Razvoj mobilne aplikacije za procjenu i praćenje rizika od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom

Vidović, Dario

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

RAZVOJ MOBILNE APLIKACIJE ZA PROCJENU I PRAĆENJE RIZIKA OD MOŽDANOG UDARA KLASIFIKACIJSKIM POSTUPCIMA STROJNOG UČENJA I GENERATIVNOM UMJETNOM INTELIGENCIJOM

Diplomski rad

Dario Vidović

Osijek, 2024.

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

Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju

Ocjena diplomskog rada na sveučilišnom diplomskom studiju

Ime i prezime pristupnika:	Dario Vidović
Studij, smjer:	Sveučilišni diplomski studij Računarstvo
Mat. br. pristupnika, god.	D1337R, 07.10.2022.
JMBAG:	0165082622
Mentor:	prof. dr. sc. Goran Martinović
Sumentor:	
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	izv. prof. dr. sc. Alfonzo Baumgartner
Član Povjerenstva 1:	prof. dr. sc. Goran Martinović
Član Povjerenstva 2:	prof. dr. sc. Krešimir Nenadić
Naslov diplomskog rada:	Razvoj mobilne aplikacije za procjenu i praćenje rizika od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom
Znanstvena grana diplomskog rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Zadatak diplomskog rada:	U teorijskom dijelu diplomskog rada potrebno je opisati probleme i izazove pri procjeni i praćenju rizika obolijevanja od moždanog udara, te mogućnosti korištenja strojnog učenja i generativne umjetne inteligencije za potporu pacijentu i liječniku. Na temelju analize stanja u području i postojećih rješenja, treba definirati funkcionalne i nefunkcionalne zahtjeve na mobilnu aplikaciju, predložiti model, arhitekturu i dizajn aplikacije. S ciljem procjene i praćenja rizika obolijevanja od moždanog udara, treba izabrati i prilagoditi prikladne skupove podataka, definirati značajke
Datum ocjene pismenog dijela diplomskog rada od strane mentora:	06.09.2024.
Ocjena pismenog dijela diplomskog rada od strane mentora:	Izvrstan (5)
Datum obrane diplomskog rada:	12.09.2024.
Ocjena usmenog dijela diplomskog rada (obrane):	Izvrstan (5)
Ukupna ocjena diplomskog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:	12.09.2024.

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

IZJAVA O IZVORNOSTI RADA

Osijek, 12.09.2024.

Ime i prezime Pristupnika:	Dario Vidović
Studij:	Sveučilišni diplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D1337R, 07.10.2022.
Turnitin podudaranje [%]:	13

Ovom izjavom izjavljujem da je rad pod nazivom**: Razvoj mobilne aplikacije za procjenu i praćenje rizika od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom**

izrađen pod vodstvom mentora prof. dr. sc. Goran Martinović

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD

Moždani udar je po život ugrožavajuće stanje do kojeg dolazi kada dio mozga ne dobiva dovoljno krvi. Postoje dvije vrste moždanog udara: ishemijski i hemoragijski. Ishemijski moždani udar je uzrokovan ugruškom koji začepi arteriju i onemogući protok krvi kroz nju dok je hemoragijski uzrokovan puknućem krvne žile i prodiranjem krvi u okolno tkivo. Kako bi se spriječio moždani udar, potrebno je pratiti čimbenike rizika koji vode do obolijevanja od moždanog udara. Razvoj mobilnih aplikacija pomaže u praćenju i procjeni rizika od moždanog udara.

Cilj ovog diplomskog rada je prikupiti sve medicinske čimbenike koji utječu na moždani udar te analizom sveukupnih podataka opisati mogućnosti korištenja strojnog učenja i generativne umjetne inteligencije u potpori pacijentu i liječniku. Funkcionalnosti mobilne aplikacije potrebno je implementirati programskim jezikom Kotlin dok će se za izradu dizajna aplikacije koristiti Jetpack Compose. Na temelju parametara, značajnih za moždani udar, testirat će se različiti klasifikacijski postupci te na temelju rezultata predložit će se model strojnog učenja i prikladan sustav stvaranja preporuka generativnom umjetnom inteligencijom. Također, opisuju se korišteni alati i tehnologije prilikom izrade programskog rješenja te sam rad aplikacije testira se kroz različite korisničke slučajeve.

U drugom poglavlju opisuje se problematika moždanog udara koja uključuje rad mozga, moždani udar i čimbenike rizika obolijevanja od moždanog udara te se prikazuju slična programska rješenja u području koja se bave navedenom tematikom. Treće poglavlje daje uvid u model i arhitekturu mobilne Android aplikacije te prikladne klasifikacijske postupke koji se koriste prilikom treniranja modela strojnog učenja. U četvrtom poglavlju opisuju se alati, programski jezik i razvojna okruženja koji se koriste prilikom razvoja i implementacije mobilne aplikacije. Također, prikazuje se programsko rješenje pojedinih funkcionalnosti aplikacije. Ispitivanje rada aplikacije, kroz različite korisničke slučajeve, prikazuje se u petom poglavlju.

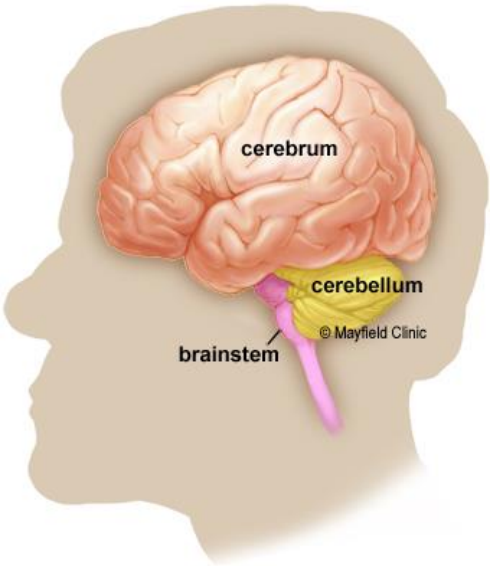
2. PROCJENA RIZIKA I PREGLED STANJA U PODRUČJU PRAĆENJA RIZIKA OBOLIJEVANJA OD MOŽDANOG UDARA

Unutar ovog poglavlja opisan je rad mozga, moždani udar te čimbenici rizika obolijevanja od moždanog udara. Također, prikazana su postojeća rješenja s gledišta medicine i računarstva te predloženo je idejno rješenje vlastite mobilne Android aplikacije.

2.1.Rad mozga

Mozak (znanstv. lat. *encephalon*) je najviši dio središnjega živčanoga sustava kod viših mnogostaničnih životinja i čovjeka [1]. Mozak prima informacije iz osjetnih organa, obrađuje ih i šalje upute izvršnim organima. Mozak, sjedište inteligencije i pamćenja, smješten je u lubanji, zaštićen moždanim ovojnicama i uronjen u cerebrospinalni likvor. Anatomski, mozak se dijeli na moždano deblo (lat. *truncus encephalicus*), koje obuhvaća produženu moždinu, most i srednji mozak, mali mozak (lat. *cerebellum*) i veliki mozak (lat. *cerebrum*). Glavni dijelovi mozga prikazani su na slici 2.1 preuzetoj s [2].

Veliki mozak, najveći dio mozga, sastoji se od desne i lijeve hemisfere. Izvršava brojne funkcije poput tumačenja dodira, vida i sluha, kao i govora, razmišljanja, emocija, učenja i precizne kontrole pokreta. Mali mozak se nalazi ispod velikog mozga. Njegova funkcija je koordinacija pokreta mišića, održavanje držanja i ravnoteže. Moždano deblo šalje poruke u oba smjera između mozga i ostatka tijela. Obavlja mnoge automatske funkcije kao što su disanje, rad srca, tjelesna temperatura, ciklusi budnosti i spavanja, probava, kihanje, kašljanje, povraćanje i gutanje.



Slika 2.1 Glavni dijelovi mozga [2]

2.2.Moždani udar

Moždani udar je po život opasno stanje, a nastaje kada dio mozga ne prima dovoljnu količinu krvi. Najčešći je uzrok tome začepljene arterije ili krvarenja u mozgu. Bez stalne opskrbe krvlju, moždane stanice u tom dijelu mozga počinju umirati zbog nedostatka kisika [3]. Postoje dva glavna načina na koja može doći do moždanog udara: ishemija i krvarenje.

Ishemija nastaje kada stanice nemaju dovoljno krvi za opskrbu kisikom, što je najčešće uzrokovano blokadom krvne žile u mozgu, prekidajući protok krvi. Ishemijski moždani udari čine 80% svih moždanih udara.

Hemoragijski moždani udar, koji uzrokuje krvarenje u ili oko mozga, događa se na jedan od dva načina:

- Krvarenje unutar mozga (intracerebralno). To se događa kada se krvna žila unutar mozga pokida ili otvori, uzrokujući krvarenje koje vrši pritisak na okolno moždano tkivo.
- Krvarenje u subarahnoidalni prostor (prostor između mozga i njegove vanjske ovojnice). Arahnoidna membrana, tanak sloj tkiva s uzorkom poput paukove mreže, okružuje mozak. Prostor između njega i mozga je subarahnoidalni prostor. Oštećenje krvnih žila koje prolaze kroz arahnoidnu membranu može uzrokovati subarahnoidalno krvarenje, koje krvari u subarahnoidalni prostor, vršeći pritisak na moždano tkivo ispod.

2.3. Čimbenici rizika obolijevanja od moždanog udara

Identifikacija pojedinaca s visokim rizikom od moždanog udara dodatno pomaže u odabiru osoba za opće i specifične intervencije za modificiranje čimbenika rizika [4].

Prema [5], čimbenici koji se mogu kontrolirati čine 82% do 90% svih moždanih udara:

- Visoki krvni tlak
- Pretilost
- Tjelesna neaktivnost
- Loša prehrana
- Pušenje

Ostali čimbenici rizika se temelje na načinu života, genetici i okolišu:

- Dob
- Anksioznost, depresija i visoka razina stresa
- Obiteljska povijest i genetika
- Život ili rad u područjima s onečišćenim zrakom
- Medicinska stanja (apneja u snu, bolesti bubrega, migrenske glavobolje)
- Nezdrave životne navike (prekomjerno konzumiranje alkohola, previše spavanja i upotreba ilegalnih droga)
- Rasa i etnička pripadnost
- Spol
- Virusne infekcije ili stanja, poput lupusa ili reumatoidnog artritisa, mogu uzrokovati upalu

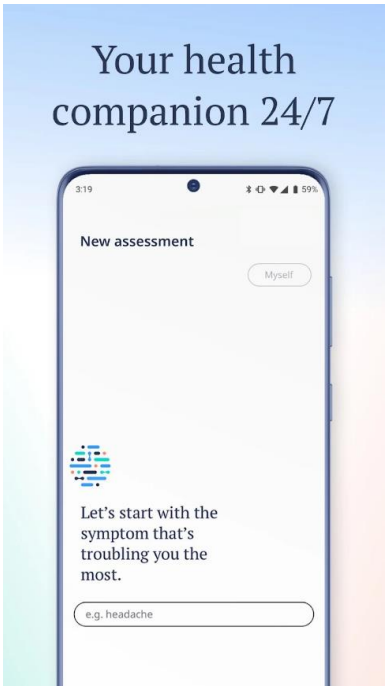
2.4. Postojeća rješenja s gledišta medicine

Ada Health osnovana je 2011. godine, kroz suradnju dr. Claire Novorol, profesora Martina Hirscha i Daniela Nathratha [6]. Ada Health je brzorastuća globalna zdravstvena mobilna aplikacija s misijom pomoći ljudima da aktivno upravljaju svojim zdravljem i podrže medicinske stručnjake u pružanju učinkovite skrbi.

Ada Health aplikacija vodeće je rješenje umjetne inteligencije za provjeru simptoma i procjenu zdravlja. Tehnologija, koja se temelji na probabilističkom zaključivanju nad odabranom medicinskom bazom znanja, predstavlja najsuvremeniju automatiziranu, točnu i sigurnu procjenu simptoma [7]. Trenutno, većina slučajeva upotrebe za rješenja temeljena na dubokom

učenju umjetne inteligencije je za analizu slike - na primjer, analiziranje MRI skeniranja za prepoznavanje potencijalnih karcinoma i djelovanje kao alat za podršku odlučivanju.

Ada Health u svom sustavu koristi tehnike strojnog učenja (engl. *Machine Learning*) i obrade prirodnog jezika (engl. *Natural Language Processing – NLP*) [8]. Koriste se algoritmi strojnog učenja, posebno konvolucijska neuronska mreža (engl. *Convolutional Neural Network – CNN*) kako bi detektirali slučajeve depresivnih karaktera i pružili rješenja za probleme mentalnog zdravlja. Dodatno, koriste NLP tehnike za pretvaranje audio i video ulaza u tekst za daljnju analizu. Kombinirajući ove tehnologije, Ada Health ima za cilj razviti detaljnu nadzornu ploču statusa i povijesti bolesti korisnika, koja se može vizualizirati putem web aplikacije. Ovaj pristup omogućuje predviđanje anksioznosti i depresije na temelju slika izraza lica i simptoma, što u konačnici pomaže u dijagnozi i liječenju stanja mentalnog zdravlja. Zaslon navedene aplikacije, prikazan na slici 2.2, preuzet je s [9].



Slika 2.2 Ada Health [9]

2.5. Postojeća rješenja s gledišta računarstva

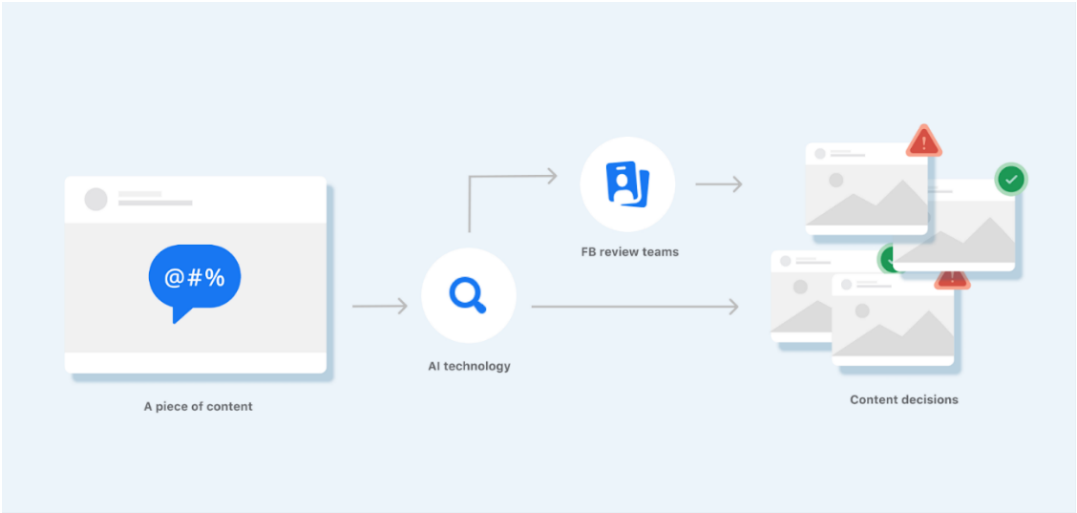
Facebook je američka online platforma društvenih medija i servis društvenih mreža koji je dio tvrtke Meta Platforms [10].

Za pronalazak, pregled i poduzimanje radnji u vezi sa sadržajem koji može biti protiv standarda zajednice, koriste se tehnologija i ljudski recenzenti (engl. *human reviewers*). Tehnologija umjetne inteligencije (engl. *Artificial Intelligence – AI*) ključna je za postupak pregleda

sadržaja. AI može otkriti i ukloniti sadržaj koji je u suprotnosti sa standardima zajednice prije nego što ga bilo tko prijavi.

Timovi umjetne inteligencije (engl. *AI teams*) počinju izgradnjom modela strojnog učenja koji mogu raditi stvari poput prepoznavanja onoga što je na fotografiji ili analize teksta objave. Na primjer, modeli umjetne inteligencije mogu se izraditi kako bi saznali sadrži li dio sadržaja golotinju ili eksplicitan sadržaj. Navedeni modeli zatim mogu odlučiti treba li poduzeti radnje u vezi sa sadržajem, kao što je njegovo uklanjanje s platforme ili smanjenje njegove distribucije.

Ponekad dio sadržaja zahtijeva daljnji pregled i tehnologija umjetne inteligencije ga šalje ljudskom timu za pregled da ga pomnije pogleda. U tim slučajevima timovi za pregled donose konačnu odluku, a tehnologija uči i poboljšava se iz svake odluke. S vremenom, nakon učenja na tisućama ljudskih odluka, tehnologija postaje bolja. Upotreba tehnologije umjetne inteligencije od strane Facebooka, prikazana na slici 2.3, preuzeta je s [11].



Slika 2.3 Upotreba tehnologije umjetne inteligencije od strane Facebooka [11]

Facebook koristi strojno učenje za generiranje procijenjene stope radnje i ocjene kvalitete oglasa koja se koristi u jednadžbi ukupne vrijednosti. Kako bi se pronašla procijenjena stopa radnje, modeli strojnog učenja predviđaju vjerojatnost da će određena osoba poduzeti željenu radnju oglašivača, na temelju poslovnog cilja koji oglašivač odabere za svoj oglas, poput povećanja posjeta web-lokaciji ili povećanja kupnje. Modeli, da se to učini, uzimaju u obzir ponašanje osobe na Facebooku i izvan njega, kao i druge čimbenike, poput sadržaja oglasa, doba dana i interakcije između ljudi i oglasa. Računanje ukupne ocjene vrijednosti oglasa na dražbi oglasa, prikazano na slici 2.4, preuzeto je s [12].

Primjeri ponašanja osobe na Facebooku koje modeli razmatraju uključuju stvari koje osoba radi dok koristi Facebook aplikacije, poput pritiska na oglas ili označavanje objave sa „sviđa mi se“, dok primjeri ponašanja osobe izvan Facebooka koje modeli razmatraju uključuju stvari koje osoba radi izvan Facebooka koje tvrtke dijele s ljudima putem poslovnih alata, poput posjete web stranici, kupnje proizvoda ili instaliranja aplikacije.

Kako bi se generirala ocjena kvalitete oglasa, modeli strojnog učenja uzimaju u obzir povratne informacije ljudi koji gledaju ili skrivaju oglas, kao i procjene atributa niske kvalitete (kao što je previše teksta na slici oglasa, senzacionalistički jezik ili mamac za angažman).

Slika 2.4 Računanje ukupne ocjene vrijednosti oglasa na dražbi oglasa [12]

2.6. Idejno rješenje vlastite mobilne aplikacije za praćenje tijeka bolesti

Na temelju prethodno opisanih sličnih programskih rješenja u području računarstva i medicine, mobilna Android aplikacija za procjenu i praćenje rizika od moždanog udara trebala bi zadovoljiti uvjete u vezi generiranja preporuka na temelju korisničkih osobnih i zdravstvenih čimbenika koristeći model strojnog učenja u kombinaciji s generativnom umjetnom inteligencijom.

Putem izrade osobnog profila i proizvoljnog unosa nalaza ispunjavanjem zdravstvenog upitnika, aplikacija korisniku treba pomoći u praćenju tijeka bolesti dajući mu prikladne preporuke na temelju njegovih sveukupnih zdravstvenih i osobnih parametara.

3. MODEL I ARHITEKTURA MOBILNE APLIKACIJE

Unutar ovog poglavlja, opisuju se funkcionalni i nefunkcionalni zahtjevi na mobilnu aplikaciju. Opisuju se parametri simptoma i pokazatelji bolesti koje korisnik ispunjava unutar zdravstvenog upitnika te se predlažu prikladni klasifikacijski postupci koji se koriste prilikom treniranja modela strojnog učenja. Također, prikazuje se građa mobilne aplikacije.

3.1.Funkcionalni zahtjevi na mobilnu aplikaciju

Za korištenje aplikacije za procjenu i praćenje rizika od moždanog udara, korisnik se prvo treba registrirati. Da bi se korisnik uspješno registrirao, treba unijeti ime, prezime, adresu e-pošte, zaporku te ponovno potvrditi zaporku. Nakon toga, korisnik se prijavljuje u aplikaciju unosom adrese e-pošte i zaporke prethodno napravljenog računa. Prije nego korisnik krene ispunjavati zdravstveni upitnik, treba izraditi osobni profil. Nakon što korisnik unese zdravstvene podatke, zdravstveni podaci zajedno s danim preporukama aplikacije pohranjuju se u bazu podataka te se povijest unosa može pregledati u bilo kojemu trenutku pritiskom na tipku kartice „*History*“.

3.1.1. Prijava korisnika

Kada se mobilna aplikacija pokrene, korisniku se prikazuje zaslon prijave. Navedeni zaslon sastoji se od polja za unos adrese e-pošte, zaporke te tipke za prijavu. Ispod navedenih polja, nalaze se dvije poruke. Pritiskom na poruku „*Forgot your password*?“, korisniku se otvara zaslon za ponovno postavljanje zaporke dok pritiskom na „*Register*“ dio poruke, korisniku se otvara zaslon za registriranje.

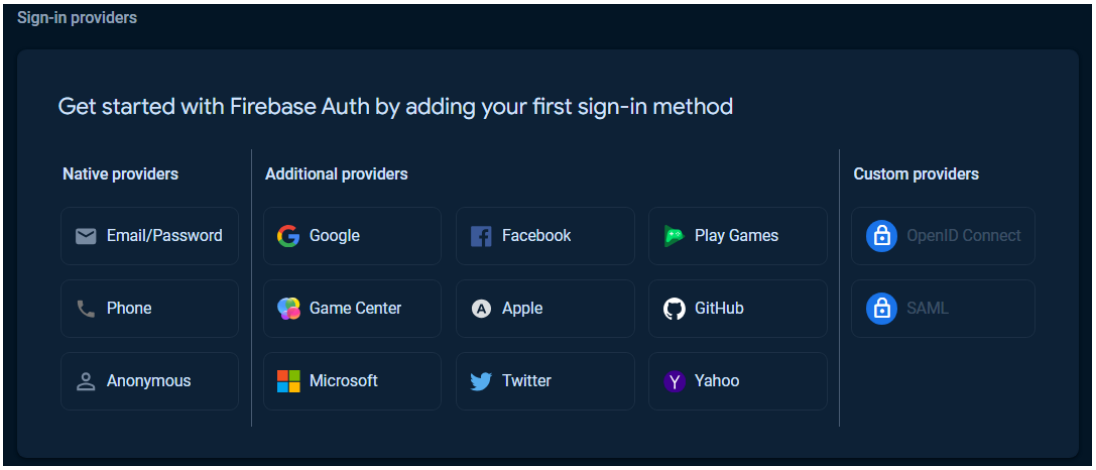
Nakon uspješne prijave, korisniku se otvara glavni zaslon aplikacije koji se sastoji od popisa kartica od kojih korisnik može odabrati odlazak na korisnički profil, unos zdravstvenih podataka, povijest unosa zdravstvenih podataka ili odjavu iz aplikacije.

3.1.2. Registriranje korisnika

Kada korisnik pritisne na „*Register*“ dio poruke, koja je prikazana na zaslonu za prijavu, otvara se zaslon za registriranje. Navedeni zaslon sastoji se od polja za unos imena, prezimena, adrese e-pošte, zaporke, ponovni unos zaporke te tipke za registriranje. Ispod navedenih polja, nalazi se poruka „*Already have an account? Login*“. Pritiskom na „*Login*“ dio poruke, korisniku se otvara zaslon za prijavu.

Izrada korisničkog računa omogućena je pomoću *Firebase Autha*, a njegovi pružatelji usluge prijave su prikazani na slici 3.1. Nakon uspješnog registriranja, korisniku se otvara zaslon za

prijavu dok se na poslužiteljskoj strani, u *Realtime Databaseu*, spremaju njegovi identifikacijski podaci.



Slika 3.1 Pružatelji usluge prijave *Firebase Autha*

3.1.3. Ponovno postavljanje zaporke

Ako je korisnik zaboravio zaporku korisničkog računa, ima mogućnost ponovno je postaviti. Pritiskom poruke „*Forgot your password?*“, koja se nalazi na zaslonu za prijavu, otvara se zaslon za ponovno postavljanje zaporke. Navedeni zaslon sadrži polje za unos adrese e-pošte i tipku za podnošenje zahtjeva. Ukoliko postoji korisnički račun s navedenom adresom e-pošte, korisnik će na adresi e-pošte dobiti poveznicu na kojoj može izvršiti ponovno postavljanje zaporke. Nakon uspješne promjene, korisnik se može prijaviti u aplikaciju s novom zaporkom.

3.1.4. Izrada korisničkog profila

Ako se korisnik prvi puta prijavljuje u aplikaciju, prije unosa zdravstvenih podataka, treba izraditi profil. Profil uključuje osobne podatke koji se rjeđe mijenjaju, a uključuju sljedeće: spol (muško/žensko), dob, srčana bolest (da/ne), bračni status (da/ne), vrsta posla (državni posao, nikada nije radio/la, privatni sektor, samozaposlen, rad s djecom) te vrsta prebivališta (ruralno/urbano). Nakon unosa podataka i pritiska na tipku „*Save*“, podaci se spremaju u bazu podataka.

Navedeni osobni podaci se mogu promijeniti u bilo kojemu trenutku te se koriste prilikom procjene i praćenja rizika od moždanog udara.

3.1.5. Unos zdravstvenih podataka

Nakon izrade profila, pritiskom na tipku kartice „*Add health report*“, korisnik ima mogućnost unijeti zdravstvene podatke. Unos zdravstvenih podataka se može unijeti proizvoljno puta,

ovisno o promjeni s vremenom ili situacijom korisnika te se povijest detalja svakog unosa bilježi u bazi podataka. Korisnik treba unijeti sistolički tlak, dijastolički tlak, visinu, težinu te razinu šećera u krvi. Zdravstveni podaci, zajedno s osobnim podacima čine ključne parametre pri procjeni i praćenju rizika od moždanog udara.

3.1.6. Preporuke s obzirom na zdravstvene podatke

Nakon što korisnik unese svoje zdravstvene podatke, zajedno s dohvaćenim osobnim podacima, šalje se upit OpenAI modelu koji na temelju navedenih podataka generira prilagođene preporuke koje se šalje u aplikaciju kao odgovor, pohranjuju te ispsuje korisniku na zahtjev.

3.1.7. Pregled povijesti unosa podataka

Korisnik ima uvid u povijest unosa zdravstvenih podataka u bilo kojemu trenutku. Pritiskom na tipku kartice „*History*“, korisnik može vidjeti popis datum unosa zdravstvenih podataka. Pritiskom na tipku kartice pojedinog datuma, korisnik može vidjeti rezultate unosa podataka za navedeni datum. Praćenjem tijeka bolesti, korisnik ima mogućnost praćenja napretka, tj. kontinuirani uvid u promjenu zdravstvenog stanja tijekom vremena te se povijest podataka može upotrijebiti kao dokumentacija za medicinske potrebe.

3.1.8. Odjava korisnika

Pritiskom na tipku kartice za odjavu, korisnik se odjavljuje iz aplikacije te mu se pojavljuje zaslon za prijavu nakon čega ima mogućnost ponovne prijave ili izrade novog korisničkog računa.

3.2. Nefunkcionalni zahtjevi na mobilnu aplikaciju

Prema [13], nefunkcionalni zahtjevi skup su specifikacija koje opisuju radne mogućnosti i ograničenja sustava. To su, u osnovi, zahtjevi koji evaluiraju koliko sustav dobro radi, uključujući stvari poput brzine, sigurnosti, pouzdanosti te integriteta podataka.

Aplikacija se radi sa svrhom da bude brza, sigurna, pouzdana, skalabilna te intuitivna. Osobni podaci te povijest unosa zdravstvenih podataka zaštićeni su adresom e-pošte i zaporkom koji su pohranjeni u bazi podataka. Unos zdravstvenih podataka, procjena rizika od moždanog udara i generiranje preporuka odvijaju se u pozadini i bez djelovanja na ostale funkcionalnosti mobilne aplikacije. Promjena zaslona je brza, a sama upotreba mobilne aplikacije je jednostavna i intuitivna.

3.3.Parametri simptoma i pokazatelja bolesti u zdravstvenom upitniku

Nakon što korisnik izradi profil, ima mogućnost unosa zdravstvenih podataka u bilo kojem trenutku. Pritiskom na tipku kartice „*Add health report*“, korisniku se otvara zdravstveni upitnik koji se sastoji od unosa sljedećih parametara: sistolički tlak (mmHg), dijastolički tlak (mmHg), srčana bolest (DA/NE), status pušenja (prije pušio/la, nikada pušio/la, puši, nepoznato), tjelesna visina (m), tjelesna masa (kg) te razina glukoze u krvi (mg/dL). Pomoću gornjeg (sistoličkog) tlaka i donjeg (dijastoličkog) tlaka određuje se kategorija krvnog tlaka. Indeks tjelesne mase (engl. *Body mass index - BMI*) može se izračunati jednadžbom (3-1) koristeći tjelesnu visinu i tjelesnu masu korisnika. Navedeni parametri simptoma i pokazatelja bolesti koriste se pri procjeni rizika od moždanog udara.

$$Indeks\ tjelesne\ mase = \frac{Tjelesna\ masa\ (kg)}{Tjelesna\ visina^2\ (m^2)} \tag{3-1}$$

Nakon unosa svih zdravstvenih podataka, pritiskom tipke „*Submit*“, podaci se spremaju u bazu podataka te su dostupni za pregled u bilo kojemu trenutku pritiskom na tipku kartice „*History*“.

Kategorije krvnog tlaka prikazane su u tablici 3.1, a kategorije Indeksa tjelesne mase su prikazane u tablici 3.2.

Tablica 3.1: Kategorije krvnog tlaka [14]

Kategorija	**Sistolički krvni tlak (mmHg)**	**Dijastolički krvni tlak (mmHg)**
Normalan	< 120	< 80
Povišen	120 – 129	< 80
1. stupanj hipertenzije	130 – 139	80 – 89
2. stupanj hipertenzije	140 >	90 >
Hipertenzivna kriza	180 >	120 >

Tablica 3.2: Kategorije Indeksa tjelesne mase [15]

Indeks tjelesne mase	**Kategorija**
< 18.5	Pothranjenost
18.5 – 24.9	Idealna tjelesna masa
25.0 – 29.9	Prekomjerna tjelesna masa
30 >	Pretilost

3.4. Prikladni klasifikacijski postupci analize podataka strojnog učenja

Prema [16], [17], [18], [19], opisani su prikladni klasifikacijski postupci analize podataka koji se mogu koristiti prilikom treniranja modela strojnog učenja unutar ove mobilne Android aplikacije.

3.4.1. Stablo odluke

Stablo odluka, neparametarski nadzirani algoritam učenja, koristi se za zadatke klasifikacije i regresije. Ima hijerarhijsku, stabloliku strukturu koja se sastoji od korijenskog čvora (engl. *root node*), grana (eng. *branches*), unutarnjih čvorova (engl. *internal nodes*) i listova (engl. *leaf*) [16].

Stablo odluke počinje s korijenskim čvorom, koji se grana na različite moguće ishode. Svaki od tih ishoda također dolazi s dodatnim čvorovima koji proizlaze iz njih i mogu se dalje granati. Kada se svi mogući ishodi razgranaju, stvara se dijagram oblika stabla.

Pojačano stablo odluke, algoritam korišten unutar ove aplikacije, metoda učenja je s ansamblima u kojoj drugo stablo ispravlja pogreške prvog stabla, treće stablo ispravlja pogreške prvog i drugog stabla i tako dalje. Predviđanja se temelje na cijelom ansamblu stabala koja zajedno donose predikciju.

3.4.2. Slučajna šuma

Slučajna šuma (engl. *random forest*) je algoritam za strojno učenje koji stvara ansambl više stabala odluka kako bi došao do jedinstvene, preciznije predikcije ili rezultata [17]. Prema [18], slučajna šuma počinje sa stablom odluka, kao što je binarno stablo, kako bi pronašao par varijabla-vrijednost iz podataka na kojima je treniran. Proces je rekurzivan i ponavlja se sve dok se ne postigne maksimalna dubina ili dok se više ne može izvršiti podjela podskupa. Slučajna šuma radi korištenjem metrika poput dobivanja informacija, *Gini impurity* i srednje kvadratne pogreške za procjenu novog razdvajanja. Slučajna šuma spaja slaba stabla odluka i pretvara ih u snažnije modele.

3.4.3. Neuronska mreža

Neuronska mreža je program za strojno učenje, ili model, koji donosi odluke na način sličan ljudskom mozgu, koristeći procese koji oponašaju način na koji biološki neuroni zajedno rade kako bi prepoznali pojave, procijenili opcije i donijeli zaključke [19].

Svaka neuronska mreža sastoji se od slojeva čvorova, ili umjetnih neurona - ulaznog sloja, jednog ili više skrivenih slojeva i izlaznog sloja. Svaki čvor je povezan s drugim čvorovima i ima svoju pridruženu težinu i prag. Ako je izlaz pojedinog čvora iznad određene granične vrijednosti, taj čvor se aktivira i šalje podatke sljedećem sloju mreže. U suprotnom, podaci se ne prenose u sljedeći sloj mreže.

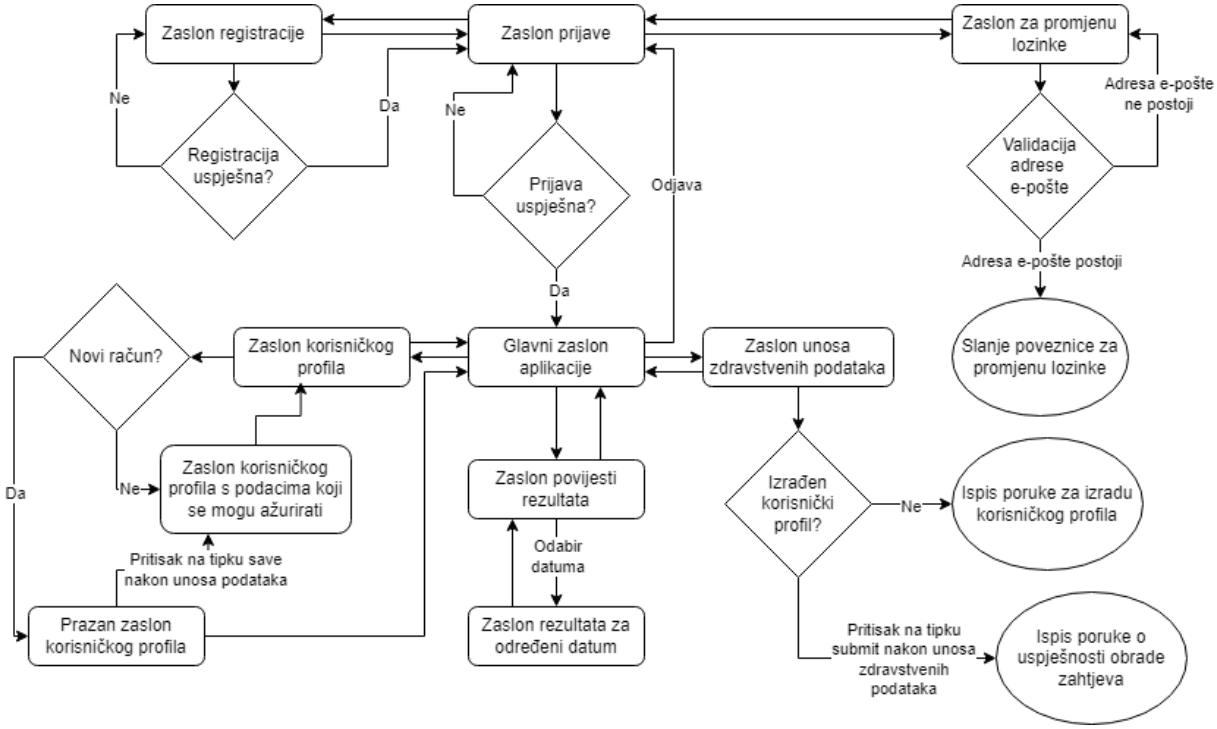
Prema [20], prednosti neuronskih mreža su: nelinearnost, mapiranje ulaz-izlaz, prilagodljivost, evidentni odgovor, kontekstualne informacije, otpornost na greške, mogućnost implementacije u VLSI (engl. *very-large-scale integration*), uniformnost analize i dizajna i neurobiološka analogija.

3.5. Građa mobilne aplikacije

Klijentski dio aplikacije uključuje sljedeće zaslone: zaslon registriranja, zaslon prijave, zaslon za promjenu zaporke, glavni zaslon aplikacije, zaslon izrade korisničkog profila, zaslon korisničkog profila, zaslon izmjene osobnih podataka, zaslon povijesti rezultata, zaslon unosa zdravstvenih podataka te zaslon rezultata za odabrani datum unesenog nalaza.

Poslužiteljski dio aplikacije uključuje bazu podataka *Firebase Realtime Database*, uslugu za jednostavno i sigurno upravljanje autentikacijom korisnika u aplikaciji *Firebase Authentication*, procjena rizika putem modela strojnog učenja te sustav generiranja preporuka slanjem korisničkog upita preko API-ja (engl. *Application Programming Interface*) OpenAI-ju.

Tijek aktivnosti pri korištenju mobilne Android aplikacije prikazan je na slici 3.2.



Slika 3.2 Tijek aktivnosti pri korištenju mobilne Android aplikacije

4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE

Unutar ovog poglavlja, opisuju se korišteni alati i tehnologije prilikom razvoja mobilne Android aplikacije. Također, opisuje se postupak treniranja modela strojnog učenja kroz analizu korištenog skupa podataka i razvoj, treniranje i ispitivanje samog modela. Osim postupka stvaranja preporuka korištenjem generativne umjetne inteligencije, bit će prikazano programsko rješenje na strani korisnika i programsko rješenje na strani poslužitelja.

4.1. Korišteni alati i tehnologije

4.1.1. Operacijski sustav Android

Operacijski sustav Android, snažna je i prilagodljiva platforma razvijena prvenstveno za mobilne uređaje poput pametnih telefona, tableta, pametnih satova i drugih nosivih uređaja. To je operativni sustav otvorenog koda baziran na *Linux* jezgri, koji pruža snažnu i fleksibilnu osnovu za širok raspon aplikacija i funkcionalnosti [21]. U svojoj srži, operacijski sustav Android, služi kao softverski okvir koji korisnicima omogućuje interakciju sa svojim uređajima, pristup raznim uslugama i neometano pokretanje različitih aplikacija. Njegova arhitektura dizajnirana je tako da bude modularna i prilagodljiva, omogućujući proizvođačima uređaja i razvojnim programerima da prilagode korisničko iskustvo specifičnim hardverskim konfiguracijama i korisničkim preferencijama.

4.1.2. Android Studio

Android Studio [22] službeno je integrirano razvojno okruženje (engl. *Integrated Development Environment - IDE*) za razvoj Android aplikacija. Temeljen na snažnom uređivaču koda i alatima za razvoj iz *IntelliJ IDEA*, Android Studio nudi brojne značajke koju povećavaju produktivnost prilikom izrade Android aplikacija, a neke od njih uključuju: emulator koji je brz i bogat značajkama, ažuriranje i prikaz promjena *Composable* funkcija u emulatorima i fizičkim uređajima u stvarnom vremenu, predlošci koda i integracija s GitHubom, opsežni alati i okviri za testiranje i druge. Android Studio dostupan je za Windows, macOs i Linux operacijske sustave.

4.1.3. Programski jezik Kotlin

Prema [23], Kotlin je jezik temelj na JVM-u razvijen od JetBrains, tvrtke koje je poznata po izradi IntelliJ IDEA, snažnog integriranog razvojnog okruženja za razvoj u Javi. Kotlin je statički programski jezik otvorenog koda koji podržava i objektno i funkcionalno programiranje

te nudi sličnu sintaksu i koncepte iz drugih jezika, uključujući C#, Java i Scala, među mnogim drugima. Kotlin je brz, sažet, siguran i fleksibilan jezik te potpuno je interoperabilan s kodom pisanim u Javi.

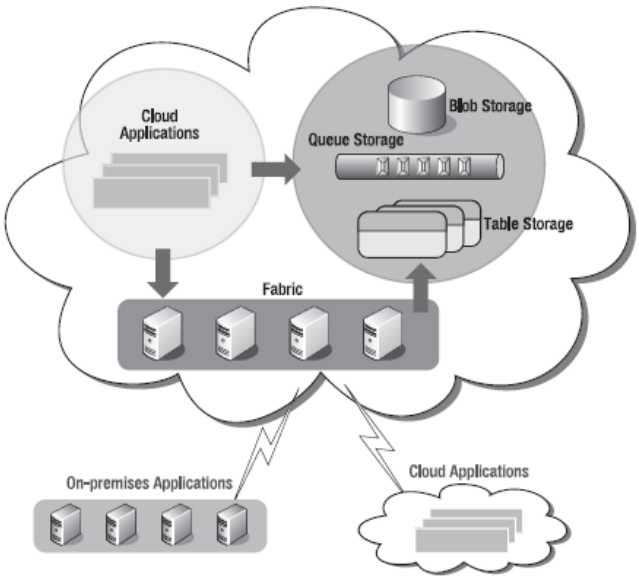
4.1.4. Jetpack Compose

Jetpack Compose je moderan alat za izradu korisničkih sučelja kojeg je pokrenuo Google i koristi se za izradu nativnog Android sučelja. Pojednostavljuje i ubrzava razvoj sučelja s manje koda, Kotlin API-jima i snažnim alatima [24]. Potpuno je deklarativan tako da se može korisničko sučelje opisati pozivom niza funkcija koje transformiraju podatke u hijerarhiju korisničkog sučelja. Kada se podaci promijene ili ažuriraju, okvir automatski poziva te funkcije i ažurira pogled (engl. *view*). Prednosti Jetpack Compose-a su: deklarativan je, kompatibilan je, povećana brzina razvoja, sažet i idiomatski je, lagan za održavanje te pisan je u Kotlinu.

4.1.5. Microsoft Azure

Microsoft Azure, prije poznat kao Windows Azure, Microsoftova je javna računalna platforma u oblaku. Pruža širok raspon usluga u oblaku koje korisnici mogu birati za razvoj i skaliranje novih aplikacija ili pokretanje postojećih aplikacija u javnom oblaku [25]. Neke od najpopularnije kategorije usluga uključuju: računalne usluge, pohrana, umrežavanje, baze podataka, umjetna inteligencija i strojno učenje, *DevOps,* sigurnost i brojne druge.

Infrastruktura Azure platforme u oblaku, preuzeta s [26] prikazana je slikom 4.1.



Slika 4.1 Infrastruktura Azure platforme u oblaku [26]

4.1.6. OpenAI

OpenAI, tvrtka i istraživački laboratorij umjetne inteligencije, osnovan je 2015. s fokusom na razvoj umjetne inteligencije i alata za strojno učenje za različite aktivnosti. Njegova prva ponuda bio je alat otvorenog koda za razvoj algoritama za učenje s pojačanjem (engl. *reinforcement learning algorithms*) (OpenAI Gym), što ga je potaknulo da se usredotoči na istraživanje umjetne inteligencije za općenitije svrhe. OpenAI nudi značajan broj proizvoda poput: *ChatGPT, DALL-E 2, Codex, Whisper, Scholar, OpenAI Gym* te *OpenAI API* koji je bio korišten unutar ove mobilne aplikacije [27].

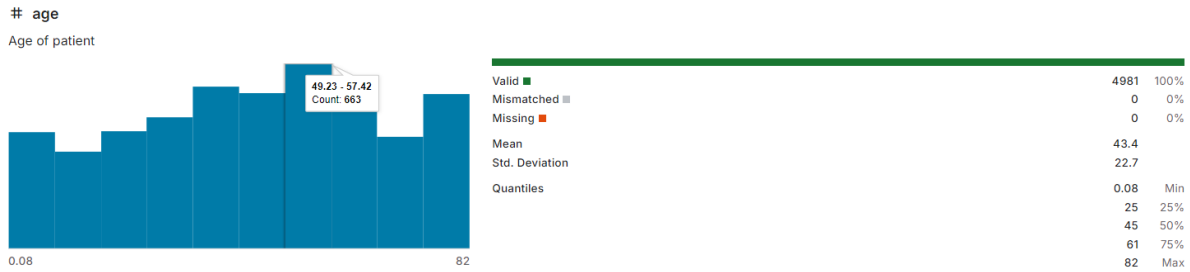
4.1.7. Firebase

Firebase je proizvod tvrtke Google koji olakšava razvoj funkcionalnosti za upravljanje bazom podataka te na taj način pomaže programerima da lako izgrade, upravljaju i unaprijede svoje aplikacije [28]. Osim što olakšava programerima da brže i sigurnije razvijaju svoje aplikacije, pruža pohranu u oblaku te koristi NoSQL bazu podataka za pohranu podataka. Za izradu boljih aplikacije, Firebase uključuje usluge na poslužiteljskoj strani koje pomažu programerima da bolje izgrade i upravljaju svojim aplikacijama, a one su: *Realtime Database, Cloud Firestore, Authentication, Remote Config, Hosting* te *Firebase Cloud Messaging(FCM)*. Za poboljšanje kvalitete aplikacije, Firebase nudi sljedeće usluge: *Crashlytics, Performance Monitoring, Test lab* i *App Distribution*. Za razvoj aplikacije, koji uključuje analitiku aplikacije te funkcionalnosti koje pomažu u interakciji s korisnicima i predviđanjima, dostupne usluge su: *Google Analytics, Predictions, Dynamic Links* te *A/B Testing*. Usluge korištene unutar ove mobilne aplikacije su *Firebase Authentication* i *Firebase Realtime Database*.

4.2. Analiza skupa podataka za treniranje modela strojnog učenja

Skup podataka, preuzet s [29], upotrijebljen je prilikom treniranja modela strojnog učenja koji je implementiran unutar ove aplikacije. Skup podataka je prijašnje obrađen tako da su uklonjene vrijednosti koje nedostaju, vrijednosti koje su značajno odstupale od drugih te vrlo rijetke kategorijske vrijednosti. Skup podataka se sastoji od 12 značajki: spol (m/ž), dob (numerička vrijednost), hipertenzija (0/1), srčana bolest (0/1), bračni status (da/ne), tip posla (državni posao, nikada nije radio/la, privatni sektor, samozaposlen, rad s djecom), tip prebivališta (ruralno/urbano), razina glukoze u krvi (numerička vrijednost), indeks tjelesne mase (numerička vrijednost), status pušenja (bivši pušač, pušač, nikada pušio/la, nepoznato) te moždani udar (0/1). Prema [29], u detaljima skupa podataka, može se vidjeti za svaku značajku u postotcima različite statistike poput: jedinstvene vrijednosti, najčešće vrijednosti, srednje

vrijednosti, standardna devijacija, broj važećih unosa, broj unosa koji nedostaju i druge. Navedene statistike prikazane su u obliku grafova te primjer za jedan od statistika prikazan je na slici 4.2. Skup podataka sastoji se od 4981 zapisa, od kojih 4733 rezultira da osoba nije imala moždani udar, a 248 da je osoba imala moždani udar. Navedeni podaci naznačuju da je skup podataka neuravnotežen te da slučaj kada osoba nema moždani udar predstavlja većinsku klasu. Kako bi se skup podataka uravnotežio, s ciljem poboljšanja performansi modela, koristi se jedna od strategija za uravnoteženje skupa podataka koja će biti opisana u potpoglavlju 4.3.



Slika 4.2 Graf raspodjele dobi u skupu podataka predviđanja moždanog udara

4.3. Razvoj, treniranje i ispitivanje modela strojnog učenja

Skup podataka, detaljnije opisan u potpoglavlju 4.2, preuzet je u *.csv* formatu te isječak koji prikazuje informacije vezane za predviđanje moždanog udara prikazan je slikom 4.3.

```
gender,age,hypertension,heart_disease,ever_married,work_type,Residence_type,avg_glucose_level,bmi,smoking_status,stroke
Male,67.0,0,1,Yes,Private,Urban,228.69,36.6,formerly smoked,1
Male,80.0,0,1,Yes,Private,Rural,105.92,32.5,never smoked,1
Female,49.0,0,0,Yes,Private,Urban,171.23,34.4,smokes,1
Female,79.0,1,0,Yes,Self-employed,Rural,174.12,24.0,never smoked,1
Male,81.0,0,0,Yes,Private,Urban,186.21,29.0,formerly smoked,1
Male,74.0,1,1,Yes,Private,Rural,70.09,27.4,never smoked,1
Female,69.0,0,0,No,Private,Urban,94.39,22.8,never smoked,1
Female,78.0,0,0,Yes,Private,Urban,58.57,24.2,Unknown,1
Female,81.0,1,0,Yes,Private,Rural,80.43,29.7,never smoked,1
Female,61.0,0,1,Yes,Govt_job,Rural,120.46,36.8,smokes,1
Female,54.0,0,0,Yes,Private,Urban,104.51,27.3,smokes,1
Female,79.0,0,1,Yes,Private,Urban,214.09,28.2,never smoked,1
Female,50.0,1,0,Yes,Self-employed,Rural,167.41,30.9,never smoked,1
Male,64.0,0,1,Yes,Private,Urban,191.61,37.5,smokes,1
Male,75.0,1,0,Yes,Private,Urban,221.29,25.8,smokes,1
Female,60.0,0,0,No,Private,Urban,89.22,37.8,never smoked,1
Female,71.0,0,0,Yes,Govt_job,Rural,193.94,22.4,smokes,1
Female,52.0,1,0,Yes,Self-employed,Urban,233.29,48.9,never smoked,1
Female,79.0,0,0,Yes,Self-employed,Urban,228.7,26.6,never smoked,1
```

Slika 4.3 Isječak skupa podataka za predviđanje moždanog udara u .csv formatu

Kako bi se skup podataka koristio za razvoj, treniranje i ispitivanje unutar dizajnera (engl. *Designer*) *Azure Machine Learning Studija* potrebno ga je prvo učitati kao resurs. Nakon što je skup podataka učitan, može ga se koristiti tako da se resurs, iz kartice s podacima, povuče i

ispusti na praznu površinu novoizrađenog eksperimenta. S ciljem postizanja boljih rezultata, dodaju se različite komponente koje transformiraju inicijalni skup podataka. Komponente korištene u modelu koji je implementiran unutar ove aplikacije su: *Edit Metadata*, *SMOTE*, *Normalize Data*, *Split Data*, *Train Model*, *Score Model*, *Evaluate model* te različiti algoritmi strojnog učenja.

Edit Metadata komponenta služi za promjenu metapodataka povezanih sa stupcima u skupu podataka. Postavke komponente uključuju postavljanje svih stupaca koji nisu numeričkog tipa na kategorijski tip kako bi se osiguralo da model strojnog učenja ispravno prepoznao navedene značajke kao diskretne skupine te na taj način se sprječava pogrešna numerička interpretacija i omogućava ispravna obrada podataka.

Problem neuravnoteženosti klasa rješava se primjenom SMOTE-a [30]. *SMOTE* (engl. *Synthetic Minority Oversampling Technique*) statistička je tehnika za povećanje broja slučajeva u skupu podataka na uravnotežen način. Komponenta radi tako da generira nove instance iz postojećih manjinskih slučajeva – slučajeva kada korisnik je dobio moždani udar. Nove instance nisu samo kopija manjinskih slučajeva, već algoritam uzima uzorke iz prostora značajki za svaku ciljnu klasu i njezine najbliže susjede te zatim generira nove primjerke koji kombiniraju značajke ciljnog slučaja sa značajkama njegovih susjeda. Za model koji se koristi u aplikaciji, postavke *SMOTE* komponente uključuju 700% postotak SMOTE-a te broj najbližih susjeda – 1. Nakon postavljanja komponente na navedene postavke, generira se 1736 novih instanci logičke jedinice (slučaj kada je korisnik dobio moždani udar).

Normalize Data – cilj normalizacije je promijeniti vrijednosti numeričkih stupaca u skupu podataka tako da koriste zajedničku skalu bez narušavanja razlika u rasponima vrijednosti ili da se izgube informacije. Postavke komponente uključuju primjenu *ZScore* transformacije na stupce koji su numeričkog tipa.

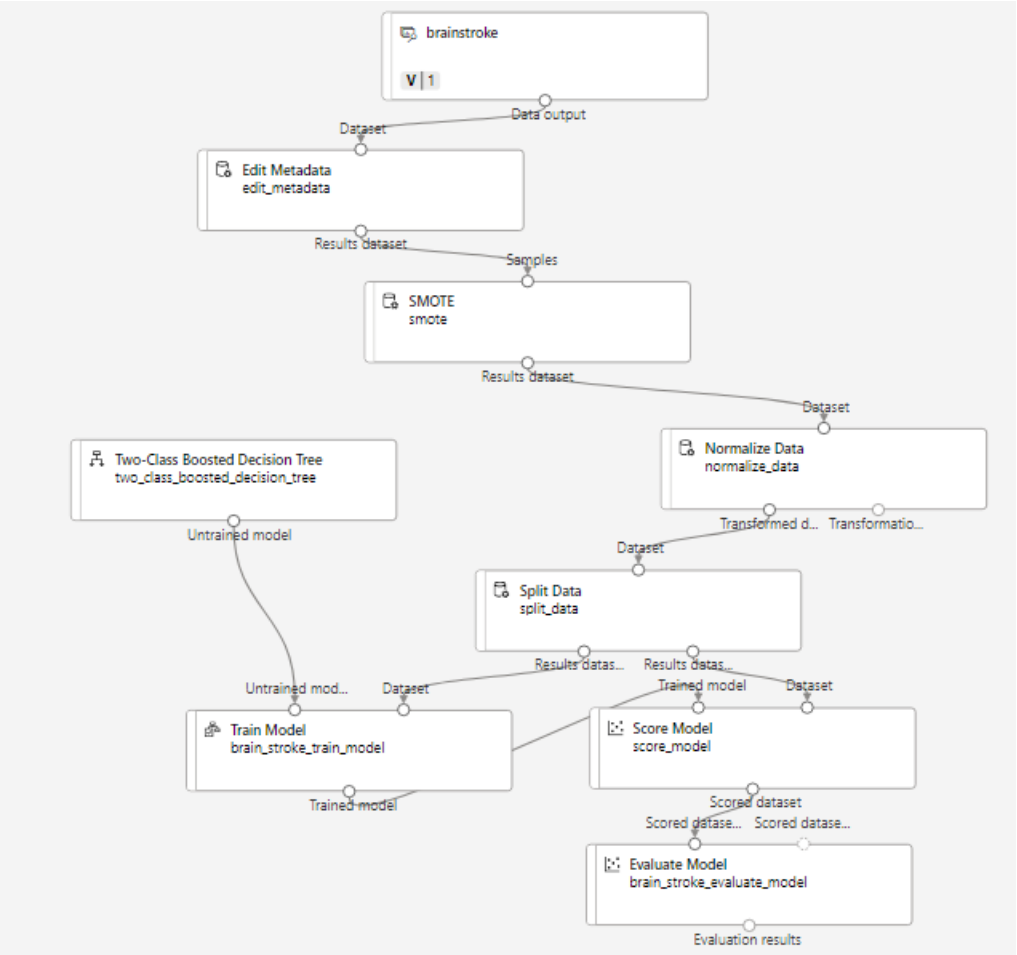
Split Data komponenta služi za razdvajanje podataka u skup za treniranje i skup za testiranje. Skup podataka podijeljen je omjerom 70:30, pri čemu je 70% podataka korišteno za treniranje moela, a preostalih 30% podataka za testiranje.

Train Model komponenta se koristi za treniranje klasifikacijskog ili regresijskog modela. Za stupac oznake odabran je moždani udar – stupac koji sadrži rezultate koje model koristi za treniranje.

Score Model komponenta se koristi za generiranje predviđanja pomoću treniranog klasifikacijskog ili regresijskog modela.

Evaluate Model komponenta se koristi za mjerenje točnosti treniranog modela. Na temelju pruženog skupa podataka, koji sadrži podatke generirane iz modela, komponenta izračunava skup metrika evaluacije industrijskog standarda.

Prije nego što je odabran konačni algoritam, koji je korišten unutar modela strojnog učenja, testirani su rezultati korištenjem različitih klasifikacijskih algoritama koje uključuju: šuma odluke, pojačano stablo odluke, logistička regresija te neuronska mreža. Na temelju rezultata, odabran je algoritam pojačanog stabla odluke. Konačni model strojnog učenja prikazan je na slici 4.4.



Slika 4.4 Model pojačanog stabla odluke

Prema [31], kako bi se provela objektivna evaluacija različitih atributa u skupu podataka, izračunata je preciznost, odziv, F1-mjera i točnost. U zadatku binarne klasifikacije, TP, FP, TN i FN označavaju točne pozitivne, lažne pozitivne, točne negativne i lažne negativne slučajeve.

TP označava broj pacijenata koji su dobili moždani udar i koje je model točno prepoznao (engl. *true positive*). TN označava broj pacijenata koji nisu dobili moždani udar i koje je model točno prepoznao (engl. *true negative*). FP označava broj pacijenata koji nisu dobili moždani udar, a model identificira kao da jesu (engl. *false positive*). FN označava broj pacijenata koji su dobili moždani udar, a model ih identificira kao da nisu (engl. *false negative*). Matrica zabune prikazana je tablicom 4.1.

Tablica 4.1 Matrica zabune

		Stvarni podaci	
		Dobio/la moždani udar	Nije dobio/la moždani udar
Predviđeni podaci	Dobio/la moždani udar	TP	FP
	Nije dobio/la moždani udar	FN	TN

Preciznost (engl. *precision*) se dobije jednadžbom (4-1), a definira se kao omjer točno predviđenih primjera u odnosu na ukupne predviđene primjere.

$$Preciznost = \frac{TP}{(TP + FP)} \tag{4-1}$$

Odziv (engl. *Recall*) se dobije jednadžbom (4-2), a predstavlja omjer točno predviđenih primjera u odnosu na sve primjere u identificiranoj klasi.

$$Odziv = \frac{TP}{(TP + FN)} \tag{4-2}$$

F1-mjera (engl. *F1-Score*) se dobije jednadžbom (4-3), a definira se kao harmonijska sredina preciznosti i odziva.

$$F1 - mjera = \frac{2 * Preciznost * Odziv}{(Preciznost + Odziv)} \tag{4-3}$$

Točnost (engl. *Accuracy*) se dobije jednadžbom (4-4), a definira se kao omjer točno predviđenih primjera u odnosu na ukupni broj primjera.

$$Točnost = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (4\text{-}4)$$

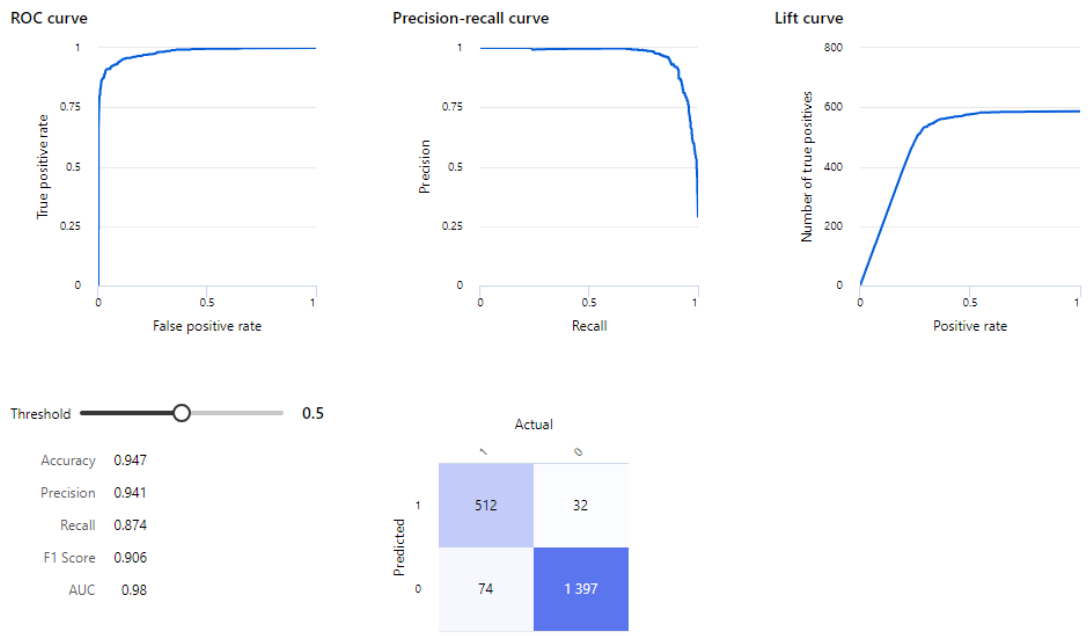
Tablica 4.2 prikazuje vrednovanje razmatranih modela dobivenih uz pomoć razvojnog okruženja *Microsoft Machine Learning Studio*.

Tablica 4.2 Vrednovanje razmatranih modela

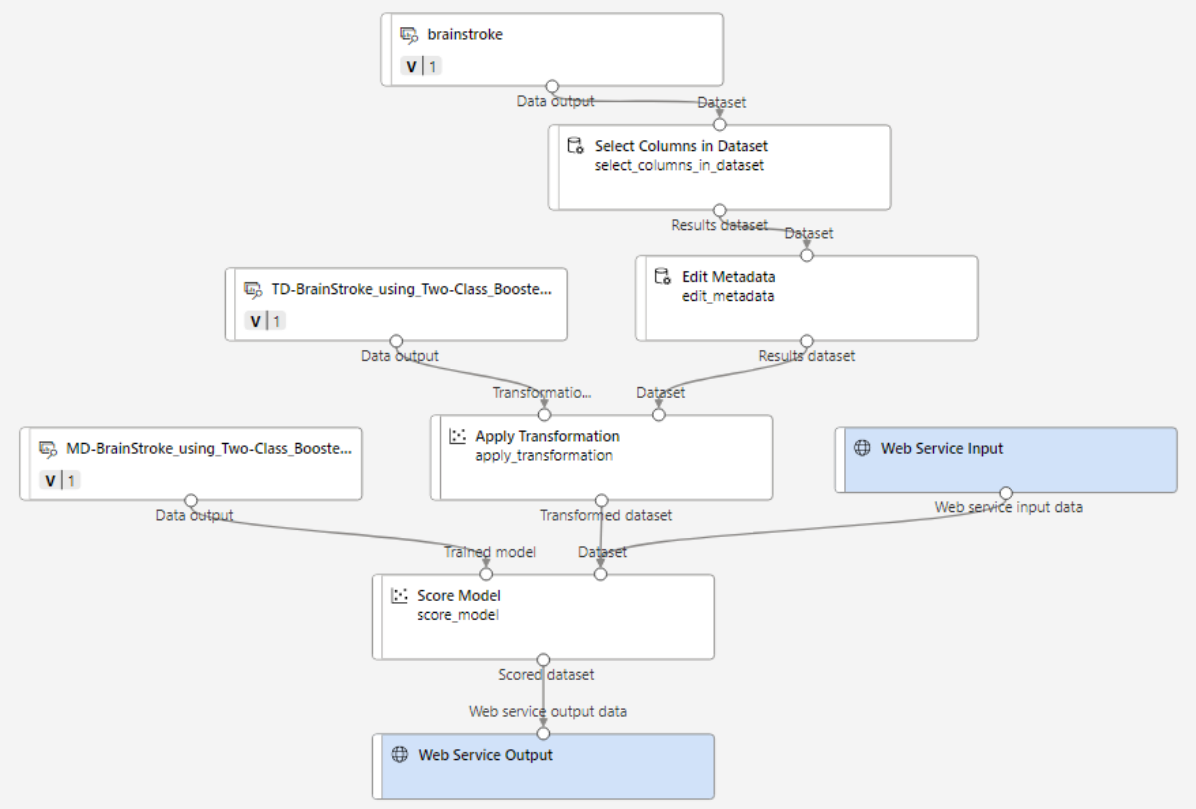
	Šuma odluke	*Neuronska mreža*	*Pojačano stablo odluke*
Preciznost	0.867	0.8213	0.9412
Odziv	0.9232	0.6706	0.8737
F1-mjera	0.8942	0.6859	0.9062
Točnost	0.9365	0.8213	0.9474

Rezultati evaluacije modela koji se koristi unutar aplikacije prikazani su na slici 4.5. S područjem ispod ROC (engl. *receiver operating characteristic*) krivulje od 0.98, model pokazuje visoku sposobnost razlikovanja između pozitivnih i negativnih klasa. Na grafu preciznosti i odziva, preciznost od 0.941 i odziv od 0.874 pokazuju da model održava dobru ravnotežu između točnosti predviđanja pozitivnih slučajeva i sposobnosti prepoznavanja svih stvarnih pozitivnih slučajeva. Točnost modela od 0.947 i F1-mjera od 0.906 dodatno potvrđuju njegovu sveukupnu učinkovitost.

Nakon što je razvijen model za procjenu rizika na moždani udar, pokreće se proces objave navedenog treniranog modela kako bi bio korišten za predviđanja u stvarnom vremenu. Rezultat navedene operacije je verzija modela koja je implementirana i izložena putem API-ja za interakciju u stvarnom vremenu. Infrastruktura koja upravlja zahtjevima prema inferencijskoj krajnjoj točki (engl. *Inference endpoint*) i obrađuje predviđanja prikazan je slikom 4.6.



Slika 4.5 Rezultati evaluacije implementiranog modela



Slika 4.6 Infrastruktura inferencije u stvarnom vremenu

Prije nego što se model implementirao u mobilnu aplikaciju, putem *API razvojnog okruženja Postman* testirana je dobivena krajnja točka slanjem POST zahtjeva kako bi se provjerilo kako

razvijeni model reagira na različite ulaze te kako izgleda odgovor koji će model vratiti. *Postmanu* se predaje adresa krajnje točke (pružena od strane *Microsoft Azure-a*), metoda HTTP zahtjeva (POST), zaglavlje (*API* ključ generiran od strane *Microsoft Azure-a*) te tijelo zahtjeva koje sadrži podatke koje će se slati modelu za obradu. Programsko rješenje implementacije postupka procjene rizika na moždani udar prikazano je u potpoglavlju 4.6.2.

4.4. Postupak stvaranja preporuka za liječenje generativnom umjetnom inteligencijom

Postupak stvaranja preporuka za liječenje sastoji se od nekoliko koraka. Prvi korak je izrada upita koji se, u slučaju mobilne aplikacije za procjenu rizika na moždani udar, sastoji od poruke koja traži od modela da generira preporuke na temelju predanih podataka te podataka koji uključuju korisničke podatke iz profila i podatke koje je korisnik unio na zaslonu unosa zdravstvenih podataka.

Nakon što se upit sastavio, šalje se *OpenAI* modelu putem HTTP POST zahtjeva na određenu API adresu krajnje točke (engl. *endpoint*). Poslani zahtjev sastoji se od upita, API ključa generiranog od strane *OpenAI* platforme te dodatnih parametara poput odabranog modela koji će obraditi zahtjev, duljine poruke te broj odgovara.

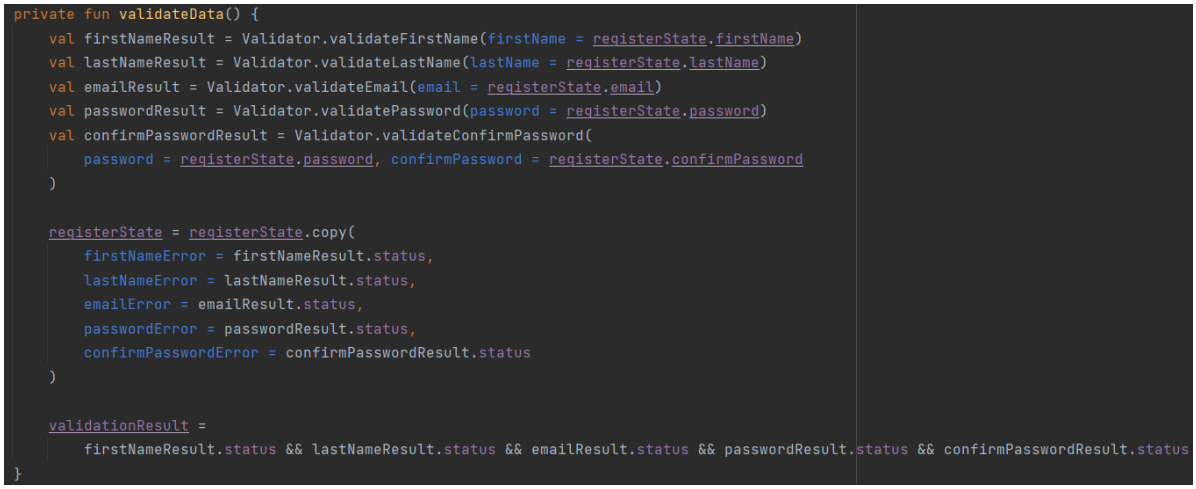
Nakon što je upit poslan, OpenAI model generira preporuku koja se vraća kao odgovor unutar aplikacije. Navedeni odgovor se zatim, nakon spremanja u bazu podataka, dohvaća i obrađuje kako bi bio prikladno prikazan unutar aplikacije.

Programsko rješenje implementacije postupka stvaranja preporuka za liječenje generativnom umjetnom inteligencijom prikazano je u potpoglavlju 4.6.3.

4.5. Programsko rješenje na strani korisnika

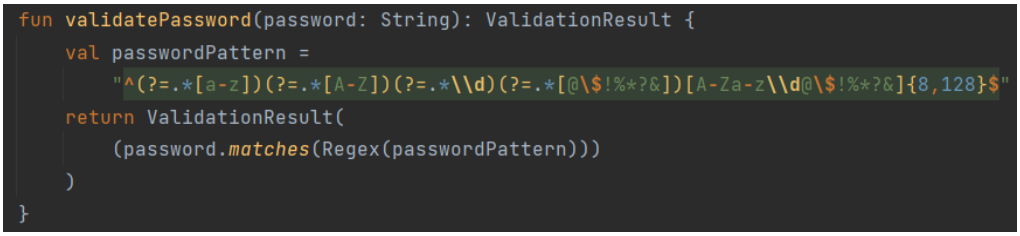
4.5.1. Registriranje korisnika

Prilikom unosa podataka potrebnih za izradu računa, okida se metoda *validateData*, prikazana slikom 4.7, koja provjerava ispravnost unesenih podataka. Ukoliko uneseni podaci nisu ispravni, korisniku se prikazuje odgovarajuća poruka ispod polja za unos. Tipka za registriranje je omogućena tek kada su svi unosi ispravni. Primjer validacije zaporke prikazan je slikom 4.8.



```
private fun validateData() {
    val firstNameResult = Validator.validateFirstName(firstName = registerState.firstName)
    val lastNameResult = Validator.validateLastName(lastName = registerState.lastName)
    val emailResult = Validator.validateEmail(email = registerState.email)
    val passwordResult = Validator.validatePassword(password = registerState.password)
    val confirmPasswordResult = Validator.validateConfirmPassword(
        password = registerState.password, confirmPassword = registerState.confirmPassword
    )

    registerState = registerState.copy(
        firstNameError = firstNameResult.status,
        lastNameError = lastNameResult.status,
        emailError = emailResult.status,
        passwordError = passwordResult.status,
        confirmPasswordError = confirmPasswordResult.status
    )

    validationResult =
        firstNameResult.status && lastNameResult.status && emailResult.status && passwordResult.status && confirmPasswordResult.status
}
```

Slika 4.7 Implementacija metode *validateData*



```
fun validatePassword(password: String): ValidationResult {
    val passwordPattern =
        "^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[@\$!%*?&])[A-Za-z\\d@\$!%*?&]{8,128}$"
    return ValidationResult(
        (password.matches(Regex(passwordPattern)))
    )
}
```

Slika 4.8 Primjer validacije zaporke

Pritiskom na tipku „*Register*“ poziva se metoda *register* koja poziva metodu *createUser*, prikazana slikom 4.9, predajući joj unesenu adresu e-pošte i zaporku.

```
private fun createUser(email: String, password: String) {
    registerStatusLoading = true
    FirebaseAuth.getInstance().createUserWithEmailAndPassword(
        email, password
    ).addOnCompleteListener { task ->
        if (task.isSuccessful) {
            registerStatusLoading = false
            saveUser(
                registerState.firstName,
                registerState.lastName,
                registerState.email,
                registerState.password
            )
            requestStatusMessage = "Registration successful!"
        } else {
            registerStatusLoading = false
            requestStatusMessage = "Registration failed!"
        }

    }.addOnFailureListener { it: Exception
        registerStatusLoading = false
        requestStatusMessage = "Registration failed!"
    }
}
```

Slika 4.9 Implementacija metode *createUser*

Ukoliko adresa e-pošte nije prije registrirana, *Firebase Authentication* stvara novog korisnika i dodaje ga u bazu podataka. Također, metodom *saveUser* prikazanoj na slici 4.10, korisnički podaci se spremaju u *Realtime Database* pod njegovim jedinstvenim ključem. Prije nego se spreme korisnički podaci, zaporka računa se šifrira koristeći metodu *hashPassword*. Nakon uspješnog registriranja, čisti se stanje zaslona za registriranje, korisniku se ispisuje poruka o rezultatu te ga se usmjerava na zaslon prijave.

```
private fun saveUser(firstName: String, lastName: String, email: String, password: String) {

    val hashedPassword = hashPassword(password)

    val user = User(firstName, lastName, email, hashedPassword)
    val currentUserId = FirebaseAuth.getInstance().currentUser?.uid

    currentUserId?.let { it: String
        db.child(it).setValue(user)
    }?.addOnCompleteListener { task ->
        if (task.isSuccessful) {
            resetState()
            AppRouter.navigateTo(Screen.LoginScreen)
        }
    }

}
```

Slika 4.10 Spremanje korisničkih podataka u *Realtime Database*

4.5.2. Prijava korisnika

Prilikom unosa podataka potrebnih za prijavu u aplikaciju, vrši se ista validacija podataka kao na zaslonu za registriranje. Pritiskom na tipku „*Login*“ poziva se metoda *login* prikazana slikom 4.11.

```
private fun login() {
    loginStatusLoading = true
    FirebaseAuth.getInstance().signInWithEmailAndPassword(loginState.email, loginState.password)
        .addOnCompleteListener { task ->
            if (task.isSuccessful) {
                resetState()
                loginStatusLoading = false
                AppRouter.navigateTo(Screen.HomeScreen)
            } else {
                loginStatusLoading = false
                errorMessage = "Failed to connect to the db."
            }

        }.addOnFailureListener { it: Exception
            loginStatusLoading = false
            errorMessage = "Invalid credentials."
        }
}
```

Slika 4.11 Implementacija metode *login*

Prijava u korisnički račun odvija se metodom *signInWithEmailAndPassword* kojoj se kao parametri predaju unesena adresa e-pošte i zaporka. Ukoliko je prijava uspješna, poziva se

metoda *resetState* koja čisti stanje zaslona za prijavu te se korisnika usmjerava na glavni zaslon aplikacije.

4.5.3. Izrada korisničkog profila

Prije nego što korisnik može započeti s unosom zdravstvenih podataka, treba izraditi korisnički profil. Kada korisnik prvi put otvori profil, prikazuju mu se prazna polja koja treba popuniti. Tipka za spremanje podataka omogućena je samo ako su svi elementi označeni ili popunjeni i kada korisnik upali način uređivanja. Kada se pritisne tipka za spremanje podataka poziva se metoda *saveProfileData*, prikazana slikom 4.12, kojoj se prosljeđuju podaci profila za spremanje. Nakon izrade profila, korisniku se prikazuju njegovi podaci koji se dohvaćaju prilikom svakog odlaska na zaslon profila metodom *getProfileData* prikazanoj na slici 4.13.

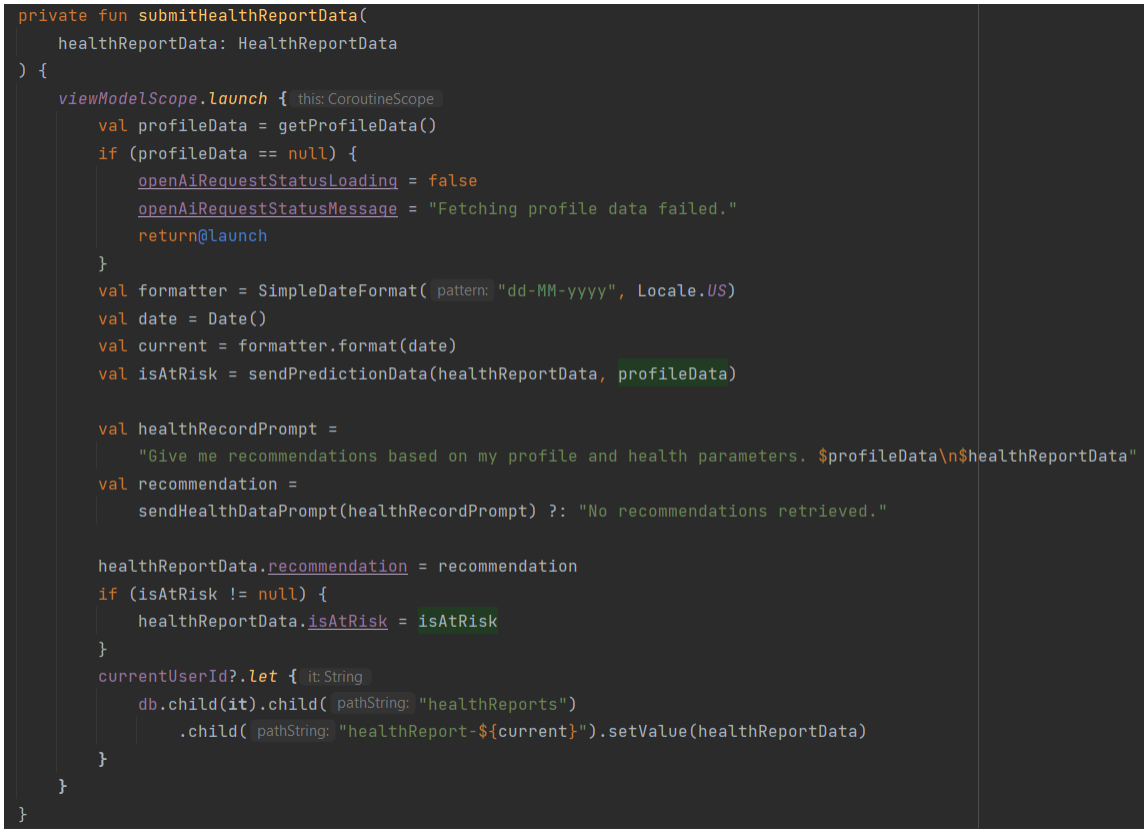
```
private fun saveProfileData(
    gender: String,
    age: String,
    heartDisease: String,
    maritalStatus: String,
    residenceType: String,
    workType: String,
    smokingStatus: String
) {

    val profileData = ProfileData(
        gender, age.toInt(), heartDisease, maritalStatus, residenceType, workType, smokingStatus
    )

    currentUserId?.let { it: String
        db.child(it).child( pathString: "profile").setValue(profileData)
    }
    isProfileSetUp = true
    isEditMode = false

}
```

Slika 4.12 Spremanje korisničkog profila

Slika 4.13 Dohvaćanje korisničkog profila

4.5.4. Unos zdravstvenih podataka

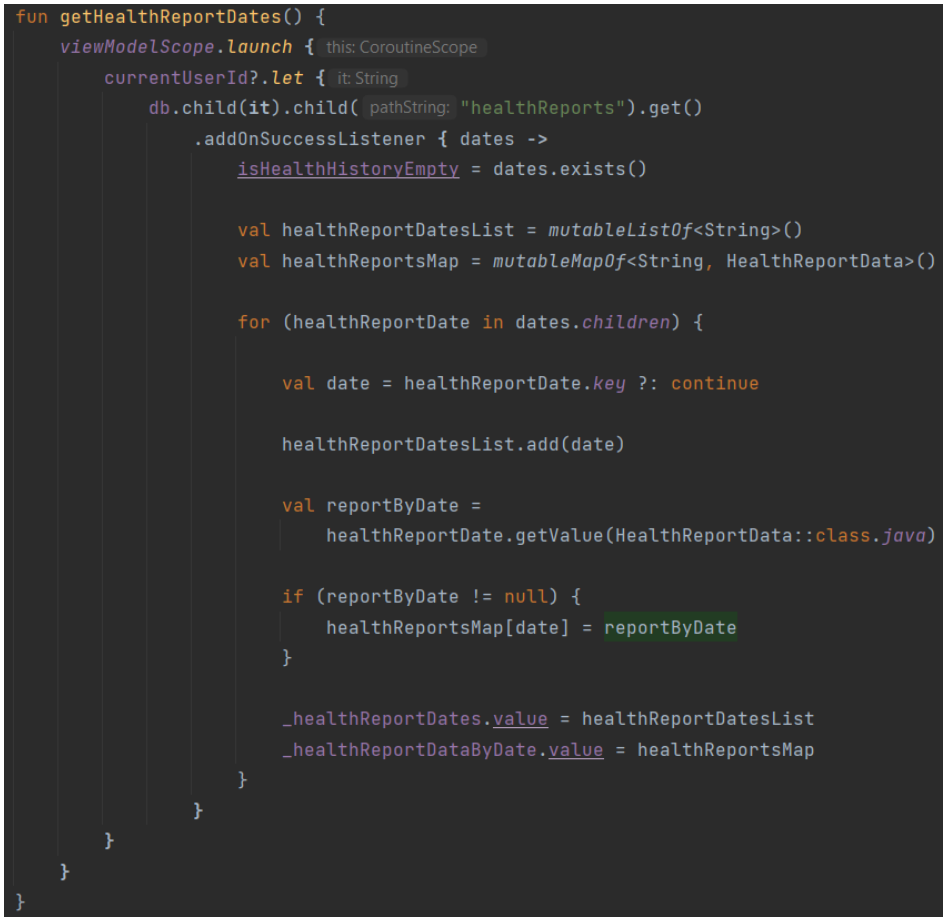
Nakon što je korisnik izradio profil, može krenuti s unosom zdravstvenih podataka. Zdravstveni podaci se mogu unijeti proizvoljno puta. Kada korisnik otvori zaslon unosa zdravstvenih podataka, prikazuju mu se prazna polja koja treba popuniti. Tipka za pohranu podataka je omogućena samo ako su unosi svih polja ispravni. Ukoliko korisnik unese neispravnu vrijednost u bilo koje polje, prikazuje mu se odgovarajuća poruka greške. Kada korisnik pritisne tipku pohrane, poziva se metoda *submitHealthReportData*, prikazana slikom 4.14, kojoj se predaju zdravstveni podaci iz forme te nakon čega se vrši procjena rizika na moždani udar, a nakon toga slanje upita OpenAi-ju. Metoda *submitHealthReportData* pokreće asinkrono izvršavanje navedenih zadataka unutar *ViewModel-a* koristeći korutine, kako se ne bi blokiralo glavnu nit korisničkog sučelja. Dok korisnik čeka da se zadaci završe, prikazuje mu se pokazatelj obrade zahtjeva.



Slika 4.14 Implementacija metode *submitHealthReport*

4.5.5. Pregled povijesti unosa podataka

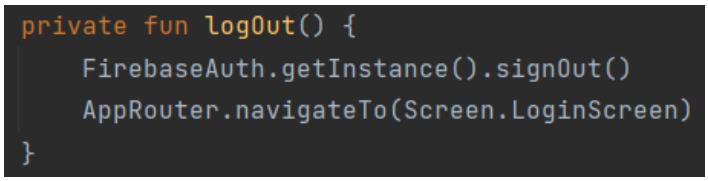
Korisnik svoje rezultate može pregledati u bilo kojemu trenutku odlaskom na zaslon povijesti. Prilikom svakog odlaska na zaslon povijesti, poziva se metoda *getHealthReportDates* kako bi se unosi mogli odgovarajuće ažurirati. Ako korisnik nema prijašnjih unosa, ispisuje mu se odgovarajuća poruka. Metoda *getHealthReportDates*, prikazana na slici 4.15, prikuplja datume unosa te njihove pripadajuće podatke kako bi se mogli proslijediti zaslonu detalja rezultata na koji se usmjerava pritiskom na tipku kartice rezultata.



Slika 4.15 Dohvaćanje unosa zdravstvenih podataka

4.5.6. Odjava korisnika

Pritiskom na tipku „*Log out*“, poziva se metoda *logOut* koja korisnika odjavljuje iz korisničkog računa te ga se usmjerava na zaslon prijave gdje se može ponovno prijaviti s postojećim ili drugim korisničkim računom. Metoda *logOut* prikazana je slikom 4.16.



Slika 4.16 Implementacija metode *logOut*

4.5.7. Ponovno postavljanje zaporke

Ukoliko postoji unesena adresa e-pošte, pritiskom na tipku slanja poziva se metoda *sendPasswordResetEmailRequest*, prikazana na slici 4.17, unutar koje *Firebase Auth* na navedenu adresu e-pošte šalje poveznicu na kojoj se nalazi upute za ponovno postavljanje zaporke.

```
private fun sendPasswordResetEmailRequest(email: String, onComplete: (String) -> Unit) {
    isRequestStatusLoading = true
    FirebaseAuth.getInstance().sendPasswordResetEmail(email).addOnCompleteListener { task ->
        isRequestStatusLoading = false
        requestStatusMessage = if (task.isSuccessful) {
            "If an account is created with that e-mail address, a link to reset password will be sent!"
        } else {
            "Invalid e-mail address."
        }
        onComplete(requestStatusMessage)
    }
}
```

Slika 4.17 Implementacija metode za ponovno postavljanje zaporke

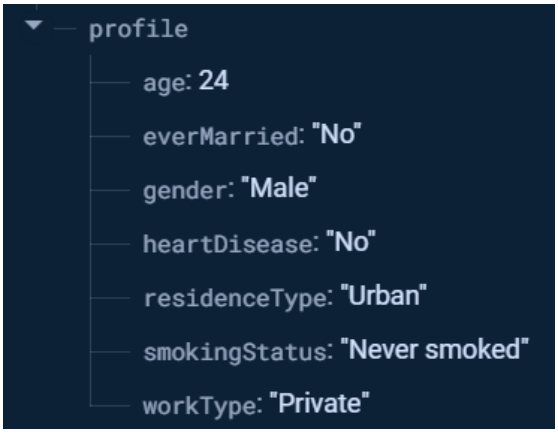
4.6. Programsko rješenje na strani poslužitelja

4.6.1. Pohrana podataka u Realtime Database

Nakon što je korisnik izradio račun, njegovi osnovni podaci su spremljeni u *Realtime Database* pod „*Users*“ granom pod njegovim jedinstvenim identifikatorom. Slika 4.18 prikazuje arhitekturu osnovnih pohranjenih podataka u *Realtime Databaseu*.

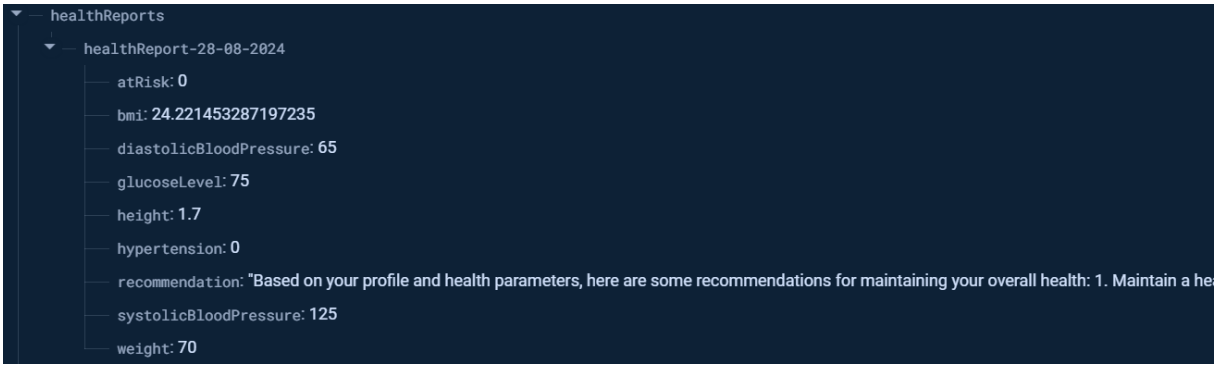
Slika 4.18 Izgled pohranjenih podataka korisničkog računa u *Realtime Database*

Nakon što korisnik prvi put napravi račun, izrađuje se nova pod grana imena „*profile*“. Svakim sljedećim odlaskom u profil, ako korisnik odluči izmijeniti neki od podataka, nakon pritiska tipke „*Save*“, ažurira se grana profila. Grana profila prikazana je slikom 4.19.



Slika 4.19 Izgled pohranjenih podataka korisničkog profila u *Realtime Database*

Kada korisnik krene s unosom zdravstvenih podataka, izrađuje se nova pod grana imena „*healthReports*“. Unutar navedene grane, svakim unosom zdravstvenih podataka kreira se pod grana koja u imenu ima *healthReport* i datum unosa. Grana *healthReports* prikazana je slikom 4.20.



Slika 4.20 Izgled pohranjenih podataka korisničkih unosa zdravstvenih podataka u *Realtime Database*

4.6.2. Procjena rizika na moždani udar modelom strojnog učenja

Kada korisnik pritisne na tipku „*Submit*“, nakon pohrane zdravstvenih podataka u *Realtime Database*, poziva se metoda *sendPredictionData* kojoj se prosljeđuju zdravstveni podaci te podaci iz profila. Prije nego što se pošalje zahtjev na *Azure-ovu* krajnju točku, određeni podaci se mapiraju kako bi odgovarali podacima iz skupa podataka prema kojemu je rađen model strojnog učenja. Nakon slanja zahtjeva, ako je odgovor uspješan, iz odgovora se izvlači rezultat procjene rizika – *scored_labels*. Ovisno o rezultatu zahtjeva, korisniku se ispisuje odgovarajuća poruka na ekran. Opisana metoda procjene rizika prikazana je na slikama 4.21 i 4.22.

```
private suspend fun sendPredictionData(
    healthReportData: HealthReportData,
    profileData: ProfileData
): Int? {
    azureRequestStatusLoading = false
    return try {
        val input = listOf(
            AzureInputData(
                stroke = 0,
                gender = profileData.gender,
                age = profileData.age,
                hypertension = healthReportData.hypertension,
                heart_disease = if (profileData.heartDisease == "Yes") 1 else 0,
                ever_married = profileData.everMarried,
                work_type = when (profileData.workType) {
                    "Working with children" -> "children"
                    "Government job" -> "Govt_job"
                    "Never worked" -> "Never_worked"
                    "Private" -> "Private"
                    else -> "Self_employed"
                },
                Residence_type = profileData.residenceType,
                avg_glucose_level = healthReportData.glucoseLevel,
                bmi = healthReportData.bmi,
```

Slika 1.21 Procjena rizika za moždani udar - 1.dio

```
                smoking_status = when (profileData.smokingStatus) {
                    "Formerly smoked" -> "formerly_smoked"
                    "Never smoked" -> "never_smoked"
                    "Currently smoking" -> "smokes"
                    else -> "Unknown"
                }
            )
        )
        val azureInputs = AzureInputs(input)
        val azureInputRequest = AzureInputRequest(azureInputs)
        val response = AzureRetrofitInstance.getRetrofitInstance()
            .predictBrainStroke(azureAuthHeader, azureInputRequest)

        if (response.isSuccessful) {
            azureRequestStatusLoading = false
            openAiRequestStatusMessage = "Prediction successful."
            response.body()?.Results?.WebServiceOutput0?.firstOrNull()?.scored_labels?.toInt()

        } else {
            azureRequestStatusLoading = false
            azureRequestStatusMessage = "Failed to retrieve the prediction."
            null
        }
    } catch (e: Exception) {
        azureRequestStatusLoading = false
        azureRequestStatusMessage = "Failed to retrieve the prediction."
        null
    }
}
```

Slika 4.22 Procjena rizika za moždani udar - 2.dio

4.6.3. Postupak stvaranja preporuka generativnom umjetnom inteligencijom

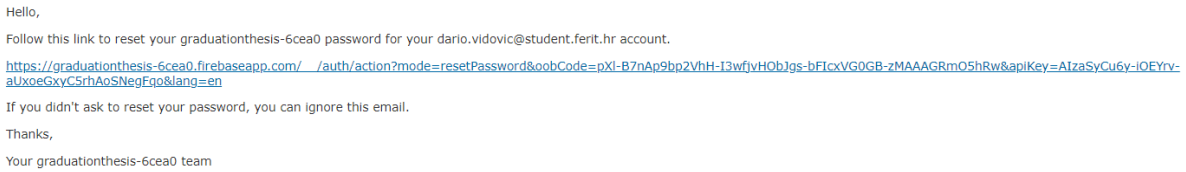
Nakon što se završi procjena rizika na moždani udar, priprema se upit za *OpenAI*. Upit za *OpenAI* se sastoji od poruke koja uključuje korisničke unesene zdravstvene podatke te podatke iz profila. Zahtjev koji se šalje *OpenAI-u* sastoji se od sljedećih podataka: *OpenAI* model koji će se koristiti za generiranje preporuka, poruka koja se šalje modelu – upit s ulogom korisnika, broj tokena koji model može generirati kao odgovor te broj ponuđenih odgovora. Nakon što se izradi zahtjev, šalje se *OpenAI-u*. Ovisno o rezultatu zahtjeva, korisniku se ispisuje odgovarajuća poruka na ekran. Opisana metoda slanja upita, koja sadržava opisanu proceduru generiranja preporuka – *sendHealthDataPrompt*, prikazana je slikom 4.23.

```
private suspend fun sendHealthDataPrompt(prompt: String): String? {
    openAiRequestStatusLoading = true
    return try {
        val request = OpenAiRequest(
            model = "gpt-3.5-turbo",
            messages = listOf(Message(role = "user", content = prompt)),
            max_tokens = 1000,
            n = 1
        )
        val response =
            OpenAiRetrofitInstance.getRetrofitInstance().getRecommendation(authHeader, request)
        if (response.isSuccessful) {
            openAiRequestStatusLoading = false
            openAiRequestStatusMessage = "Data saved. Check history for recommendations."
            response.body()?.choices?.firstOrNull()?.message?.content?.trim()

        } else {
            openAiRequestStatusLoading = false
            openAiRequestStatusMessage = "Failed to retrieve the recommendations"
            null
        }
    } catch (e: Exception) {
        openAiRequestStatusLoading = false
        openAiRequestStatusMessage = "Failed to retrieve the recommendations"
        null
    }
}
```

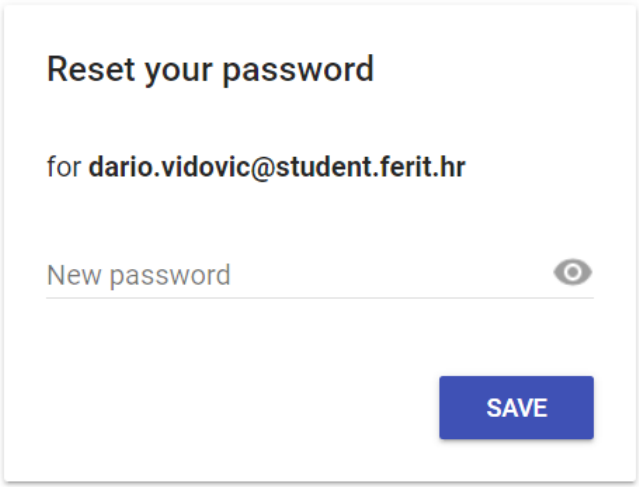
Slika 4.23 Generiranje poruka

4.6.4. Ponovno postavljanje zaporke

Kada korisnik pošalje zahtjev za ponovno postavljanje zaporke, ukoliko adresa e-pošte postoji, *Firebase Auth* šalje korisniku poveznicu prikazanu na slici 4.24. Nakon što korisnik otvori poveznicu prikazuje mu se forma prikazana na slici 4.25. Nakon unosa nove zaporke i pritiska na tipku „*Save*“, korisnik se može prijaviti u račun s novo postavljenom zaporkom.



Slika 4.24 Poveznica za ponovno postavljanje zaporke



Slika 4.25 Forma za ponovno postavljanje zaporke

5. PRIKAZ RADA I NAČIN KORIŠTENJA MOBILNE APLIKACIJE

Ovo poglavlje prikazuje način korištenja mobilne aplikacije prolazeći kroz sve zaslone prikazujući implementirane funkcionalnosti. Isto tako, testira se rad aplikacije unosom različitih korisničkih podataka.

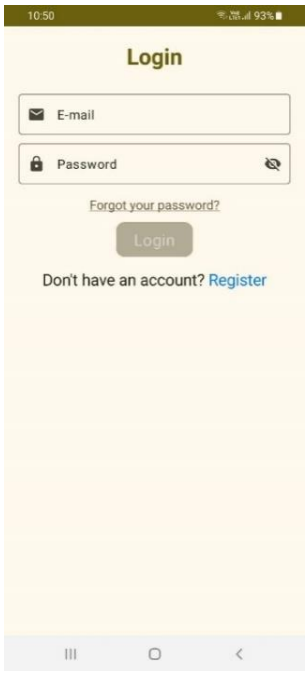
5.1. Način korištenja mobilne aplikacije

Prilikom otvaranja mobilne aplikacije korisniku se pojavljuje zaslon prijave sa poljima za unos adrese e-pošte i zaporke, porukom za ponovno postavljanje zaporke te porukom za registriranje. Ukoliko je korisnik zaboravio zaporku svog računa, odabirom poruke „*Forgot your password?*“ odlazi na zaslon ponovnog postavljanja zaporke koji sadrži polje za unos adrese e-pošte te tipke za slanje zahtjeva. Nakon unosa adrese e-pošte i pritiska tipke „*Submit*“, ukoliko postoji račun s unesenom adresom, korisnik na adresi e-pošte dobiva poveznicu na kojoj može ponovno postaviti zaporku. Ukoliko korisnik nema izrađen račun u aplikaciji, odabirom poruke „*Register*“ odlazi na zaslon registriranja gdje se za izradu računa treba unijeti ime, prezime, adresa e-pošte, zaporka te potvrda zaporke. Nakon uspješnog izrađenog računa, korisniku se otvara zaslon prijave gdje unosom podataka računa prijavljuje se u aplikaciju. Kada se korisnik uspješno prijavi u aplikaciju, otvara mu se glavni zaslon aplikacije koji se sastoji od naslova dobrodošlice te sljedećih kartica: profil, dodavanje zdravstvenih podataka, povijest te odjava iz aplikacije. Za unos zdravstvenih podataka, korisnik treba prvo izraditi račun pritiskom na tipku kartice „*Profile*“. Kada korisnik prvi put otvori zaslon profila, treba popuniti svoje osobne podatke. Nakon popunjavanja podataka profila, svakim sljedećim odlaskom na zaslon profila, korisnik ima opciju ažurirati svoje podatke pritiskom na tipku „*Edit*“. Nakon što je profil izrađen, korisnik može popunjavati zdravstvene podatke. Pritiskom na tipku kartice „*Add health report*“, korisniku se otvara zaslon na kojemu popunjava formu zdravstvenih podataka. Nakon popunjavanja forme i pritiska tipke „*Submit*“, korisniku se na temelju osobnih i zdravstvenih podataka generira procjena rizika na moždani udar te odgovarajuće preporuke. Uvid u navedene podatke, zajedno sa unesenim zdravstvenim podacima, korisnik ima na zaslonu povijesti na kojeg može doći pritiskom na tipku kartice „*History*“. Ako korisnik želi izraditi novi korisnički račun ili odjaviti se iz postojećeg, to može učiniti pritiskom na tipku kartice „*Log out*“.

5.2. Ispitivanje rada mobilne aplikacije

5.2.1. Prijava korisnika

Prilikom pokretanja mobilne aplikacije, prikazuje se početni zaslon aplikacije – zaslon prijave, prikazan na slici 5.1. U slučaju da je korisnik izgubio pristup zaporci svog računa, može je ponovno postaviti odlaskom na zaslon ponovnog postavljanja zaporke, prikazan na slici 5.2, pritiskom na poruku „*Forgot your password?*“.

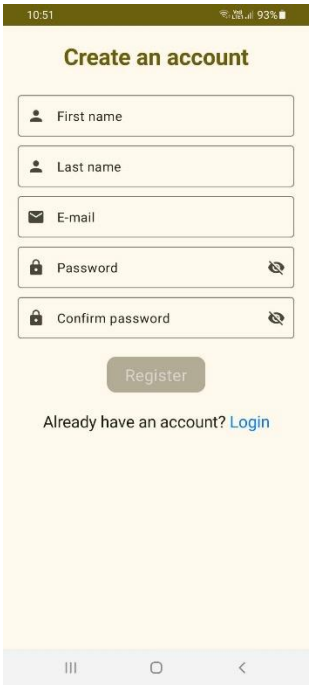


Slika 5.1 Zaslon prijave

Slika 5.2 Zaslon ponovnog postavljanja zaporke

5.2.2. Registriranje korisnika

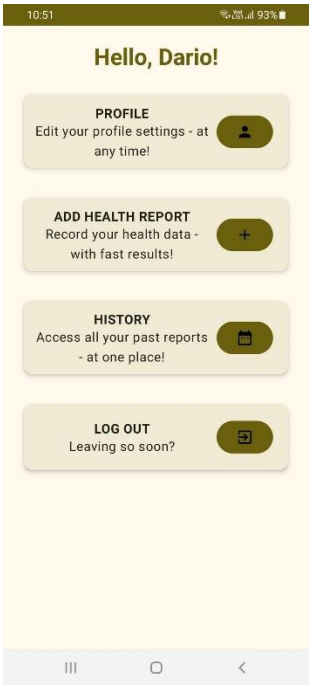
Ukoliko korisnik nije prethodno izradio korisnički račun, može ga izraditi na zaslonu registriranja, koji je prikazan na slici 5.3, na koji se može doći sa zaslone prijave pritiskom na „*Register*“ dio poruke. Nakon unosa podataka i pritiska na tipku „*Register*“, ako je registriranje uspješno, korisnika se usmjerava na zaslon prijave.



Slika 5.3 Zaslon registriranja

5.2.3. Izrada korisničkog profila

Nakon što se korisnik prijavio u aplikaciju, otvara se glavni zaslon aplikacije prikazan na slici 5.4. Prije nego što korisnik može započeti s unosom zdravstvenih podataka, treba izraditi profil. Zaslon izrade profila prikazan je slikama 5.5 i 5.6.



Slika 5.4 Glavni zaslon aplikacije

Slika 5.5 Zaslon izrade profila - 1.dio

Slika 5.6 Zaslon izrade profila - 2.dio

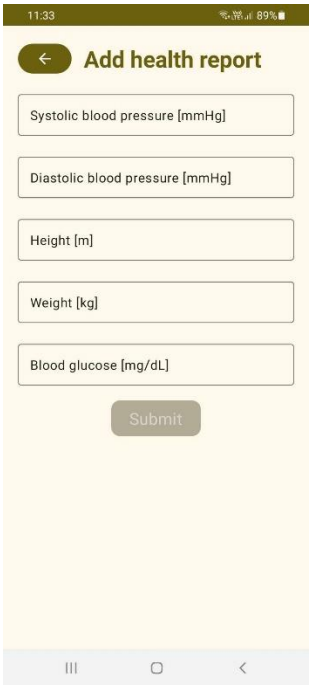
Nakon što je korisnik popunio i spremio osobne podatke, svakim sljedećim odlaskom u profil prikazuju mu se njegovi podaci koje može ažurirati u bilo kojem trenutku pritiskom na tipku „*Edit*“. Pritiskom na tipku „*Edit*“, forma za unos podataka omogućuje korisniku izmjenu bilo kojeg od prikazanih parametara. Zaslon profila s popunjenim podacima prikazan je slikama 5.7 i 5.8.

Slika 5.7 Zaslon profila - 1. dio

Slika 5.8 Zaslon profila - 2. dio

5.2.4. Unos zdravstvenih podataka

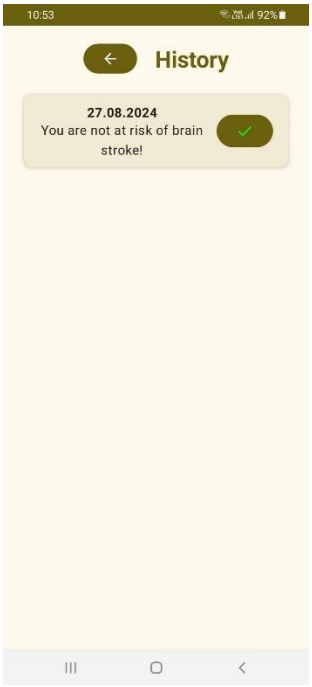
Nakon što je korisnik izradio profil, može započeti s unosom zdravstvenih podataka. Na zaslon unosa zdravstvenih podataka, prikazan na slici 5.9, usmjerava se korisnika pritiskom na tipku kartice „*Add health report*“. Nakon što korisnik unese zdravstvene podatke i pritisne na tipku „*Submit*“, korisniku se prikazuje poruka o uspješnosti obrade zahtjeva.



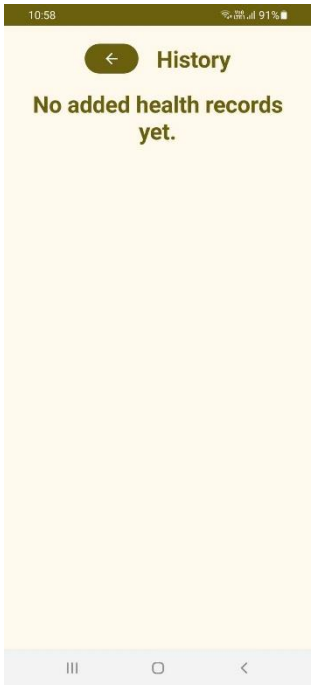
Slika 5.9 Zaslon unosa zdravstvenih podataka

5.2.5. Pregled povijesti unosa podataka

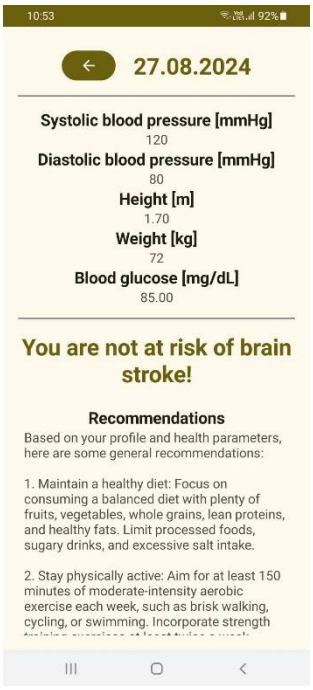
Na zaslon povijesti rezultata, prikazan na slici 5.10, usmjerava se korisnika pritiskom na tipku kartice „*History*“. Ukoliko korisnik nije napravio ni jedan unos podataka, prikazuje mu se poruka „*No added health records yet.*“ prikazana na slici 5.11. Ukoliko korisnik ima zabilježene unose podataka prikazuju se kartice rezultata koje se sastoje od datuma, procjene rizika te tipke koja vodi na detalje rezultata. Zaslon detalja rezultata prikazan je na slikama 5.12 i 5.13.



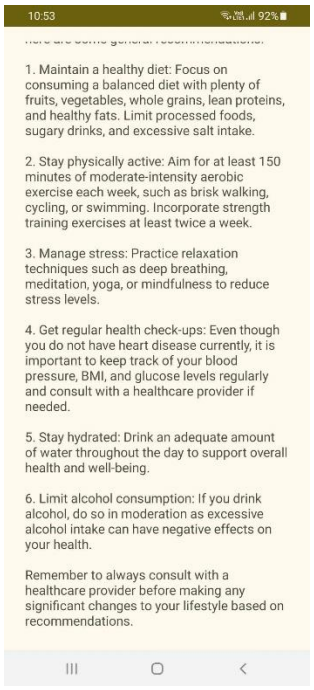
Slika 5.10 Zaslon povijesti sa zabilježenim podacima



Slika 5.11 Zaslon povijesti bez zabilježenih podataka



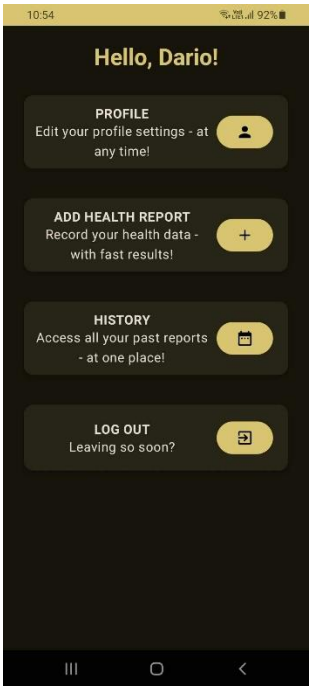
Slika 5.12 Zaslon rezultata 1.dio



Slika 5.13 Zaslon rezultata 2.dio

5.2.6. Tamni način rada aplikacije

Aplikacija podržava svijetli i tamni način rada. Ukoliko korisnik promijeni temu mobilnog uređaja na tamni način rada, boje unutar aplikacije odgovarajuće se ažuriraju. Primjer jednog od zaslona aplikacije u tamnom načinu rada prikazan je slikom 5.14.



Slika 5.14 Glavni zaslon aplikacije u tamnom načinu rada

5.3. Analiza programskog rješenja i rezultata ispitivanja mobilne aplikacije

5.3.1. Ispitni slučaj 1

Podaci korišteni za prvi ispitni slučaj prikazani su tablicama 5.1 i 5.2. Pregled rezultata za navedene podatke prikazan je prema slikama 5.15 i 5.16.

Tablica 5.1 Podaci korisničkog profila za 1. ispitni slučaj

PODACI KORISNIČKOG PROFILA	
Spol	M
Dob	67
Srčana bolest	Da
Bio/la u braku	Da
Tip posla	Privatna djelatnost
Tip prebivališta	Urbano područje
Status pušenja	Bivši pušač

Tablica 5.2 Zdravstveni podaci za 1. ispitni slučaj

ZDRAVSTVENI PODACI	
Hipertenzija	Ne
Razina glukoze u krvi [mg/dL]	228.69
Indeks tjelesne mase	36.48

Slika 5.15 Rezultati za 1. ispitni slučaj 1.dio

Slika 5.16 Rezultati za 1. ispitni slučaj 2.dio

Procjena rizika na moždani udar i generirana preporuka odgovaraju očekivanim rezultatima temeljenim na osobnim i zdravstvenim unesenim podacima.

5.3.2. Ispitni slučaj 2

Podaci korišteni za prvi ispitni slučaj prikazani su tablicama 5.3 i 5.4. Pregled rezultata za navedene podatke prikazan je prema slikama 5.17 i 5.18.

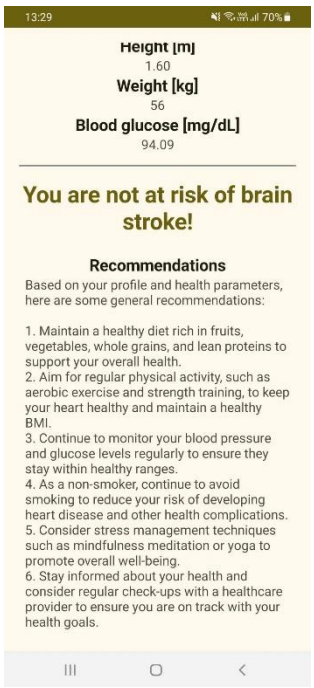
Tablica 5.3 Podaci korisničkog profila za 2. ispitni slučaj

PODACI KORISNIČKOG PROFILA	
Spol	Ž
Dob	23
Srčana bolest	Ne
Bio/la u braku	Ne
Tip posla	Privatna djelatnost
Tip prebivališta	Urbano područje
Status pušenja	Nepušač

Tablica 5.4 Zdravstveni podaci za 2. ispitni slučaj

ZDRAVSTVENI PODACI	
Hipertenzija	Ne
Razina glukoze u krvi [mg/dL]	94.09
Indeks tjelesne mase	21.87

Slika 5.17 Rezultati za 2. ispitni slučaj 1.dio



Slika 5.18 Rezultati za 2. ispitni slučaj 2.dio

Procjena rizika na moždani udar i generirana preporuka odgovaraju očekivanim rezultatima temeljenim na osobnim i zdravstvenim unesenim podacima.

5.3.3. Ispitni slučaj 3

Podaci korišteni za prvi ispitni slučaj prikazani su tablicama 5.5 i 5.6. Pregled rezultata za navedene podatke prikazan je prema slikama 5.19 i 5.20.

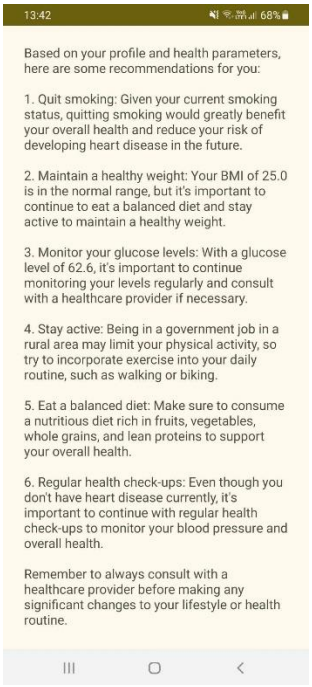
Tablica 5.5 Podaci korisničkog profila za 3. ispitni slučaj

PODACI KORISNIČKOG PROFILA	
Spol	M
Dob	32
Srčana bolest	Ne
Bio/la u braku	Da
Tip posla	Državni posao
Tip prebivališta	Ruralno područje
Status pušenja	Pušač

Tablica 5.6 Zdravstveni podaci za 3. ispitni slučaj

ZDRAVSTVENI PODACI	
Hipertenzija	Ne
Razina glukoze u krvi [mg/dL]	62.6
Indeks tjelesne mase	25

Slika 5.19 Rezultati za 3. ispitni slučaj 1.dio



Slika 5.20 Rezultati za 3. ispitni slučaj 2.dio

Procjena rizika na moždani udar i generirana preporuka odgovaraju očekivanim rezultatima temeljenim na osobnim i zdravstvenim unesenim podacima.

6. ZAKLJUČAK

U diplomskom radu opisan je način rada mozga, moždani udar te čimbenici rizika obolijevanja od moždanog udara. Nakon toga, osmišljena je i razvijena mobilna aplikacija za procjenu i praćenja rizika od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom. Procjena rizika za moždani udar odvija se klasifikacijskim postupkom pojačanog stabla odluke. Osim postupka pojačanog stabla, koristio se i postupak stabla odluke i postupak neuronske mreže. Nakon vrednovanja razmatranih modela, utvrdilo se da pojačano stablo odluke daje najbolje rezultate.

Aplikacija je razvijena unutar integrirane razvojne okoline Android Studio zajedno s jezikom Kotlin. Pomoću Kotlina, implementirane su sve opisane funkcionalnosti mobilne aplikacije dok je za izradu samog dizajna aplikacije korišten Jetpack Compose. Spremanje podataka realizirano je korištenjem baze podataka *Firebase Realtime Database*, a upravljanje autentikacijom korisnika osigurano je od strane *Firebase Authentication*. Za izradu modela strojnog učenja, koji procjenjuje rizik na moždani udar, korištene su usluge Microsoft Azure platforme, a za generiranje preporuka koristi se API OpenAI modela.

Ispitivanje rada mobilne aplikacije provedeno je kroz tri korisnička slučaja prilikom kojih se testirala procjena rizika na različite unesene parametre te preporuke koje generira OpenAI model sukladno korisničkom sveukupnom zdravstvenom stanju. Rezultati korisničkih slučajeva upućuju da aplikacija ispravno procjenjuje i prati rizik od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom.

Aplikacija je izrađena s fleksibilnošću koja omogućava jednostavne dorade poput prilagodbe modela OpenAI API-ja ili korištenje drugih klasifikacijskih postupaka strojnog učenja kako bi procjena rizika bila preciznija. Također, aplikacija se može dodatno unaprijediti implementacijom podrške za više jezika.

LITERATURA

[1] mozak. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2013. – 2024. [24.6.2024.]

[2] Mayfield Clinic, Anatomy of the Brain, dostupno na: https://mayfieldclinic.com/pe-anatbrain.htm [23.6.2024]

[3] Cleveland Clinic, Stroke, dostupno na: https://my.clevelandclinic.org/health/diseases/5601-stroke [23.6.2024.]

[4] J.C. Grotta, G.W. Albers, J.P. Broderick, A.L. Day, S.E. Kasner, E.H. Lo, R.L. Sacco, L.K.S. Wong, Stroke: Patophysiology, Diagnosis, and Management: Seventh edition, Elsevier, United States of America, 2021.

[5] National Heart, Lung, and Blood Institute, Stroke: Causes and Risk Factors, dostupno na: https://www.nhlbi.nih.gov/health/stroke/causes

[6] Wikipedia, Ada Health, dostupno na: https://en.wikipedia.org/wiki/Ada_Health [22.6.2024]

[7] Ada, What Does the Emergence of Large Language Models like ChatGPT Mean for Healthcare and AI?, dostupno na: https://ada.com/editorial/what-llms-mean-for-healthcare/ [23.6.2024]

[8] SciSpace, How Does Ada Health Use ML and NLP?, dostupno na: https://typeset.io/questions/how-does-ada-health-use-ml-and-nlp-39z8eurf30 [22.6.2024]

[9] Mobilna aplikacija, Ada – Check your health, dostupno na: https://play.google.com/store/apps/details?id=com.ada.app&hl=en

[10] Britannica Money, Facebook, dostupno na: https://www.britannica.com/money/Facebook [22.6.2024]

[11] Facebook, How Does Facebook Use Artificial Intelligence to Moderate Content?, dostupno na: https://www.facebook.com/help/1584908458516247 [22.6.2024]

[12] Facebook Business, Good Questions, Real Answers: How Does Facebook Use Machine Learning to Deliver Ads?, dostupno na: https://www.facebook.com/business/news/good-questions-real-answers-how-does-facebook-use-machine-learning-to-deliver-ads [22.6.2024.]

[13] AltexSoft, Nonfunctional Requirements in Software Engineering: Examples, Types, Best Practices, dostupno na: https://www.altexsoft.com/blog/non-functional-requirements/ [21.6.2024.]

[14] Healthline, Everything You Need to Know About High Blood Pressure (Hypertension), dostupno na: https://www.healthline.com/health/high-blood-pressure-hypertension#definition [21.6.2024.]

[15] CDC, About Adult BMI, dostupno na: https://www.cdc.gov/healthyweight/assessing/bmi/adult_bmi/index.html [21.6.2024.]

[16] IBM, What is a decision tree?, dostupno na: https://www.ibm.com/topics/decision-trees [31.8.2024]

[17] Builtin, Random Forest: A Complete Guide for Machine Learning, dostupno na: https://builtin.com/data-science/random-forest-algorithm [31.8.2024]

[18] A.F. Bulagang, N.G. Weng, J. Mountstephens, and J. Teo, "A Review of Recent Approaches for Emotion Classification Using Electrocardiography and Electrodermography Signals," Informatics in Medicine Unlocked, sv. 20, str. 100363, 2020.

[19] IBM, What is a neural network?, dostupno na: https://www.ibm.com/topics/neural-networks [30.8.2024]

[20] S. Haykin, Neural Networks and Learning Machines, 3rd ed., Pearson, New Jersey, 2008.

[21] Spiceworks, What Is Android OS? History, Features, Versions, and Benefits, dostupno na: https://www.spiceworks.com/tech/tech-general/articles/android-os/ [30.8.2024]

[22] Android Developers, Meet Android Studio, dostupno na: https://developer.android.com/studio/intro [30.8.2024]

[23] A. Leiva, Kotlin for Android Developers, Leanpub, 2017.

[24] GeeksforGeeks, Basics of Jetpack Compose in Android, dostupno na: https://www.geeksforgeeks.org/basics-of-jetpack-compose-in-android/ [30.8.2024]

[25] TechTarget, Microsoft Azure, dostupno na: https://www.techtarget.com/searchcloudcomputing/definition/Windows-Azure [30.8.2024]

[26] H. Li, Introduction to Windows Azure: An Introduction to Cloud Computing Using Microsoft Windows Azure, Apress, USA, 2009.

[27] Coursera, What Is OpenAI? Everything You Need to Know, dostupno na: https://www.coursera.org/articles/what-is-openai [30.8.2024]

[28] GeeksforGeeks, Firebase – Introduction, dostupno na: https://www.geeksforgeeks.org/firebase-introduction/ [30.8.2024]

[29] Korišteni skup podataka, dostupan na: https://www.kaggle.com/datasets/zzettrkalpakbal/full-filled-brain-stroke-dataset

[30] N. Alturki, A. Altamimi, M. Umer, O. Saidani, A. Alshardan, et al., "Improving Prediction of Chronic Kidney Disease Using KNN Imputed SMOTE Features and Trionet Model," Computer Modeling in Engineering & Sciences, sv. 139, br. 3, str. 3519, 2024.

[31] M. S. Pathan, Z. Jianbiao, D. John, A. Nag and S. Dev, "Identifying Stroke Indicators Using Rough Sets," in IEEE Access, sv. 8, str. 210323, 2020.

SAŽETAK

U ovom diplomskom radu razvijena je aplikacija za procjenu i praćenje rizika klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom. Mobilna aplikacija ostvarena je pomoću integrirane razvojne okoline Android Studio zajedno s jezikom *Kotlin*. Za izradu korisničkog sučelja korišten je okvir *Jetpack Compose*. Za izradu korisničkog računa korišten je *Firebase Auth,* a za pohranu osobnih i zdravstvenih podataka *Firebase Realtime Database*. Za klasifikaciju su korišteni postupci šume odluke, pojačanog stabla odluke i neuronske mreže te je na temelju rezultata vrednovanja za implementaciju odabran algoritam pojačanog stabla odluke. Procjena rizika odvija se putem modela izrađenog unutar *Microsoft Azure Machine Learning Studija*, a generiranje preporuka slanjem upita *OpenAi* modelu. Pohranjeni korisnički osobni podaci zajedno s unesenim zdravstvenim podacima koriste se pri procjeni rizika od moždanog udara i generiranju preporuka od strane *OpenAi-ja*. Ispitivanje mobilne aplikacije provedeno je kroz razne korisničke slučajeve prilikom čega je testirana i utvrđena njena točnost korištenjem različitih korisničkih podataka.

Ključne riječi: generativna umjetna inteligencija, klasifikacija, mobilna aplikacija, moždani udar, strojno učenje, procjena rizika.

ABSTRACT

Development of a mobile application for the assessment and monitoring of stroke risk using machine learning classification and generative AI

In this thesis, an application for risk assessment and monitoring using classification machine learning techniques and generative artificial intelligence has been developed. The mobile application was created using the Android Studio integrated development environment along with the Kotlin programming language. The user interface was designed using the Jetpack Compose framework. For user account management, Firebase Auth was used, while personal and health data are stored in Firebase Realtime Database. Classification was performed using decision tree, boosted decision tree, and neural network techniques, with the boosted decision tree algorithm being selected based on the results. Risk assessment is conducted through a model developed within Microsoft Azure's Machine Learning Studio, and recommendations are generated by querying the OpenAI model. Stored user personal data, along with entered health data, are used in the risk assessment for stroke and in generating recommendations by OpenAI. The mobile application was tested through various user cases, evaluating its accuracy using different user data.

Keywords: generative artificial intelligence, classification, mobile aplication, stroke, machine learning, risk assessment.

PRILOZI

Prilog 1. Diplomski rad „Razvoj mobilne aplikacije za procjenu i praćenje rizika od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom“ u *.docx* formatu

Prilog 2. Diplomski rad „Razvoj mobilne aplikacije za procjenu i praćenje rizika od moždanog udara klasifikacijskim postupcima strojnog učenja i generativnom umjetnom inteligencijom“ u *.pdf* formatu

Prilog 3. Programski kod aplikacije

POPIS SLIKA

POPIS TABLICA