

**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Kristina Ivatović

**Analitika velikih podataka kao
višeagentni sustav**

DIPLOMSKI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU

**FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Kristina Ivatović

Matični broj: 46321/17-R, 0016096556

Studij: Baze podataka i baze znanja

Analitika velikih podataka kao višeagentni sustav

DIPLOMSKI RAD

Mentor:

Izv. prof. dr. sc. Markus Schatten

Varaždin, kolovoz 2020.

Kristina Ivatović

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

„Znanje je moć.“

Fransis Bejkon

„Znanje nije određište, ono je proces.“

Albert Einstein

Zahvale :

Zahvalila bih svom mentoru Izv. prof. dr. sc. Markusu Schattenu koji je svojim predavanjima, radom i primjerima otkrio čari podataka. Hvala na svim smjernicama i podršci.

Zahvalila bih cijelom Fakultetu organizacije i informatike na dobivenom znanju, prilikama i potporama.

Hvala svima koji su vjerovali u mene i onda kad ja nisam.

Hvala vam!

Sažetak

U ovom radu bit će objašnjeni i opisani veliki podaci, njihovo korištenje i prisutnost u današnje vrijeme. Objašnjena je usko povezana analiza velikih podatak, uz koju se može doći do korisnih rezultata na temelju prikupljenih podataka. Zatim je opisan višeagentni sustav, njegove karakteristike, primjene i kako je došlo do njih.

Nakon teorijskog dijela rada nalazi se detaljno opisani praktični dio rada, gdje se sustav za analizu radi po logističkoj regresiji, koja je opisana. Svaki korak je detaljno opisan od same izrade, do pokretanja sustava u korištenom alatu. Za praktični dio korišten je *Google Colaboratory* i biblioteka *scikit-learn* pomoću kojih je izgrađen sustav za ovaj rad. Prikazane su tri varijante analize velikih podataka.

Ključne riječi: veliki podaci; baze podatak; analiza podataka; agenti; višeagentni sustav; umjetna inteligencija; logistička regresija; treniranje;

Sadržaj

Sadržaj.....	iv
1. Uvod.....	1
2. Veliki podaci.....	2
2.1. Definicija velikih podataka.....	2
2.2. Karakteristike velikih podataka.....	3
2.3. Primjena velikih podataka.....	4
3. Analiza velikih podataka.....	5
3.1. Proces analize velikih podataka.....	5
3.2. Razvoj tehnologija.....	6
4. Višeagentni sustav.....	7
4.1. Karakteristike višeagentnih sustava.....	10
4.1.1. Rukovođenje.....	10
4.1.2. Funkcija odlučivanja.....	11
4.1.3. Heterogenost.....	11
4.1.4. Parametri dogovaranja.....	11
4.1.5. Uzimanje kašnjenja u obzir.....	11
4.1.6. Topologija.....	12
4.1.7. Učestalost prijenosa podataka.....	12
4.1.8. Mobilnost.....	12
4.2. Agenti.....	13
4.2.1. Primjena.....	14
4.3. Umjetna inteligencija.....	14
4.3.1. Povijest umjetne inteligencije.....	16
4.3.2. Umjetna inteligencija u znanostima.....	16
5. Distribuirani sustavi.....	17
5.1. Korištenje distribuiranih sustava.....	18
5.1.1. Paralelni sustavi.....	18

5.2. Komunikacija unutar distribuiranog sustava	19
6. Praktični dio rada	20
6.1. Korištene tehnologije.....	20
6.1.1. Programski jezik Python.....	20
6.1.2. Alati.....	20
6.1.3. scikit-learn.....	21
6.1.4. Podaci.....	21
6.2. Logistička regresija	22
6.3. Proces izgradnje	23
6.3.1. Dodavanje biblioteka	24
6.3.2. Dodavanje podataka	24
6.3.3. Kodiranje podataka	25
6.3.4. Podjela podataka	26
6.3.5. Skaliranje	26
6.3.6. Treniranje.....	27
6.3.7. Predviđanje	27
6.3.8. Izrada matrice	28
6.3.9. Rezultati	28
6.3.9.1. Grafički prikaz.....	29
6.3.9.2. Validacija točnosti pomoću k-Fold Cross	30
6.4. Prikaz sustava u Google Colaboratory	31
6.4.1. Dodavanje podataka u Google Colaboratory	31
6.4.2. Pokretanje sustava u Google Colaboratory	32
6.5. Primjena korištenja	33
6.5.1. Podaci iz prvog seta podataka	34
6.5.2. Podaci iz drugog seta podataka	34
6.5.3. Podaci iz trećeg seta podataka	34
6.6. Prikaz rezultata	35
6.6.1. Rezultati na temelju prvog seta podataka	36

6.6.2. Rezultati na temelju drugog seta podataka.....	37
6.6.3. Rezultati na temelju trećeg seta podataka	38
7. Zaključak	39
Popis literature	40
Popis slika	43
Popis tablica	44
Prilozi	45

1. Uvod

Ovaj rad bavi se analitikom velikih podataka pomoću višeagentnog sustava. Kako bi svaki dio ovog rada bio što jasnije definiran podijeljen je na dva dijela, u prvom opisan je teorijski dio koji je potrebno objasniti i opisati prije drugog dijela rada koji je praktični dio. Opisani su veliki podaci njihove karakteristike i njihova primjena. Analiza velikih podataka je opisana, kao i njen proces i sam razvoj. Na koncu teorijskog dijela su višeagentni sustav i distribuirani sustavi, koji su detaljno opisani. Praktični dio ovog rada sadrži se od opisa tehnologija i alata s kojima se služi, detaljno opisan pristup analize podataka pomoću logističke regresije. Proces izgradnje opisan je prema koracima koji su potrebni za izgradnju sustava. Prikazan je rad u *Google Colaboratory*, kako bi se rezultati izgrađenog sustava mogli prikazati i na koncu opisati.

Tema ovog diplomskog rada značajna je jer se može primijetiti u mnogim područjima gdje je potrebno analizirati podatke. U današnje vrijeme tehnologija se brzo razvija i ima sve veće mogućnosti, tako da analiza pomoću višeagentnog sustava je samo mali pomak u tom razvoju umjetne inteligencije.

Diplomski rad odabran je na ovu temu kako bi se pristupilo učenju sustava, promatranjem njegovih rezultata na temelju podatak iz okoline koji su na raspolaganju.

2. Veliki podaci

Razvoje tehnologije, interneta raste i razmijene podataka. Prema Blodget (2014) 1956. godine uređaji za pohranu podataka su se iznajmljivali, bili su glomazni, imali su tek 5 megabajta, a težili su oko 900 kilograma. Danas podatke možemo spremiti na male memorijske kartice koje su manje i lakše od kovanice, a mogućnost pohrane je velika, pogotovo ako se uspoređuje s prvim uređajima. Također postoji mogućnost pohrane podataka i putem mreže, više o tome napisano je u radu Vujnovac, gdje kaže „Pohrana velike količine podataka na mreži popularizirala je takozvane 'računalne oblake' (engl. cloud computing). 'Računalni oblaci' organizacijama, ali i pojedincima, dopuštaju pohranu velike količine podataka“ Vujnovac (2015, str. 13) Zahvaljujući raznim mogućnostima za pohranom velikog broja podataka, došlo je do mogućnosti za Velike Podatke (*eng. Big Data*). Samim razvojem prikupljanja podataka, dolazi i do potrebe za razvoja drugih disciplina, kao što su analiza, učenje i slično.

2.1. Definicija velikih podataka

Prema Stepinac (2014) „Big Data je tehnologija koja omogućava prikupljanje i obradu velikih količina strukturiranih i nestrukturiranih podataka u realnom vremenu.“ Podaci mogu biti u raznim oblicima kao na primjer slike, tekst, videozapisi. Također mogu međusobno biti strukturirani, no veći postotak podataka danas je ne strukturiran.

Zahvaljujući internetu skupljanje podataka postalo je lakše, svaki korisnik za sobom ostavlja određene podatke. Bio korisnik svjestan ili ne ostavlja podataka za sobom, zahvaljujući njegovim navikama na internetu mogu se pronaći mnogi odgovori. Na temelju tih podataka moguće mu je predložiti neke promocije, bili to proizvodi, usluge ili prilike koje bi ga mogle zanimati. Podataka je sve više, potreba za analizom je sve veća i društveno korisna od znanstvenih istraživanja do društvenih. „Velika količina podataka i analitika spominje se unutar mnogih disciplinarnih skupina, a od kojih neke jesu: Statistika, ekonomija, sociologija, psihologija i ostalo.“ Stihović (2015, 1 str) Zahvaljujući velikim podacima moguće su razne analize, simulacije i razvoj. Mnoge poznate firme prepoznale su potencijale koje nude veliki podaci koji se kvalitetno obrade za potrebe rješavanja njihovih potreba za razvoj uspješnog poslovanja.

2.2. Karakteristike velikih podataka

Velikih podataka poznate su po tri karakteristike, prema Stepinac (2014) poznate su i pod „3V“ što bi bili :

- *eng. Volume* – Volumen – što obilježava veliku količinu podataka, koja se prikuplja, obrađuje kako bi podaci bili pogodni za analizu
- *eng. Velocity* – Brzina – što obilježava kontinuirano prikupljanje velike količine podataka, u realnom vremenu
- *eng. Variety* – Raznolikost – što obilježava raznolikost oblika i izvora podataka

Zahvaljujući svom volumenu, brzini i raznolikosti danas imamo mogućnost analizirati te podatke, te pomoću tih podataka možemo trenirati i graditi nove sustave. Zbog velikog volumena moguće je obrađivanje podataka s područja biologije, fizike i slično, gdje se znanstvenici susreću s velikom količinom podataka i ograničenjima obrade istih, kako ističe Batinica (2018). Prema podacima Morabito (2015) postoje još dvije dimenzije :

- Dostupnost – što obilježava veliki broj mogućnosti za razvoj podatkovnih izvora
- Istinitost – što obilježava kvalitetu, bitna je kod analitike

Postoje još neke varijacije na ovu podjelu karakteristika velikih podataka, ali najraširenije i poznatije u današnje vrijeme je zasigurno 3V.

Korijeni velikih podataka postavljeni su prije petnaestak godina i dalje se usavršavaju prema Kluk (2018), te postoje dvije vrste podataka:

- strukturirani podaci,
- nestrukturirani podaci.

Strukturirani podaci najčešće se skupljaju u kontroliranim slučajevima, kao na primjer u poduzećima i organizacijama. Do nestrukturiranih podataka dolazi iz ne organiziranih izvora. Također postoje i podaci koji su polustrukturirani, te se također mogu pronaći u velikim podacima.

Bitno je istaknuti za velike podatke i njihovih pet karakteristika po Schneider (2013) :

- Velika količinu informacija,
- Razni tipovi datoteka i formata,
- Sastavljeni iz različitih izvora,
- Zadržavaju se duže vremena,
- Upotrijebljeni od strane novih aplikacija.

2.3. Primjena velikih podataka

Veliki podaci koriste se za rješavanje raznih pitanja i predviđanja, s njima se radi u poslovnom svijetu te su neke tvrtke prepoznale njihovu prednost i iskoristile te pogodnosti za svoje uspješno poslovanje. Prema Kluk (2018) poduzeća koja primjenjuju mogućnosti velikih podataka su :

- CERN
 - Koristi podatke kod istraživanja.
- Netflix
 - Koristi podatke kako bi predložio sadržaj korisniku.
- Facebook
 - Koristi podatke kod promatranja korisničkih navika.
- Rolls-Royce
 - Koristi podatke kako bi poboljšao automobile

Platforme koje nude besplatno korištenje za rukovanje s velikim podacima bi prema Kumar (2018) bile sljedeće:

- Apache Hadoop
- Elasticsearch
- MongoDB
- Cassandra
- Apache HBase

Najpopularnija platforma je Apache Hadoop, koja se prema Stihović (2015) pretvara u operativni sustav za poslovanje. Hadoop nudi mogućnosti pokretanja različitih upita, operacija nad podacima koje sadrži, uz to ima veliki kapacitet za sve vrste podataka i dostupan je svima, prema Stihović (2015).

Veliki podaci sami za sebe mogu se skupljati na razne načine, no sami za sebe samo su velika hrpa podataka. Kako bi od njih bilo veće koristi potrebno ih je obraditi, a najčešće se nad njima vrši analiza podataka o kojoj će više biti u poglavlju koje slijedi.

3. Analiza velikih podataka

Sami veliki podaci za sebe, samo zauzimaju prostor, no uz dobro rukovanje s njima, tj. analizu i obradu može se doći do brojanih rezultata i odgovora, koji su potrebni. Prema radu Đorđević (bez dat.) analiza podataka se koristi u raznim akademskim disciplinama kao što su ekonomija, statistika, psihologija, sociologija te u prirodnim i društvenim znanostima. Zahvaljujući prikupljenim podacima i mogućnostima koje nam danas računala nude moguće je vršiti analize. Prema Vujnovac (2015) čovjek uz računala i alate koje ima, može obraditi velike podatke i to se naziva 'analitika velike količine podataka' (*engl. big data analytics*). Za kvalitetne rezultate potrebno je pripremiti svoje podatke za obradu, tj. slijediti proces analize, koji je u nastavku detaljnije opisan.

3.1. Proces analize velikih podataka

Postoje razni modeli po kojima se podaci mogu promatrati, za svaki set podataka koji se promatra može vrijediti drugačiji model, ovisno o podacima i traženim rezultatima koji se traže od određenog seta podataka s kojim se radi. Kako bi podatke na što kvalitetniji i bolji način mogli obraditi i iskoristiti njihovo prikupljanje potrebne su faze po Morabito(2015) koje navodi Stihović (2015):

- definiranje ciljeva analize,
- odabir podataka,
- procesiranje podataka,
- rudarenje podataka,
- evaluacija i interpretacija obrazaca,
- vizualizacija i povratne informacije

Prema ovim smjernicama dolazi se do rezultata analize podataka, te se pronalazi odgovor na postavljena pitanja, tj. na definirane ciljeve analize. Potrebno je dobro definirati ciljeve, pitanja na koja se traže odgovori, što je cilj jasnije postavljen to je cijeli proces lakši, tj. rezultat je bolji. Kad je cilj definiran, može se odabrati skupina podataka koja se promatra. Nakon toga mogu se procesirati, tj. obraditi podaci. Na temelju obrade podaci se mogu sortirati, grupirati po potrebi ili kako je ranije navedeno s njima se može rudariti. Poslije toga dolazi do zadnjih koraka gdje se na temelju rezultata dolazi do traženih odgovora pomoću pregleda i zaključaka, te na koncu preostaje sama vizualizacija i zaključivanje provedene analize.

Prema radu autora Vujnovac (2015) analiza dovodi do korisnik informacija :

„Analitika velike količine podataka odnosi se na upravljanje velikom količinom podataka, jer upravo 'analitika velike količine podataka' omogućuje da podatak postane korisna informacija. Analitika velike količine podataka odnosi se na procese koje je potrebno poduzeti u svrhu otkrivanja korisnih informacija u podatcima. Analitikom se pokušavaju utvrditi vrijedne korelacije među podatcima, 'skriveni obrasci' (engl. hidden patterns) te druge korisne informacije u svrhu donošenja odluka na temelju tih informacija.“ Vujnovac, (2015, str. 14).

Analizom velikih podataka može se doći do korisnih informacija koje mogu dati odgovore na razna pitanja, ili smjernice. Zahvaljujući analizi velikih podataka mnogi rezultati su postignuti, kao na primjer u političkim kampanjama, simulacije najvećeg svjetskog akcelerator čestica u CERNu. Također pomoću analize i daljnjeg korištenja rezultata moguća je i izgradnja novih sustava.

3.2. Razvoj tehnologija

Analiza velikih podataka koristi se na raznim područjima, odgovara na mnoga pitanja i donosi mnoga rješenja. Većina organizacija danas koristi mogućnosti koje nudi analiziranje, za svoje potrebe i razvoj. Kako potrebe postaju sve veće i potreba za raznim odgovorima sve bitnija, dolazi i do samog razvoja tehnologija vezanih za analizu velikih podataka. Autor Stihović za tehnologije analize velikih podataka navodi :

„tehnologije za obradu velike količine podataka napreduju velikom brzinom, te da je rizik upotrebe velike količine podataka iz dana u dan sve manji. Sa mnogim rastućim trendovima na područjima analitike i velike količine podataka, organizacije trebaju stvoriti uvjete u kojima će analitičari i znanstvenici moći eksperimentirati. Potrebno je pronaći način za testiranje i ocjenu, te uvođenje prototipa, a tako da bi se u budućnosti testirane tehnologije mogle integrirati u poslovanje.“ Stihović (2015, str. 16)

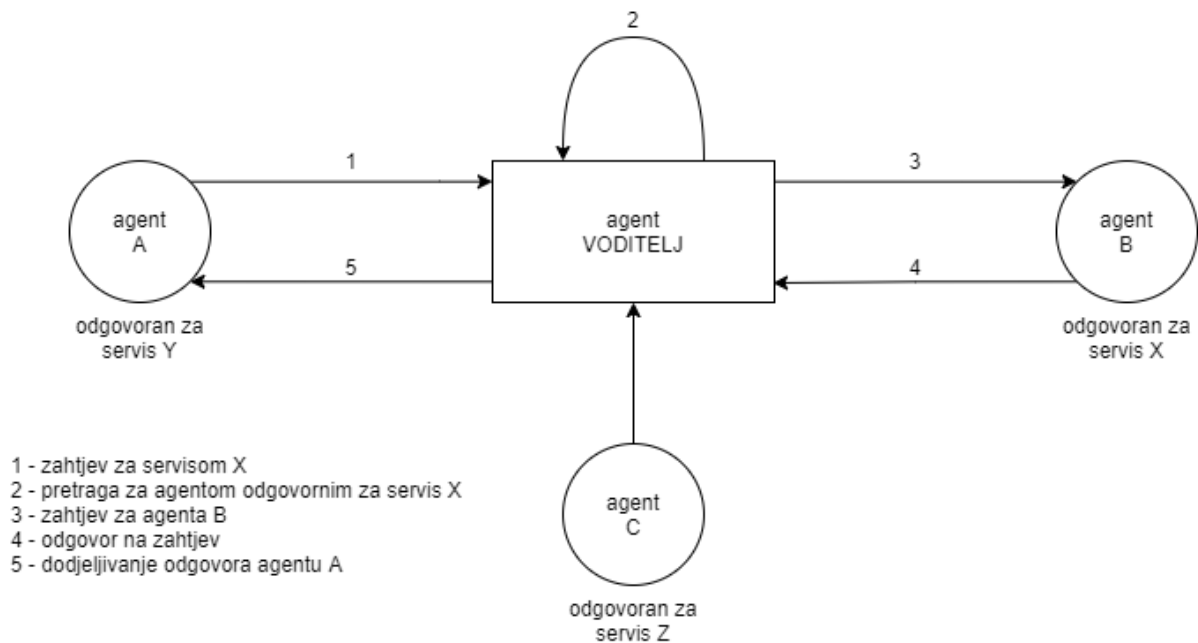
Pomoću velikih podataka dobiva se mogućnost učenja, treniranja novih sustava, te se otvara nova grana u informatičkom svijetu.

4. Višeagentni sustav

Sustav koji se sastoji od više agenata koji su međusobno interaktivni, kako što i sam naziv sustava navodi radi se o višeagentnom sustavu. Zbog potrebe za što većom automatizacijom dolazi do sve kompleksnijih zadataka, koji zbog toga dijele na manje zadatke, koji se zasebno rješavaju te njihovom suradnjom i rješavanjem dolazi se do traženih rezultata. „Mnoge računalne simulacije su implementirane pomoću višeagentnih sustava.“ Strahija, (2018, str. 25). Prema Dorri, Jurdak i Kanhere višeagentni sustavi prepoznatljivi su po svojoj učinkovitosti, niskoj potrošnji, fleksibilnosti i naravno pouzdanosti. Učinkovitost ovog sustava jer zbog podijele zadatka na manje zahtjeve, koji su raspodijeljeni među agentima, što rezultira manjom potrošnjom energije. Agent ima fleksibilnost da unatoč ne imanju znanja može riješiti svoj zadatak, ako je za njega namijenjen. Pouzdanost se prepoznaje u mogućnosti prebacivanja zadatka na agenta koji je u mogućnosti riješiti zadani problem. Više o prepoznatljivostima višeagentnih sustava piše autor Hlevnjak :

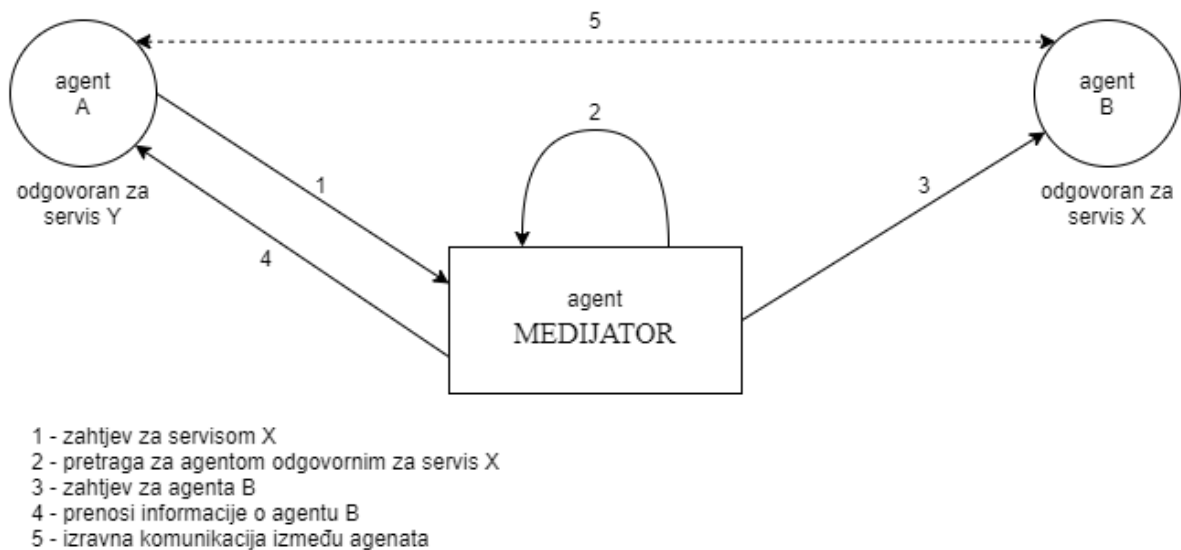
„Efikasnost leži u već ranije spomenutom razdvajanju zadatka na više manjih zadataka koji se dodjeljuju agentima. Raspodjela također donosi i smanjenju troškova jer se potrošnja energije, procesiranje i slično također dijeli na agente što ispada jeftinije nego kada bi cijeli problem trebala riješiti jedna moćna jedinica. Pouzdanost je također vezana uz ovakav pristup distributivnog rješavanja problema jer takav pristup omogućuje prebacivanje zadatka na druge agente u slučaju da jedan agent zakaže. Fleksibilnost predstavlja mogućnost agenta da riješi dodijeljeni zadatak bez obzira na razinu inicijalnih znanja.“ Hlevnjak, (2019, str 23).

Već je navedeno da agenti međusobno komunicira, i prosljeđuju zadatke. Agenti komuniciraju sa susjednim agentima, no to rješenje se unaprijedilo s uvođenjem središnjeg agenta (*eng. middle agent*). Postoje dva pristupa središnjeg agenta, kao voditelj ili kao medijator, prema Dorri, Jurdak i Kanhere. Na slikama 1. i 2. prikazani su središnji agenti svaki sa svojom ulogom, slike su napravljene po uzoru na slike iz rada Dorri, Jurdak i Kanhere. Protokol komunikacije agenta s podacima prema Huhnsu i Stephensu se dijele na pošiljatelja, primatelja, jezik samog protokola, funkcije odgovorne za kodiranje i dekodiranje i na koncu sama akcija primatelja. Protokol će se također moći prepoznati na slikama, gdje će cijeli sustav protokola za ovdje navedene komunikacije agenta biti prikazane.



Slika 1. Agent Voditelj

Slika 1. prikazuje središnjeg agenta voditelja, koji je jedini odgovoran za prijenos zahtjeva, odgovora, te može doći do opterećena. Jedan od mogućih scenarija prikazan je na slici. Svaki agent A, B i C odgovorni su za svoje servise Y, X i Z. Agent A šalje zahtjev za servisom X, te se taj zahtjev zaustavlja kod agenta Voditelja, koji zatim provjerava i pretražuje gdje je i tko odgovoran za traženi servis X. Pronalazi agenta B, te mu šalje zahtjev, na koji agent B odgovara. Tako da se odgovor šalje agentu Voditelju, koji to onda prosljeđuje agentu A, koji je poslao zahtjev.



Slika 2. Agent Medijator

Slika 2. prikazuje središnjeg agenta medijatora, koji prima zahtjeve, traži rezultate te daje informacije, no uz to omogućuje agentima izravnu komunikaciju. Jedan od mogućih scenarija između agenata prikazan je na slici. U ovom slučaju agenti A i B odgovorni su za svoje servise Y i X. Agent A šalje zahtjev za servisom X, te se taj zahtjev zaustavlja kod agenta Medijatora, koji zatim provjerava i pretražuje gdje je i tko odgovoran za traženi servis X. Pronalazi agenta B, te mu šalje zahtjev. Agent Medijator zatim šalje informacije o agentu B prema agentu A, koji je poslao zahtjev. Tako da agenti A i B mogu izravno komunicirati.

Kako bi protokoli donijeli zadovoljavajuće rezultate, potrebno je odrediti ciljeve, zadaci agenta trebaju biti raspodijeljeni da se izbjene konflikti te da se znanje agenta može postati jedna cjelina. Ovisno o kakvim zadacima se radi, tako se i mijenjaju pristupi i postavljanje načina na koje sustav funkcionira. Agenti međusobno imaju nekoliko vrsta poruka koje izmjenjuju, prema Huhnsu i Stephensu agenti izmjenjuju :

- Prijedlog načina djelovanja
- Prihvatanje načina djelovanja
- Odbacite način akcije
- Uklonite tijek akcije
- Ne slaganje s predloženim postupkom
- Protivljenje prijedloga postupka

4.1. Karakteristike višeagentnih sustava

Višeagentni sustav ima više kategorija, a u ovom radu bit će podijeljene na dvije vrste. Podijela na karakteristike okruženja višeagentnog sustava i na same karakteristike višeagentnih sustava. Karakteristike višeagentnih okruženja, koja su preuzeta iz rada Huhnsu i Stephensu.

- Okruženje pruža infrastrukturu koja određuje komunikaciju i protokole interakcije
- Obično je okruženje otvoreno.
- Okruženje sadrži agente koji mogu raditi samo za sebe ili surađivati međusobno

Višeagentni sustav ovisi o svom okruženju koje mu može pridonijeti u radu, no ovaj sustav ima osam svojih važnih karakteristika koje navode Dorri, Jurdak i Kanhere.

- Rukovođenje (*engl. leadership*)
- Funkcija odlučivanja (*engl. decision function*)
- Heterogenost
- Parametri dogovaranja (*engl. agreement parameters*)
- Uzimanje kašnjenja u obzir (*engl. delay consideration*)
- Topologija
- Učestalost prijenosa podataka (*engl. data transmission frequency*)
- Mobilnost

Svaka karakteristika ima svoju ulogu, te kako bi svaka od njih bila jasna u nastavku će svaka biti opisana.

4.1.1. Rukovođenje

Agent koji određuje zadatke i ciljeve za ostale agente u sustavu. Sustav ne mora nužno imati agenta koji je vođa, no u tom slučaju svaki agent samostalno odlučuje prema vlastitim ciljevima. Dok postoji vođa, on definira akcije za ostale agente ili drugim riječima sljedbenicima. Do vođe dolazi ili s predodređenosti za taj zadatak ili ga agenti sami odabiru. Također u sustavu ne mora biti samo jedan vođa, može ih biti više, ali međusobno moraju surađivati, kako bi zajedno mogli voditi sljedbenike.

4.1.2. Funkcija odlučivanja

Ova karakteristika agenta se temelji na ulaznim i izlaznim funkcijama, dijeli na linearne i ne linearne. Kada agent odlučuje proporcionalno iz parametra iz okoline, za primjer možemo promatrati agenta koji na određeni stupanj temperature gasi grijanje, tada je riječ o linearnom agentu, koji je najpogodniji za matematičke analize. Dok kad promatramo ne linearnog agenta primjećuje se da odluke nisu proporcionalne jer su i sami mjerni podaci ne linearni, što dovodi do primjera većeg broja zrakoplova koji se trebaju složiti, pri tome treba paziti na luk i kut zemlje i slično.

4.1.3. Heterogenost

Kod ove karakteristike agenta dolazi do podijele na dva načina, to su homogeni ili heterogeni agenti. Homogeni agenti uključuju sve agente koji imaju ista svojstva. Agenti koji nemaju sva svojstva ista spadaju pod heterogeni sustav.

4.1.4. Parametri dogovaranja

Do parametara dogovaranja dolazi u slučajevima gdje se agenti međusobno moraju dogovoriti oko parametara. Ovisno o broju parametara oko kojih se agenti moraju dogovoriti klasificiramo ih kao prvi red, drugi red ili viši red. U slučaju da agenti imaju jedan parametra za dogovoriti riječ je o prvom redu. Kada imaju dva parametra za složiti, riječ je o drugom redu, a kad se radi o sustavu višeg reda traže se približne zajedničke vrijednosti. Za primjer mogu se promatrati jata ptica u letu, gdje imamo primjer položaja, brzine i sam timski rad za postizanje odgovarajućeg smjera, brzine, za dolazak do cilja.

4.1.5. Uzimanje kašnjenja u obzir

U današnje vrijeme postoje česta kašnjenja u komunikaciji, tako u slučaju da agent komunicira treba biti spreman za moguća kašnjenja. U ovom slučaju sustavi se dijele na sustave s kašnjenjem i sustave bez kašnjenja. Gdje se gleda jesu li kašnjenja značajna i bitna za određenu zadaću ili su zanemariva i nema potrebe da utječu na rad agenta.

4.1.6.Topologija

Radi se o mjestu i vezama između agenata, gdje topologija može biti statička ili dinamička. Kada se promatra statička topologija, položaj i veze agenata su u cijelo vrijeme postojanja sustava iste i nepromijenjene. Kod dinamičke topologije situacija je drugačija, položaj i veze agenata su promjenjive, agenti napuštaju ili pokreću novu komunikaciju međusobno.

4.1.7.Učestalost prijenosa podataka

Prema ovoj karakteristici agenti promatraju okruženja, te podatke prijenose na dva načina potaknuti vremenom ili potaknuti nekim događajem. Agent koji je potaknuti vremenom skupljaju podatke, te ih u određenom vremenskom razdoblju dijele dalje. Dok agent koji je potaknut nekim događajem odmah dijele podatke drugim agentima u sustavu.

4.1.8.Mobilnost

Mobilnost agenta dijeli se na statičke i mobilne agente. Prema samom nazivu statički agenti nalaze se na istom položaju u svojoj okolini. Mobilni agenti može se kretati da ima priliku nadgledati i prikupiti podatke iz različitih okolina. Mobilni agenti koriste se na primjer u otkrivanju napada, gdje se kreću kako bi analizirali procese i komunikaciju, te ukoliko ima nepravilnosti obavijestili o njima.

4.2. Agenti

Samo značenje riječi agent govori da se radi o posredniku ili zastupniku. Agent u ovom radu je računalni sustav koji ima mogućnost donošenja odluka, kako bi ostvarili svoje zadaće, prema Strahiji. Riječ je dakle o inteligentnim agentima koji imaju sposobnost učenja i korištenja svog znanja za ostvarivanje svog cilja. „Inteligentni agentni, da bi postigli ciljeve dizajna kroz niz autonomnih akcija trebaju neka imati sljedećih svojstava: reaktivnost, proaktivnost (engl. *pro-activeness*) i društvene sposobnosti (engl. *social ability*)“ Strahija (2018, str 7.) Agenti svoju reaktivnost postižu na način što doživljavaju svoje okruženje, te svoje odluke donose na temelju njega. Društvene sposobnosti kod agenta prepoznaju se kod komunikacije koju imaju agenti međusobno. Agenti mogu sami sebi određivati zadatke koje trebaju poduzeti iz skupljenih podataka iz svog okruženja. Mogu predvidjeti situacije, ako su tako izgrađeni. Agenti imaju mnoštvo mogućnosti koje mogu ponuditi, no to ovisi ne samo o njihovoj arhitekturi nego i o samom okruženju u kojem se nalaze. Arhitektura agenata dijeli se na konkretnu i apstraktnu arhitekturu prema Wooldridge.

Konkretna arhitektura dijeli se na :

- Temelju simboličke logike
- Reaktivnu arhitekturu
- Belief-Desire-Intention arhitekturu
- Slojevitou arhitekturu

Dok se apstraktna arhitektura agenta dijeli na:

- Čisto reaktivni agenti
- Percepciju
- Agente sa stanjem

Inteligentni agenti klasificiraju se prema složenosti i načinu rješavanja svojih zadataka, te su podijeljeni u pet klasa, prema Strahiji.

Klasifikacija agenata :

- Jednostavni reflektivni agenti
- Reflektivni agenti na temelju modela
- Agenti na temelju ciljeva
- Agenti na temelju korisnosti
- Agenti sa sposobnošću učenja

4.2.1. Primjena

Agenti se primjenjuju u mnogim sustavima, oni su proizašli od otkrivanja i razvoja umjetne inteligencije. Agenti mogu biti jednostavni kao što je već po karakteristikama, njihovoj arhitekturi moglo biti primijećeno, no mogu tvoriti i kompleksne sustave i rješavati izazovne zadatke. U suradnji s razvojem ostatka tehnologije, agenti ulaze i u stvarnu okolinu ljudi u obliku robotike, asistencija preko raznih servisa i slično.

4.3. Umjetna inteligencija

Umjetna inteligencija nema opću prihvaćenu definiciju, no mnogi su znanstvenici dali svoje definicije za umjetnu inteligenciju, prikupio ih je autor Dalbelo Bašić.

D. W. Patterson (1990.)

„Umjetna inteligencija je grana računarske znanosti koja se bavi proučavanjem i oblikovanjem računarskih sustava koji pokazuju neki oblik inteligencije. Takvi sustavi mogu učiti, mogu donositi zaključke o svijetu koji ih okružuje, oni razumiju prirodni jezik te mogu spoznati i tumačiti složene vizualne scene te obavljati druge vrste vještina za koje se zahtijeva čovjekov tip inteligencije.“

Patrick H. Winston (MIT)

“Proučavanje postupaka koji čine mogućim percipiranje, umovanje i reagiranje”

Elain Rich

“Umjetna inteligencija bavi se izučavanjem kako računalo učiniti sposobnim da obavlja poslove koje u ovom času ljudi obavljaju bolje.”

Definicija umijete inteligencije može se podijeliti na četiri grupe, te se tako najčešće i definiraju. Grupe ovise na čemu se temelje, kao na primjer na misaonom procesu koji zahtjeva zaključke. Također može očekivati ponašanje što se postiže kopiranjem ljudskog ponašanja, prema Marković napravljena je i tablica 1 po uzoru Marković.

SUSTAVI KOJI	
RAZMIŠLJA LJUDSKI	RAZMIŠLJA RACIONALNO
SE PONAŠAJU LJUDSKI	SE PONAŠAJU RACIONALNO

Tablica 1. Umjetna inteligencija

Svaka od ovih grupa ovisi o cilju kakav se sustav gradi i ima svoju zadaću. Mnogi znanstvenici dali su svoje viđenje za grupe, što se od njih traži da omoguće. Preuzeto od autora Dalbelo Bašić :

RAZMIŠLJA LJUDSKI

- “Uzbudljivi novi pokušaj da se omogući razmišljanje računalima... strojevi s umovima, u punom i doslovnom smislu” (Haugeland, 1985)
- “Automatizacija aktivnosti koje asocijamo s ljudskim razmišljanjem, aktivnosti poput donošenja odluka, rješavanja problema, učenja...” (Bellman, 1978)

RAZMIŠLJA RACIONALNO

- “Proučavanje mentalnih svojstava kroz uporabu računalnih modela” (Charniak i McDermott, 1985)
- “Proučavanje postupaka koji čine mogućim percipiranje, umovanje i reagiranje” (Winston, 1992)

PONAŠA SE LJUDSKI

- “Proces stvaranja strojeva koji obavljaju funkcije koje zahtijevaju inteligenciju koju imaju ljudi” (Kurzweil, 1990)
- “Proučavanje kako učiniti da računala rade stvari u kojima su, trenutno, ljudi bolji” (Rich i Knight, 1991)

PONAŠA SE RACIONALNO

- “Polje rada koje želi objasniti i emulirati inteligentno ponašanje u smislu računalnih procesa” (Schalkoff, 1990)
- “Grana računalnih znanosti koja se bavi automatizacijom inteligentnog ponašanja” (Luger i Stubblefield, 1993)

4.3.1. Povijest umjetne inteligencije

Umjetna inteligencija svoj razvoj započinje još 1943. godine kada je napravljen prvi model umjetnog neurona. Od tada dolazi do mnogih pomaka, jedan od značajnijih bio bi Turingov test 1950. godine pomoću kojeg započinje strojno učenje. Prvi chatbot ELIZA napravljen je 1965. godine. Polako se razvijala, u 70-tim godinama se radilo na sustavima znanja, tako da je umjetna inteligencija u 80-tima dobila mjesto u znanosti i sa sobom donijela inteligentne agente, prema Dalbelo Bašić.

4.3.2. Umjetna inteligencija u znanostima

Inteligencija se promatra u mnogim znanostima, no najpromatranija je u humanističkim znanostima kao što je psihologija, filozofija i tako dalje. „inteligencija (latinski intellegentia, intelligentia: razboritost, razum; vještina), u psihologiji, sposobnost mišljenja koja omogućuje snalaženje u novim prilikama u kojima se ne koriste (ili nemaju dobar ishod) nagonsko ponašanje, ni učenjem stečene navike, vještine i znanja.“ Hrvatska enciklopedija. Spoznaja znanja i umjetna inteligencija promatrani iz humanističkih znanosti i računalnih prikazana je u tablici koja je napravljena prema Dalbelo Bašić.

Humanistička znanost	Računalna znanost
Inteligencija	Umjetna inteligencija
Znanje	Baza znanja
Spoznaja	Obrada informacija
Učenje	Strojno učenje
Govorni jezici (prirodni)	Obrada prirodnog jezika

Tablica 2. Umjetna inteligencija u humanističkoj i računalnoj znanosti

Iz podataka koji su prikazani u tablici može se reći da računalna znanost uspješno zrcali humanističke uvijete za spoznaju znanosti. Turingov test služi za ispitivanje računalnih sustava, tj. za ispitivanje umjetne inteligencije.

5. Distribuirani sustavi

Distribuirani sustav omogućuje rješavanje problema koji samostalni entiteti, računala ili agenti (ovisno o promatranom sustavu) ne bi mogli samostalno izvršiti. U radu autora Vidaković, distribuirani sustav opisan je na primjeru ljudskog tijela, koje se sastoji od više organa te samostalno ne mogu postići puno, no u suradnji mogu napraviti puno. U prirodi ima puno takvih primjera od mrava, ptica, vukova, riba i tako dalje, te do samog ljudskog društva gdje se dijele razna zanimanja, poslovi i odgovornosti, kako bi društvo opstalo i napredovalo. Definiciju koju nudi Tanenbaum u svom radu za distribuirani sustav glasi : Distribuirani sustav je skup neovisnih računala koja se svojim korisnicima pojavljuje kao jedinstveni koherentni sustav.

U računalnom sustavu distribuirani sustavi se definiraju kao nezavisni procesi koji pomoću komunikacijske mreže imaju određena svojstva. Prema Kshemkalyani & Singhal razlikuju se četiri karakteristike :

- Bez usklađenog vremena (*eng. No common physical clock*)
 - Ova karakteristika je važna jer ima ulogu predstavljanja, upoznavanja (*eng. introduces*) elemenata u sustavu i na taj način gradi asinkroni sustav, u kojem elementi u stvarnom vremenu nisu usklađeni.
- Bez zajedničke memorije (*eng. No shared memory*)
 - Misli se na fizičku memoriju koja zbog asinkronosti sustava nije moguća, koristi se distribuirana memorija kojoj se šalju poruke.
- Geografska odvojenost (*eng. Geographical separation*)
 - Najpoznatiji primjer povezivanja kod odvojenosti u sustavu je Internet. Sustav zbog toga gubi na brzini, što su procesi udaljeniji to im je brzina i učinkovitost manji.
- Autonomnost i heterogenost (*eng. Autonomy and heterogeneity*)
 - Ova karakteristika označava mogućnost rada procesa u različitim brzinama i mogućnost da se obavljaju razne operacije. Ne rade neophodno u specifičnom distribuiranom sustavu.

5.1. Korištenje distribuiranih sustava

Postoje razni razlozi i svrha korištenja i samog razvoja distribuiranih sustava, prema radu Vidaković navedene su sljedeća svojstva:

- Inherentno distribuirano računanje
- Dijeljenje resursa
- Pristup podacima i resursima koji su geografski udaljeni
- Povećanje pouzdanosti
- Bolji omjer performanse i cijene
- Skalabilnost i modularnost

Primjeri ovih sustava mogu se pronaći u transferu sredstava u bankama, kod uskog grla u sustavu, sigurnost podataka s centralnim sjedištem, sigurne replike podataka za izbjegavanje gubitka podataka. Radi podijele resursa na više računala dolazi do mogućnosti iskorištavanja te prilike da se zadaci također podijele i brzina obrade zadatka je brža. Mogućnost da se procesi u sustavu mogu bez problema zamijeniti novim.

5.1.1. Paralelni sustavi

Sustavi s paralelnim procesiranjem su razvijeni za brze obrade podatke, a povezani su LAN mrežom. Njihova prednost je mogućnost bolje performanse u izvršavanju zadataka, te se bazira na kalkulacijama numeričkih vrijednosti. Postiže se podjelom na više procesa, koji ne zahtijevaju puno međusobne komunikacije, kako bi sinkronizacija bila što uspješnija. Paralelni sustavi imaju neke svojstva koja su prethodno navedena u poglavlju 5.1., a dijele se na tri skupine :

- multiprocesorski sustavi
 - Paralelni sustavi koji imaju više procesa s izravnim pristupom dijeljenoj memoriji.
- višeračunalni sustavi
 - Paralelni sustavi koji imaju više procesa koji nemaju izravni pristupom dijeljenoj memoriji.
- array procesore
 - Paralelni sustav u kojem su računala na istoj lokaciji, usko povezani, imaju isti radni ritam, memoriju ne dijele, a komuniciraju pomoću poruka.

5.2. Komunikacija unutar distribuiranog sustava

Komunikacija unutar distribuiranog sustava se postiže pomoću funkcija, koje dolaze s dijeljenom memorijom. Sustavi dijeljene memorije sadrže dijeljeni adresni prostor, dok oni koji to nemaju komuniciraju pomoću slanja poruka. Simulacijom je moguće u distribuiranom sustavu stvoriti zajednički adresni prostor, distribuirana zajednička memorija. Razlikuju se dva načina komunikacije unutar distribuiranog sustava.

- Sinkronizacija procesora
 - Procesori koji izvršavaju naredbe u isto vrijeme, nisu mogući u distribuiranom sustavu, ali je moguće da procesi ne zaostaju jedan za drugim u radu i da se međusobno čekaju kako bi izvršili svoj zadatak.
- Izbor vođe
 - Kako bi sustav imao što bolju sinkronizaciju, potreban je proces prema kojem se ostali procesi mogu ravnati, tj. pratiti ga. Vođa se izabire uspoređivanjem ID, numeričke vrijednosti, koji svaki proces ima, te obavještava ostale o svom ID. „to znači da nakon što svaki proces pokaže svoj ID, onaj s najvećom numeričkom vrijednošću ID-a je vođa, a svi koji su sudjelovali u odabiru prate njegove instrukcije.“ (Vidaković, 2019, 9.str)

6. Praktični dio rada

U ovom dijelu rada promatrat će se veliki podaci i prikazat će se načini na koje se podaci pomoću agenta mogu obraditi. Pokazat će se kako na temelju velikih podataka agenta trenirati, tj. učiti, da bi mogao dati što točnije odgovore na tražena pitanja koja mu se postavljena.

Za potrebe praktičnog djela rada korištena su znanja i kod stečeni u Prilogu 1., Prilogu 2., Prilogu 3 i Prilogu 4. Za samu izradu korišten je Prilog 5. i Prilog 6. Podaci koji su korišteni za potrebe testiranja i prezentiranja mogućnosti su u Prilogu 1, Prilogu 7 i Prilogu 8.

6.1. Korištene tehnologije

Sve pomoću čega se izradi praktični dio rada opisano je u slijedećim poglavljima. Jezik u kojem je programiran praktični dio je Python. Programirano je u Google Colaboratory ili po poznatijem imenu Colab. Korišten je scikit-learn za učenje agenta. Podaci su različitih tematika i izvora.

6.1.1. Programski jezik Python

Programski jezik Python nastao je krajem osamdesetih godina. Viziju za novi jezik imao je Guido van Rossuma. Njegova ideja je bila da jezik bude jednostavan i pogodan za korištenje. Uz to Python ima ciljeve kao što su produktivnost što omogućuje na način da je kod vrlo lako za pisati, održavati i pregledavati, kvaliteta pokazuje svojom lakoćom čitanja i duljinom koda, modularnost postiže raznim bibliotekama koje proširuju mogućnosti rada i portabilnost koja je realizirana mogućnošću rada neovisno o operacijskom sustavu, prema Šantić (2017).

6.1.2. Alati

Google ima svoj servis Colaboratory ili kraće Colab. Moguće ga je pokrenuti sa bilo kojeg preglednika, uz to moguće ga je preuzeti kao aplikaciju na svoje mobilne uređaje kao što su mobitel ili tablet. Prednost Colab su što ima većinu biblioteka već instaliranu, ima slobodan pristup GPU, vrlo lako se može dijeliti. Uz to dobar je za rad s podacima i istraživanje i razvoj umjetne inteligencije. Kod je lako podijeliti i Colab nudi mnoge prijedloge s objašnjenjima tokom samog pisanja koda. Podaci o samom alatu preuzeti su iz Prilog 6.

6.1.3.scikit-learn

Riječ je o bibliotekama za strojno učenje u Python jeziku. Na biblioteci se radi od 2007. godine u sklopu Google projekta. Od 2010. godine Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort i Vincent Michel preuzeli su projekt i omogućili ga javno dostupnim u veljači 2010. godine. Pomoću ove biblioteke moguće je rješavanje mnogih problema, složeni su na način da lako mogu raditi s numeričkim i znanstvenim bibliotekama, poznatim kao Python: NumPy i SciPy. Uz to scikit-learn sadrži razne algoritme za grupiranje, klasifikaciju, regresiju, klase, funkcije znanstvenog učenja i još mnogo toga. Svi podaci za opis biblioteke scikit-learn preuzeti su iz Priloga 5.

6.1.4.Podaci

Prvi set podataka koji se koristi je preuzet iz Priloga 1., riječ je o tesianim podacima nekog poduzeća koje ima podatke o navikama svojih kupaca, te na temelju njih traži odgovore na pitanja. Kojoj skupini svojih korisnika treba više promovirati proizvode, dok koju grupu korisnika možda treba zaštititi i ne prikazivati promocije kako ne bi imali dodatne troškove. Set podataka koji se koristi za drugi prikaz, preuzet je s Priloga 2, gdje je prikazano na podacima o raku dojke kako se može predvidjeti dijagnoza. Sami podaci nalaze se u Prilogu 7, preuzeti su sa stranice koja ima mnoštvo baza podataka na kojima se mogu trenirati sustavi kako bi njihova točnost postala što veća.

Zadnji set podataka koji se koristi u ovom radu se nalazi u Prilogu 8, a riječ je o podacima o avokadu. Podaci su skupljani o cijeni, količini, mjestu prodaje i naravno radili se o organsko uzgoju ili ne.

Opis svih setova podataka nalazi se u tablici 3. gdje su prikazani nazivi, broj atributa i količina podataka koju svaki set posjeduje, te koja će biti analizirana.

Set	Naziv	Broj atributa	Broj redova
1.	Social_Network_Ads_1.csv	3	400
2.	breast_cancer.csv	11	683
3.	avocado.csv	14	18250

Tablica 3. Prikaz korištenih setova podataka

6.2. Logistička regresija

Kako bi predviđanja sustava, na temelju podataka koje dobiva bila što točnija, radit će pomoću logističke regresije. Prema Kostić logistička regresija opisuje vezu između ovisne i neovisne varijable, koje mogu biti u različitim oblicima. Za primjer može se gledati kako je zavisna varijabla binarna, te govori hoće li se nešto dogoditi ili ne, gleda se vjerojatnost za događaj. Prema zavisnoj varijabli imamo tri moguća tipa :

- Binarnu logističku regresiju
 - Zavisna varijabla je binarna
 - npr. proći ili pasti test, odgovor da ili ne
- Nominalna logistička regresija
 - Ima tri ili više kategorija, koje se ne mogu uspoređivati po vrijednostima
 - npr. na boje (crvena, plava, žuta ...), na pretraživače (Google, Yahoo!, Bing ...)
- Ordinarna logistička regresija
 - Ima tri ili više kategorija, koje se mogu međusobno uspoređivati, no nisu nužno jednako rangirani
 - npr. ocjena (1-5), završeno obrazovanje i slično.

Logistička regresija koristi se kad zavisna promjenjiva varijabla uzima neki konačni skup podataka, prema Kostić. Logistička regresija temelji se na linearnoj regresiji. Postoji jednostavna linearna regresija, višestruka linearna regresija . Kad se na jednostavnu linearnu regresiju doda Sigmoidova funkcija (matematička funkcija koja ima karakterističnu krivulju u obliku slova S ili poznata još pod nazivom sigmoidna krivulja). Dobiva se formula za logističku regresiju. Sve formule su prikazane u tablici 4. gdje su sve spomenute formule i funkcije prikazane.

Regresija	Formula
Jednostavna linearna regresija	$y = b_0 + b_1 \cdot x$
Višestruka linearna regresija	$y = b_0 + b_1 \cdot x_1 + \dots + b_n \cdot x_n$
Sigmoidova funkcija	$p = \frac{1}{1 + e^{-y}}$
Logistička regresija	$\ln\left(\frac{p}{1-p}\right) = b_0 + b_1 \cdot x$

Tablica 4. Formule regresija

6.3. Proces izgradnje

Kako bi se izgradio višeagentni sustav koji ima mogućnost analizirati veliki broj podataka potrebno je podijeliti proces izgradnje na manje dijelove. Tako da za potrebe ovog rada imamo sustav koji se sastoji od 9 dijelova. Kako bi bilo što jasnije što se događa, svaki dio će zasebno biti opisan i objašnjen. Svi dijelovi se ne koriste za svaki sustav koji će na koncu biti prikazan i s kojim će se analizirati pripremljeni podaci. Sustav je građen da bude što više fleksibilan i da ima mogućnosti obraditi različite baze podataka, sa sličnim karakteristikama. Programski kod preuzet je iz Priloga 1. i Priloga 2. te je za potrebe analize zadnjeg seta podataka prilagođen za tu namjenu.

Proces izgradnje sustava podijeljen je na sljedeće dijelove:

- Dodavanje biblioteka
- Dodavanje podataka
- Kodiranje podataka
- Podjela podataka
- Skaliranje
- Treniranje
- Predviđanje
- Izrada matrice
- Rezultati

6.3.1. Dodavanje biblioteka

Kako bi sustav mogao uspješno raditi potrebno je uključiti tri biblioteke:

- numpy – za rad s redovima
- matplotlib.pyplot – za prikazivanje grafova
- pandas – omogućuje rad s vektorima i maticama, manipulaciju i analizu podataka

Svaka biblioteka je dodana pomoću naredbe `import`, naziv biblioteke te na koncu je dodana skraćunica kako bi se lakše pozvala biblioteka ubuduće.

Prikaz dodavanja biblioteka :

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

6.3.2. Dodavanje podataka

Može se primijetiti da se ovdje koristi biblioteka `pandas`, koja je bila dodana na početku. Pomoću nje se poziva funkcija koja omogućuje čitanje podataka iz `csv` datoteka. Podatke spremamo u dvije varijable `x` i `y`. U `x` se spremaju neovisni podaci koji pomažu u predviđanju, dok se u `y` spremaju podaci po kojima se predviđa i odgovara na postavljena pitanja, tj. ciljanje vrijednosti. Ovaj programski kod napravljen je da prati najčešću praksu, gdje su podaci neovisni u prvim stupcima, dok je u zadnjem stupcu nalaze podaci koji su ovisni za predviđanja. Pomoću `iloc` lociraju se indeksi i prema njima se dodaju određeni redovi i stupci. U prikazanom kodu u nastavku za `x` dodani su svi redovi, i svi stupci osim zadnjeg, koji je isključen, dok su za `y` dodani svi redovi, ali samo za zadnji stupac.

Prikaz dodavanja podataka :

```
dataset = pd.read_csv('naziv_datoteke.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```


6.3.3. Kodiranje podataka

Ovisno radi li se o podacima koji su neovisni ili ovisni, rade se različita kodiranja tih zapisa. Kod obadva kodiranja koristi se biblioteka `sklearn`, no koriste se drugačije funkcije.

Kada je riječ o kodiranju nekih od stupaca u neovisnoj skupini koriste se funkcije :

- `ColumnTransformer` – koristi se za transformaciju reda,
 - prvo očekuje podatke o tome kakvo kodiranje treba napraviti, poziva funkciju, te na kojem mjestu se nalazi stupac koji potrebno kodirati
 - zatim je potrebno naglasiti `remainder='passthrough'` da nema potrebe za dodatnim promjenama
- `OneHotEncoder` – koristi se za kodiranje pomoću sheme kodiranja „*one-of-K*“, stvara binarni stupac za svaku kategoriju (vraća rijetku matricu ili gusti niz)

Red koji se kodira na ovaj način u slučaju da nije na prvom mjestu, nakon kodiranja postavlja se na prvo mjesto. Transformacija se ostvaruje pomoću `fit_transform` gdje se na jednostavan način primjenjuje objekt `ct` nad podacima koji se pohranjuju u svoju varijablu.

Prikaz kodiranja podataka koji su neovisni :

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = np.array(ct.fit_transform(X))
```

Kodiranje podataka na ovaj način nešto je manje kompleksno, zahtjeva samo funkciju :

- `LabelEncoder` – koja bavi kodirajte ciljne oznake s vrijednošću između 0 i n

Objekt `le` poziva funkciju, i kako imamo samo jedan stupac nema zabune, tako da se na sličan način kao i u prošlom primjeru pomoću `fit_transform` podaci kodiraju i pohranjuju u svoju varijablu.

Prikaz kodiranja podatka koji su ovisni :

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

6.3.4. Podjela podataka

Za podjele podataka koristi se biblioteka `sklearn` sa svojom funkcijom `train_test_split` koja omogućuje da se podaci podijele. Podatke dijelimo na četiri seta: `X_train, X_test, y_train, y_test` koja se dobivaju iz `x` i `y`, s testnom veličinom 0.2, što je 20% podataka koji se ostavljaju za testiranje, a biraju se slučajnim odabirom.

Prikaz podjele podataka :

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 1)
```

6.3.5. Skaliranje

Skaliranje se radi nakon podjele podataka, da se ne bi došlo do izljeva podataka, tj. kako bi testiranje pokazalo što točnije rezultata, treniranog sustava. Skaliranje je potrebno napraviti kako bi se podaci izjednačili, da ne dođe do dominacije nekih podataka. Potrebna je ili standardizacija ili normalizacija podataka, kako bi se moglo započeti treniranje. Standardizacija odgovara za više slučajeva, za razliku od normalizacije, za potrebe ovog rada će se koristiti skaliranje pomoću standardizacije, koja se zove `StandardScaler` te se pomoću nje skaliraju svi neovisni podaci koji se promatraju, podaci za treniranje i testni podaci. U slučaju kodiranih podataka, njih bi trebalo izbjegavati dodatno skalirati, jer su one već skalirane.

Prikaz skaliranja podataka :

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

6.3.6. Treniranje

Logistička regresija nalazi se u modlu `linear_model` gdje se poziva klasa `LogisticRegression`. Model se naziva `classifier` jer klasifikacija rješava pitanje identifikacije novih parametara po kategorijama, naravno na temelju prethodno skupljenih podataka koje promatra i po kategorijama koje su poznate. Zove se klasa, kako bi se stvorila instanca za željeni mode analize. Parametar se biraju prema modelima *k-Fold Cross* ili *Grid*, no za sada koristit će se parametar slučajnog odabira. Kako bi podaci bili pohranjeni u `classifier` koristi se `fit` koji koristi samo podatke koji su namijenjeni za treniranje.

Prikaz treniranja :

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
```

6.3.7. Predviđanje

Predviđanja se mogu raditi za specifične upite ili općenito. Prvo je prikazano predviđanje na zahtjev, koje se odnosi na prvi set podataka. Traži se osoba koja ima 30 godina s doprinosima od 87000. Kako bi se došlo do tog podatka potrebna je skalirati traženi zahtjev. Prvo imamo `classifier` gdje se nalaze sve klasifikacije, zatim metoda `predict`, kako bi se predvidjela odluka, te na koncu skaliranje traženih podataka. Pomoću ovog rezultata moći će se provjeriti koliko je sustav treniran, hoće li se rezultat koji će dati trenirani sustav poklapati.

Prikaz predviđanja traženog zahtjeva nad pravim podacima:

```
print(classifier.predict(sc.transform([[30, 87000]])))
```

Pitanje koje se postavlja je hoće li tražena osoba biti potencijalni kupac. Kako bi došli do predviđanja od treniranog sustava potrebni su sljedeći koraci. Varijabla `y_pred` koja pohranjuje sve klasificirane, predviđene neovisne podatke, sličan prethodnom kodu. Zatim imamo ispis, koji se prikazuje u vertikalnoj matrici pomoću `concatenate` gdje se prvo ispisuje `y_pred` predviđeni, a onda se od testnih `y_test`. Kako bi podaci bili vertikalno ispisani koristi se `reshape(len(), 1)` i na koncu se može odabrati `0` ili `1` gdje `1` označuje da će biti horizontalno, a `0` vertikalno. Pošto se radi o dva vektora koja su prikazana vertikalno, potrebno je da ispisi budu horizontalni.

Prikaz predviđanja traženog pitanja nakon treniranja :

```
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

U slučaju da nema posebnih zahtjeva, dovoljno je napraviti predviđanje koje je prikazano u nastavku. Sastoji se od `y_pred` koja pohranjuje sve predviđene neovisne podatke, slično kao i u prethodnim primjerima koda.

Prikaz predviđanja :

```
y_pred = classifier.predict(X_test)
```

6.3.8. Izrada matrice

Riječ je o matrici konfuzije, koja daje broj točnih predviđanja, i broj krivih predviđanja, koji se rade s testnim podacima i predviđenim. Potrebno je dodati `confusion_matrix`, `accuracy_score`, te se `confusion_matrix` koristi kako bi se dobila matrica, a `accuracy_score` kako bi dobili točnost predviđanja.

Prikaz matrice :

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

6.3.9. Rezultati

Sami rezultati mogu se vidjeti iz matrice i točnosti predviđanja, ali kako bi se stvorila cijela slika promatranih podataka u analizi mogu se prikazati grafički, no to ima svoje nedostatke. Osim grafičkog prikaza podaci se mogu i provjeriti na način da se prikaže točnost treniranog predviđanja sustava.

6.3.9.1. Grafički prikaz

Da se podaci grafički prikažu, potrebno je promatrati samo dvije neovisne vrijednosti. Prva promatrana vrijednost na grafu postaje x-os, dok je druga y-os. Ovisna vrijednost prikazana je kao linija na grafu. Graf sadrži i točke koje prikazuju gdje podaci zaista pripadaju, unatoč podijeli koja je napravljena, na taj način mogu se primijetiti iznimke, tj. odstupanja. Klasa `ListedColormap` omogućava bojanje po grafu. Lokalne varijable `X_set`, `y_set` povezuju se s podacima koji se prikazuju na grafu, u ovom slučaju su to testni podaci `X_test`, `y_test`, no ne moraju nužno biti samo oni prikazani. Kod `X_set` mogu se primijetiti oznake minus i plus koje sprječava preklapanja, a `step=0.01` označava da su svi pikseli rezolucijom 0.01. Kako bi odgovarajuće pikseli imali svoju boju odgovorna je `plt.contourf`, riječ je o pozadinskom bojanju grafa. Nakon toga definirane su x i y osi sa `plt.xlim()`, `plt.ylim()`, nakon toga nalazi se petlja pomoću koje se točke prikazuju na grafu, u svojoj odgovarajućoj boji. Kako bi prikaz grafa bio što jasniji potrebno je dodijeliti imena i nazive, grafu, osima, doraditi legenda, te naravno sam prikaz.

Prikaz grafičkog prikaza:

```
from matplotlib.colors import ListedColormap
X_set, y_set = X_test, y_test
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X
_set[:, 0].max() + 1, step = 0.01),
                    np.arange(start = X_set[:, 1].min() - 1, stop = X
_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(), X2.ravel
()]).T).reshape(X1.shape),
            alpha = 0.75, cmap = ListedColormap(('red', 'green')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
    plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1],
                c = ListedColormap(('red', 'green'))(i), label = j)
plt.title('')
plt.xlabel('')
plt.ylabel('')
plt.legend()
plt.show()
```

6.3.9.2. Validacija točnosti pomoću k-Fold Cross

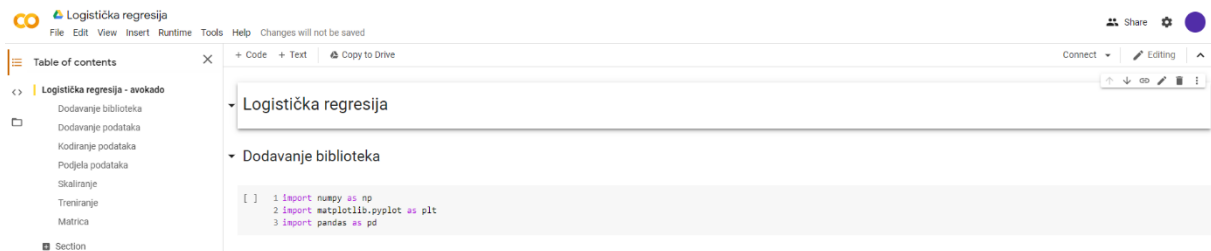
Pomoću `cross_val_score` realizira se unakrsna validacija (k-Fold Cross). Promatrat će se točnost na uzorku od 10 podataka. Podaci koji se promatraju su oni koji su prošli kroz treniranje, a to su `X_train`, `y_train`. Kako bi ispis bio prikazan u postocima s dvije decimale potrebno je formatirati podatke, detaljno je prikazano niže u prikazu koda. Točnost podataka prikazuje se u postocima i sa standardnom devijacijom koja ukazuje prosječno odstupanja.

Prikaz validacije točnosti :

```
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator = classifier, X = X_train, y =
y_train, cv = 10)
print("Točnost: {:.2f} %".format(accuracies.mean()*100))
print("Standardna devijacija: {:.2f} %".format(accuracies.std()*100))
```

6.4. Prikaz sustava u Google Colaboratory

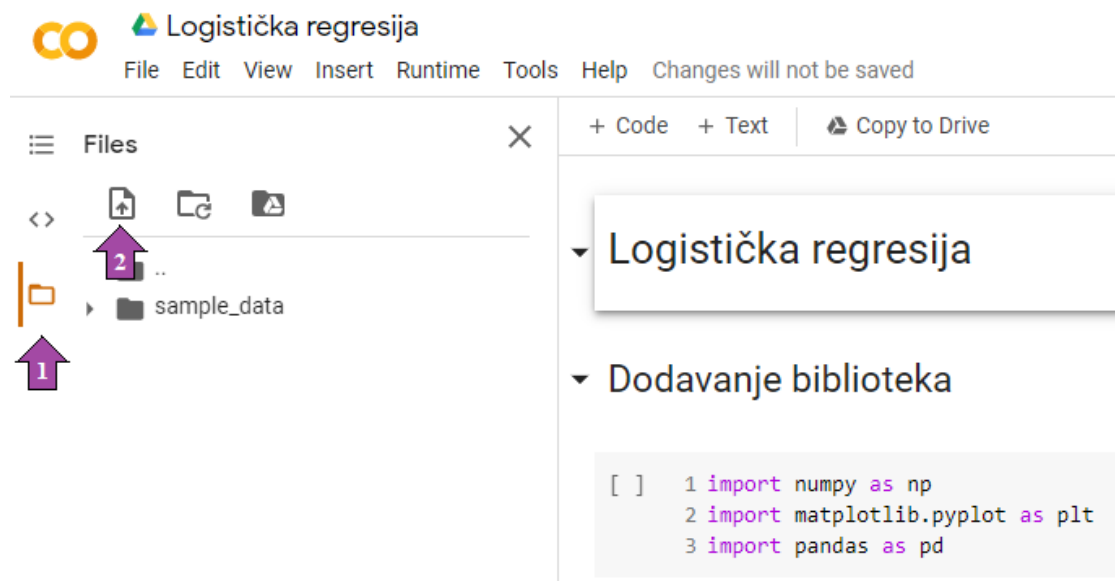
U Prilogu 9. nalazi se programski kod i podaci koji su namijenjeni za izgrađeni sustav. Podatke je potrebno prethodno skinuti na svoje računalo. Pokretanje programskog koda je intuitivno, te se može otvoriti na dva načina dvostrukim klikom te odabirom opcije „Open with Google Colaboratory“ ili desni klik odabir opcije „open with“ koja zatim nudi „Google Colaboratory“. Nakon toga trebao bi se otvoriti novi prozor u kojem se nalazi programski kod, prikazano je na slici 3. kako izgleda pokrenuti kod u *Google Colaboratory*.



Slika 3. Prikaz pokrenutog Google Colaboratory

6.4.1. Dodavanje podataka u Google Colaboratory

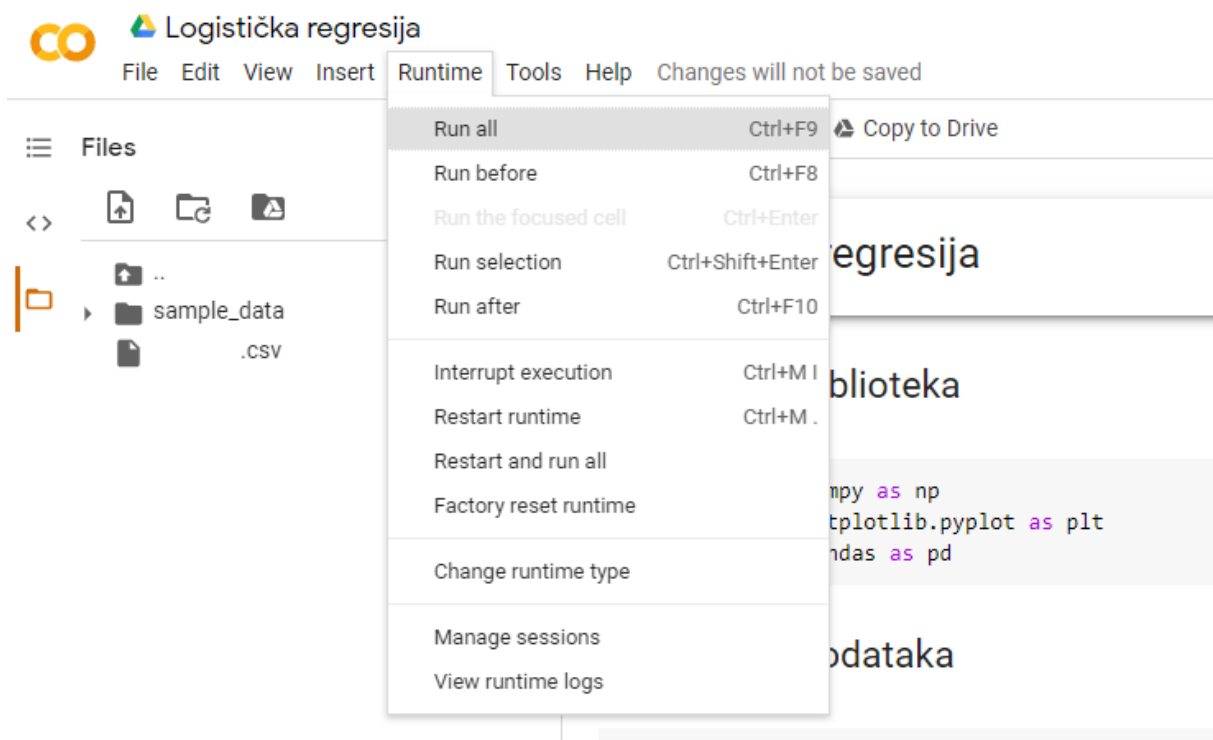
Kako bi sustav mogao analizirati podatke potrebno mu ih je dodati. Podaci se dodaju odabirom na ikonu „Files“, te zatim odabirom na „Upload to session storage“ odabire prethodno skinuta datoteka za taj sustav. Na slici 4. prikazano je pomoću strelica postupak dodavanja podataka.



Slika 4. Prikaz dodavanja podataka u Google Colaboratory

6.4.2. Pokretanje sustava u Google Colaboratory

Postoji više načina za pokretanje koda u *Google Colaboratory*. Najjednostavniji način za pokretanje cijelog sustava je odabirom „Runtime“, nakon toga „Run all“. Nakon toga svaki će se programski prozor pokrenuti jedan za drugim, kako koji završi sa svojim zadatkom. Taj način prikazan je na slici 5. Osim tog načina, postoji i ručna mogućnost gdje se svaki prozor ručno može pokrenuti od strane korisnika. Veoma je korisno u samoj izradi sustava, kod testiranja i isprobavanja rada sustava.



Slika 5. Prikaz pokretanja koda u Google Colaboratory

6.5. Primjena korištenja

Svaki dio koda nije jednako primijenjen za svoj primjer, negdje nije potrebno kodiranje podataka, jer su podaci pripremljeni. Negdje nije nužno skaliranje, dok negdje nije prikazano predviđanje. Kako bi bilo jasno što se događa za koji set podataka u tablici 5 su prikazani svi dijelovi sustava po setovima podataka. Kako bi oznake koje se koriste u tablici bile jasne, ispod se nalazi objašnjenje za svaku oznaku. Sav programski kod i setovi podatak mogu se pronaći u Prilogu 9.

	Set podataka 1	Set podataka 2	Set podataka 3
Dodavanje biblioteka	+	+	+
Dodavanje podataka	+	+*	+*
Kodiranje podataka	-	-	+
Podjela podataka	+	+	+
Skaliranje	+	-	+
Treniranje	+	+	+
Predviđanje	+	+	-
Izrada matrice	+	+	+
Rezultati	+	+	+

Tablica 5. Dijelovi sustava po setovima podataka

- + sadrži programski kod
- ne sadrži programski kod
- +* sadrži programski kod, ali s preinakama

6.5.1. Podaci iz prvog seta podataka

Promatraju se potrošači nekog poduzeća. Podaci koji se promatraju su godine, plaća korisnika i ovisni podatak koji promatramo jesu li kupili proizvod od promatranog poduzeća.

Traži se predviđanje hoće li potrošač kupiti proizvod koji se nudi. Na temelju tih podataka poduzeće može utjecati na svoje promocije putem društvenih mreža i ciljati na željenu publiku kako bi ostvario što veću prodaju svog proizvoda. Set podataka preuzet je iz Priloga 1.

6.5.2. Podaci iz drugog seta podataka

Promatraju se podaci o raku dojke koji su prikupljeni od 1989. do 1991. godine, te donirani 1992. godine. Skupljani su podaci o nalazima i njihovi konačna dijagnoza. Podaci su preuzeti iz Priloga 7.

Na temelju ovog seta podataka sustav će biti treniran da može prema nalazima procijeniti ima li pacijent rak ili ne. Na temelju ovog seta moguće je u sustavu analizirati slične setove i dobiti predviđanja.

6.5.3. Podaci iz trećeg seta podataka

Promatraju se podaci o avokadu. Podaci su bilježeni kroz 2015. godinu do 2018. godinu, u različitim mjestima. Bilježena je količina prodaje i jesu li organski uzgojeni ili nisu. Podaci su preuzeti iz Priloga 8.

Ovaj set podataka promatran je da se usporedi prodaja i količina organski uzgojenih avokada. Na temelju izvršene analize moći će se zaključiti koje vrste avokada ima najviše u prikupljenim podacima.

6.6. Prikaz rezultata

Kod programiranja za pregled podataka s kojima se radi, pogotovo kod kodiranja, kako bi se provjerilo stanje koristi se `print()`. Rezultati tog tipa neće ovdje biti prikazani, već samo najbitniji dijelovi za ovaj rad.

Glavni dio ovog sustava bio bi treniranje, koje je u svim slučajevima identično. Na slici 6. prikazan je izgled nakon izvršavanja treniranja.

▼ Treniranje

```
[ ] 1 from sklearn.linear_model import LogisticRegression
     2 classifier = LogisticRegression(random_state = 0)
     3 classifier.fit(X_train, y_train)

↳ LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, l1_ratio=None, max_iter=100,
multi_class='auto', n_jobs=None, penalty='l2',
random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
warm_start=False)
```

Slika 6. Prikaz rezultata treniranja

Nakon izvršavanja dijela treniranja sustava s podacima za treniranje, treba se prikazati ovakav ispis što pokazuje da je logistička regresija uspješno napravljena, te se može nastaviti daljnji procesi u sustavu.

U nastavku su tri poglavlja, svako za posebni set podataka. Svi imaju dva prikaza svojih rezultata, prikaz matrice je u svima detaljno objašnjen i opisan. Nakon pregleda rezultata svako pitanje koje se postavilo za promatrani set podataka je odgovoreno i prokomentirano.

6.6.1. Rezultati na temelju prvog seta podataka

Rezultati su prikazani na slikama 7. i 8., gdje je prikazana matrica, te grafovi na temelju analize podataka koji se promatraju.

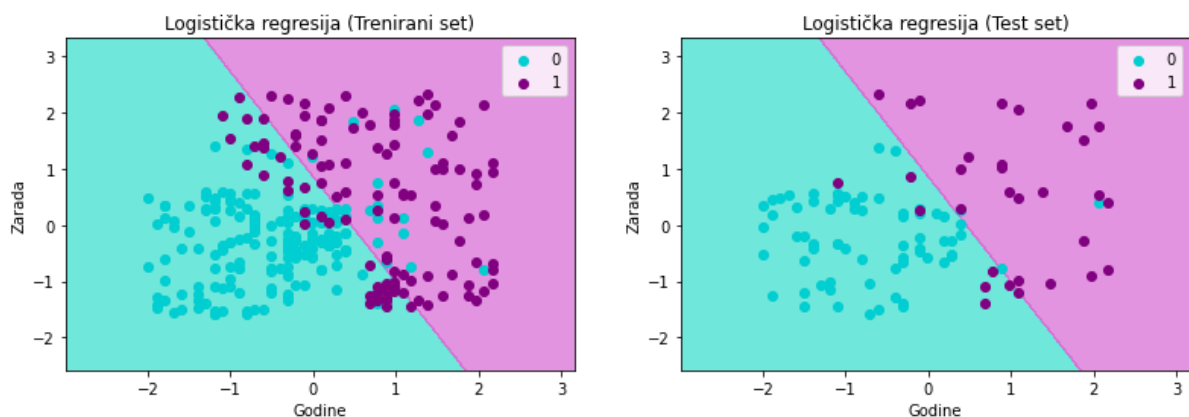
▼ Matrica

```
1 from sklearn.metrics import confusion_matrix, accuracy_score
2 cm = confusion_matrix(y_test, y_pred)
3 print(cm)
4 accuracy_score(y_test, y_pred)
```

[[65 3]
 [8 24]]
0.89

Slika 7. Prikaz matrice za prvi set

Ova matrica ukazuje da je 65 procjena da će potrošač kupiti njihov proizvod, 24 procjene da korisnici neće kupiti proizvod. Među njima ima i 8 korisnika koji iako jesu u skupini koja mi mogla biti zainteresirana ipak neće kupiti proizvod. Također postoji i 3 korisnika koji unatoč obilježjima koji ne nagovještaju kupnju proizvoda, su si ga priuštiti. Prema svim ovim podacima imamo 89% točno predviđanje na temelju seta podataka.



Slika 8. Prikaz grafova za prvi set

0 – nisu kupili proizvod

1 – jesu kupili proizvod

Grafovi prikazuju podatke nakon treniranja nad podacima (lijevo) i na podacima koji su uzeti za testiranje (desno). Može se primijetiti da su podaci iz matrice prikazani na grafu desno. Na temelju ove analize, nad ovim setom podataka, može se reći da većina kupaca njihovog proizvoda je starije, što naravno povlači i činjenicu veće financijske mogućnosti.

6.6.2. Rezultati na temelju drugog seta podataka

Rezultati su prikazani na slikama 9. i 10., gdje je prikazana matrica, te točnost treniranja s određenim odstupanjem.

▼ Matrica

```
[10] 1 from sklearn.metrics import confusion_matrix, accuracy_score
      2 y_pred = classifier.predict(X_test)
      3 cm = confusion_matrix(y_test, y_pred)
      4 print(cm)
      5 accuracy_score(y_test, y_pred)
```

```
↳ [[84  3]
    [ 3 47]]
0.9562043795620438
```

Slika 9. Prikaz matrice za drugi set

Ova matrica pokazuje da je 84 osoba kojima je točno dijagnosticirano da nemaju rak i 47 osoba kojima je točno dijagnosticirano da imaju rak. U oba slučaja da je rak pogrešno dijagnosticiran je 3. Na koncu dolazimo do 131 točne procjene dijagnoze i 6 pogrešnih, što nam daje oko 95% točnosti predviđanja po ovim rezultatima.

```
[11] 1 from sklearn.model_selection import cross_val_score
      2 accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10)
      3 print("Točnost: {:.2f} %".format(accuracies.mean()*100))
      4 print("Standardna devijacija: {:.2f} %".format(accuracies.std()*100))
```

```
↳ Točnost: 96.70 %
Standardna devijacija: 1.97 %
```

Slika 10. Prikaz točnosti i standardne devijacije za drugi set

Pomoću unakrsne validacija (k-Fold Cross) došlo se do podatak točnosti i standardne devijacije, tj. odstupanja. Točnost predviđanja ovog sustava nakon treniranja je 96.70%. Odstupanje koje se događa za 1.97%. Na temelju ovih rezultata, izgrađen je sustav koji ima veliki postotak točnosti u procjeni dijagnostike, na temelju podataka nalaza koje obrađuje.

6.6.3. Rezultati na temelju trećeg seta podataka

Rezultati su prikazani na slikama 11. i 12., gdje je prikazana matrica, te grafovi na temelju analize podataka koji se promatraju.

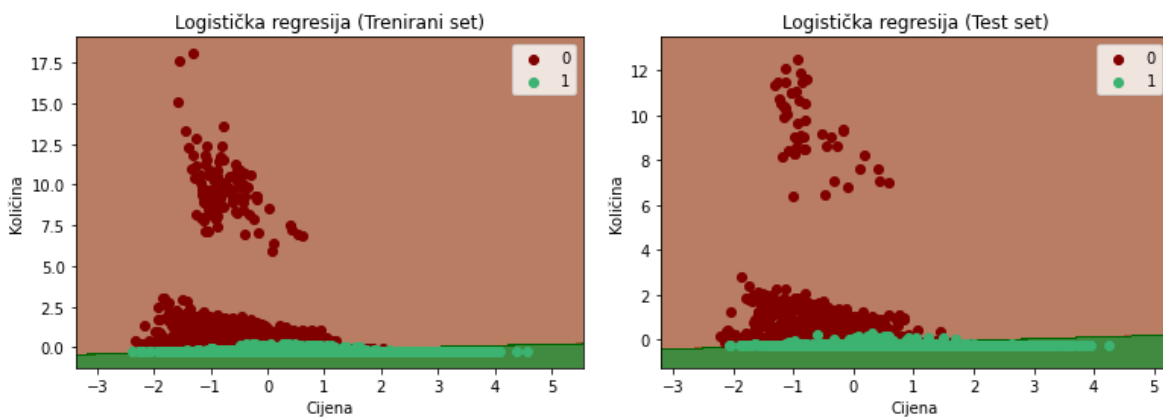
▼ Matrica

```
[10] 1 from sklearn.metrics import confusion_matrix, accuracy_score
      2 y_pred = classifier.predict(X_test)
      3 cm = confusion_matrix(y_test, y_pred)
      4 print(cm)
      5 accuracy_score(y_test, y_pred)
```

```
[[1946 287]
 [ 193 2137]]
0.8948060486522025
```

Slika 11. Prikaz matrice za treći set

Iz matrice se može pročitati da je 1946 organskih avokada točno predviđeno, dok je 193 koja su trebala po predviđanjima biti organska, nisu. Ne organski uzgojenih avokada ima 2137, a 287 procjena da avokado nije organski uzgojen, je pogrešno. Točnost predviđanja je oko 89%.



Slika 12. Prikaz grafova za treći set

0 – neorganski uzgojeni avokado

1 – organski uzgojen avokado

Grafovi prikazuju podatke nakon treniranja nad podacima (lijevo) i na podacima koji su uzeti za testiranje (desno). Može se primijetiti da su podaci iz matrice prikazani na grafu desno. Iz ove analize može se zaključiti kako je na tržištu, u promatrano vrijeme na određenim mjestima, veliki broj ne organskih avokada, kojima je cijena niža, a količina zavidno veća.

7. Zaključak

Podaci se prikupljaju svake sekunde u velikim količinama, za razne svrhe i na različite načine zahvaljujući internetu. Uz to raste i ljudska znatiželja i želja za razvojem, što postavlja sve veća i teža pitanja u znanosti i društvu. Pomoću tehnologija koje danas imamo moguće je izvršiti razne analize i simulacije, koje čovjek samostalno nije u mogućnosti realizirati ili bi mu za to trebao velik vremenski period. Iz potrebe za pomoć računalnih rješenja, došla je i do razvijanja umjetne inteligencija, koja je u svom razvoju donijela inteligentne agente, koji zajedno tvore višeagentne sustave pomoću kojih se mogu graditi razni sustavi.

Jedno od mnogih mogućnosti sustava prikazano je u ovom radu, kroz analizu podataka logističkom regresijom. Sustav je na temelju seta podataka treniran, kako bi mogao imati što bolja predviđanja i kako bi na temelju rezultata mogli doći do zaključaka u znanosti, marketingu i slično. Zahvaljujući ovom sustavu može se reagirati te prema tim podacima nastaviti ostvarivanje svojih ciljeva, ovisno o promatranim podacima i samim pitanjima.

Praktični dio je izrađen pomoću jednog agenta koji se bavi analitikom podataka, a kada bi se koristili na većem setu podataka u drugačijem okruženju, dolazi do potrebe distribucije zadataka, tj. do distribuiranih sustava i njegovih mogućnosti. Ovisno o uvjetima u kojima je sustav, na koji način se komunicira unutar sustava, potrebno je podijeliti zadatke. Jedna od mogućnosti je paralelni sustav, u kojem bi došla do izražaja njihova prednost baziranja na numeričkim kalkulacijama. Postigla bi se podjela na više procesa, koji ne bi zahtijevali puno komunikacije, te bi sinkronizacija bila uspješnija.

Iz ovog rada moglo se upoznati s *Google Colaboratory*, kako raditi u njemu, koje su njegove mogućnosti. Također istaknute su i prednosti biblioteka *scikit-learn* pomoću koje su izgrađeni sustavi u ovom radu.

Razvojem ovakvih sustava olakšani su i ubrzani mnogi procesi, a neki sustavi će biti značajni za način svakidašnjeg života i načina življenja. Polako ovakvi sustavi postaju dostupni svima, te mnogi postaju ovisni o mogućnostima koje im ti sustavi pružaju, npr. *Google Assistant* koji je svim korisnicima Google servisa veoma poznat.

Popis literature

- [1.] Batinica, S. (2018). *PRIMJENA BIG DATA TEHNOLOGIJE U ZDRAVSTVU (Doctoral dissertation, Polytechnic of Sibenik. Management.)*. Preuzeto 22.06.2020. sa : <https://zir.nsk.hr/islandora/object/vus%3A921/datastream/PDF/view>
- [2.] Blodget, H. (2014) *You'll Enjoy This Picture Of An IBM Hard Drive Being Loaded Onto An Airplane In 1956*, Preuzeto 20.06.2020. sa : <https://www.businessinsider.com/picture-of-ibm-hard-drive-on-airplane-2014-1>
- [3.] Dalbelo Bašić B. (2010). *Umjetna inteligencija* , Preuzeto 22.06.2020. sa : https://www.ieee.hr/download/repository/UI_1_uvod.pdf
- [4.] Dorri, A., Jurdak, R i Kanhere, S. S. „*Multi-Agent Systems: A survey*” (2018.), Preuzeto 25.06.2019. sa : https://www.researchgate.net/publication/324847369_Multi-agent_Systems_A_survey
- [5.] Đorđević, D. P., & savez Srbije, F. *ANALITIKA VELIKIH PODATAKA U SPORTSKOJ INDUSTRIJI BIG DATA ANALYTICS IN SPORTS INDUSTRY*. Preuzeto 20.06.2020. sa : <http://infotech.org.rs/pdf/121.pdf>
- [6.] Hlevnjak, T. (2019). *Pametna zgrada kao višeagentni sustav (Doctoral dissertation, University of Zagreb. Faculty of Organization and Informatics. Department of Theoretical and Applied Foundations of Information Sciences.)*. Preuzeto 20.06.2020. sa : <https://zir.nsk.hr/islandora/object/foi%3A5324>
- [7.] Hrvatska enciklopedija, (2020). *mrežno izdanje. Leksikografski zavod Miroslav Krleža*, Preuzeto 29.06. 2020. sa : <http://www.enciklopedija.hr/Natuknica.aspx?ID=27600>
- [8.] Huhns, M. N., & Stephens, L. M. (1999). *Multiagent systems and societies of agents. Multiagent systems: a modern approach to distributed artificial intelligence*, 1, 79-114. Preuzeto 20.06. 2020. sa : <https://doc.lagout.org/science/Artificial%20Intelligence/General/Multiagent%20systems%20a%20modern%20approach%20to%20distributed%20artificial%20intelligence%20-%20Gerhard%20Weiss.pdf>
- [9.] Kluk, V. (2018). *Usporedba Data Warehouse i Big Data principa te njihova primjenjivost na sustave daljinskog očitavanja potrošnje prirodnog plina i električne energije (Doctoral dissertation, Josip Juraj Strossmayer University of Osijek. Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Department of Core Courses. Chair of Mathematics and Physics.)*. Preuzeto 20.06.2020. sa : <https://zir.nsk.hr/islandora/object/etfos%3A1942/datastream/PDF/view>

- [10.] Kostić A. (2018) *Logistička regresija*. Preuzeto 27.06.2020. sa : <http://www.matf.bg.ac.rs/p/files/69-logisticka.html>
- [11.] Kshemkalyani, A. D., & Singhal, M. (2011). *Distributed computing: principles, algorithms, and systems*. Cambridge University Press. Preuzeto 23.07.2020. sa : [https://books.google.hr/books?id=G7SZ32dPuLgC&lpg=PR7&ots=fMcwr1KA7e&dq=Ajay%20D.Kshemkalyani%2C%20M.%20S.%20\(2008\).%20Distributed%20Computing%3A%20Principles%2C%20Algorithms%2C%20and%20Systems.%20Cambridge%3A%20Cambridge&lr&hl=hr&pg=PR10#v=onepage&q&f=false](https://books.google.hr/books?id=G7SZ32dPuLgC&lpg=PR7&ots=fMcwr1KA7e&dq=Ajay%20D.Kshemkalyani%2C%20M.%20S.%20(2008).%20Distributed%20Computing%3A%20Principles%2C%20Algorithms%2C%20and%20Systems.%20Cambridge%3A%20Cambridge&lr&hl=hr&pg=PR10#v=onepage&q&f=false)
- [12.] Kumar, A. (2018) *5 Open Source Big Data Platforms*. Preuzeto 20.06.2020. sa : https://linuxhint.com/open_source_big_data_platforms/
- [13.] Marković D. (2015) : *Uvod. Inteligentni agenti*. Preuzeto 22.06.2020. sa : <http://www.mathos.unios.hr/oui/p1.pdf>
- [14.] Michel, F., Ferber, J., & Drogoul, A. (2009). *Multi-agent systems and simulation: a survey from the agents community's perspective*. *Multi-Agent Systems: Simulation and Applications*, 5, 3-52.
- [15.] Morabito, V. (2015). *Big Data and Analytics: Strategic and Organizational Impacts*. Preuzeto 20.06.2020. sa : <https://1library.net/document/7q09wd3q-big-data-and-analytics-bb-pdf.html?tab=pdf>
- [16.] Schneider, R.D. (2013). *Hadoop Buyer's Guide*. *Ubuntu*, Preuzeto 20.06.2020. sa : http://insights.ubuntu.com/wp-content/uploads/HadoopBuyersGuide_sm.pdf
- [17.] Strahija, F. (2018). *Pametni grad kao višeagentni sustav (Doctoral dissertation, University of Zagreb. Faculty of Organization and Informatics. Department of Theoretical and Applied Foundations of Information Sciences.)*. Preuzeto 22.06.2020. sa : <https://repozitorij.unizg.hr/en/islandora/object/foi%3A3899>
- [18.] Stepinac, L. (2018). *Što je to zapravo Big Data i gdje se primjenjuje*. Preuzeto 22.06.2020. sa : <https://www.ictbusiness.info/poslovna-rjesenja/sto-je-to-zapravo-big-data-i-gdje-se-primjenjuje>
- [19.] Stihović, V. (2015). *Alati za analitiku velike količine podataka (big data) (Doctoral dissertation, University of Pula. Faculty of economics and tourism" Dr. Mijo Mirković".)*. Preuzeto 22.06.2020. sa : <https://zir.nsk.hr/islandora/object/unipu%3A230/datastream/PDF/view>
- [20.] Šantić, T. (2017). *Programski jezik Python (Doctoral dissertation, Josip Juraj Strossmayer University of Osijek. Faculty of Electrical Engineering, Computer Science and Information Technology Osijek. Department of Computer Engineering and Automation. Chair of Computer Engineering.)*. Preuzeto 23.06.2020. sa : <https://zir.nsk.hr/islandora/object/etfos:1230>

- [21.] Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed systems: principles and paradigms*. Prentice-Hall. Preuzeto 23.07.2020. sa : <https://slidelegend.com/distributed-systems-principles-and-paradigms-software-engineering-5a8db0651723dd45ca95d0f3.html>
- [22.] Vidaković, D. (2019). *SUSTAVI S DISTRIBUIRANIM PROCESIRANJEM (Doctoral dissertation, Algebra University College.)*. Preuzeto 23.06.2020. sa : <https://zir.nsk.hr/islandora/object/algebra%3A199>
- [23.] Vujnovac, E. (2015). *Utjecaj velike količine podataka na znanstvenonastavni rad (Doctoral dissertation, Josip Juraj Strossmayer University of Osijek. Faculty of Humanities and Social Sciences. Department of Information Sciences.)*. Preuzeto 24.06.2020. sa : <https://repositorij.ffos.hr/islandora/object/ffos:102/datastream/PDF/download>
- [24.] Wooldridge, M. (1999). *Intelligent agents. Multiagent systems*, 6. Preuzeto 22.06.2020. sa : <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.7456&rep=rep1&type=pdf>

Popis slika

Slika 1. Agent Voditelj.....	8
Slika 2. Agent Medijator.....	9
Slika 3. Prikaz pokrenutog Google Colaboratory	31
Slika 4. Prikaz dodavanja podataka u Google Colaboratory	31
Slika 5. Prikaz pokretanja koda u Google Colaboratory	32
Slika 6. Prikaz rezultata treniranja	35
Slika 7. Prikaz matrice za prvi set.....	36
Slika 8. Prikaz grafova za prvi set.....	36
Slika 9. Prikaz matrice za drugi set.....	37
Slika 10. Prikaz točnosti i standardne devijacije za drugi set	37
Slika 11. Prikaz matrice za treći set.....	38
Slika 12. Prikaz grafova za treći set.....	38

Popis tablica

Tablica 1. Umjetna inteligencija	15
Tablica 2. Umjetna inteligencija u humanističkoj i računalnoj znanosti.....	16
Tablica 3. Prikaz korištenih setova podataka	21
Tablica 4. Formule regresija	22
Tablica 5. Dijelovi sustava po setovima podataka.....	33

Prilozi

Prilog 1. : Machine Learning A-Z™: Hands-On Python & R In Data Science (Pristupljeno 30.6.2020.) <https://www.udemy.com/course/machinelearning/>

Prilog 2. : Logistic Regression Practical Case Study (Pristupljeno 30.6.2020.) <https://www.udemy.com/course/logistic-regression-cancer-detection-case-study/>

Prilog 3. : Natural Language Processing (NLP) with BERT (Pristupljeno 30.6.2020.) <https://www.udemy.com/course/natural-language-processing-with-bert/>

Prilog 4. : Artificial Neural Network for Regression (Pristupljeno 30.6.2020.) <https://www.udemy.com/course/linear-regression-with-artificial-neural-network/>

Prilog 5. : scikit-learn Machine Learning in Python (Pristupljeno 30.6.2020.) <https://scikit-learn.org/stable/modules/classes.html>

Prilog 6. : Google Colaboratory <https://colab.research.google.com/notebooks/intro.ipynb>

Prilog 7. : UC Irvine Machine Learning Repository, Breast Cancer Wisconsin (Original) Data Set (Pristupljeno 30.6.2020.) <https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>

Prilog 8. : Kaggle Simulations Avocado Prices (Pristupljeno 30.6.2020.) <https://www.kaggle.com/neuromusic/avocado-prices>

Prilog 9. : Google Drive datoteka sa svim programskim rješenjima i setovima podataka (Pristupljeno 30.6.2020.) https://drive.google.com/drive/folders/1-i8ooF_liafkxNRMq-jXZDE_1hBiFsXz?usp=sharing