Strojno učenje kao alat za zaključivanje

Gregorić, Marin

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak. god. 2018./2019.

Marin Gregorić

Strojno učenje kao alat za zaključivanje

Završni rad

Mentor: prof. dr. sc. Sanja Seljan

Zagreb, rujan 2019.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

_____

(potpis)

Sadržaj

1. Uvod

Ovaj rad zamišljen je kao prezentacija strojnog učenja u suvremenom svijetu. U radu je objašnjeno strojno učenja kao interdisciplinarni oblik djelovanja na kojeg su utjecale mnoge znanstvene discipline poput statistike i umjetne inteligencije te je prikazano kako postaje sve važniji dio suvremenog znanstvenog rada. Opisani su pojmovi koji se često koriste u kontekstu strojnog učenja kako bi se omogućilo razumijevanje strojnog učenja kao alata za odlučivanje i izvlačenje podataka iz novih izvora koje je omogućilo informacijsko doba. Strojno učenje je prezentirano kroz klasifikaciju sustava strojnog učenja te kroz pobliže objašnjavanje osnovnih algoritama strojnog učenja. Prikazane su suvremene primjene strojnog učenja, kao i utjecaj pojave dubokog učenja na suvremenu znanstvenu metodologiju, ali i na neznanstveni svijet kroz primjere. Također, u radu se prezentira nekoliko alata temeljenih na strojnom učenju koji se koriste za generiranje, to jest, za stvaranje sadržaja putem generativnih neuronskih mreža. Cilj ovog rada jest izrada programa moguće dijagnoze dijabetesa koristeći strojno učenje.

U drugome dijelu rada opisana je izrada sustava nenadziranog strojnog učenja za klasifikaciju potencijalnih dijabetičara. Objašnjen je i opisan postupak analize podataka, odabira modela, treniranja modela i interpretacije rezultata testiranja sa svrhom izrade dijagnostičkog programa baziranog na strojnom učenju. U projektu su korištena četiri algoritma: logistička regresija, k najbliži susjedi te stroj potpornih vektora linearnog i RBF kernela (eng. *kernel*). Rezultati su potom uspoređeni i interpretirani te su navedena moguća poboljšanja.

2. Strojno učenje - uvodno

Aurelien Geron (2017) navodi dvije definicije strojnog učenja. Po jednoj definiciji, strojno učenje je područje istraživanja koje daje računalima mogućnost učenja bez da su eksplicitno programirani, a po drugoj definiciji Tom Mitchella iz 1997. godine, strojno učenje podrazumijeva računalni program koji uči iz iskustva E za zadatak T gdje P označava uspješnost, ako njegova uspješnost P na zadatku T raste sa iskustvom E. Alpaydin (2004) definira strojno učenje kao proces programiranja računala da optimiziraju izvođenje kriterija koristeći podatke ili stečeno iskustvo. Strojno učenje omogućuje strojevima, najčešće računalima, da uče, odnosno, da postaju sve bolji u rješavanju određenih problema u odnosu na „iskustvo“. Drugim riječima, stroj uči na podacima, a što više podataka je dostupno stroju to je veća mogućnost da sa boljom preciznošću ili točnošću riješi problem - to jest, raste uspješnost.

Potrebno je biti upoznat sa širom slikom, pojmovima i znanostima koje su utjecale na stvaranje strojnog učenja. Strojno učenje je bazirano na metodama iz raznih disciplina poput statistike, umjetne inteligencije i rudarenja podataka (eng. *data mining*), prema (Alpaydin, 2004). Da bi stroj ili računalo moglo učiti, mora imati dostupne određene resurse iz kojih može učiti, a to su u ovom slučaju podaci. Uspješnost programa za strojno učenje uvelike ovisi o količini, kvaliteti i relevantnosti podataka na kojima je treniran prema (Geron, 2017). Treniranje je u ovome kontekstu proces u kojemu sustav uz pomoć raznih algoritama uči iz podataka i na temelju tih podataka stvara određeni model uz pomoć kojega će moći rješavati zadatke.

S obzirom da strojno učenje ovisi o podacima i da cijeli proces strojnog učenja može biti kompromitiran lošim podacima, valja znati što je podatak. „Podatak [je] poznata ili pretpostavljena činjenica na osnovi koje se oblikuje informacija. Sastoji se od skupa kvantitativnih parametara koji se mogu zapisivati kao nizovi znakova ili nizovi brojeva“. (enciklopedija.hr).

Prema izvoru Enciklopedije Leksikografskog zavoda „Miroslav Krleža“ - „Statistika [je] grana primijenjene matematike koja se bavi prikupljanjem, uređivanjem, analizom i tumačenjem podataka i donošenjem zaključaka o pojavama i procesima koje ti podaci predočuju. U širem smislu, [to su] podaci o različitim prirodnim, društvenim i drugim pojavama i procesima“. Statistika je osnova čiji alati i metode omogućuju postojanje znanosti poput podatkovne znanosti (Weihs & Ickstadt, 2018).

Podatkovna znanost (eng. *data science*) nastala je iz statistike kao šira disciplina (Longbing, 2017). Longbing Cao navodi više definicija podatkovne znanosti (eng. *data science*), poput šire definicije po kojoj je podatkovna znanost - znanost o podacima ili studija podataka. Na podatkovnu znanost kao disciplinu utječu mnoge druge znanosti, poput informatike, računalne znanosti, matematike i statistike (Weihs & Ickstadt, 2018), što je vidljivo iz definicije Caoa po kojoj podatkovnu znanost tvore statistika, informatika, računanje, komunikacija, sociologija i upravljanje, primijenjeni na podacima u okolišu s ciljem transformacije podataka na principu podatak-znanje-mudrost. Russ i suradnici (2019) pak navode da definicija podatkovne znanosti nije jasna. Odnosno, navode da je definirana kao četvrta paradigma znanosti, kao koncept za ujedinjenje statistike, analize podataka i njihovih metoda, kao sinonim statistike, a i kao aktivnost koja sadrži multidisciplinarna istraživanja modela i metoda podataka te također navode da nema dogovora oko nastavnih planova raznih smjerova podatkovne znanosti. Također, neki autori tvrde da statistika i strojno učenje imaju središnju ulogu u podatkovnoj znanosti (Longbing, 2017) čime strojno učenje spada pod metode podatkovne znanosti. Kao i strojno učenje, veliki podaci (eng. *big data*) predmet su rasprava u polju podatkovne znanosti (Longbing, 2017).

Pejić Bach i suradnici (2019) navode da velika količina podataka (eng. *big data*) obuhvaća različite vrste podataka u strukturiranim, polustrukturiranim i nestrukturiranim dokumentima. Veliki podaci (eng. *big data*) najčešće se definiraju kao 3V: što uključuje volumen, to jest, velike količine podataka (eng. *volume*), raznolikost (eng. *variety*) i promjenjivost (eng. *velocity*). Dakle, podaci koji su velikog opsega, dobiveni iz širokog raspona izvora, ili kao podaci koji su prikupljeni velikom brzinom (eng. *velocity*). Pojedini istraživači navode i karakteristike vrijednosti (eng. *value*), varijabilnosti (eng. *variability*) i istinitosti (eng. *veracity*), prema (Bach, Krstić & Seljan, 2019). Često određene znanosti imaju svoje definicije pojma kao, na primjer, u kontekstu istraživanja pretilosti gdje se naziv veliki podaci (eng. *big data)* odnosi na podatke koji su prikupljeni u neke druge svrhe, ali mogu pridonijeti vrijednosti tradicionalnih podatkovnih izvora (Vogel et al., 2019).

Rudarenje podataka (eng. *data mining*) još je jedan pojam koji se često nalazi u literaturi. Rudarenje podataka ili dubinska analiza podataka je naziv nastao u poslovnom svijetu kao naziv za primjenu algoritama strojnog učenja na velikim količinama podataka (Alpaydin, 2004). Odnosno, rudarenje podataka je proces otkrivanja uzoraka u velikim nestrukturiranim skupovima

podataka pritom koristeći razne metode (najčešće strojnog učenja) za stvaranje modela prema (Marrara, Pejić, Seljan, Topalovic, 2019). Dubinska analiza teksta (eng. *text mining, text analytics*) je podvrsta rudarenja podataka gdje je cilj analizirati tekstualni dokument i izvući podatke koji će omogućiti neki oblik odlučivanja (Bach, Krstić, Seljan & Turulja, 2019).

Strojno učenje nije samo problem podataka, nego je i dio umjetne inteligencije. Alpaydin (2004.) tvrdi - da bi sustav bio inteligentan potrebno je da ima sposobnost učenja u promjenjivoj okolini. Prema web enciklopediji leksikografskog zavoda „Miroslav Krleža“ - “Umjetna inteligencija [je] dio računalne znanosti (informatike) koji se bavi razvojem sposobnosti računala da obavljaju zadaće za koje je potreban neki oblik inteligencije, to jest, da se mogu snalaziti u novim prilikama, učiti nove koncepte, donositi zaključke, razumjeti prirodni jezik, raspoznavati prizore i drugo“. Iz navedenog može se vidjeti zašto je strojno učenje velik dio umjetne inteligencije.

3. Klasificiranje sustava strojnog učenja

Po Geronu (2017), klasificiranje sustava baziranih na strojnom učenju može se vršiti u više kategorija. U prvoj kategoriji uzima se u obzir jesu li trenirani sa ljudskim nadzorom ili ne. Po tome sustav se dijeli na: nadzirano (eng. *supervised*), nenadzirano (eng. *unsupervised*) i polunadzirano (eng. *semisupervised)* učenje te učenje uz podršku (eng. *reinforcement learning*, „podržano učenje" ili učenje pojačavanjem). Druga kategorija razvrstava sustave po tome jesu li sposobni učiti inkrementalno (eng. *online learning*) ili isključivo uče odjednom (eng. *batch learning*). Treća podjela prema Geronu (2017.) je po tome može li sustav stvoriti model predviđanja s obzirom na uzorke u podacima za treniranje ili jednostavno uspoređuje nove podatke sa poznatim podacima. Prema toj podjeli postoje sustavi kojima je učenje temeljeno na primjerima (eng. *instance-based*) i sustavi kojima je učenje temeljeno na modelima (eng. *model-based*).

3.1. Vrste sustava strojnog učenja s obzirom na razinu nadzora

U ovome poglavlju prikazat će se osnovna podjela sustava strojnog učenja s obzirom na razinu nadzora, to jest, na nadzirano, nenadzirano, polunadzirano učenje i učenje uz podršku.

3.1.1. Nadzirano učenje (eng. *supervised learning*)

U nadziranom učenju sustav prima označene podatke s ciljem da pronađe određenu vrijednost ili da nadolazeće podatke svrsta u klase. Najčešći zadatak kojeg sustavi nadziranog učenja rješavaju je klasifikacija (Geron, 2017).

Klasifikacija podrazumijeva „pospremanje" novih podataka u neku od prethodno definiranih kategorija ili klasa s obzirom na svojstva tih podataka. Primjer klasifikacije na djelu je spam filter. Spam filter se trenira na e-mailovima koji su prethodno označeni ili kao spam ili kao siguran mail te potom pokušava klasificirati nadolazeće e-mailove u jednu od te dvije prethodno definirane kategorije (Geron, 2017).

Drugi čest zadatak za sustave nadziranog učenja jest regresija (Geron, 2017). Regresija je problem u statistici i strojnom učenju gdje se pokušava zaključiti vrijednost regresivne funkcije (eng. *regression function*) čije vrijednosti odgovaraju prosjeku izlazne varijable (eng. *response variable, output variable, dependant*) na koju utječu jedna ili više ulazne varijable (eng. *input variable*) (Sammut & I. Webb, 2017). Jednostavnije, regresijom se pokušava dobiti određena brojčana vrijednost s obzirom na odnose među svojstvima podataka; odnosno, sustav pokušava otkriti povezanost između svojstava te time predvidjeti traženu brojčanu vrijednost (Geron, 2017). Kao primjer može se navesti hipotetski sustav za procjenu cijene nekretnina. Takvom sustavu dani su određeni podatci o postojećim nekretninama koji sadrže svojstva poput godine izgradnje nekretnine, veličine prostora, pozicije i cijene. Tom sustavu je cilj regresijom predvidjeti cijenu nadolazećih nekretnina.

Neki od algoritama nadziranog učenja koji se primjenjuju na probleme klasifikacije i regresije su: k-najbliži susjedi (eng. *k-nearest neighbors*, KNN), linearna regresija (eng. *linear regression*), logistička regresija (eng. *logistic regression*), algoritmi stroja potpornih vektora (eng. *support vector machines*, SVM), algoritmi stabla odluka (eng. *decision trees*), nasumične šume (eng. *random forests*) i neuronske mreže (eng. *neural networks*, NN), prema (Geron, 2017).

K-najbliži susjedi ili skraćeno KNN je algoritam koji pretpostavlja da su slične stvari blizu jedna drugoj. Na toj pretpostavci algoritmom se izračuna udaljenost između točaka u prostoru. Što je ta udaljenost manja, dvije točke su sličnije. KNN ima široku primjenu u zadacima nadziranog učenja (Harrison, 2018).

Linearna regresija (eng. *linear regression*) je algoritam kojim se pokušava modelirati odnos između ulazne varijable i izlazne varijable koristeći linearnu ili vektorsku funkciju. Logistička regresija je algoritam kojim se metode linearne regresije primjenjuju na klasifikacijske zadatke (Sammut & I. Webb, 2017). Logističkom regresijom može se, u slučaju klasifikacije, dobiti vrijednost vjerojatnosti pripadanja određenoj klasi (Geron, 2017).

SVM je naziv za skup linearnih algoritama koji se mogu koristiti na mnogim zadacima poput klasifikacije, regresije i procjene gustoće vrijednosti. SVM algoritmi se uspješno primjenjuju na probleme u bioinformatici, odnosno, u obradi prirodnog jezika (Sammut & I. Webb, 2017).

3.1.2. Nenadzirano učenje (eng. *unsupervised learning*)

Geron (2017) navodi nenadzirano učenje kao naziv za pristup zadatku bez nadzora u kojem sustav pokušava sam naučiti nad, podacima za učenje koji nisu nužno označeni. S obzirom na to da podaci nisu označeni, algoritmi nenadziranog učenja mogu ukazati na dosad nepoznate korelacije. Stoga, takav pristup najčešće se koristi za grupiranje (eng. *clustering*, klastering) ili za vizualizaciju (eng. *visualization*) podataka, (Krstić, Seljan & Zoroja, 2019). Također se koristi i za redukciju dimenzionalnosti (eng. *dimensionality reduction*), detekciju anomalija (eng. *anomaly detection*) i asocijativno učenje (eng. *association rule learning*), prema (Geron, 2017).

Grupiranje ili klastering je proces kojim algoritam sam pokušava skup podataka grupirati po sličnosti s obzirom na neka njihova svojstva. Takav proces predstavlja način za pronalaženje neopaženih veza između podataka. Neki od algoritama i metoda koje se koriste za grupiranje su: K prosječne vrijednosti (eng. *K-Means*), hijerarhijska klaster analiza (eng. *Hierarchical Cluster Analysis*, HCA) i maksimizacija očekivanja (eng. *Expectation Maximization*, EM), prema (Geron, 2017). *K-means* je popularna metoda grupiranja podataka, korištena u raznim situacijama sa širokim rasponom primjena. Radi na principu da uzme osnovno, a ne optimalno, grupiranje te premjesti svaku točku na njezin novi najbliži centar, ažurira klastering centre tako što izračuna prosječnu vrijednost točaka članova, te ponavlja taj proces sve dok se ne postigne unaprijed definirana vrijednost pokrivenosti. (Sammut & I. Webb, 2017).

Algoritmi za vizualizaciju rade na način da uzmu veliku količinu kompleksnih neoznačenih podataka te „izbace“ 2D ili 3D prikaz tih podataka. Oni omogućuju istraživačima da vizualno uoče nove odnose između naizgled nasumičnih podataka. Redukcija dimenzionalnosti je pak proces kojim sustav pokušava pojednostaviti skupove podataka bez prevelikih gubitaka informacija. Najčešće se to postiže spajanjem više povezanih značajka u jednu. Algoritmi i metode koje se koriste za vizualizaciju i redukciju dimenzionalnosti su: analiza glavnih klastera (eng. *Principal Cluster Analysis*, PCA), „kernel“ analiza glavnih klastera (eng. *Kernel PCA*), lokalno-linearno ugrađivanje (eng. *Locally-Linear Embedding*, LLE) i t-SNE (eng. *t-distributed Stohastic Neighbor Embedding*), prema (Geron, 2017).

Detekcija anomalija je proces koji se ponekad primjenjuje na podacima kao korak prije treniranja nekog sustava jer omogućuje pronalaženje anomalija u skupu podataka. Također, može se primijeniti i na druge probleme poput primjećivanja neobičnih bankovnih transakcija (Geron, 2017).

Metode asocijativnog učenja (eng. *Association rule learning*) idealne su za pronalaženje veza između atributa u velikim količinama podataka. Algoritmi koji se koriste za metode asocijativnog učenja (eng. *association rule learning*) su Apriori i Eclat (Geron, 2017). Apriori algoritam je metoda rudarenja podataka kojom dobivamo sve podatke iznad određene čestote (eng. *frequency*) kao i asocijativna pravila (Sammut & I. Webb, 2017).

3.1.3. Polunadzirano učenje (eng. *semisupervised learning*)

U algoritme polunadziranog (eng. s*emisupervised*) učenja spadaju svi algoritmi koji su sposobni raditi sa djelomično označenim podacima i velikim količinama neoznačenih podataka. Najčešće su to kombinacije algoritama nadziranog i nenadziranog učenja.

Primjeri algoritama polunadziranog učenja su duboke mreže vjerovanja (eng. *deep belief networks*, DBN) i ograničeni Boltzmann strojevi (eng. *restricted Boltzmann machines*, RBM), prema (Geron, 2017). S obzirom na to da algoritmi polunadziranog učenja pokušavaju riješiti isti problem kao i algoritmi nadziranog učenja, mogu se koristiti gotovo u svim situacijama gdje se koristi i nadzirano učenje. Često se koriste na problemima obrade prirodnog jezika u obliku polunadzirane obrade teksta (eng. *semi-supervised text processing*) (Sammut & Webb, 2017). Za razliku od nadziranih i nenadziranih sustava, polunadzirani sustav može djelovati i na označenim i na neoznačenim skupovima podataka.

Često se uzima manji skup za treniranje sastavljen od označenih podataka i veći radni skup neoznačenih podataka. Sustav se onda evaluira na testnom setu sastavljenom od neoznačenih podataka. S obzirom na to da se takav sustav može trenirati na označenim podacima i poboljšati sa neoznačenim podacima, predstavlja dobro rješenje za probleme obrade teksta i jezika zato što su velike količine neoznačenih tekstualnih podataka puno dostupnije od onih označenih, prema (Sammut & I. Webb, 2017).

3.1.4. Učenje uz podršku ili podržano učenje (eng. *reinforcement learning*)

Za probleme gdje nije moguće koristiti nadzirano učenje, a poznato je, donekle, kakav rezultat priželjkujemo od sustava, najbolje je koristiti sustave podržanog učenja (Alpaydin, 2004). Podržano učenje (eng. *reinforcement learning*) je naziv za veliku skupinu metoda učenja u kojima sustav ili agent (eng. *agent*) može promatrati okolinu (eng. *environment*) i s obzirom na nju izabrati i vršiti radnje koje rezultiraju u nagradi (eng. *reward*) ili kazni (eng. *penalty*) za agenta. S obzirom na nagrade i kazne, sustav treba stvoriti strategiju ili politiku (eng. *policy*) kojom će dobiti što više nagrada ili što manje kazni. Na taj način sustav uči sam i teoretski može doći do idealnih rješenja za određene probleme. Strategija ili politika (eng. *policy*) definira što će agent ili sustav učiniti u određenoj situaciji (Geron, 2017). Za razliku od nadziranog učenja, u podržanom učenju nema označenih primjera dobrog i lošeg djelovanja, no za razliku od nenadziranog učenja, čovjek sudjeluje u ovom načinu učenja time što dodjeljuje nagrade, odnosno, određuje u kojim uvjetima te za što će sustav ili agent biti nagrađen (Sammut & I. Webb, 2017). Podržano učenje često se primjenjuje na robotima za učenje radnji poput hodanja (Geron, 2017).

3.2. Podjela na offline učenje (eng. *offline learning*) i online učenje (eng. *online learning*)

3.2.1. Sustavi grupnog ili offline učenja (eng. *batch learning*, eng. *offline learning*)

Sustavi grupnog učenja (eng. *batch learning*) ili offline učenja (eng. *offline learning*) su sustavi koji nisu sposobni učiti inkrementalno, nego se treniraju na svim dostupnim podacima odjednom. Treniranje ili „učenje“ sustava strojnog učenja je resursno intenzivan proces. Stoga, pristup grupnog učenja predstavlja mnoge probleme poput otežane nadogradnje sustava novim podacima zbog toga što je potrebno ponovo trenirati sustav nad svim podacima uključujući i nove podatke i podatke na kojima je već treniran. No, danas se taj proces nadogradnje lako može automatizirati, iako i dalje ostaju problemi povezani s potrošnjom računalnih resursa, to jest, takve je sustave gotovo nemoguće primijeniti na pametne telefone ili ostale uređaje koji ne raspolažu sa velikom količinom računalne moći (Geron, 2017).

3.2.2. Sustavi online učenja (eng. *online learning*)

Navedene probleme sustava grupnog učenja (eng. *batch learning*) mogu riješiti sustavi online učenja (eng. *online learning*). Sustavi online učenja uče inkrementalno sekvencijalnim primanjem podataka u manjim količinama (eng. *mini-batches*). S obzirom na malu količinu podataka koju prihvaća, svaka sekvenca je jeftina i brza pa sustav može učiti kako podaci stižu (eng. *on the fly*). Takvi sustavi dobri su za primjenu na zadacima gdje se podaci konstantno mijenjaju ili na uređajima s ograničenom količinom računalnih resursa poput pametnih telefona. Također, algoritmi online učenja mogu se primijeniti na velikim količinama podataka koji ne stanu u memoriju sustava. Bitna značajka sustava online učenja je stopa učenja (eng. *learning rate*). Stopa učenja diktira koliko će brzo sustav „zaboraviti" prijašnje podatke, ali i koliko će ih brzo ili često učiti. Nedostatak sustava online učenja jest to što, u slučaju da nauči na lošoj skupini podataka, kvaliteta sustava pada te, ako je stopa učenja velika, kvaliteta može brzo pasti (Geron, 2017).

3.3. Podjela na učenje na temelju primjera (eng. *instance-based learning*) i učenje na temelju modela (eng. *model-based learning*)

3.3.1. Učenje na temelju primjera (eng. *instance-based learning*)

Učenje na temelju primjera (eng. *instance-based learning*) odnosi se na skup metoda za klasifikaciju i regresiju gdje je rezultat baziran na sličnosti upita (eng. *query*) najbližem susjedu (ili najsličnijem podatku) u skupu za treniranje. U tom slučaju, sustav ne stvara model s obzirom na podatke za učenje, nego, jednostavno, pohranjuje sve podatke te kod upita pregledava sve podatke u potrazi za rješenjem (Sammut & I. Webb, 2017). Na taj način djeluje algoritam K najbližih susjeda (Varghese, 2018).

3.3.2. Učenje na temelju modela (eng. *model-based learning*)

Još jedan način na koji mogu raditi sustavi strojnog učenja je da se izgradi model stvoren na skupu podataka. To je učenje na temelju modela (eng. *model-based learning*). Algoritmi učenja na temelju modela generaliziraju s obzirom na uzorke ili trendove koje pronalaze među podacima. Na primjer, linearna regresija stvara linearni model po kojem se predviđaju slučajevi koji slijede (Geron, 2017). Jednostavnije, u sustavu učenja na temelju modela stvara se model zasnovan na podacima za treniranje, što znači da sustav ne mora „pamtiti“ sve podatke iz skupa za treniranje nego se novi upiti samo uspoređuju sa stvorenim modelom.

4. Suvremena primjena strojnog učenja

Danas, strojno učenje postaje jedna od najpopularnijih i najpotrebitijih disciplina. Poslovi poput inženjera strojnog učenja (eng. *machine learning engineer*) i specijalista strojnog učenja (eng. *machine learning specialist*) su među pet najbrže nastajućih poslova prema (Columbus, 2018); također, poslovi u području strojnog učenja jedni su od najbolje plaćenih prema (Indeed Editorial Team, 2019). O popularnosti i potrebi za strojnim učenjem svjedoči i to što su zajednice koje rade na projektima strojnog učenja jedne od najbrže rastućih (Elliott, 2018). Uz pomoć strojnog učenja, umjetna inteligencija dostigla je dosad nezamislivu razinu. Danas računala uz pomoć strojnog učenja mogu izvršavati zadatke poput vizualnog prepoznavanja na razini prosječnog odraslog čovjeka. Takvu razinu umjetne inteligencije omogućila je dostupnost ogromnih količina podataka koji su popločili put za razvitak i primjenu takozvanog dubokog učenja (Sejnowski, 2018).

4.1. Duboko učenje (eng. *deep learning*)

Procesiranje podataka u njihovom sirovom obliku nemoguć je zadatak za tehnike konvencionalnog strojnog učenja. Za obradu podataka konvencionalnim strojnim učenjem potrebno je prethodno obraditi podatke na način da su „čitljivi“ sustavu ili algoritmu, što zahtijeva mnogo stručnog znanja o predmetu (LeCun, Bengio & Hinton, 2015). Jedan od primjera tog nedostatka je na polju medicine u klasifikaciji tumora na maligni i benigni. U slučaju da je izabran broj stanica kao značajka po kojoj klasificiramo tumor, konvencionalni algoritam strojnog učenja će po toj značajci i klasificirati. No, ne znamo da li je značajka, koja je izabrana važna ili ne. (Eraslan, Avsec, Gagneur & Theis, 2019). Taj problem pokušava se riješiti reprezentativnim učenjem.

Reprezentativno učenje (eng. *representation learning*) je naziv za skup metoda koje omogućuju sustavu da prima podatke u sirovom obliku i samostalno otkrije reprezentacije potrebne za otkrivanje i klasifikaciju. To je osnova dubokog učenja. Metode dubokog učenja su metode reprezentativnog učenja sa više razina reprezentacije. Jednostavnije, duboko učenje je naziv za metode koje od sirovih podataka kroz razne razine transformacije stvaraju apstrakcije koje „pojednostave“ te podatke. Takvim pristupom, kroz više razina transformacija, moguće je naučiti vrlo kompleksne funkcije poput „razumijevanja“ fotografija ili samovozećih automobila (LeCun,

Bengio & Hinton, 2015). Pojava dubokog učenja je olakšavanjem obrade podataka povećala dostupnost podataka, odnosno, povećala broj obradivih podataka.

4.2. Strojno učenje u bio-znanostima i medicini

Metode strojnog učenja široko su primjenjivane u polju genetike i genomike. Strojno učenje je najutjecajnije na problemu interpretacije i klasifikacije velikih skupova podataka o genomima s ciljem označavanja elemenata genskog slijeda. Prema (Libbrecht & Stafford, 2015), algoritmi za strojno učenje mogu naučiti prepoznavati uzorke u DNK i RNA sekvencama. Kao i u drugim disciplinama, pojava dubokog učenja kao grane strojnog učenja predstavlja potencijalno rješenje nekih nedostataka konvencionalnog strojnog učenja.

Najveći nedostatak konvencionalnog strojnog učenja je to da uspješnost algoritma strojnog učenja uvelike ovisi o podacima, odnosno, o tome kako su podaci reprezentirani. To se pokušava riješiti primjenom metoda dubokog učenja kojima je sustav sposoban sam odrediti koje su značajke podataka „bitne“ (Eraslan, Avsec, Gagneur & Theis, 2019). Strojno učenje se primjenjuje i u farmakologiji za razvoj i otkrivanje lijekova. To je omogućila veća dostupnost podataka, kao i primjena metoda dubokog učenja (Ekins et al., 2019). Strojno učenje se primjenjuje i u drugim aspektima medicine kao, na primjer, u predviđanju tijeka bolesti poput dijabetesa (Makino et al., 2019) ili u procjeni stanja bolesnika (Cosgriff et al., 2019).

4.3. Strojno učenje u obradi prirodnog jezika

Strojno učenje omogućilo je razvoj obrade prirodnog jezika. Od 1980-ih strojno učenje je dio obrade prirodnog jezika - (Otter, Medina & Kalita, 2018). Jedna od metoda koja se primjenjuje u obradi prirodnog jezika je SVM (stroj potpornih vektora). SVM aktivnog učenja omogućio je klasifikaciju teksta bez oslanjanja na isključivo označene podatke. Odnosno, sustav može učiti uz pomoć lako dostupnih neoznačenih podatka (Tong & Koller, 2001). No, u zadnje vrijeme, pojavom dubokog učenja, obrada prirodnog jezika doživljava još jednu evoluciju (Otter, Medina & Kalita, 2018).

4.4. Strojno učenje kao alat za generiranje

GAN (eng. *Generative Adversarial Networks*) je naziv za skupinu algoritama koji putem neuronskih mreža generiraju sadržaj (Nicholson, 2019). Primjer strojnog učenja koje se koristi za generiranje bio bi alat GauGAN. GauGAN alat baziran je na radu „Semantic Image Synthesis with Spatially-Adaptive Normalization“ (nvidia-research-mingyuliu.com). Na „GitHub“ repozitoriju programa dostupnom na: https://github.com/NVlabs/SPADE moguće je preuzeti program i koristiti ga u nekomercijalne svrhe. GauGAN alat je dostupan u „beta“ verziji na: http://nvidia-research-mingyuliu.com/gaugan/ . Alat omogućuje stvaranje foto-realističnih grafika jednostavnim skiciranjem krajolika (Slika 1, Slika 2).

Slika 1: Skica

Slika 2: Generirana grafika

Drugi projekt u kojemu se strojnim učenjem generiraju slike koristeći GAN je StackGAN. StackGAN omogućuje stvaranje visokokvalitetnih slika iz tekstualnih opisa istih. StackGAN projekt dostupan je, kao i GauGAN, na „GitHub“ repozitoriju (Zhang et al, 2016). Također, GAN algoritmi mogu se koristiti i za generativno predviđanje. U radu „Generating Videos with Scene Dynamics“ autori prezentiraju program koji uči iz velike količine neoznačenih video podataka kako bi prepoznao što se događa u određenoj sličici videa (klasifikacija) te kako bi mogao predvidjeti što slijedi i generirati novu sličicu (Vondrick, Pirsiavash & Torralba, 2016).

4.5. Otvorena znanost

Otvorena znanost (eng. *open science*) ili znanost 2.0 (eng. *science 2.0*) je koncept koji objedinjuje više ideja poput otvorenog pristupa znanju, problema pristupačnosti stvaranju znanja, suradničkog istraživanja, ali i ideja otvorenih podataka (Fecher & Frieske, 2014).

Otvoreni podaci (eng. *open data*) su naziv za koncept koji podrazumijeva odvajanje znanstvenih podataka istraživanja od izdavača i općenito otvaranje pristupa podacima svima (Fecher & Frieske, 2014). Povećan pristup znanstvenim podacima i radovima kroz ideju o otvorenim podacima i otvorenoj znanosti mogao bi pomoći razvoju znanosti putem primjene metoda strojnog učenja. O tome svjedoči rad „Unsupervised word embeddings capture latent knowledge from materials science literature“ gdje su autori korištenjem metoda nadziranog i nenadziranog strojnog učenja pokazali mogućnost ekstrakcije znanja i odnosa iz znanstvene literature (Tshitoyan et al, 2019). U pogledu samog strojnog učenja postoji napor da se podaci učine dostupnijima putem otvorene znanosti u strojnom učenju putem platformi poput „OpenML“ i „mldata“ (Vanschoren, Braun & Ong 2014).

5. Projektni rad

Projektni rad obuhvaća izradu sustava za detekciju mogućnosti u domeni dijabetesa. Projekt je izrađen u Python programskom jeziku verzije 3.7. Python je izabran zbog toga što je interpretirani jezik visoke razine s jednostavnom sintaksom (Python.org). S obzirom na te značajke, Python je lako razumljiv i pogodan za brzo programiranje i stvaranje prototipa.

Također, Python se može proširiti raznim paketima (eng. *packages*) i knjižnicama (eng. *library*), poput NumPy, pandas ili sickit-learn. Ovaj program izrađen je u distribuciji otvorenog koda (eng. *open-source*) pod nazivom Anaconda. Anaconda distribucija bazirana je na Python programskom jeziku i sadrži mnoštvo paketa dizajniranih za olakšavanje procesa strojnog učenja i alata za razvoj projekata podatkovne znanosti (Anaconda.com). Program je napisan u „Spyder“ (Scientific Python Development Environment) razvojnom okruženju zbog njegovih brojnih mogućnosti koje olakšavaju rad u Python programskom jeziku poput označavanja sintakse i drugih alata.

5.1 Skup podataka

Podaci korišteni u programu preuzeti su s javno dostupnog izvora, s platforme koja nudi veliki broj skupova podataka te resursa za učenje i primjenu strojnog učenja u obliku natjecanja. Skup podataka koji je korišten je „Pima Indians Diabetes Database“. Navedeni skup izvorno je dio skupa podataka američkog Nacionalnog instituta za dijabetes i probavne i bubrežne bolesti (eng. *National Institute of Diabetes and Digestive and Kidney Diseases*) te samo sadrži podatke ženskih pacijenata starijih od 21 godine Pima indijanskog podrijetla.

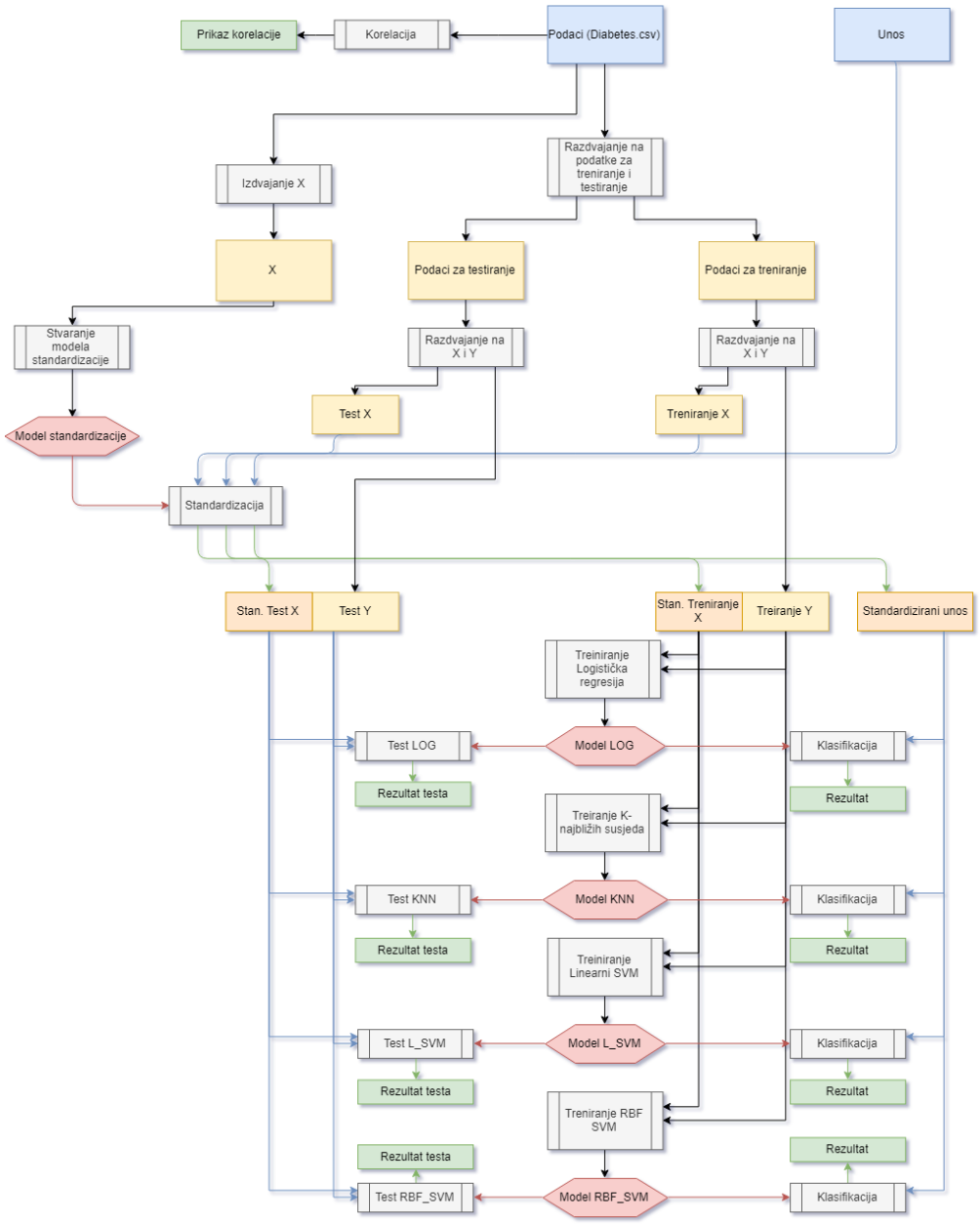
Skup podataka sastavljen je od 768 pojava (pacijenata) sa devet značajki ili dimenzija. Značajke u skupu su: broj trudnoća, razina glukoze, krvni tlak, debljina kože, inzulin, BMI (eng. *body mass indeks*, indeks tjelesne mase), DPF (Diabetes Pedigree Function), broj godina i, najbitnija, ishod.

Cilj programa je podatke ulaznog prostora (X) koristiti za treniranje modela koji će klasificirati nove pojave u skup oznaka (Y). Drugim riječima, cilj programa bio je s obzirom na prvih osam značajki (X) predvidjeti nove pojave, odnosno, klasificirati nove pojave u skupinu „Y“ što je u

ovom slučaju skupina ishod („Outcome“) gdje su podaci u obliku 1 ili 0 u kojem 1 i 0 predstavljaju ima li osoba dijabetes (1) ili nema (0). S obzirom na zadatak projekta ovaj program spada u programe nenadziranog učenja.

Navedeni skup podataka dobar je za prve projekte strojnog učenja jer je dovoljno malen (9kb) da je moguće trenirati model na prosječnom računalu, a oblikovanje i značajke skupa omogućavaju zadovoljavajuće rezultate. Ograničenje ovoga rada jest što modeli stvoreni nad skupom takvog uskog opsega, nisu idealni za primjenu u problemima stvarnoga svijeta, što se vidi iz kasnijih rezultata, ali omogućuju razumijevanje osnovnih procesa strojnog učenja.

5.2 Proces izrade sustava



Slika 3: Proces projekta

Slika 3 pokazuje proces kojim je izrađen sustav za klasifikaciju, to jest, kako je tekao proces treniranja modela i podjele skupova te same klasifikacije.

5.2.1 Kratka analiza podataka

Prvi korak u projektu bio je napraviti kratku analizu podataka. Prvo su ispisani podaci da bi se utvrdio točan broj pojava i značajki (Slika 4). Skup podataka se sastoji od 768 redova što pokazuje da se radi o 768 osoba te 9 stupaca što su u ovom slučaju značajke. Potom, podaci su vizualizirani koristeći „seaborn“ knjižnicu (eng. *library*) koja omogućuje jednostavnu vizualizaciju podataka u Pythonu (Slika 5).

```
[768 rows x 9 columns]
```

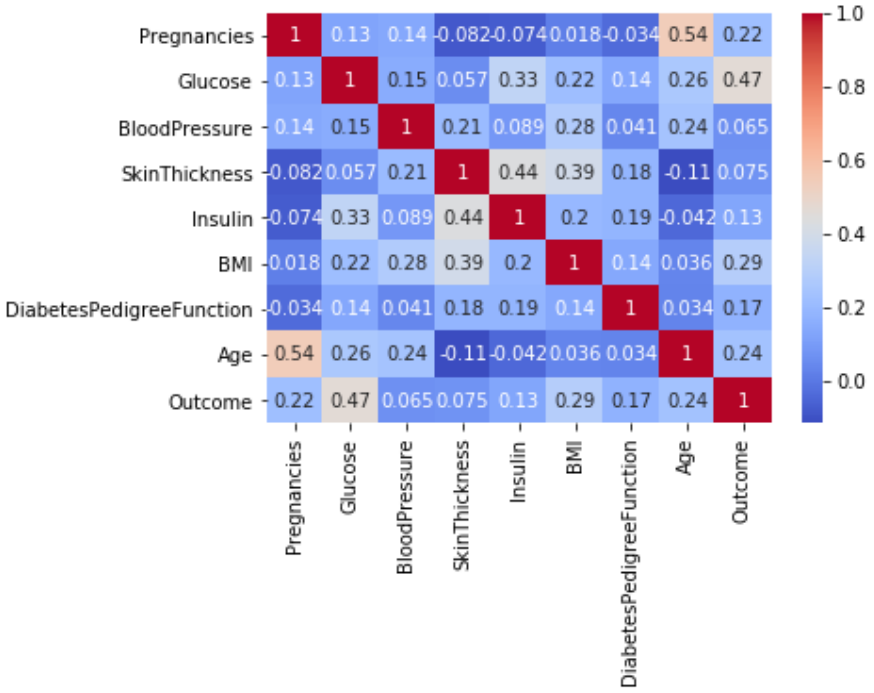
Slika 4: Ispis podataka

Nakon toga generiran je još jedan grafički prikaz. Također je generiran koristeći „seaborn“ knjižnicu (eng. *Seaborn library*) i „pandas“ knjižnicu (eng. *Pandas library*) koja olakšava rad na podacima kao i računanje korelacija podataka pomoću metode „.corr“ (Slika 6).

Slika 5: Vizualizacija podataka koristeći "seaborn"

Slika 5 pokazuje međusobne odnose značajki i vrijednost značajke ishod („Outcome“). Ishod gdje osoba ima dijabetes (1) označen je narančastom bojom, dok je ishod gdje osoba nema dijabetes (0) označen plavom bojom. Moguće je utvrditi da niti u jednom odnosu podaci nisu u potpunosti

odvojeni, to jest, u nijednom odnosu nije lako povući zamišljenu liniju koja bi odvajala ishod što može stvarati probleme u treniranju modela.



Slika 6: Vizualizacija korelacija među značajkama

Slika 6 pokazuje korelacije između značajki. Iz navedenog prikaza može se iščitati da na skup oznaka (Y) ili na značajku „Outcome“ (ishod) najviše utječe vrijednost značajke „Glucose“ (glukoza), potom najviše utječe vrijednost „BMI“ značajke te zatim godine života. To jest, što je veća vrijednost glukoze, veći broj godina i što je veći „BMI“ ili indeks tjelesne mase osobe, veća je mogućnost da osoba ima dijabetes.

5.2.2 Standardizacija

S obzirom na to da skup podataka na kojemu je program treniran i testiran ne sadrži podatke koji su normalno i standardno raspodijeljeni te skalirani - neke značajke sadrže dvoznamenkaste vrijednosti dok druge sadrže jednoznamenkaste ili troznamenkaste vrijednosti, a neke čak i decimalne brojeve – podatke je potrebno skalirati (normalizirati).

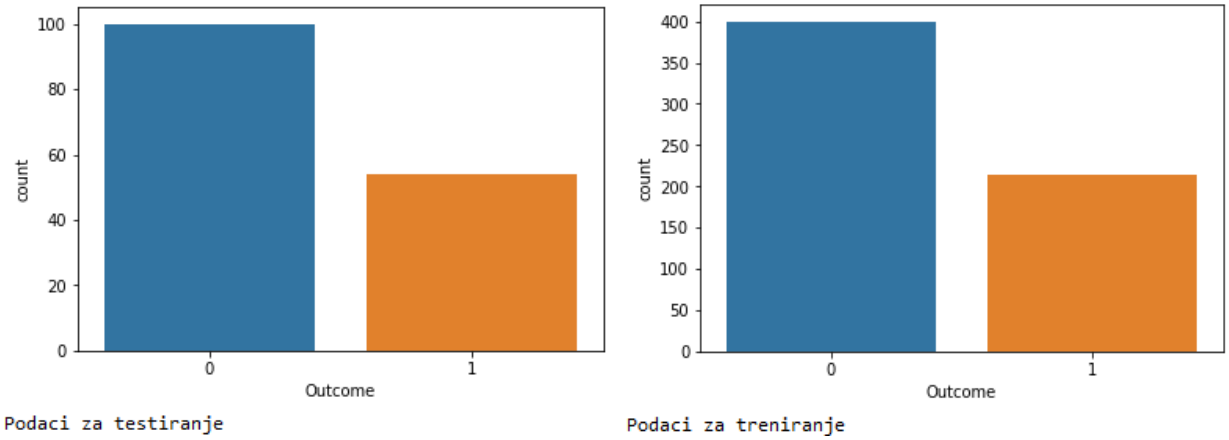
Neki algoritmi poput RBF „kernel“ stroja potpornih vektora ne mogu stvarati dobre modele na neskaliranim podacima jer pretpostavljaju da su sve značajke centrirane oko 0 te variraju na isti način (sickit-learn.org). Skaliranje podataka vršeno je korištenjem formule za podatke „x“ $z = \frac{x-u}{s}$ gdje je „u“ srednja vrijednost (eng. *mean*) podataka, a „s“ standardno odstupanje (eng. *standard deviation*) podataka.

Iz skupa podataka izdvojen je X, odnosno, podaci ulaznog prostora (prostor primjera). Drugim riječima, iz skupa podataka izdvojeni su svi podaci iz prvih osam stupaca, to jest, svi podaci osim onih koji predstavljaju rezultat (u ovom slučaju to su binarni podaci iz stupca „Outcome“). Potom su ulazni podaci (X) standardizirani koristeći „StandardScaler“ funkciju knjižnice (eng. *library*) „scikit-learn“ koja radi na principu navedene formule. Korištenjem „StandardScaler“ funkcije omogućeno je stvaranje i pohranjivanje modela skaliranja nastalog nad X. Pohranjivanjem modela skaliranja omogućeno je primjenjivanje tog istog modela i na drugim podacima što se pokazalo korisnim u nastavku rada.

5.2.3 Razdvajanje skupa podataka

Potom je skup podataka podijeljen na podatke koji će se koristiti za testiranje i podatke koji će se koristiti za treniranje u omjeru od 20 : 80 gdje je manji skup, skup za treniranje.

Skup podatka podijeljen je nasumično, ali je stratificiran po Y ili po dimenziji ishoda (Outcome), što znači da je omjer podataka označenih kao 1 i 0 u stupcu „Outcome“ jednak u dijelu podataka odvojenih za treniranje i u podacima za testiranje (Slika 7, Slika 8).



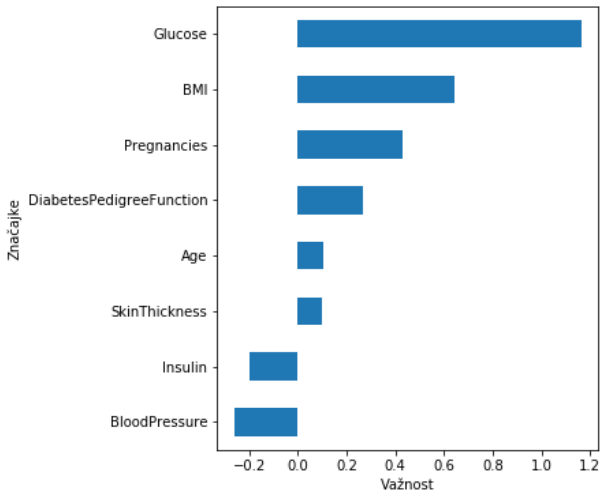
Slika 7: Odvojeni podaci za testiranje (lijevo) i Slika 8: Odvojeni podaci za treniranje (desno)

5.2.4 Treniranje modela

Nakon što su skupovi podijeljeni na skup za treniranje i skup za testiranje, oba se skupa dijele još jednom na X i Y, to jest, na podatke ulaznog prostora (X, prvih osam značajki) i izlazni rezultat (Y). Podaci ulaznog prostora (X) skupa za treniranje i skupa za testiranje skaliraju se po modelu napravljenom u prethodnom poglavlju čime se dobivaju jednako skalirani podaci u oba skupa. Nakon skaliranja stvaraju se modeli.

Prvi algoritam koji je korišten jest logistička regresija. Kao što je prije navedeno, logistička regresija je algoritam nadziranog učenja regresijskog tipa koji se često primjenjuje i na problemima klasifikacije (Geron, 2014). Logistička regresija je korištena u ovom projektu zbog toga što, osim same klasifikacije po Geronu (2014.), kao rezultat daje i vrijednost vjerojatnosti pripadanja određenoj klasi.

Koristeći podatke skupa za treniranje, model logističke regresije treniran je i pohranjen te je potom taj model primijenjen na skupu za testiranje (više o rezultatima testiranja u idućem poglavlju). Koristeći stvoreni model također je generiran i grafički prikaz koji ukazuje na koeficijent značajki u funkciji odluke (Slika 9). Iz generiranog prikaza moguće je utvrditi da je model logističke regresije za najvažniju značajku u procesu odlučivanja izabrao značajku „Glucose“ (razina glukoze) što se poklapa sa rezultatima iz poglavlja 5.3.1 i slike 6.



Slika 9: Značajke koje naviše utječu na odluku po logističkoj regresiji

U projektu je korišteno više algoritama za stvaranje više različitih modela kako bi se ti modeli mogli usporediti te kako bi se mogli primijeniti u kasnijem periodu razvoja programa.

Drugi algoritmi korišteni u projektu su algoritam K najbližih susjeda (eng. *K-nearest neighbors*) ili skraćeno KNN, te stroj potpornih vektora (eng. *support vector machine*) ili SVM u dva oblika: linearni i RBF. Svim navedenim algoritmima generirani su modeli koji su kasnije testirani na testnom skupu podataka.

Cjelokupni proces olakšan je korištenjem „scikit-learn“ knjižnice (eng. *library*). „Scikit-learn“ omogućuje da se algoritam primijeni i trenira zvanjem predefiniranih funkcija i metoda. Algoritam se pozove funkcijom (u slučaju logističke regresije funkcija glasi „LogisticRegression“) i trenira metodom „.fit“, te je potom moguće model pohraniti funkcijom „joblib.dump“ u „.pk1“ datoteku. Taj se proces može ponoviti za svaki algoritam.

Algoritam stroja potpornih vektora (linearni) korišten je zbog toga što se njegova uspješnost često može usporediti s uspješnosti algoritma logističke regresije jer su oba linearna modela (Drakos, 2018). RBF (eng. *radial basis function*) stroj potpornih vektora je pak izabran zbog svojeg nelinearnog načina klasifikacije isto kao i KNN (k najbližih susjeda) (Varghese 2018).

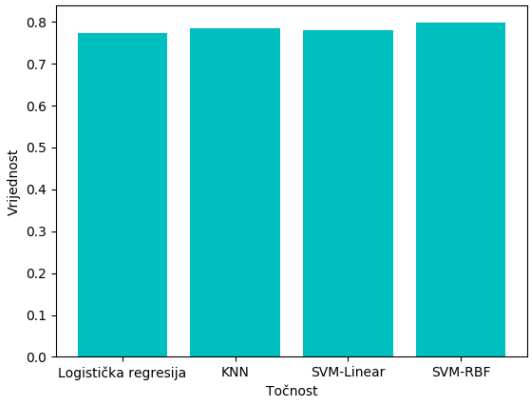
5.3 Testiranje i rezultati

Bez adekvatnog testiranja modela nije moguće znati koliko je model pouzdan i primjenjiv. U ovome projektu testiranje je provedeno nad svakim modelom čime se dobivaju usporedivi rezultati.

Prethodno odvojeni skup podataka za testiranje koristi se tako što model klasificira podatke ulaznog prostora (X) u skup oznaka (Y) te se potom uspoređuju njegove klasifikacije sa točnim odgovorima. Mjera koja se dobiva naziva se točnost (eng. *accuracy*) (Mishra, 2018).

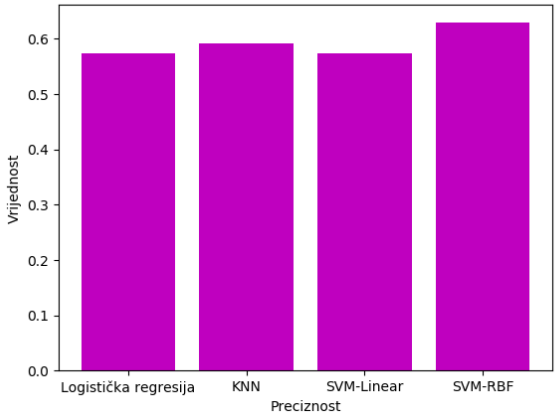
U procesu testiranja također se izračunaju i vrijednosti preciznosti (eng. *precision*) te odziva (eng. *recall*). Preciznost mjeri koliko je od predviđenih pozitiva stvarnih pozitiva, čime preciznost može dati uvid u to koliko je negativnih pozitiva - što je veća vrijednost preciznosti manje je negativnih pozitiva. Odziv pokazuje koliko pozitiva je model označio kao pozitiv, to jest, mjera odziva ukazuje na to koliko je lažnih negativa (Shung, 2018).

Svi algoritmi pokazali su točnost između 70% i 80% (Slika 10), što u ovome slučaju predstavlja zadovoljavajuću razinu točnosti (eng. *accuracy*).



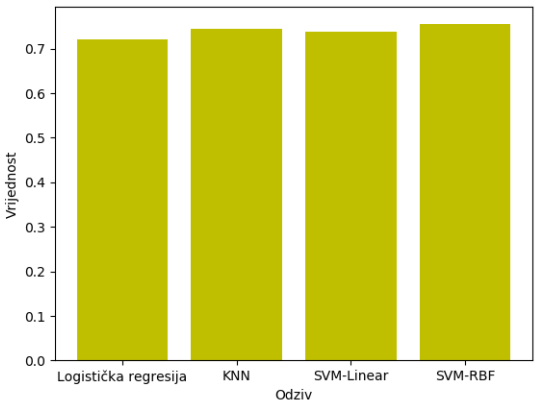
Slika 10: Vrijednosti točnosti

Svi algoritmi pokazali su poprilično lošu preciznost. Većina algoritama nema preciznost veću od 60% osim RBF stroja potpornih vektora, no niti jedan nema preciznost ispod 50%, to jest, u procesu klasifikacije velik je broj lažnih pozitiva (Slika 11).



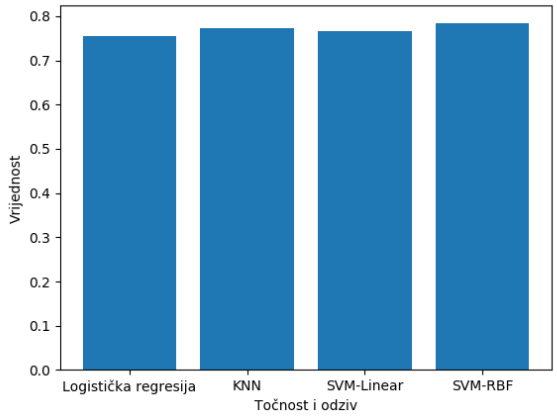
Slika 11: Vrijednosti preciznosti

Svi su algoritmi pokazali dobru razinu odziva. Niti jedan model nema odziv manji od 70% (Slika 12). Odziv se u ovom slučaju predstavlja kao bitna mjera zbog cilja programa. Uzevši u obzir to da je cilj programa ukazati na mogućnost dijabetesa, vrijednost odziva pokazuje se važnim jer odziv ukazuje na količinu lažnih negativa. U slučaju dijagnoza, lažni negativi predstavljaju potencijalno veću štetu nego lažni pozitivi.



Slika 12: Vrijednosti odziva

S obzirom na to predstavljam je još jednu mjeru (Slika 13) koja označava spoj točnosti i odziva. Prema formuli $Z = \frac{2*x+y}{3}$ dobiva se rezultat Z između 0 i 1. U ovome slučaju „x“ je vrijednost točnosti, dok je „y“ vrijednost odziva. Jasne granice poput 0 i 1 olakšavaju interpretaciju rezultata. Točnost i odziv odabrani su zbog toga što točnost kao mjera točno klasificiranih slučajeva predstavlja važnu mjeru, dok se odziv pokazuje važnim u ovom specifičnom slučaju zbog toga što ukazuje na razinu lažnih negativa. Ta mjera se, zbog svojih prednosti, kasnije koristi u izvođenju programa kao mjera koja označava „najbolji“ algoritam za ovu primjenu.



Slika 13: Vrijednosti točnosti i odziva

Rezultati mjere spoja odziva i točnosti su, na svim modelima, iznad 70% i ispod 80% (Slika 13). Najbolji algoritam prema mjeri spoja odziva i točnosti jest RBF stroj potpornih vektora sa 78% posto, a najmanje uspješan je algoritam logističke regresije sa 75%.

Iz ovog testiranja može se zaključiti da u ovom slučaju linearni algoritmi (logistička regresija i linearni stroj potpornih vektora) imaju slične ocjene (Slika 10, 11, 12 i 13), dok nelinearni algoritmi (RBF stroj potpornih vektora i k najbližih susjeda) imaju bolje rezultate (Slika 10, 11, 12 i 13). Prednost nelinearnih algoritama u ovom primjeru moguća je zbog prirode podataka, to jest, kao što je moguće vidjeti na „Slika 5“, podaci u ovome skupu nisu lako linearno odvojivi.

6. Program

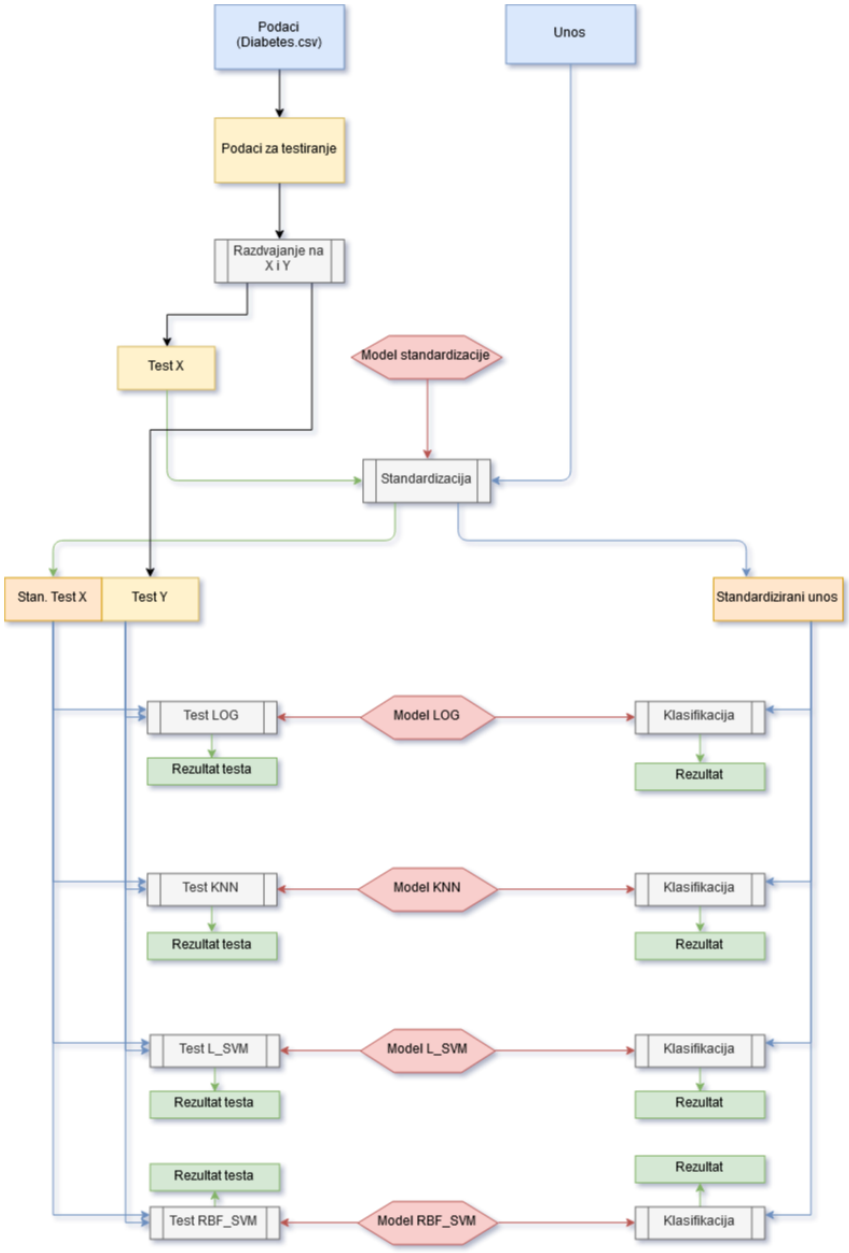
Navedeni postupci omogućili su rad na programu strojnog učenja kojim se predviđa vjerojatnost pojave dijagnoze dijabetesa. Program postoji u dvije inačice. Klasična verzija koja se izvodi u Python konzoli i u samom radu prolazi kroz proces testiranja što na ovako malom skupu podataka ne predstavlja problem, no u slučaju većih skupova podataka može predstaviti problem zbog trošenja računalnih resursa. Taj problem pokušava se riješiti drugom, jednostavnijom inačicom. Druga inačica također se može izvoditi i u konzoli, ali za razliku od prve verzije programa, ova verzija se oslanja samo na učitavanje već pohranjenih modela što ju čini bržom i „prjenosnijom“.

6.1 Prva verzija programa

Prva inačica programa (Slika 14) učitava skup podataka „diabetes.csv“ te modele standardizacije, logističke regresije, algoritma K najbližih susjeda, linearnog stroja potpornih vektora i RBF stroja potpornih vektora. Potom, korisnik unosi podatke za značajke ulaznog prostora (broj trudnoća, razina glukoze, krvni tlak, debljina kože, inzulin, BMI, DPF i broj godina). U međuvremenu, program skup podataka iz datoteke „diabetes.csv“ razdvaja na X i Y te potom, koristeći učitan model standardizacije, skalira X. Zatim se testira točnost, preciznost i odziv svih modela.

Podaci koje korisnik unese također prolaze i proces skaliranja koristeći učitan model te se provodi klasifikacija prema učitanim modelima algoritama. Korisniku su potom prezentirani rezultati modela logističke regresije u obliku odgovora „DA“ ili „NE“ (ima li korisnik dijabetes ili ne) te vjerojatnost pripadanja odgovora određenoj klasi po logističkoj regresiji.

Također, korisniku je ponuđen i odgovor od „najboljeg“ algoritma. Najbolji algoritam određen je uzimajući u obzir njegovu točnost i njegov odziv prema formuli $Z = \frac{2*x+y}{3}$. Uz to, korisnik može vidjeti i više podataka o odluci upisivanjem pojma „više“ u predviđeno polje. Upisivanjem pojma „više“ korisniku se ispisuju i odgovori ostalih algoritama zajedno s njihovim rezultatima točnosti, preciznosti, odziva te spoja točnosti i odziva. Iz programa se izlazi pritiskom bilo koje tipke na tipkovnici.

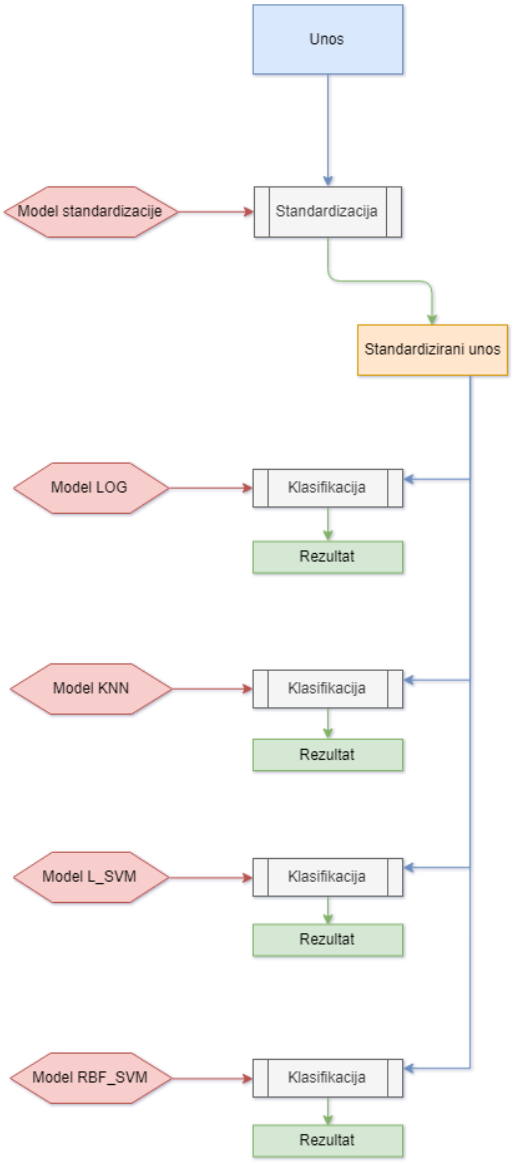


Slika 14: Tijek izvođenja prve verzije programa

6.2 Druga verzija programa

Druga inačica programa (Slika 15) zamišljena je kao jednostavnija verzija programa, „lakša“ na računalnim resursima. U drugoj inačici programa učitavaju se samo modeli skaliranja i modeli algoritama. Potom korisnik unosi svoje podatke koji se skaliraju prema modelu i kasnije klasificiraju prema učitanim modelima. U ovoj inačici korisniku su prezentirani rezultati svih

algoritama u obliku jednostavnog „DA“ ili „NE“, osim logističke regresije gdje je prikazana i vjerojatnost. S obzirom na to da u ovoj inačici program ne prolazi kroz fazu testiranja, nije moguće dobiti rezultate uspješnosti algoritama, stoga su svi rezultati jednostavno ispisani korisniku. Ova verzija se izvodi brže i koristi manje knjižnica (eng. *library*) te stoga čini gotovo prijenosnu verziju, no korisnik je i dalje dužan imati instaliran Python sa „scikit-learn“ knjižnicom na svom računalu.



Slika 15: Tijek izvođenja druge verzije programa

6.3 Moguća poboljšanja

Najveći nedostatak u ovome su projektu sami podaci koji su korišteni u procesu treniranja i testiranja. S obzirom na to da je broj „pojava“ vrlo malen, program nema zavidnu razinu „uspješnosti“ te zbog toga nije dovoljno pouzdan za stvarnu primjenu u polju medicine kao alat za dijagnostiku. Širim skupom podataka, to jest, skupom podataka sa više značajki ili, možda, sa više relevantnih značajki i većim brojem pojava postigli bi se bolji rezultati time što bi modeli bili robusniji. Također, sam program ima svoje nedostatke poput toga što niti jedna verzija programa nije u potpunosti prijenosna, odnosno, potrebno je da korisnik ima instaliran širok raspon knjižnica (eng. *library*) te da ima instaliran Python na svom računalu. Drugi navedeni problem bilo bi moguće riješiti korištenjem nekog drugog programskog jezika, stvaranjem izvršne datoteke (eng. *executable file*) ili korištenjem optimizirajućeg statičnog kompajlera (eng. *optimising static compiler*) i Cython programskog jezika. Cython predstavlja validnu opciju zbog toga što omogućuje pisanje Python koda u kombinaciji sa C i C++ programskim jezicima čime se izvođenje programa može ubrzati te je moguće program učinit prijenosnim stvaranjem izvršnih datoteka poput „.exe“ (Cython.org).

7. Zaključak

Rezultati testiranja pokazali su da, u ovom projektu, linearni algoritmi (logistička regresija i linearni stroj potpornih vektora) imaju slične ocjene, dok algoritmi nelinearnog pristupa klasifikaciji (k najbliži susjedi i RBF stroj potpornih vektora) imaju bolje rezultate. Ta prednost nelinearnih algoritama moguća je zbog prirode korištenih podataka, to jest, podaci u korištenom skupu nisu lako linearno odvojivi.

S obzirom na nedostatke skupa podataka, rezultati su zadovoljavajući, ali program u svojem trenutnom obliku nije primjenjiv na problemima stvarnoga svijeta kao alat za dijagnozu dijabetesa.

Strojno učenje je danas važan dio znanstvene djelatnosti jer predstavlja način za korištenje ogromnih količina podataka koji su generirani iz mnoštva digitalnih uređaja temeljem čega se stvaraju nove informacije. Pristupačnost Pythona i dostupnost raznih paketa specijaliziranih za podatkovnu znanost omogućuje korištenje metoda strojnog učenja u gotovo svim znanostima, a pokazuje se i kao koristan način djelovanja izvan informacijskih i računalnih znanstvenih krugova. Strojno učenje se danas više ne koristi samo za predviđanje ili klasificiranje, nego se može koristiti i za stvaranje novog sadržaja putem, primjerice, GAN algoritama. No, primjena strojnog učenja mogla bi se proširiti sa razvojem ideja poput koncepta otvorene znanosti koja bi omogućila još veći pristup podacima i time otvorila prostor za nova istraživanja bazirana na strojnom učenju.

8. Literatura

1. Alpaydin, E. (2004). Introduction to Machine Learning. Cambridge: The MIT Press.
2. Anaconda, Inc. (2019). Anaconda Python/R Distrubution. Anaconda.com. Preuzeto sa https://www.anaconda.com/distribution/ [13. rujna 2019.]
3. Columbus L. (2017 Prosinac 11). LinkedIn's Fastest-Growing Jobs Today Are In Data Science And Machine Learning [blog post]. Preuzeto sa https://www.forbes.com/sites/louiscolumbus/2017/12/11/linkedins-fastest-growing-jobs-today-are-in-data-science-machine-learning/#5a0d01af51bd [1. rujna 2019.]
4. Cosgriff, C.V., Celi, L.A., Ko, S., Sundaresan, T., Armengol de la Hoz, M.A., Kaufman, A.R., ... Deliberato, R.O. (2019). Developing well-calibrated illness severity scores for decision support in the critically ill. npj Digital Medicine, 2(1). doi:10.1038/s41746-019-0153-6
5. Cython.org (2019.) Cython: C-extensions for Python. Cython.org. Preuzeto sa https://cython.org/ [15. rujna 2019.]
6. Drakos, G. (2018, Kolovoz 12). Support Vector Machine vs Logistic Regression [blog post]. Preuzeto sa https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f [13. rujna 2019.]
7. Ekins, S., Puhl, A.C., Zorn, K.M., Lane, T.R., Russo, D.P., Klein, J.J., ... Clark A.M. (2019). Exploiting machine learning for end-to-end drug discovery and development. Nature Materials, 18(5), 435-441. doi:10.1038/s41563-019-0338-z
8. Elliott, T. (2018, Veljača 8). Open source project trends for 2018 [blog post]. Preuzeto sa https://github.blog/2018-02-08-open-source-project-trends-for-2018/ [2. rujna 2019.]
9. Eraslan, G., Avsec, Ž., Gagneur, J. & Theis, F.J. (2019). Deep learning: new computational modelling techniques for genomics. Nature Reviews Genetics, 20(7), 389-403. doi:10.1038/s41576-019-0122-6
10. Fecher, S. & Friesike, S. (2014). Open Science: One Term, Five Schools of Thought. U: Bartling, S., Frieske, S. (ur.). Opening Science The Evolving Guide on How the Internet is Changing Research, Collaboration and Scholarly Publishing. Chem: Springer. 17-49. doi:10.1007/978-3-319-00026-8

11. Geron, A. (2017). Hands-on machine learning with with Sickit-Learn and TensorFlow. Sebastopol: O'Reilly Media
12. Harrison, O. (2018, Rujan 10). Machine Learning Basics with the K-Nearest Neighbors Algorithm [blog post]. Preuzeto sa https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 [27. kolovoza 2019.]
13. Indeed Editoral Team (2019, Ožujak 14). The Best Jobs in the U.S.: 2019 [blog post]. Preuzeto sa http://blog.indeed.com/2019/03/14/best-jobs-2019/ [29. kolovoza 2019.]
14. Kaggle.com (2019). Pima Indians Diabetes Database. Preuzeto sa https://www.kaggle.com/uciml/pima-indians-diabetes-database [2. srpanja 2019.]
15. Krstić, Ž., Seljan, S. & Zoroja, J. (2019). Visualization of Big Data Text Analytics in Financial Industry: A Case Study of Topic Extraction for Italian Banks. Proceedings of ENTRENOVA.
16. Lardinois, F., Lynley, M. & Mannes, J. (2017, Ožujak 8). Google is acquiring data science community Kaggle [blog post]. Preuzeto sa https://techcrunch.com/2017/03/07/google-is-acquiring-data-science-community-%20kaggle/ [12. rujna 2019.]
17. LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep Learning. Nature, 521. doi:10.1038/nature14539
18. Leksikografski zavod Miroslav Krleža. (2019). Podatak. Enciklopedija.hr. Preuzeto sa http://www.enciklopedija.hr/natuknica.aspx?id=48887 [3. rujna 2019.]
19. Leksikografski zavod Miroslav Krleža. (2019). Statistika. Enciklopedija.hr. Preuzeto sa http://www.enciklopedija.hr/Natuknica.aspx?ID=57896 [3. rujna 2019.]
20. Leksikografski zavod Miroslav Krleža. (2019). Umjetna inteligencija. Enciklopedija.hr. Preuzeto sa http://www.enciklopedija.hr/Natuknica.aspx?ID=63150 [3. rujna 2019.]
21. Libbrecht, M.W. & Stafford, W.N. (2015). Machine learining applications in genetics and genomics. Nature Reviews Genetics, 16. doi:10.1038/nrg3920
22. Makino, M., Yoshimoto, R., Ono, M., Itoko, T., Katsuki, T., Koseki, A., ... Suzuki, A. (2019). Artificial intelligence predicts the progression of diabetic kidney disease using big data machine learning. Scientific Reports, 9(1). doi:10.1038/s41598-019-48263-5
23. Marrara, S., Pejić, M.B., Seljan, S. & Topalovic, A. (2019). FinTech and SMEs: The Italian Case U: Rafay, A. (ur.). FinTech as a Disruptive Technology for Financial

Institutions. Hershey, Pennsylvania: IGI Global. 42-60. doi:10.4018/978-1-5225-7805-5.ch002

24. Mishra, A. (2018, Veljača 24). Metrics to Evaluate your Machine Learning Algorithm [blog post]. Preuzeto sa https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 [10. rujna 2019.]
25. Nicholson, C. (2019). A Beginner's Guide to Generative Adversarial Networks (GANs). skymind.ai. Preuteto sa https://skymind.ai/wiki/generative-adversarial-network-gan
26. GauGAN (2019.) nvidia-research-mingyuliu.com. Preuzeto sa http://nvidia-research-mingyuliu.com/gaugan [14. rujna 2019.]
27. Otter, D.W., Medina, J.R. & Kalita, J.K. (2018). A Survey of the Usages of Deep Learning in Natural Language Processing. Preuzeto sa https://arxiv.org/abs/1807.10854v1 [10. srpnja 2019.]
28. Pejić, M.B, Krstić, Ž. & Seljan, S. (2019). Big data text mining in the financialsector. U: Metawa, N., Elhoseny, M., Hassanien, A. & Hassan, M. (ur.). Expert Systems inFinance: Smart Financial Applications in Big Data Environments. London, Routledge. 80-96. doi:10.4324/9780429024061
29. Pejić, M.B, Krstić, Ž., Seljan, S. & Turulja, L. (2019). Text Mining for Big Data Analysis in Financial Sector: A Literature Review. doi:10.3390/su11051277
30. Python Software Foundation (2019). What is Python? Executive Summary. Python.org Preuzeto sa https://www.python.org/doc/essays/blurb/ [15. rujna 2019.]
31. Russ, T.C., Woelbert, E., Davis, K.A.S., Hafferty, J.D, Ibrahim, Z., Inkster, B., ... Steward, R. (2019). How data science can advance mental health research. Nature human behaviour, 3(1), 24-32. doi:10.1038/s41562-018-0470-9
32. Sammut, C. & Webb, G.I. (ur.) (2017). Encylopedia of Machine Learning and Data Mining (2nd ed.). New York: Springer Nature.
33. Sejnowski, T.J. (2018) The Deep Learning Revolution. Cambridge: The MIT Press.
34. Shung, K.P. (2018, Ožujak 15). Accuracy, Precision, Recall or F1? [blog post]. Preuzeto sa https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 [9. rujna 2019.]
35. Sickit-learn developers (2019). sklearn.preprocessing.StandardScaler - scikit-learn 0.21.3 documentation. scikit-learn.org. Preuzeto sa https://scikit-

learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html [4. rujna 2019.]

36. Tong, S. & Koller, D. (2001) Support Vector Machine Active Learning with Applications to Text Classification. Journal of Machine Learning Research. 2, 45-66. Preuzeto sa http://www.jmlr.org/papers/volume2/tong01a/tong01a.pdf [4. rujna 2019.]
37. Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., ... & Jain, A. (2019). Unsupervised word embeddings capture latent knowledge from materials science literature. Nature, 571, 95-98. doi:10.1038/s41586-019-1335-8
38. Vanschoren, J., Braun, M.L. & Ong, C.S. (2014). Open science in machine learning. Preuzeto sa https://arxiv.org/abs/1402.6013 [14. rujna 2019.]
39. Varghese, D. (2018, Prosinac 6). Comparative Study on Classic Machine learning Algorithms [blog post]. Preuzeto sa https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222 [2. rujna 2019.]
40. Vogel, C., Zwolinsky, S., Griffiths, C., Hobbs, M., Henderson, E. & Wilkinsm E. (2019). A Delphi study to build consensus on the definition and use of big data in obesity research. International Journal of Obesity. doi:10.1038/s41366-018-0313-9
41. Vondrick, C., Pirsiavash, H., Torralba, A. (2016). Generating Videos with Scene Dynamics. Preuzeto sa https://arxiv.org/abs/1609.02612 [13. rujna 2019.]
42. Weihs, C. & Ickstadt, K., (2018). Data Science: the impact of statistics. Internatioanal Journal of Data Science and Analytics, 6(3), 189-194. doi:10.1007/s41060-018-0102-5
43. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X. & Metaxas, D. (2016). StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks. Preuzeto sa https://arxiv.org/abs/1612.03242 [13. rujna 2019.]

7.1. Popis slika

Strojno učenja kao alat za zaključivanje

Sažetak

Strojno učenje postaje sve češći alat u većini znanstvenih disciplina. Ovaj tekst pokušava objasniti strojno učenje i njegove osnovne koncepte kroz klasifikaciju sustava strojnog učenja i kroz objašnjenja najkorištenijih algoritama. Također prikazuje kako stvoriti sustav strojnog učenja za klasifikaciju korištenjem klasifikatora K najbližih susjeda, logističke regresije te linearnog i RBF stroja potpornih vektora. Također, objašnjeno je kako interpretirati rezultate najkorištenijih metoda za evaluaciju sustava strojnog učenja.

Ključne riječi: strojno učenje, klasifikacija, nadzirano učenje, nenadzirano učenje, polunadzirano učenje, učenje uz podršku, duboko učenje, dijabetes

Machine learning as a tool for inference

Summary

Machine learning is becoming a more prevelant tool in most science fields. This text tries to explain machine learning and its basic concepts through classification of machine learning systems and through explanations of most used algorithms. It is also presents how to create a machine learning system for classification using K- nearest neighbors classifier, logistic regression, and linear and RBF kernel support vector machines. Also, it is shown how to interpret results of most used methods for evaluating machine learning systems.

Key words: machine learning, classification, supervised learning, unsupervised learning, semisupervised learning, reinforcement learning, deep learning, diabetes