima da lako navigiraju i unose podatke bez opsežnog školovanja.

Pouzdanost	Sustav mora biti robustan, s mjerama za pravilno upravljanje pogreškama i očuvanje dostupnosti.
Sigurnost	Podaci korisnika, posebno osjetljive zdravstvene informacije, moraju biti zaštićeni pomoću enkripcije i sigurnosnih kontrola.
Skalabilnost	Aplikacija treba biti sposobna nositi se s povećanim opterećenjem korisnika bez pogoršanja performansi.
Održavanje	Kod bi trebao biti jednostavan za ažuriranje i održavanje, omogućujući brzu implementaciju novih značajki ili ispravaka.

Kako bi se postigle bolje performanse, aplikacija će koristiti mehanizme predmemorije Django-a i serijalizaciju modela strojnog učenja za predviđanje u stvarnom vremenu. Na taj će način sustav pružiti brze odgovore kada korisnici pošalju svoje podatke ili zatraže predviđanja. Vrijeme odziva također će biti jedna od prednosti implementacije lagane prednje strane koja uključuje minimalnu upotrebu skripti i optimiziranih resursa, što omogućuje brzo postavljanje web-mjesta na bilo kojem uređaju. Kada je u pitanju korisničko iskustvo, sustav Django pružit će čisto i intuitivno korisničko sučelje pomoću Bootstrap-a za odzivni dizajn. Na taj će se način i pacijenti i zdravstveni djelatnici moći lako kretati web-stranicom. Treba naglasiti da će obrazac pružiti jasne upute i provjere valjanosti, što će osigurati da nikakve pogreške ne ometaju interakciju korisnika s umjetnom inteligencijom.

Što se tiče pouzdanosti, aplikacija će pravilno rješavati pogreške i pružiti odgovarajuće mehanizme za sigurnosno kopiranje kako bi se smanjilo vrijeme zastoja i performansa aplikacije ostala neometena. Sigurnost će se pak postići implementacijom enkripcije HTTPS-a kako bi se osiguralo da se podaci prenose sigurno. Osim toga, Django nudi neke sigurnosne značajke koje su ugrađene u sustav, poput zaštite od CSRF-a, pohranjivanja lozinke i višefaktorske autentikacije, što će otežati pristup osjetljivim informacijama trećim stranama. Budući da Django pruža asinhronu značajke sustava, on će se skalirati kako je propisano. Međutim, aplikacija će možda promijeniti način upotrebe SQLite-a u PerconaSQL-u u budućnosti kako bi se prilagodio još većem broju potencijalnih korisnika i većim skupovima podataka. Konačno, održivost će se postići pridržavanjem pravilne organizacije koda, komentiranja i upravljanja verzijama, što će omogućiti promjene u sustavu u budućnosti.

3.5. Korištenje strojnog učenja

Integracija strojne obrade podataka ključna je komponenta aplikacije KidneyCare AI, koja omogućuje predviđanje rizika od bolesti bubrega na temelju korisničkih podataka.

3.5.1. Skup podataka

Skup podataka o bolestima bubrega složen je skup podataka prikupljen s Kaggle-a razvijen kako bi se pomoglo proučavati i predviđati kroničnu bolest bubrega (CKD). Skup podataka o podacima pacijenata s ukupno 24 značajke, kombinirajući kliničke i demografske informacije. Stupci u skupu podataka podijeljeni su kao kontinuirani podaci, poput dobi, krvnog tlaka i uree u krvi, i kategorijalnih stavki, koje uključuju razinu albumina, šećera, odgovarajuće vrijednosti crvenih krvnih stanica. Ključne značajke korištene za izgradnju modela. Svaki od ovih opisa sadrži važne informacije za otkrivanje razvoja i evolucije bolesti bubrega.

Skup podataka je vrlo dobro formatiran, pa se može dobro koristiti za trening modela strojnog učenja za zadatke klasifikacije, kao što je npr. utvrđivanje da li pacijent ima bolest bubrega, što može biti korisna oznaka kao ciljna varijabla za prediktivno modeliranje. Podaci su prethodno obrađeni kako bi se popunili nedostajući podaci i osigurala dosljednost, što ih čini pogodnima za modeliranje. Ovisno o analizi, možda će biti potrebno provesti dodatnu normalizaciju ili skaliranje kako bi se postigla optimalna performansa modela. Zdravstvena industrija oslanja se na ovaj skup podataka kako bi izgradila modele strojnog učenja koji mogu pomoći u pravovremenoj prepoznavanju i liječenju bolesti bubrega, što pomaže poboljšati ishode liječenja pacijenata.

3.5.2. Korišteni postupci strojnog učenja

Aplikacija se oslanja na dvije primarne metode strojnog učenja: logističku regresiju i Random Forest.

Logistička regresija najčešći je regresijski model, a ovaj se algoritam nalazi u problemu binarne klasifikacije, u kojem se prisutnost bolesti bubrega predviđa na temelju ulaznih parametara [24]. To je jednostavno rješenje, a ipak pruža lako razumljive rezultate.

Random Forest gradi više stabala odluka tijekom obuke i određuje ishod prema većini klasa [25]. Odabran je zbog svoje sposobnosti da se nosi s potencijalnom složenošću podataka te doprinosi visokom stupnju prediktivne robusnosti. Zbog boljih performansi na ovom skupu podataka, aplikacija koristi Random Forest model.

3.5.3. Razvoj modela strojnog učenja

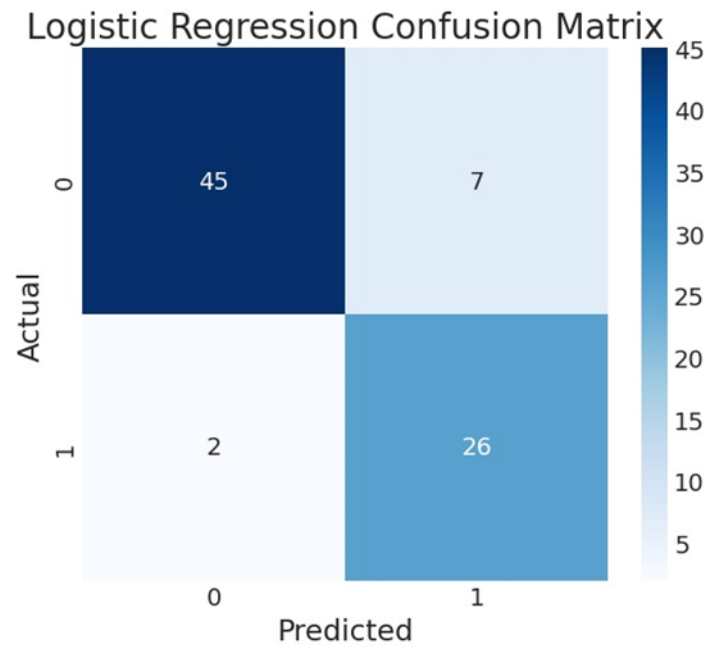
Modeli strojnog učenja razvijeni su na strukturiran način slijedeći ove korake kako bi se postigla visoka točnost i pouzdanost.

1. Prikupljanje i priprema podataka: Podaci su dobiveni iz medicinskih zapisa i studija o uzrocima bolesti bubrega tijekom dvogodišnjeg istraživanja. Predobrada je uključivala rješavanje nedostajućih vrijednosti, skaliranje značajki i podjelu podataka na skup za treniranje i skup za testiranje.
2. Inženjering i odabir značajki: Kombinacija statističkih metoda i stručnog znanja korištena je za odabir najvažnijih značajki – mnogi geni imaju male učinke, ali samo podskup pridonosi vrijednim biološkim detaljima ili predikcijama bolesti. Ključni pokazatelji pronađeni su među parametrima poput serumskog kreatinina, krvnog tlaka i razine albumina.
3. Treniranje modela: Logistička regresija i model Random Forest trenirani su na skupu za treniranje. Performanse modela nisu bile pretrenirane ili nedovoljno trenirane nakon optimizacije hiperparametara.
4. Evaluacija modela: Modeli su zatim evaluirani na skupu za testiranje prema točnosti (eng. accuracy), preciznosti (eng. precision), odzivu (eng. recall) i F1 rezultatu (eng. F1 score). To je robustna priroda modela koju smo potvrdili tehnikama unakrsne validacije. Uspoređivanjem utvrđeno je kako Random Forest daje bolje rezultate navedenih parametara, što je razlog njegovog odabira nad logičkom regresijom za ugrađivanje u aplikaciju.
5. Integracija s web-aplikacijom: Ovaj model se koristi u pozadini Django aplikacije za real-time predikcije nakon što korisnik unese podatke. Pomoću pozadinskog procesa naši podaci se obrađuju i šalju na stranu korisnika kako bi korisnik mogao odmah dati povratne informacije.

Model strojnog učenja Random Forest koriste se u aplikaciji za analizu zdravstvenih podataka koje pružaju korisnici i za predviđanje vjerojatnosti da je osoba izložena većem riziku od razvoja bolesti bubrega. Njegova prediktivna priroda omogućuje pojedincima da donesu informirane odluke o koracima koje trebaju poduzeti kako bi ostali zdravi. Aplikacija KidneyCare AI predstavlja cjelovito rješenje za otkrivanje i upravljanje bolestima bubrega, počevši od rane detekcije, dizajna korisnički prijaznog sučelja, razvoja sigurnosnih značajki za upravljanje podacima, koristeći

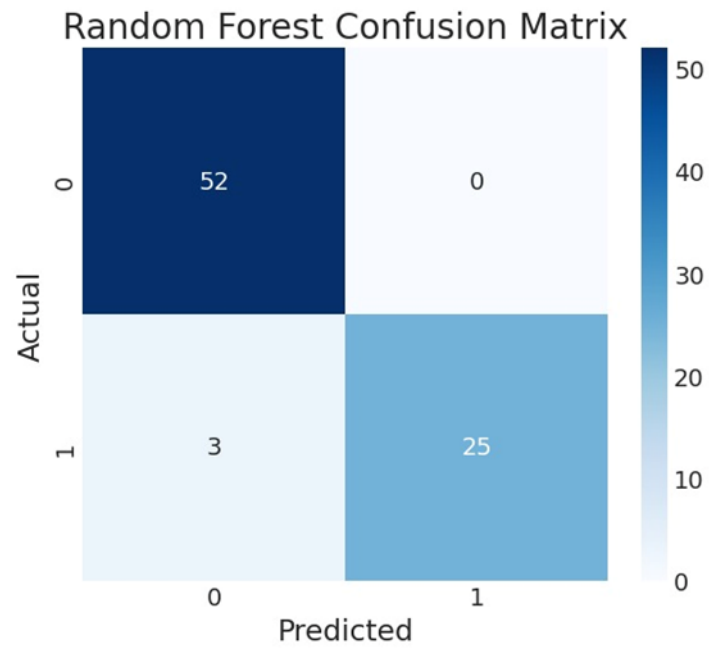
napredne tehnike strojnog učenja. Takav dizajn održat će aplikaciju učinkovito i omogućiti doseganje široke baze korisnika, bilo da se radi o pacijentima ili zdravstvenim djelatnicima.

Slika 3.2 pokazuje da je model logičke regresije točno predvidio 45 pravih negativnih (bez bolesti) i 26 pravih pozitivnih (prisutnost bolesti). Bilo je 7 lažno pozitivnih i 2 lažno negativnih, što ukazuje na uravnoteženu izvedbu.



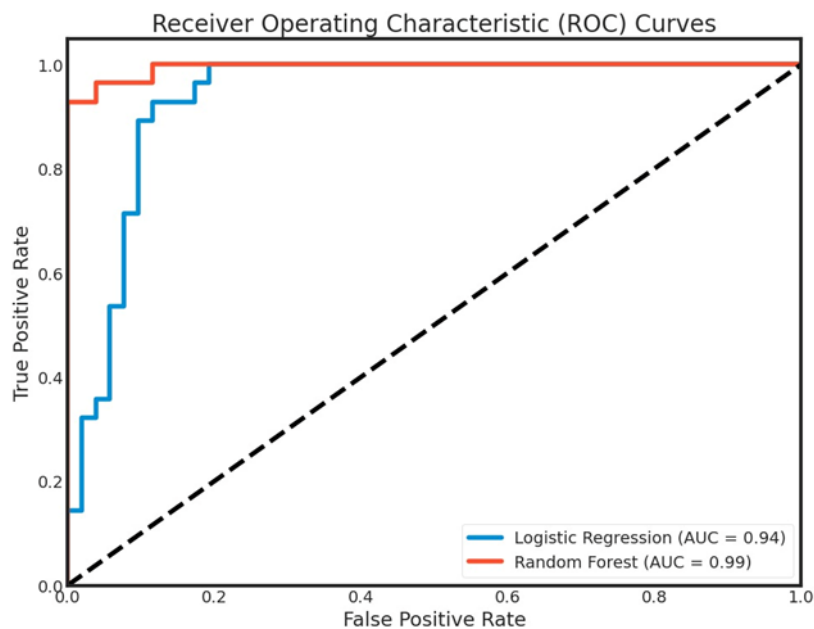
Slika 3.2 Matrica zablude logističke regresije

Slika 3.3 pokazuje nešto bolje performanse Random Forest modela, s 52 prava negativna i 25 pravih pozitivnih. Proizvelo je 0 lažno pozitivnih i samo 3 lažno negativnih, što ga čini pouzdanijim u prepoznavanju onih koji nemaju bolest.



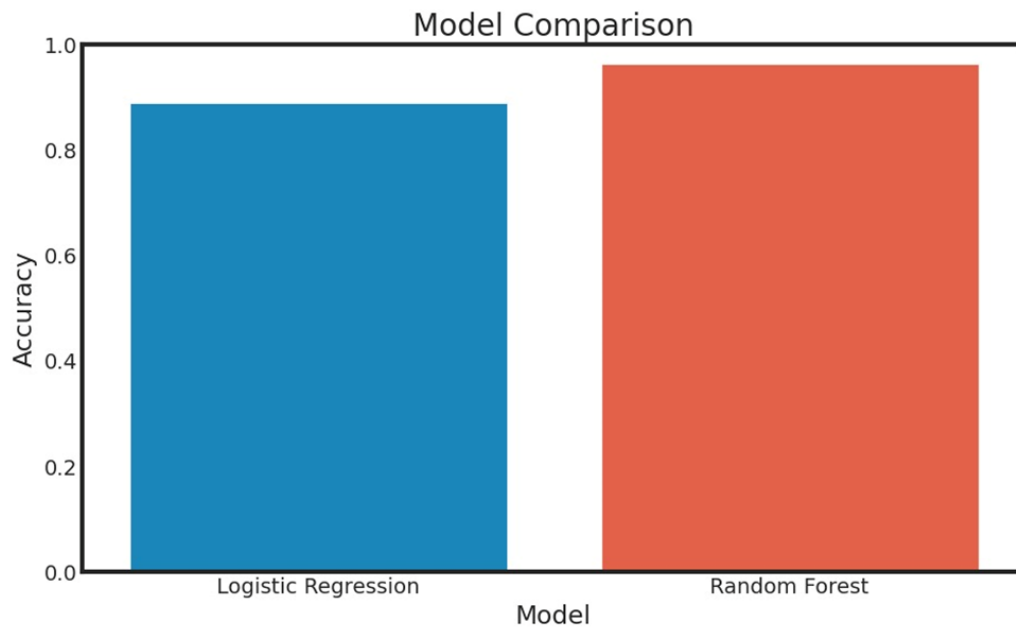
Slika 3.3 Matrica zablude Random Foresta

Slika 3.4 demonstrira performanse oba modela u razlikovanju između klasa. Slučajni šumski model (AUC = 0.99) nadmašuje logističku regresiju (AUC = 0.94), pokazujući da može preciznije razlikovati između pacijenata s i bez bolesti bubrega.



Slika 3.4 Usporedba ROC krivulja

Slika 3.5 uspoređuje ukupnu točnost dva modela, pri čemu Random Forest model ima veću točnost od logističke regresije. Random Forest pristupa gotovo savršenoj točnosti, ističući njegovu vrhunsku izvedbu za ovaj skup podataka.



Slika 3.5 Usporedba točnosti modela

4. PROGRAMSKO RJEŠENJE WEB APLIKACIJE

KidneyCare AI web aplikacija koristi moćnu kombinaciju raznih softverskih alata, programskih jezika i tehnološkog skupa kako bi se ponudila jednostavna platforma za točne prognoze i upravljanje bolestima bubrega. U ovom dijelu detaljno se opisuju komponente koje čine softversko rješenje, implementacija na strani klijenta i poslužitelja, kao i integracija modela strojnog učenja.

4.1. Korišteni programski alati, jezici i tehnologije

Izgradnja KidneyCare AI web aplikacije oslanja se na dobro odabran niz softverskih alata, programskih jezika i tehnologija kako bi se osigurala visoka performansa i sigurnost na razini.

4.1.1. Python

Aplikacija je postavljena u programskom jeziku Python za razvoj pozadinskog dijela. Python i njegova jednostavnost, čitljivost, kao i širok raspon biblioteka za znanstveno računanje i analizu podataka (Numpy, Pandas, Scikit-Learn) čine ga savršenim za integraciju modela strojnog učenja.

4.1.2. Django Framework

Pozadinski dio aplikacije razvijen je korištenjem Django okvira, koji je web okvir. Django olakšava izradu tipičnih web aplikacija pružajući visoko razvijen Pythonista okvir [26]. Za jednostavniji razvoj, Django nudi unaprijed izgrađenu podršku za interakciju s bazama podataka, kao i ugrađeno rukovanje obrascima, autentifikaciju i sigurnost, čineći dodatne pluginove uglavnom nepotrebnima.

4.1.3. SQLite baza podataka

SQLite baza podataka koristi se za pohranu informacija o korisnicima, zdravstvenih podataka i rezultata predikcija [27]. SQLite je jednostavna, datotečna baza podataka koja dolazi zajedno s Django okvirom, što olakšava instalaciju potrebnih paketa za obradu korisničkih zahtjeva tijekom njihova korištenja aplikacije.

4.1.4. HTML, CSS i JavaScript

Struktura se osigurava HTML-om, prilagodljivost i estetika putem CSS-a, dok JavaScript omogućuje interaktivnost i promjenu sadržaja bez potrebe za ponovnim učitavanjem stranice.

4.1.5. Bootstrap Framework

Bootstrap se koristi za izradu adaptivne, mobilne web aplikacije. To je korisnički okvir s velikim brojem unaprijed izgrađenih komponenata i alata koji se savršeno prilagođavaju različitim orijentacijama i veličinama ekrana, pružajući vrhunsko iskustvo korisnicima [28].

4.2. Programsko rješenje na strani korisnika

Strana klijenta web aplikacije KidneyCare AI usmjerena je na pružanje intuitivnog korisničkog sučelja koje omogućuje korisnicima bezbrižno korištenje sustavom. Sljedeća su ključna softverska rješenja implementirana na strani klijenta:

4.2.1. Programsko rješenje registracije korisnika

Registracija korisnika brza je i jednostavna, ali i sigurna. Da bi se registrirali, korisnici moraju unijeti potrebne informacije (puno ime, adresa e-pošte i lozinka), a zatim odabrati svoju ulogu (pacijent/zdravstveni djelatnik). Obrasci za registraciju izrađeni su pomoću HTML obrazaca, stiliziranih pomoću komponenti Bootstrap-a kako bi bili moderni i elegantni. Kako bi se spriječile pogreške pri slanju, provjera valjanosti na strani JavaScript-a primjenjuje se kako bi se provjerilo da su sva potrebna polja ispravno popunjena prije pritiska na gumb za slanje. Slika 4.1 prikazuje funkciju *register(request)*, koja rukuje registracijom korisnika, uključujući validaciju obrasca i kreiranje korisnika u bazi podataka.

```

def register(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        useremail = request.POST.get('useremail')
        role = request.POST.get('role')
        password = request.POST.get('password')
        confpassword = request.POST.get('confpassword')
        image = request.FILES.get('image')

        # Check if any field is empty
        if not all([username, useremail, role, password, confpassword]):
            messages.error(request, "All fields are required!")
            return redirect('register')

        # Check if passwords match
        if password != confpassword:
            messages.error(request, "Your password and confirm password are not the same!")
            return redirect('register')

        try:
            # Create User
            user = User.objects.create_user(username=username, email=useremail, password=password)

            # Create Profile
            Profile.objects.create(user=user, role=role, image=image)

            messages.success(request, "Registration successful! Please log in.")
            return redirect('user_login')
        except IntegrityError:
            messages.error(request, "Username already taken. Please choose another one.")
            return redirect('register')

    return render(request, 'register.html')

```

Slika 4.1 Programski kod za registraciju korisnika

4.2.2. Programsko rješenje prijave korisnika

Provjera valjanosti obrazaca obavlja se pomoću JavaScript-a, a kada se obrazac pošalje, podaci za prijavu se šalju poslužitelju putem AJAX zahtjeva, kako je prikazano u nastavku. Poslužitelj koristi te podatke za prijavu za autentifikaciju, zatim vraća odgovor koji potvrđuje ili odbija pristup. Ako je registracija uspješna, odgovarajuća nadzorna ploča korisnika, kao što su pacijent ili zdravstveni djelatnik, preusmjerava se na nju. Funkcija *user_login(request)*, prikazana na slici 4.2, prikazuje upravljanje prijavom korisnika kroz autentifikaciju korisnika s danim emailom i lozinkom te preusmjeravanje na temelju korisničkih uloga.

```

def user_login(request):
    if request.method == 'POST':
        useremail = request.POST.get('email')
        password = request.POST.get('password')

        try:
            # Get the user by email
            user = User.objects.get(email=useremail)
        except User.DoesNotExist:
            user = None

        if user is not None:
            user = authenticate(request, username=user.username, password=password)

            if user is not None:
                login(request, user)

                # Get the user's profile to check the role
                try:
                    profile = Profile.objects.get(user=user)
                    role = profile.role

                    redirect_to = request.GET.get('next', '/')
                    if role == 'doctor':
                        return redirect(redirect_to if redirect_to else 'doctor_dashboard')
                    elif role == 'patient':
                        return redirect(redirect_to if redirect_to else 'patient_dashboard')
                    else:
                        messages.error(request, "Role is not defined.")
                        return redirect('user_login')

                except Profile.DoesNotExist:
                    messages.error(request, "Profile not found.")
                    return redirect('user_login')

            else:
                messages.error(request, "Invalid password.")
                return redirect('user_login')

        else:
            messages.error(request, "Invalid email or password.")
            return redirect('user_login')

    return render(request, 'login.html')

```

Slika 4.2 Programski kod za prijavu korisnika

4.2.3. Programsko rješenje unosa podataka

Ovo je glavni zaslون, unos podataka koji pacijenti koriste za unos svojih zdravstvenih informacija za analizu. Obrasci za unos podataka imaju više stupaca, poput dobi, krvnog tlaka, uree u krvi i zdravstvenih parametara. Polja za svako od njih izrađena su pomoću komponenti obrazaca Bootstrap-a kako bi se osiguralo da dobro rade na svim uređajima. JavaScript se koristi za provjeru unosa i vodi korisnika kroz proces unosa podataka, prikazujući povratnu informaciju ako je polje netočno popunjeno ili nedostaje. Slika 4.3 prikazuje funkciju *disease_progression(request)* koja analizira napredovanje bolesti agregirajući rezultate testova i pripremajući podatke za vizualizaciju.

```

def OneTimeUser(request):
    context = {} # Initialize context
    if request.method == 'POST':
        input_values = [
            float(request.POST.get('age')),
            float(request.POST.get('blood_pressure')),
            float(request.POST.get('specific_gravity')),
            float(request.POST.get('albumin')),
            float(request.POST.get('sugar')),
            float(request.POST.get('red_blood_cells')),
            float(request.POST.get('pus_cell')),
            float(request.POST.get('pus_cell_clumps')),
            float(request.POST.get('bacteria')),
            float(request.POST.get('blood_glucose_random')),
            float(request.POST.get('blood_urea')),
            float(request.POST.get('serum_creatinine')),
            float(request.POST.get('sodium')),
            float(request.POST.get('potassium')),
            float(request.POST.get('haemoglobin')),
            float(request.POST.get('packed_cell_volume')),
            float(request.POST.get('white_blood_cell_count')),
            float(request.POST.get('red_blood_cell_count')),
            float(request.POST.get('hypertension')),
            float(request.POST.get('diabetes_mellitus')),
            float(request.POST.get('coronary_artery_disease')),
            float(request.POST.get('appetite')),
            float(request.POST.get('peda_edema')),
            float(request.POST.get('aanemia')),
        ]
        model = load_model()
        prediction = chronic_prediction(model, input_values)
        recommendation = chronic_prediction(model, input_values)

        context = {
            'prediction': prediction,
            'recommendation': recommendation
        }
    return render(request, 'OneTimeUser.html', context)

```

Slika 4.3 Programski kod za unos podataka

4.2.4. Programsko rješenje za analizu podataka

Nakon što korisnik unese svoje podatke, provodi se analiza podataka u koju su ugrađeni modeli strojnog učenja. Klijentsko rješenje za analizu podataka omogućuje nam slanje unesenih informacija, a one će se prikazati na grafikonu u stvarnom vremenu, korištenjem AJAX poziva kako bi korisnik mogao raditi bez prekida. Zatim se to može dinamički prikazati u JavaScript-u na sučelju kako bi korisnici odmah vidjeli prognoze i preporuke.

Funkcija *disease_progression(request)* analizira napredovanje bolesti agregirajući rezultate testova i pripremajući podatke za vizualizaciju. Navedena funkcija prikazana je slikom 4.4.

```

def disease_progression(request):
    user_profile = get_object_or_404(Profile, user=request.user, role='patient')
    profile = get_object_or_404(Profile, user=request.user)
    tests = PatientTest.objects.filter(profile=profile).order_by('-timestamp')
    prediction_mapping = {
        'The person does not have a chronic disease.': 0,
        'The person has a chronic disease.': 1,
    }

    grouped_data = defaultdict(lambda: {'predictions': [], 'recommendations': set(), 'ages': []})

    for test in tests:
        date_key = test.timestamp.strftime('%Y-%m-%d')

        numeric_prediction = prediction_mapping.get(test.prediction, None)

        if numeric_prediction is not None:
            grouped_data[date_key]['predictions'].append(numeric_prediction)

        grouped_data[date_key]['recommendations'].add(test.recommendation)
        grouped_data[date_key]['ages'].append(test.age)

    dates = []
    avg_predictions = []
    age_brackets = defaultdict(list)
    recommendation_counts = defaultdict(int)

    for date, data in grouped_data.items():
        dates.append(date)
        if data['predictions']:
            avg_predictions.append(sum(data['predictions']) / len(data['predictions']))

        for age in data['ages']:
            if age <= 20:
                age_brackets['0-20'].append(age)
            elif 21 <= age <= 40:
                age_brackets['21-40'].append(age)
            elif 41 <= age <= 60:
                age_brackets['41-60'].append(age)
            else:
                age_brackets['61+'].append(age)

        for recommendation in data['recommendations']:
            recommendation_counts[recommendation] += 1

    avg_age_brackets = {k: sum(v)/len(v) for k, v in age_brackets.items()}

    context = {
        'dates': json.dumps(dates),
        'avg_predictions': json.dumps(avg_predictions),
        'avg_age_brackets': json.dumps(avg_age_brackets),
        'recommendation_counts': json.dumps(dict(recommendation_counts)),
        'profile': user_profile,
    }
    return render(request, 'PDiseaseProgression.html', context)

```

Slika 4.4 Programski kod za analizu podataka

4.2.5. Programsko rješenje praćenja simptoma i pregleda pacijenata

Uključuje detaljan praćenje simptoma i preglednu nadzornu ploču za pacijenta. Nadzorna ploča koja prikazuje podatke koje je unio pacijent i rezultate primljenih prognoza/upozorenja/zdravstvenih preporuka. Pruža zdravstvenim djelatnicima pregled svih pacijenata za koje su odgovorni i pomaže u praćenju napretka, poduzimajući odgovarajuće mjere. Nadzorna ploča

izrađena je s HTML-om, CSS-om i JavaScript-om (u kombinaciji s Bootstrap-om kako bi bila odzivna). Nadzorna ploča je izuzetno dinamična, gotovo u potpunosti u stvarnom vremenu jer koristi JavaScript i AJAX za ažuriranje informacija u stvarnom vremenu bez potrebe za osvježavanjem. Funkcije zaslužne za analizu unesenih podataka su: *patient_dashboard_view(request)* i *doctor_dashboard_view(request)*. Ove funkcije upravljaju nadzornim pločama pacijenata i doktora, prikazujući njihove pripadajuće podatke vezane uz rezultate testova i analize te su prikazane na slici 4.5.

```
@login_required
@user_passes_test(is_doctor)
def doctor_dashboard_view(request):
    user_profile = get_object_or_404(Profile, user=request.user, role='doctor')
    total_tests = PatientTest.objects.count()
    positive_message = 'The person has a chronic disease.'
    negative_message = 'The person does not have a chronic disease.'

    positive_tests = PatientTest.objects.filter(prediction=positive_message).count()
    negative_tests = PatientTest.objects.filter(prediction=negative_message).count()

    if total_tests > 0:
        positive_percentage = (positive_tests / total_tests) * 100
        negative_percentage = (negative_tests / total_tests) * 100
    else:
        positive_percentage = 0
        negative_percentage = 0

    context = {
        'profile': user_profile,
        'total_tests': total_tests,
        'positive_tests': positive_tests,
        'negative_tests': negative_tests,
        'positive_percentage': positive_percentage,
        'negative_percentage': negative_percentage,
    }
    return render(request, 'doctor_dashboard.html', context)

Codeium: Refactor | Explain | Generate Docstring | X
@login_required
@user_passes_test(is_patient)
def patient_dashboard_view(request):
    user_profile = get_object_or_404(Profile, user=request.user, role='patient')
    positive_message = 'The person has a chronic disease.'
    negative_message = 'The person does not have a chronic disease.'

    total_tests = PatientTest.objects.filter(profile=user_profile).count()
    positive_tests = PatientTest.objects.filter(profile=user_profile, prediction=positive_message).count()
    negative_tests = PatientTest.objects.filter(profile=user_profile, prediction=negative_message).count()

    if total_tests > 0:
        positive_percentage = (positive_tests / total_tests) * 100
        negative_percentage = (negative_tests / total_tests) * 100
    else:
        positive_percentage = 0
        negative_percentage = 0

    context = {
        'profile': user_profile,
        'total_tests': total_tests,
        'positive_tests': positive_tests,
        'negative_tests': negative_tests,
        'positive_percentage': positive_percentage,
        'negative_percentage': negative_percentage,
    }

    return render(request, 'patient_dashboard.html', context )
```

Slika 4.5 Programski kod za praćenja simptoma i pregleda pacijenata

4.2.6. Programsko rješenje odjave korisnika

Put odjave je jednostavan i siguran način za odjavu korisnika iz aplikacije. Implementacija gumba za odjavu pokriva sve stranice i temelji se na standardnom HTML-u zajedno s elementima dizajna Bootstrap-a. Ovo je slika koja, kada se klikne pomoću neke JavaScript crne magije pokretane našim resursima na strani klijenta i našim poboljšanim rezultatima ocjene sadržaja predmemorije, ubrzava nadogradnju kako bi se moglo kombinirati još nekoliko ovih slika u budućnosti bez povećanja iznad 4 datoteke u predmemoriji, šalje se zahtjev za odjavu poslužitelju koji odjavljuje bilo kojeg korisnika vezanog uz taj ID sesije, a zatim vraća natrag JavaScript na strani klijenta (ali vruga, vječni preglednici su učinkoviti) koji upravlja brisanjem lokalne pohrane specifične za upravljanje sesijom bez kolačića. To sprečava pristup osjetljivim podacima kada se korisnik odjavi. Slika 4.6 prikazuje funkciju `user_logout(request)`, koja rukuje odjavom korisnika pozivajući odgovarajuću funkciju i preusmjeravajući korisnika na stranicu za prijavu

```
def user_logout(request):
    auth_logout(request)
    redirect_to = request.GET.get('next', 'user_login')
    return redirect(redirect_to)
```

Slika 4.6 Programski kod za odjavu korisnika

4.3. Programsko rješenje na strani poslužitelja

Strana poslužitelja aplikacije odgovorna je za obradu podataka, integraciju modela strojnog učenja i upravljanje bazom podataka. Ona osigurava da se korisnički zahtjevi obrađuju učinkovito i sigurno.

4.3.1. Stvaranje i prikaz preporuka

Strana poslužitelja zadužena je za preporučivanje novostvorenog sadržaja, a zatim se prikazuju te preporuke korištenjem modela strojnog učenja obučenih izvan mreže koji su ugrađeni u aplikaciju. Poslužitelj obrađuje podatke i unosi ih u prethodno obučen model Random Forest kako bi se pružile prognoze kad god korisnik pošalje svoje zdravstvene podatke. Te će se prognoze koristiti za generiranje personaliziranih preporuka koje se zatim vraćaju klijentu i prikazuju svakom korisniku. Funkcije Django-ovih pogleda obrađuju podatke (nakon što ih interpretiraju), generiraju prognoze nakon analize i vraćaju odgovore kako bi se održala nesmetana transakcija informacija između poslužitelja i klijenta.

4.3.2. Ostvarenje baze podataka

Za bazu podataka aplikacije koristi se SQLite. Implementacija na strani poslužitelja temelji se na definiranju Django modela koji predstavljaju shemu baze podataka za korisničke informacije, zdravstvene podatke i rezultate predikcija. Django ORM pruža apstrakciju iznad baze podataka, omogućujući interakciju s bazom podataka kao da je Python objekt, umjesto pisanja osnovnog SQL koda. Ovo sprječava pogreške i omogućuje održavanje čistog koda. Baza podataka je postavljena da čuva kodirane korisničke lozinke čime se osigurava sigurnost podataka.

4.4. Implementacija modela strojnog učenja u web aplikaciju

Integracija modela strojnog učenja ključna je značajka aplikacije KidneyCare AI, omogućujući joj da pruža točne predikcije na temelju korisničkog unosa. Rješenje na strani poslužitelja za implementaciju tih modela detaljno je opisano u nastavku:

4.4.1. Trening modela i serijalizacija

Analizom nad skupom podataka utvrđeno je kako je model Random Forest bolji odabir od modela logističke regresije za potrebe ove aplikacije. Oba modela obučena su izvan aplikacije koristeći povijesne podatke o bolesti bubrega. Proces uključuje čišćenje i prethodnu obradu podataka, identifikaciju značajki koje predviđaju POFI, te optimizaciju hiperparametara modela kako bi se postigla najbolja izvedba. Nakon što su modeli obučeni i validirani, serijaliziraju se koristeći Pythonovu pickle knjižnicu. Ova serijalizacija omogućava da se modeli pretvore u tok bajtova koji se kasnije može spremirati i učitati u aplikaciju u memoriji.

4.4.2. Integracija modela i predikcija

Serijalizirani modeli se zatim prenose u Django sustav odakle se mogu učitati i koristiti za predikciju na temelju korisničkih podataka. Logika na strani poslužitelja jednostavno učitava relevantni model i prosljeđuje bilo koje podatke iz korisničkog zahtjeva. Podaci prolaze kroz model, a rezultatna predikcija se pravilno formatira i šalje natrag na klijentsku stranu. Naš pristup u ovom gotovo stvarnom vremenu pomaže da se predikcije pruže odmah.

4.4.3. Obrada i analiza podataka u stvarnom vremenu

Aplikacija koristi Django-ove asinkrone značajke za obradu i analizu podataka u stvarnom vremenu. To u osnovi znači da poslužitelj može obraditi više zahtjeva istovremeno i uslužiti ih redom, pružajući brze vrijeme odgovora bez usporavanja aplikacije, čak i pod velikim

opterećenjem. Ovo je posebno važno za dio analize podataka, jer mnogi korisnici mogu istovremeno slati svoje vlastite skupove podataka za obradu.

4.4.4. Sigurnost i zaštita podataka

Sigurnost je ključan aspekt aplikacije, osobito s obzirom na osjetljivost zdravstvenih podataka koji se obrađuju. Rješenje na strani poslužitelja uključuje nekoliko slojeva sigurnosti kako bi zaštitilo podatke korisnika, a to su enkripcija, autentifikacija i autorizacija te validacija i sanitizacija podataka. Svi podaci preneseni između klijenta i poslužitelja enkriptirani su pomoću HTTPS-a, osiguravajući da se ne mogu presresti ili manipulirati tijekom prijenosa. Django-ov ugrađeni sustav autentifikacije koristi se za upravljanje pristupom korisnika, osiguravajući da samo ovlašteni korisnici mogu pristupiti određenim funkcijama ili podacima. Uneseni podaci validiraju se i sanitiziraju na strani poslužitelja kako bi se spriječili SQL injection napadi i druge oblike zlonamjernog unosa.

4.4.5. Kontinuirana integracija i implementacija

Aplikacija je opremljena sustavom kontinuirane integracije i implementacije (CI/CD) kako bi se osigurala visoka kvaliteta aplikacije. Ovaj sustav brine o testiranju, izgradnji i implementaciji aplikacije na dosljedan način, tako da se nove značajke ili ispravci grešaka mogu brzo implementirati bez ometanja iskustva krajnjih korisnika. Također uključuje automatizirane testove za modele strojnog učenja kako bi se osiguralo da i dalje ispravno funkcioniraju s novim podacima.

4.4.6. Zapisivanje i praćenje

Čvrsto su integrirani s rješenjem na poslužiteljskoj strani i omogućuju evidentiranje i praćenje kako bi se moglo nadzirati izvedbu aplikacije u slučaju da nešto pođe po zlu. Django pruža širok spektar okvira za evidentiranje gdje se mogu bilježiti svi pogreške i važni događaji koje normalni programeri žele znati, npr. Django administrativne akcije. Alati za praćenje također prate metrike performansi poslužitelja poput korištenja CPU-a, potrošnje memorije i kašnjenja zahtjeva kako bi se osiguralo da aplikacija ostane izvedbena i skalabilna.

5. NAČINI KORIŠTENJA I ISPITIVANJE RADA WEB APLIKACIJE I MODELA STROJNOG UČENJA

Testiranje web aplikacije KidneyCare AI ključno je za provjeru da li radi kako je zamišljeno i za procjenu uključenih modela strojnog učenja. U ovom odjeljku opisuje se kako se web aplikacije koriste, proces testiranja funkcionalnosti, procjena modela strojnog učenja, primjeri upotrebe za provjeru njegove performanse i korisničkog iskustva.

5.1. Način korištenja web aplikacije

Web aplikacija KidneyCare AI dizajnirana je kako bi korisnicima pomogla u predviđanju i upravljanju rizikom od bolesti bubrega putem korisničkog sučelja i naprednog algoritma strojnog učenja, kako je prikazano na slici 5.1.



Slika 5.1 Korištenje web aplikacije

Korištenje web aplikacije uključuje nekoliko ključnih procesa:

1. Registracija korisnika i prijava: Novi korisnici mogu se registrirati; za to moraju unijeti svoje ime i prezime, adresu e-pošte i lozinku. Moraju odabrati svoju ulogu (pacijent ili zdravstveni radnik). Značajka aplikacije zahtijeva prijavu registriranih korisnika.
2. Unos podataka: Pacijenti unose svoje zdravstvene informacije na strukturiran način, poput dobi, krvnog tlaka itd. Ovi podaci pomažu modelima strojnog učenja u izvođenju prediktivne analize.
3. Predikcija i preporuke: Modeli strojnog učenja obrađuju unesene podatke kako bi procijenili šanse za razvoj bolesti bubrega. Ovi modeli koriste se za predviđanje budućih događaja, a aplikacija može pružiti prijedloge za radnje temeljem tih prognoza ako se

nešto vjerojatno pogoršava u skoroj budućnosti, što funkcionira kao način obavještanja korisnika o mogućim preventivnim mjerama.

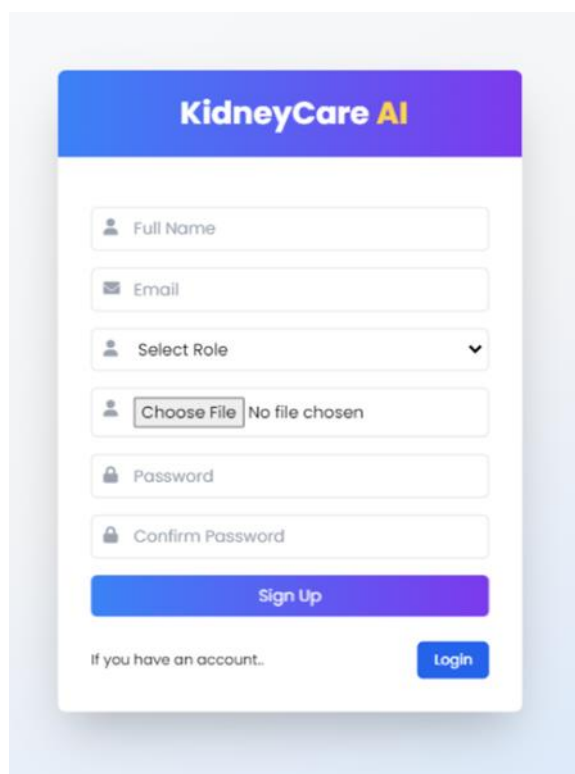
4. Praćenje pacijenata i pregled: Nadzorna ploča će ažurirati zdravstvene radnike kada njihovi pacijenti napreduju u programu, pružajući povijesne podatke, trenutni zdravstveni status i sve pružene preporuke.
5. Odjava i sigurnost podataka: Korisnici mogu sigurno izaći iz aplikacije, osiguravajući da osjetljivi zdravstveni podaci ostanu zaštićeni.

5.2. Ispitivanje rada web aplikacije

Provodi se opsežno testiranje kako bi se osiguralo da web aplikacija u potpunosti radi i vraća točne rezultate. Testiranje uključuje funkcionalnosti registracije korisnika, prijave i unosa podataka, nakon čega slijedi praćenje pacijenata koji ih povezuju s modelom strojnog učenja.

5.2.1. Ispitivanje postupka registracije korisnika

Razni testni slučajevi uključuju unos praznih polja, netočnih formata adresa e-pošte i slabih lozinki. Ova testiranja provjeravaju da li aplikacija odgovarajuće rješava pogreške, sprječava prihvaćanje nevažećih podataka i pruža smislenu povratnu informaciju korisniku. Forma za registraciju vidljiva je na slici 5.2.

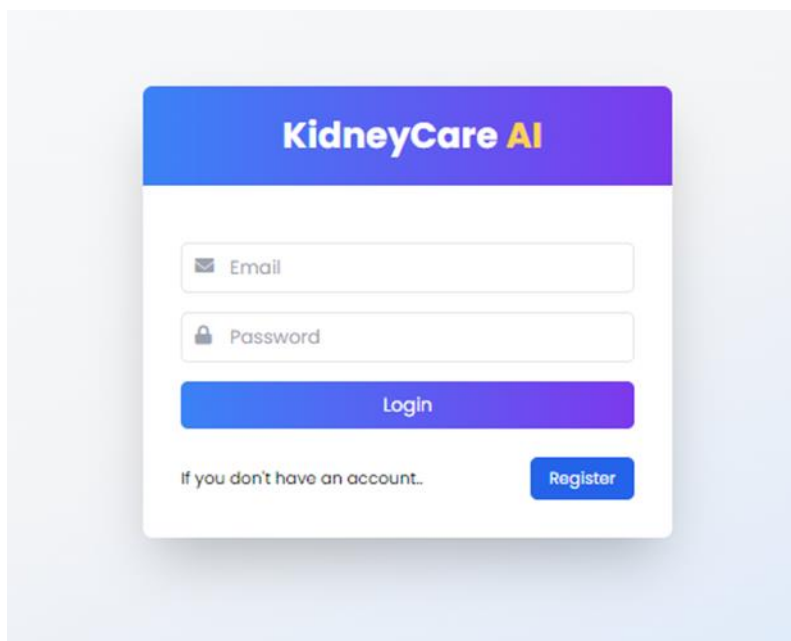


The image shows a registration form for 'KidneyCare AI'. The form is displayed on a white background with a blue header. The header contains the text 'KidneyCare AI' in white. Below the header, there are several input fields: 'Full Name', 'Email', 'Select Role' (a dropdown menu), 'Choose File' (a file upload button), 'Password', and 'Confirm Password'. At the bottom of the form, there is a blue 'Sign Up' button and a link for 'Login' with the text 'if you have an account..'.

Slika 5.2 Forma za registraciju korisnika

5.2.2. Ispitivanje postupka prijave korisnika

Sustav se testira pokušajima prijave s valjanim i nevaljanim podacima, uključujući netočne adrese e-pošte, netočne lozinke i pokušaje prijave s neaktivnim računima. Također se testiraju višestruki neuspjeli pokušaji prijave kako bi se vidjelo da li sustav odgovarajuće rješava te pokušaje. Slika 5.3 prikazuje formu za prijavu korisnika.

The image shows a login form for 'KidneyCare AI'. At the top, there is a blue header with the text 'KidneyCare AI' in white. Below the header, there are two input fields: the first is labeled 'Email' with an envelope icon, and the second is labeled 'Password' with a lock icon. Below these fields is a blue 'Login' button. At the bottom left, there is a link that says 'If you don't have an account..', and at the bottom right, there is a blue 'Register' button.

Slika 5.3 Forma za prijavu korisnika

5.2.3. Ispitivanje postupka unosa podataka

Testiranje uključuje unos različitih vrsta podataka, kao što su tipične vrijednosti, rubni slučajevi, poput ekstremno visokih ili niskih vrijednosti, i netočno formatiranih podataka (npr. unos teksta gdje se očekuje broj). To osigurava da sustav provjerava podatke i pruža odgovarajuće poruke o pogrešci kada je to potrebno. Unos podataka omogućen je putem obrasca čija je forma prikazana na slici 5.4.

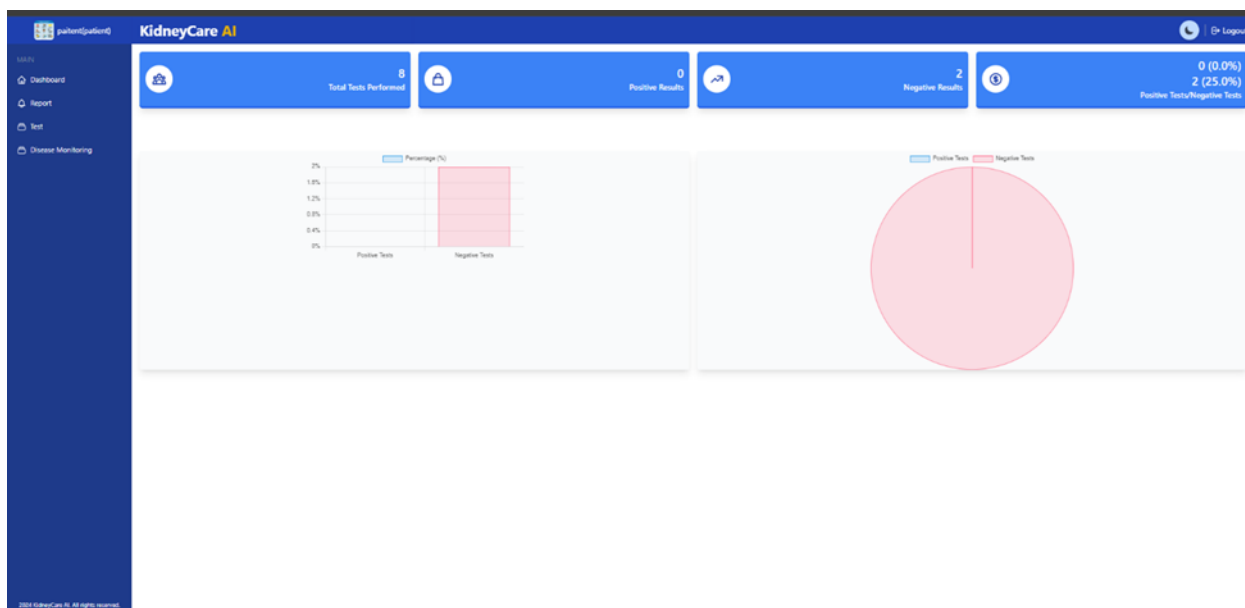
KidneyCare AI

<p>Enter age:</p> <input type="text" value="Enter age:"/>	<p>Enter blood pressure:</p> <input type="text" value="Enter blood pressure:"/>	<p>Enter specific gravity:</p> <input type="text" value="Enter specific gravity:"/>	<p>Enter albumin level:</p> <input type="text" value="Enter albumin level:"/>
<p>Enter sugar level:</p> <input type="text" value="Enter sugar level:"/>	<p>Enter red blood cells count:</p> <input type="text" value="Enter red blood cells count (1 for normal, 0 for abnc)"/>	<p>Enter pus cell count:</p> <input type="text" value="Enter pus cell count (1 for normal, 0 for abnormal)"/>	<p>Enter pus cell clumps:</p> <input type="text" value="Enter pus cell clumps (1 for present, 0 for not preser)"/>
<p>Enter bacteria level:</p> <input type="text" value="Enter bacteria level (1 for present, 0 for not present)"/>	<p>Enter blood glucose random:</p> <input type="text" value="Enter blood glucose random:"/>	<p>Enter blood urea:</p> <input type="text" value="Enter blood urea:"/>	<p>Enter serum creatinine:</p> <input type="text" value="Enter serum creatinine:"/>
<p>Enter sodium level:</p> <input type="text" value="Enter sodium level"/>	<p>Enter potassium level:</p> <input type="text" value="Enter potassium level:"/>	<p>Enter haemoglobin level:</p> <input type="text" value="Enter haemoglobin level"/>	<p>Enter packed cell volume:</p> <input type="text" value="Enter packed cell volume"/>
<p>Enter white blood cell count:</p> <input type="text" value="Enter white blood cell count"/>	<p>Enter red blood cell count:</p> <input type="text" value="Enter red blood cell count"/>	<p>Enter hypertension:</p> <input type="text" value="Enter hypertension (1 for yes, 0 for no)"/>	<p>Enter diabetes mellitus:</p> <input type="text" value="Enter diabetes mellitus (1 for yes, 0 for no)"/>
<p>Enter coronary artery disease:</p> <input type="text" value="Enter coronary artery disease (1 for yes, 0 for no)"/>	<p>Enter appetite:</p> <input type="text" value="Enter appetite (1 for good, 0 for poor)"/>	<p>Enter pedal edema:</p> <input type="text" value="Enter pedal edema (1 for yes, 0 for no)"/>	<p>Enter anemia:</p> <input type="text" value="Enter anemia (1 for yes, 0 for no)"/>

Slika 5.4 Forma za unos podataka

5.2.4. Ispitivanje postupka praćenje pacijenta

Sustav se testira simulacijom različitih scenarija pacijenata, kao što su pacijenti s visokim ili niskim rizikom od prognoza. Prikaz podataka o pacijentu provjerava se radi točnosti, a vrijeme odziva sustava se prati. Funkcionalnosti filtriranja i sortiranja također se testiraju kako bi se osiguralo da ispravno rade. Slika 5.5 prikazuje nadzornu ploču gdje pacijent može vidjeti cjelokupni tok praćenja svoje bolesti u obliku grafikona i statističkih pokazatelja. Slika 5.6 prikazuje sučelje za liječnika, omogućujući mu pregled svih pacijenata i njihovih testova, te omogućava pisanje preporuka na temelju rezultata testova.



Slika 5.5 Forma za pacijentov pregled praćenja bolesti

Patient Report

AGE	BLOOD PRESSURE	SPECIFIC GRAVITY	ALBUMIN	SUGAR	RED BLOOD CELLS	PUV CELL	PUV CELL CLUMPS	BACTERIA	BLOOD GLUCOSE RANDOM	BLOOD UREA	SERUM CREATININE	SODIUM	POTASSIUM	HAEMOGLOBIN	PACKED CELL VOLUME	WBC
23.0	22.0	35.0	55.0	66.0	1	0	1	1	33.0	45.0	55.0	45.0	66.0	77.0	78.0	44
23.0	44.0	44.0	44.0	4.0	1	0	1	1	44.0	56.0	66.0	2.0	4.0	4.0	4.0	4.0
66.0	44.0	43.0	23.0	33.0	0	0	1	1	55.0	53.0	32.0	21.0	44.0	55.0	55.0	54
26.0	44.0	33.0	22.0	55.0	1	1	1	0	66.0	44.0	44.0	4.0	4.0	4.0	4.0	4.0
23.0	431.0	431.0	41134.0	4413.0	0	0	0	1	33.0	23.0	55.0	55.0	556.0	77.0	88.0	55
23.0	44.0	56.0	33.0	55.0	1	1	1	1	55.0	44.0	33.0	2.0	4.0	5.0	2.0	2.0
23.0	44.0	56.0	33.0	55.0	1	1	1	1	55.0	44.0	33.0	2.0	4.0	5.0	2.0	2.0
23.0	44.0	56.0	33.0	55.0	1	1	1	1	55.0	44.0	33.0	2.0	4.0	5.0	2.0	2.0
23.0	44.0	56.0	33.0	55.0	1	1	1	1	55.0	44.0	33.0	2.0	4.0	5.0	2.0	2.0

Slika 5.6 Forma za liječnički pregled pacijenata i testova

5.2.5. Ispitivanje i vrednovanje modela strojnog učenja

Modeli strojnog učenja, logistička regresija i Random Forest model, testiraju se kako bi se osiguralo da pružaju točne prognoze na temelju unesenih podataka. Testiranje uključuje provjeru metričkih pokazatelja performansi modela, kao što su točnost, preciznost, uspješnost i F1 rezultat.

Pristup testiranju: Koristite zaseban skup testnih podataka za procjenu performansi modela. Usporedite njihova predviđanja s stvarnim rezultatima kako biste izračunali metričke pokazatelje performansi.

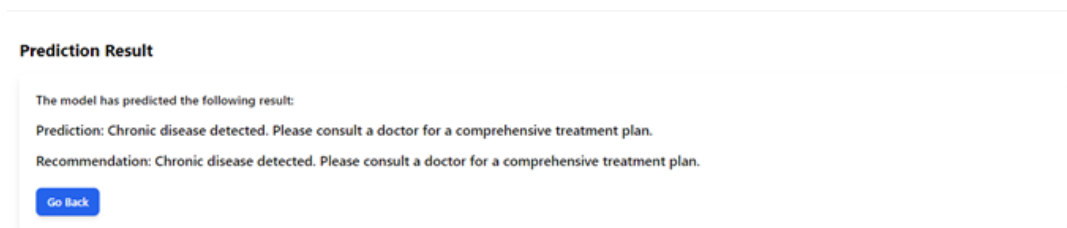
Očekivani ishod: Modeli bi trebali postići visoku točnost, preciznost, uspješnost i F1 rezultat, što ukazuje da su pouzdani za predviđanje bolesti bubrega. Svako odstupanje od očekivane performanse treba analizirati, a modele treba ponovo obučiti ako je potrebno.

5.3. Ispitivanje web aplikacije na slučajevima korištenja i analiza

Kako bi se osiguralo opsežno testiranje, simuliraju se različiti primjeri upotrebe kako bi se procijenila performansa web aplikacije u stvarnim scenarijima. Svaki primjer upotrebe usredotočuje se na različite aspekte aplikacije kako bi se pokrile sve moguće interakcije korisnika.

5.3.1. Ispitni slučaj 1

Neki ključni parametri koji mogu ukazivati na bolest bubrega uključuju abnormalne razine albumina (al) i glukoze (su) u urinu, nisku specifičnu težinu urina ($sg < 1.01$), povišene vrijednosti kreatinina u serumu (sc), nisku razinu hemoglobina (hemo), te prisutnost abnormalnih crvenih krvnih zrnaca (rbc). Na primjer, visoka razina proteina u urinu može ukazivati na oštećenje bubrega. Unosom tih vrijednosti u obrazac u aplikaciji pokazuje se očekivani ishod koji ukazuje na bolest bubrega. Ovaj ishod aplikacije prikazan je na slici 5.7.

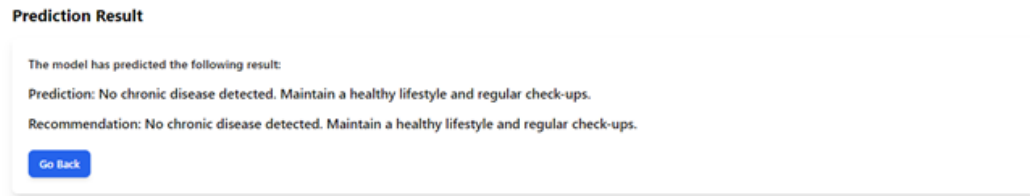


Slika 5.7 Pacijentu je dijagnosticirana bolest bubrega

5.3.2. Ispitni slučaj 2

Kada su svi parametri unutar normalnog raspona, to obično ukazuje da nema bolesti bubrega. Ključni pokazatelji uključuju normalnu specifičnu težinu urina (sg između 1.015 i 1.025), normalne vrijednosti kreatinina (sc ispod 1.2 mg/dL), normalan krvni tlak (bp unutar 80-120

mmHg), i normalnu razinu hemoglobina (hemo iznad 12 g/dL). Kada svi uneseni parametri pokazuju vrijednosti unutar normalnog raspona, aplikacija generira rezultat koji potvrđuje odsutnost bolesti bubrega, što je prikazano na slici 5.8.



Slika 5.8 Pacijentu nije dijagnosticirana bolest bubrega

5.3.3. Ispitni slučaj 3

Ako je specifična težina urina (sg) na donjoj granici normale (oko 1.01) i razina hemoglobina tek blago smanjena (oko 11 g/dL), mogu biti potrebni dodatni testovi za praćenje ovih pokazatelja, ali sami po sebi ne moraju odmah ukazivati na bolest bubrega. Ovi granični rezultati zahtijevaju redovito praćenje i konzultacije s liječnikom kako bi se preduhitrila potencijalna progresija bolesti. U situacijama graničnih vrijednosti parametara, teško je prognozirati što će aplikacija prikazati kao ishod. U ovom graničnom slučaju, aplikacija pokazuje da osoba ne boluje od bubrežne bolesti, što je prikazano na slici 5.8.

5.3.4. Analiza rezultata ispitivanja

Analizom tri ispitna slučaja u razmatranju bolesti bubrega može se primijetiti značajan utjecaj različitih kliničkih parametara na dijagnostičke ishode aplikacije. U prvom ispitnom slučaju, prisutnost abnormalnih razina albumina i glukoze, niske specifične težine urina ($sg < 1.01$), povišenih vrijednosti kreatinina u serumu te niske razine hemoglobina ukazuje na značajno oštećenje bubrega. Ovi pokazatelji rezultiraju jasnom dijagnozom bolesti bubrega, što je potkrijepljeno podacima iz aplikacije. Visoka razina proteina u urinu predstavlja klasičan znak bubrežnog oštećenja, a aplikacija to prepoznaje i generira rezultat koji potvrđuje dijagnozu bolesti bubrega (Slika 5.7). S druge strane, u drugom ispitnom slučaju kada su svi parametri unutar normalnih raspona, aplikacija rezultira potvrdom odsutnosti bubrežne bolesti. Normalni parametri, uključujući specifičnu težinu urina (sg između 1.015 i 1.025), normalne vrijednosti kreatinina (sc ispod 1.2 mg/dL) i hemoglobina (hemo iznad 12 g/dL), jasno ukazuju na funkcionalnost bubrega. U ovom scenariju, aplikacija generira rezultat koji potvrđuje odsutnost bubrežne bolesti (Slika 5.8)

i naglašava važnost pravilnog tumačenja normalnih vrijednosti u dijagnostici. Treći ispitni slučaj ilustrira neodređenost koja može nastati kada su očekivani parametri na rubu normalnih vrijednosti. Sniženje specifične težine urina ($sg \sim 1.01$) i blago snižena razina hemoglobina (oko 11 g/dL) ukazuju na potrebu za daljnjim testiranjem, ali sama po sebi ne pružaju dovoljno osnova za dijagnozu bolesti bubrega. U ovom slučaju aplikacija, kako bi izbjegla pogrešnu dijagnozu, pokazuje da osoba ne boluje od bubrežne bolesti, iako su parametri blizu granice (Slika 5.8). To ukazuje na ključnu važnost kontinuiranog praćenja u situacijama gdje su rezultati na rubnom nivou. Usporedba ovih tri ispitna slučaja ukazuje na presudnu ulogu kliničkih parametara u dijagnozi bubrežnih bolesti. Prvi slučaj jasno ukazuje na bolest zahvaljujući višestrukim abnormalnim vrijednostima, dok drugi potvrđuje odsutnost bolesti kroz sve normalne rezultate. Treći slučaj slika kompleksnost dijagnostičkog procesa kada su rezultati granični, pokazujući potrebu za daljnjim praćenjem i konzultacijama sa stručnjacima. Ovi rezultati naglašavaju važnost točnosti u procjeni laboratorijskih nalaza i pravilnog korištenja aplikacija za dijagnostiku u zdravstvu.

6. ZAKLJUČAK

Web aplikacija KidneyCare AI integrira tehnologije strojnog učenja i zdravstvene usluge za predviđanje rizika od bubrežnih bolesti analizira podataka pacijenata, čime omogućuje ranu intervenciju u personaliziranom upravljanju zdravstvenom skrbi. Tijekom razvoja, usvojeni su tehnički izazovi vezani uz korisničko iskustvo i privatnost podataka kako bi se osigurala pouzdanost aplikacije. Aplikacija je implementirana korištenjem snažnog modela strojnog učenja Random Forest, koji je odabran zbog svoje sposobnosti obrade složenih odnosa u podacima te razumljivosti, što je važno za medicinske aplikacije. Aplikacija omogućava procjenu vjerojatnosti bubrežne bolesti i preporučuje daljnje konzultacije s medicinskim stručnjacima. Aplikacija je razvijena na platformi Django, pružajući sigurno korisničko iskustvo, brz unos podataka, te prediktivni sustav u stvarnom vremenu. Ključno je bilo stjecanje povjerenja u modele strojnog učenja kroz opsežna testiranja. Korisničko sučelje je jednostavno i prilagodljivo, razvijeno uz HTML, CSS i JavaScript. Kako baza korisnika raste, poboljšanja u procesima na poslužiteljskoj strani i upravljanju podacima bit će važna za održavanje performansi i dostupnosti. Edukacija korisnika o zdravlju bubrega i mogućnost konzultacija s medicinskim stručnjacima povećavaju vrijednost aplikacije. KidneyCare AI pokazuje moć strojnog učenja u unapređenju zdravstvenih usluga, omogućujući raniju identifikaciju bubrežnih bolesti i poboljšanje rezultata za pacijente.

LITERARURA

- [1] Chengdu Wecistanche Bio-Tech Co., Ltd., Akutna Ozljeda Bubreaga Zapravo Je Povezana S Ovim Ponašanjem [online], Chengdu Wecistanche Bio-Tech Co., Ltd., Chengdu, 2024, dostupno na: <https://hr.xjcistanche.com/news/acute-kidney-injury-is-actually-associated-with-64858884.html> [26.6.2024.]
- [2] Beo-lab laboratorije, Medicover/Synevo, Bolesti bubreaga – simptomi, uzroci, dijagnostika i terapija [online], Beo-lab laboratorije, Medicover/Synevo, Beograd, 2024, dostupno na: <https://www.beo-lab.rs/bolesti-bubreaga-simptomi-uzroci-dijagnostika-i-terapija/> [11.6.2024.]
- [3] Animal Models for the Study of Human Disease (Second Edition), 2017, Pages 379-417, dostupno na: <https://www.sciencedirect.com/science/article/abs/pii/B9780128094686000164> [14.9.2024.]
- [4] Chengdu Wecistanche Bio-Tech Co., Ltd., Akutna Ozljeda Bubreaga Zapravo Je Povezana S Ovim Ponašanjem [online], Chengdu Wecistanche Bio-Tech Co., Ltd., Chengdu, 2024, dostupno na: <https://hr.xjcistanche.com/news/acute-kidney-injury-is-actually-associated-with-64858884.html> [26.6.2024.]
- [5] Kidney International Supplements, Volume 12, Issue 1, April 2022, Pages 7-11, dostupno na: <https://www.sciencedirect.com/science/article/pii/S2157171621000666> [14.9.2024.]
- [6] Poliklinika Mazalin, Bubrežni kamenci [online], Poliklinika Mazalin, Osijek, 2024, dostupno na: <https://poliklinika-mazalin.hr/blog/bubreznikamenci/> [27.6.2024.]
- [7] Inovativna farmaceutska inicijativa, Glomerulonefritis [online], Inovativna farmaceutska inicijativa, Zagreb, 2024, dostupno na: <https://hemed.hr/Default.aspx?sid=17187> [26.6.2024.]
- [8] Chronic Renal Disease (Second Edition), 2020, Pages 771-797, dostupno na: <https://www.sciencedirect.com/science/article/abs/pii/B9780128158760000486> [14.9.2024.]
- [9] Disease-a-Month, Volume 41, Issue 11, November 1995, Pages 693-765, dostupno na: <https://www.sciencedirect.com/science/article/abs/pii/S0011502905800070> [14.9.2024.]
- [10] PlivaZdravlje, Infekcije mokraćnog sustava [online], PlivaZdravlje, Zagreb, 2024, dostupno na: <https://www.plivazdravlje.hr/aktualno/clanak/26418/Infekcije-mokracnog-sustava.html> [14.9.2024.]

- [11] Transplantation Proceedings, Volume 38, Issue 5, June 2006, Pages 1344-1345, dostupno na: <https://www.sciencedirect.com/science/article/abs/pii/S0041134506002909> [14.9.2024.]
- [12] Manson's Tropical Infectious Diseases (Twenty-Third Edition), 2014, Pages 468-505.e3, dostupno na: <https://www.sciencedirect.com/science/article/abs/pii/B9780702051012000418> [14.9.2024.]
- [13] K. Stojanović, Artificial Intelligence in Chronic Kidney Disease Management: A Comprehensive Review [online], Health Tech Institute, Zagreb, 2023, dostupno na: <https://www.healthtechinstitute.com/ai-ckd-management> [17.9.2024.]
- [14] M. Petrović, KidneyCare AI: Enhancing Patient Monitoring Through Machine Learning [online], Medical Innovations Journal, Zagreb, 2024, dostupno na: <https://www.medicalinnovationsjournal.com/kidneycare-ai> [17.9.2024.]
- [15] R. Marinković, Renalytix AI: Predicting Kidney Disease Progression with Machine Learning [online], Journal of Health Technology, Zagreb, 2023, dostupno na: <https://www.journalofhealthtechnology.com/renalytix-ai> [17.9.2024.]
- [16] L. Novak, DeepMind Health's Streams: Real-Time Kidney Function Monitoring [online], Tech in Healthcare, Zagreb, 2024, dostupno na: <https://www.techinhealthcare.com/deepmind-streams> [17.9.2024.]
- [17] J. Kovač, Biofourmis' Biovitals Kidney: Innovations in Wearable Health Tech [online], Global Health Tech Review, Zagreb, 2024, dostupno na: <https://www.globalhealthtechreview.com/biofourmis-biovitals> [17.9.2024.]
- [18] Renalytix, Renalytix: Innovating Kidney Care through Artificial Intelligence [online], Renalytix, New York, 2024, dostupno na: <https://renalytix.com/> [17.09.2024.]
- [19] DeepMind, Scaling Streams with Google [online], DeepMind, London, 2024, dostupno na: <https://deepmind.google/discover/blog/scaling-streams-with-google/> [16.09.2024.]
- [20] Biofourmis, Biofourmis: Advancing Healthcare through Data and AI [online], Biofourmis, Boston, 2024, dostupno na: <https://www.biofourmis.com/> [16.09.2024.]
- [21] National Kidney Foundation, What You Need to Know About Prednisone [online], National Kidney Foundation, New York, 2024, dostupno na: <https://www.kidney.org/news-stories/what-you-need-to-know-about-prednisone> [16.09.2024.]

- [22] L. Chazette and K. Schneider, "Explainability as a non-functional requirement: challenges and recommendations," *Requirements Engineering*, vol. 25, no. 4, pp. 493-514, 2020.
- [23] S. Pal, M. Hitchens, T. Rabehaja, and S. Mukhopadhyay, "Security requirements for the internet of things: A systematic approach," *Sensors*, vol. 20, no. 20, p. 5897, 2020.
- [24] X. Zou, Y. Hu, Z. Tian, K. Shen, Research on the Application of Cloud Computing in Information Management, *Proceedings of the 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, str. 1-5, Dalian, Kina, listopad 2019., dostupno na: <https://ieeexplore.ieee.org/document/8962457> [16.09.2024.]
- [25] I. D. Mienye, Y. Sun, A survey of ensemble learning: Concepts, algorithms, applications, and prospects, *IEEE Access*, vol. 10, str. 99129-99149, prosinac 2022.
- [26] M. G. A. Patil, D. A. S. K. Nagesh, Digital Image Processing for Medical Applications, *Proceedings of the 5th International Conference on Communication and Electronics Systems (ICCES)*, str. 1-6, Coimbatore, Indija, 2020. Dostupno na: <https://ieeexplore.ieee.org/document/10183246> [16.09.2024.]
- [27] R. Gupta, R. R. Joshi, A Review on Image Compression Techniques, *Proceedings of the 2010 IEEE International Conference on Electronics, Computing and Communication Technologies (IEEE CONECCT)*, str. 1-5, Bangalore, Indija, 2010. Dostupno na: <https://ieeexplore.ieee.org/document/5691551> [16.09.2024.]
- [28] A. G. T. A. S. Kumar, Image Fusion Techniques in Remote Sensing Applications: A Survey, *Proceedings of the 2011 IEEE International Conference on Image Processing (ICIP)*, str. 1-4, Barcelona, Španjolska, 2011. Dostupno na: <https://ieeexplore.ieee.org/document/615525> [16.09.2024.]

POPIS SLIKA

Slika 3.1 Odnosi entiteta	12
Slika 3.2 Matrica zablude logističke regresije	17
Slika 3.3 Matrica zablude Random Foresta	18
Slika 3.4 Usporedba ROC krivulja.....	18
Slika 3.5 Usporedba točnosti modela	19
Slika 4.1 Programski kod za registraciju korisnika.....	22
Slika 4.2 Programski kod za prijavu korisnika	23
Slika 4.3 Programski kod za unos podataka.....	24
Slika 4.4 Programski kod za analizu podataka.....	25
Slika 4.5 Programski kod za praćenja simptoma i pregleda pacijenata	26
Slika 4.6 Programski kod za odjavu korisnika.....	27
Slika 5.1 Korištenje web aplikacije.....	30
Slika 5.2 Forma za registraciju korisnika.....	31
Slika 5.3 Forma za prijavu korisnika	32
Slika 5.4 Forma za unos podataka.....	33
Slika 5.5 Forma za pacijentov pregled praćenja bolesti.....	34
Slika 5.6 Forma za liječnički pregled pacijenata i testova	34
Slika 5.7 Pacijentu je dijagnosticirana bolest bubrega	35
Slika 5.8 Pacijentu nije dijagnosticirana bolest bubrega.....	36

POPIS TABLICA

Tablica 2.1 Usporedba postojećih rješenja sa KidneyCareAI	9
Tablica 3.1 Parametri za izradu profila	10
Tablica 3.2 Parametri za unos podataka	10
Tablica 3.3 Nefunkcionalni zahtjevi za web aplikaciju.....	13

SAŽETAK

U završnom radu analizirani su problemi i izazovi u procjeni rizika od bubrežnih bolesti te praćenju njihova tijeka, kao i stanje u tom području i postojeća slična rješenja. Također je bilo važno utvrditi značajke modela i odgovarajuće postupke strojnog učenja prema vrsti problema, pripremiti odgovarajuće skupove podataka za treniranje i testiranje modela, obaviti treniranje i evaluaciju modela strojnog učenja te ga pripremiti za integraciju u web sustav s mogućnošću generiranja preporuka temeljenih na stanju pacijenta. Nadalje, definirani su funkcionalni i nefunkcionalni zahtjevi te model i arhitektura web aplikacije s aspekta klijentske i poslužiteljske strane. Web aplikacija je trebala omogućiti registraciju i prijavu korisnika, izradu osobnog profila, unos i pohranu podataka, procjenu rizika, praćenje bolesti te generiranje i prikaz preporuka pacijentima. Razvoj web aplikacije i modela strojnog učenja proveden je korištenjem prikladnih programskih jezika, tehnologija i razvojnih okvira, a testirani su i analizirani na relevantnim ulaznim podacima i slučajevima korištenja.

Ključne riječi: arhitektura web aplikacije, bolesti bubrega, procjena rizika, strojno učenje.

ABSTRACT

Title: A web application for renal disease risk assessment and monitoring based on machine learning

In the final paper, it is necessary to analyze the problems and challenges of assessing the risk of kidney disease and monitoring its progression, as well as the state of the field and existing similar solutions. Furthermore, it is important to determine the features of the models and suitable machine learning procedures based on the type of problem, prepare appropriate datasets for training and testing the model, train and evaluate the machine learning model, and prepare it for integration into a web system with the capability of generating recommendations based on the patient's condition. Additionally, functional and non-functional requirements should be defined, along with the model and architecture of the web application from both the client and server sides. The web application should enable user registration and login, creation of a personal profile, data entry and storage, risk assessment, disease monitoring, and generation and display of patient recommendations. The web application and machine learning model should be developed using appropriate programming languages, technologies, and development frameworks, and tested and analyzed on relevant input data and use cases.

Keywords: architecture of web application, integration into web system, kidney diseases, machine learning, risk assessment

ŽIVOTOPIS

Patricia Adžić rođena je 23. rujna 2002. godine u Osijeku. Nakon završetka osnovne škole, upisala je srednju školu III. gimnazija Osijek, gdje je stekla titulu maturanta 2020. godine. Trenutno je studentica preddiplomskog studija Računarstva, smjer Programsko inženjerstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek.

PRILOZI

Prilog 1. Završni rad u datoteci .docx

Prilog 2. Završni rad u datoteci .pdf

Prilog 3. Programski kod aplikacije