

Mogućnosti primjene metoda strojnog učenja u području telekomunikacija

Aleksić, David

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:003361>

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

Download date / Datum preuzimanja: **2024-12-19**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences - Institutional Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

David Aleksić

MOGUĆNOSTI PRIMJENE METODA STROJNOG UČENJA
U PODRUČJU TELEKOMUNIKACIJA

ZAVRŠNI RAD

Zagreb, 2021.

Zagreb, 11. svibnja 2021.

Zavod: **Zavod za informacijsko komunikacijski promet**
Predmet: **Informacije i komunikacije**

ZAVRŠNI ZADATAK br. 6177

Pristupnik: **David Aleksić (0135253806)**
Studij: **Promet**
Smjer: **Informacijsko-komunikacijski promet**

Zadatak: **Mogućnosti primjene metoda strojnog učenja u području telekomunikacija**

Opis zadatka:

U okviru završnog rada potrebno je definirati i detaljno obrazložiti koncepte umjetne inteligencije i strojnog učenja te analizirati dostupne alate i platforme za primjenu algoritama strojnoga učenja. Konačno, potrebno je pružiti osvrt i analizirati mogućnosti primjene strojnoga učenja u rješavanju raznovrsnih problema u domeni telekomunikacija.

Mentor:

Predsjednik povjerenstva za
završni ispit:

dr. sc. Ivan Cvitić

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

ZAVRŠNI RAD

MOGUĆNOSTI PRIMJENE METODA STROJNOG UČENJA U
PODRUČJU TELEKOMUNIKACIJA
OPPORTUNITIES OF USING MACHINE LEARNING METHODS IN
TELECOMMUNICATIONS

Mentor: Dr. sc. Ivan Cvitić

Student: David Aleksić

JMBAG: 0135253806

Zagreb, kolovoz 2021.

MOGUĆNOSTI PRIMJENE METODA STROJNOG UČENJA U PODRUČJU TELEKOMUNIKACIJA

SAŽETAK

Ovaj završni rad prikazuje metode strojnog učenja i mogućnosti primjene strojnog učenja u području telekomunikacija. Uz primjere primjene strojnog učenja također je definiran pojam umjetne inteligencije koja je danas u sve češćoj upotrebi u svijetu. Strojno učenje kao podsustav umjetne inteligencije ima veliki broj vlastitih algoritama koji se koriste za različite primjere. Sa sve većim razvojem tehnologija kao što su veće brzine interneta, 5G mobilne mreže itd. iz npr. telekomunikacijskog područja dolazi do potrebe za boljim upravljanjem i podrškom za sustave koji se koriste novim tehnologijama. Metode strojnog učenja mogu se koristiti za poboljšanje korisničke usluge prikupljanjem podataka, za prikupljanje podataka o prometu u mreži i općenito u svim sustavima u kojima se zahtjeva prikupljanje, grupiranje i analiza podataka.

KLJUČNE RIJEČI: umjetna inteligencija, strojno učenje, nadzirano strojno učenje, nenadzirano strojno učenje, algoritmi strojnog učenja

SUMMARY

This bachelor's thesis shows methods of machine learning and opportunities of using it in telecommunications. With the examples of using machine learning methods, the term of artificial intelligence, which is today used more than ever before, is defined in this thesis as well. Machine learning as a subset of artificial intelligence has a lot of its own algorithms and methods. With the improvement of the new technologies such as higher internet speeds, 5G mobile networks etc. for e.g., in telecommunication, comes the need for the new and improved management and support for the systems that are using those new technologies. Some types of machine learning can be used for collecting data to improve user's quality of service, some are used for collecting data of traffic in networks or generally for any system that has the need to collect data, cluster data points and analysing data.

KEY WORDS: Artificial Intelligence, Machine Learning, Supervised Machine Learning, Unsupervised Machine Learning, Machine Learning algorithms

Sadržaj

1. Uvod.....	1
2. Umjetna inteligencija	3
2.1. Povijest umjetne inteligencije	3
2.2. Podsustavi umjetne inteligencije.....	3
2.3. Umjetna inteligencija u čovjekovoj svakodnevnici	5
3. Strojno učenje	7
3.1. Proces strojnog učenja.....	7
3.2. Vrste i modeli strojnog učenja.....	10
3.2.1. Nadzirano strojno učenje	10
3.2.2. Nenadzirano strojno učenje.....	12
3.2.3. Ojačano strojno učenje.....	18
4. Alati i platforme za primjenu algoritama strojnog učenja	20
4.1. Scikit Learn	20
4.2. Platforma za strojno učenje TensorFlow.....	23
4.3. PyTorch	26
4.4. Weka.....	27
4.5. RapidMiner.....	28
5. Primjena strojnog učenja u telekomunikacijama	29
5.1. Uloga strojnog učenja u sigurnosti mrežnog prometa.....	29
5.2. Strojno učenje u području klasifikacije IoT uređaja	32
5.3. Ostale mogućnosti primjene strojnog učenja u telekomunikacijama	33
6. Zaključak.....	35
Literatura.....	36
Popis kratica.....	41
Popis slika	42

1. Uvod

Zadatak i cilj ovog završnog rada je istraživanje i analiziranje mogućnosti primjene strojnog učenja u području telekomunikacija. Preduvjet za ostvarivanje zadanog cilja ovog završnog rada jest poznavanje definicije strojnog učenja i principe rada te prednosti i nedostatke njegove primjene.

Strojno učenje (engl. *Machine Learning - ML*) je podsustav umjetne inteligencije (engl. *Artificial Intelligence - AI*) koja se koristi u računalnim znanostima. Strojno učenje kao podsustav umjetne inteligencije fokusirano je na izgradnju programa i aplikacija koji će učiti iz dostupnih podataka i koji će poboljšavati svoju preciznost s vremenom bez da budu programirani da to čine.

Završni rad se sastoji od šest poglavlja:

1. Uvod
2. Umjetna inteligencija
3. Strojno učenje
4. Alati i platforme za primjenu algoritama strojnog učenja
5. Primjena strojnog učenja u telekomunikacijama
6. Zaključak

U drugom poglavlju analizira se pojam umjetne inteligencije. Tu je opisano što je to umjetna inteligencija, gdje se koristi. Ukratko je napravljena podjela inteligencije i umjetne inteligencije na podsustave koji spadaju pod umjetnu inteligenciju. Uz samu podjelu u ovom poglavlju su također neki od tih podsustava ukratko opisani.

Treće poglavlje sastoji se od više manjih podnaslova. U jednom od tih pojašnjen je princip rada strojnog učenja. Način na koji se strojevi uče i svi ostali koraci koji se moraju proći prije nego li stroj bude spreman za korištenje. U ovom poglavlju se nalazi i podjela strojnog učenja na: nadzirano strojno učenje, nenadzirano strojno učenje i ojačano strojno učenje. To su sve vrste strojnog učenja koje su definirane u tom poglavlju i opisane po njihovom principu rada i njihovim značajkama te razlikama jedne od drugih.

Četvrto poglavlje odnosi se na platforme i alate koji se koriste prilikom procesa strojnog učenja. U ovom poglavlju napravljena je podjela na alate koji se koriste. Napravljena je usporedba između

tih alata prema osnovnim karakteristikama te su prikazana dva primjera za proces *K-means* klasteriziranja u dvije od tih platformi.

Peto poglavlje obuhvaća podjelu primjene strojnog učenja i umjetne inteligencije u području telekomunikacija. Opisana je uloga strojnog učenja u zaštiti i sigurnosti mreže od kibernetičkih napada. Uz ulogu u sigurnosti mreže definiran je i način primjene strojnog učenja u klasifikaciji uređaja interneta stvari (engl. *Internet of Things - IoT*). Te su navedeni još neki od primjera primjene strojnog učenja uz slikoviti prikaz tijeka rada strojnog učenja u području interneta i telekomunikacija.

2. Umjetna inteligencija

Umjetna inteligencija odnosi se na područja računalnih znanosti koje proučavaju i bave se promatranjem i razvijanjem inteligentnih sustava, alata i strojeva koje reagiraju i uče kao ljudi [1].

2.1. Povijest umjetne inteligencije

Umjetna inteligencija je dosta mlada disciplina računalnih znanosti koja postoji šezdesetak do sedamdesetak godina. Prvi AI program koji je napravljen bio je „Logic Theorist“ kojega su 1955. godine napisali Herbert Simon, Allen Newell i John Shaw. Program je napravljen kako bi dokazao određene teoreme iz djela „Principia Mathematica“ [2].

Pojam umjetna inteligencija prvi puta se spominje 1956. godine kada se jedna mala skupina znanstvenika sastala kako bi održali projekt istraživanja o novoj znanosti koju su nazvali umjetna inteligencija. Te je tako do danas ostao taj naziv [3].

Narednih godina AI je imala uspona i padova u svojim daljnjim istraživanjima. Pa su tako bila vremenska razdoblja od nekoliko godina gdje su znanstvenici željeli dalje istraživati principe rada i problematiku umjetne inteligencije, ali su im vlasti njihovih država ograničili novčana sredstva koja su im bila potrebna za daljnjim napredovanjem u istraživanjima. Takva razdoblja događala su se dva puta. Prvi puta od 1974. do 1980. godine i drugi puta od 1987. do 1993. godine. To vrijeme u kojemu je došlo do tih ograničavanja resursa poznato je pod nazivom „AI winter“ odnosno „Zima umjetne inteligencije“ [2].

Nakon toga umjetna inteligencija doživljava svoje vrhunce pa je tako američka kompanija IBM napravila svoje računalo pod nazivom „Deep Blue“ koje 1997. godine je pomoću svoje umjetne inteligencije pobijedilo tadašnjeg svjetskog prvaka u šahu, Garya Kasparova [2].

U današnje vrijeme tj. 2000.-tih godina velike svjetske kompanije poput Googlea, Amazona, IBM-a i Facebooka ulažu ogromne novce kako bi proizveli proizvode koji bi na temelju svoje AI olakšali kupcima tih proizvoda određene poslove.

2.2. Podsustavi umjetne inteligencije

Prije same podjele na podsustave umjetne inteligencije važno je napraviti podjelu u inteligencije po njenim elementima. Tako se inteligencija sastoji od [4]:

1. Razumijevanja - element inteligencije koji omogućuje određene kriterije koji pomažu u donošenju odluka ili predviđanju nekih događaja.
2. Učenja - proces u kojemu se stječe znanje i određena vještina.
3. Rješavanja problema – proces u kojemu se mora najprije pronaći uzrok koji je stvorio nastali problem, a zatim saznavanje više od jednog načina za rješavanje tog problema.
4. Percepcije – predstavlja proces stvaranja predrasude odnosno predodžbe o nekoj pojavi ili događaju u okolini.
5. Verbalne ili lingvističke inteligencije – označava sposobnost razumijevanja, čitanja i pisanja određenih pojmova tj. informacija na nekom jeziku.

Nakon podjele inteligencije na osnovne elemente, umjetnu inteligenciju dijelimo na: strojno učenje, duboko učenje (engl. *Deep Learning*), obrada prirodnog jezika (engl. *Natural language processing - NLP*), prepoznavanje govora (engl. *Speech recognition*), robotika (engl. *Robotics*), planiranje (engl. *Planning*), strojnu viziju (engl. *Machine vision*), ekspertne sustave (engl. *Expert system*) te na umjetnu neuronsku mrežu (engl. *Artificial neural network - ANN*) [4].

Duboko učenje u načelu koristi se i u procesu strojnog učenja, ali je zapravo podsustav umjetne inteligencije isto kao i strojno učenje. Za duboko učenje može se reći kako je to proces koji je dosta sličan procesu strojnog učenja jer oba procesa imaju za cilj pomoću svojih algoritama prikupljati podatke kako bi stroj mogao razmišljati i odrađivati zadane zadatke što sličnije čovjeku i čovjekovom mozgu [5].

NLP je područje znanosti umjetne inteligencije koje se bavi istraživanjem i izradom aplikacijskih sustava koji će pomoći u razumijevanju i pronalaženju odgovora na pitanje „Kako računala mogu biti korištena za razumijevanje i manipuliranje korištenjem prirodnog jezika teksta ili govora za izvršavanje korisnih radova ? “ [6].

Robotika je područje umjetne inteligencije koje je zapravo najzanimljivije ljudima jer se ona zapravo još od davnih dana prikazuje na svim izvorima medija kao što su npr. roboti u filmovima kao jedan od najpoznatijih primjera. Robotika ima za cilj izraditi potpuno autonomne uređaje koji će pomagati ljudima u svakodnevnim poslovima i činiti njihov život lakši. Pa se tako danas produkti robotike koriste u svemiru, podmorju, logistici i proizvodnji, misijama potraga i spašavanja, u prometu u konceptima autonomnih vozila i sve češće u vojnim i ratnim misijama.

Ekspertni sustavi su podsustav AI koji se bave razvijanjem kompjutorskih programa napravljenih i dizajniranih za rješavanje kompleksnih problema i donošenja odluka. Te odluke se donose na temelju znanja koje sustavi prikupljaju tijekom istraživanja iz svojih baza znanja korištenjem razumijevanja kao elementa inteligencije [7].

Umjetna neuronska mreža je također podsustav u umjetnoj inteligenciji koji se dosta često koristi prilikom procesa strojnog učenja. Umjetna neuronska mreža je dio računalnog sustava koji je osmišljen kako bi se simulirao rad ljudskog mozga, zapravo način na koji mozak obrađuje podatke. Kao i kod biološke neuronske mreže, umjetna neuronska mreža se također sastoji od neurona koje imaju zadaću primanja i slanja signala međusobno. [8]

2.3. Umjetna inteligencija u čovjekovoj svakodnevnic

Iako pojam umjetna inteligencija djeluje kao komplicirana znanosti i kako se koristi samo u znanstvenim istraživanjima i skupim sustavima, umjetna inteligencija se koristi u svakodnevnom čovjekovom okruženju već duži niz godina. Pod svakodnevno čovjekovo okruženje podrazumijeva se svakodnevno čovjekovo korištenje mobilnog uređaja, računala, njegov odlazak na posao bilo to javnim gradskim prijevozom ili osobnim vozilom itd. Jedan od primjera umjetne inteligencije u svakodnevnom životu vidljiv je u mobilnim uređajima. Oni već dugi niz koriste programe i algoritme koji su dio ili su produkt umjetne inteligencije. Jedan od primjera je otključavanje mobilnog terminalnog uređaja uz prepoznavanje lica vlasnika uređaja. Primjer primjene umjetne inteligencije u mobilnim uređajima vidljiv je kod Apple iPhone mobilnih uređaja. Oni rade na principu da u trenutku podizanja uređaja, uređaj šalje preko 30 000 infracrvenih zraka na korisnikovo lice, koje analiziraju crte lica kako bi se mobitel otključao [9]. Taj proces otključavanja je moguć zbog svih podsustava umjetne inteligencije koji odrađuju svoj posao prilikom tog procesa otključavanja. Još jedan primjer iz čovjekove svakodnevnice korištenja umjetne inteligencije bez njegovog znanja je npr. prilikom putovanja na posao ukoliko korisnik koristi aplikaciju „Google Maps“ za navigaciju ona mu u svakom trenutku izbacuje prijedlog određnog mjesta na osnovu njegovih posljednjih pretraživanja ili mu aplikacija sama može odrediti na kojoj lokaciji se nalazi njegov posao ili kuća na temelju provedenog vremena na toj lokaciji bez da korisnik sam odabire na kojoj lokaciji se npr. nalazi njegova kuća.

Neki od ostalih primjera AI u svakodnevnom čovjekovom okruženju su tzv. virtualni asistenti na pametnim terminalnim uređajima (Siri, Bixby, Cortana itd.). Pomalo možda i zastrašujuća

činjenica je ta da današnji uređaji slušaju i preporučuju vlasniku uređaja određene stavke prilikom pretraživanja informacija na temelju toga što pamte posljednja pretraživanja. Tako za primjer možemo uzeti danas dvije najpopularnije platforme za video na zahtjev (engl. *Video On Demand-VOD*) „Netflix“ i „YouTube“. Dvije konkurentske kompanije koje rade na istom principu tako da algoritam prati kakve, u ovom slučaju videozapise, pretplaćeni korisnik pretražuje i ocjenjuje pozitivnim ocjenama. Da bi onda tom istom korisniku na osnovu tih ocjena preporučili koji sljedeći videozapis bi ga mogao zanimati. Također „Netflix“ svojim korisnicima daje mogućnost da prilikom pregleda filma prije gledanja daje informaciju o tome kolika je vjerojatnost da će se korisniku taj film svidjeti [9].

Na slici 1. nalazi se prikaz sa VOD platforme „Netflix“ koja pokazuje vjerojatnost koja je izračunata putem algoritama i podsustava unutar umjetne inteligencije koja prati i računa korisnikova pretraživanja i ocjene.



Slika 1. Prikaz izračunate vrijednosti sviđanja videozapisa nekom korisniku na platformi "Netflix"

Izvor: [10]

3. Strojno učenje

Strojno učenje je postupak u kojemu se na temelju određene aplikacije računalo pokušava naučiti da razmišlja na način kako razmišlja čovjek i da izvrši zadane zadaće na temelju naučenih podataka [11].

Računala su u povijesti radila na principu da su morala biti programirana da odrade neki zadatak i to programiranje bi morao odraditi sam programer prilikom svakog zasebnog slučaja, dok je kod strojnog učenja cilj računalo naučiti da ono već zna kako postupiti prilikom nekog događaja koji je nastao bez da mora čekati osobu koja je zadužena za njegovu kontrolu odnosno programera [12]. Jedan najobičniji primjer strojnog učenja bio bi da naučimo računalo razliku između automobila i motocikla. Taj postupak izvršio bi se na principu da se tom računalu daju fotografije automobila i da se na fotografijama označi točno što od toga na slici pripada automobilu. Tada računalo pomoću procesa i algoritama strojnog učenja uči i pamti određene uzorke koje svaka od tih slika ima za automobil. Nakon vremenskog perioda učenja i prikupljanja podataka računalo bi trebalo znati po kojim točno značajkama može prepoznati automobil i kada bi mu se pokazale slike automobila i motocikla zajedno to računalo bi analiziralo sliku i znalo prepoznati na slici automobil u odnosu na motocikl.

3.1. Proces strojnog učenja

Proces strojnog učenja je kompleksan proces. Općeniti princip rada procesa strojnog učenja može se pojednostavljeno objasniti u ovih sedam koraka:

1. Korak: Prikupljanje podataka

Prikupljanje podatka je prvi korak u procesu strojnog učenja. Kao u gore navedenom primjeru o automobilima navedeno je da se npr. računalu daju nekakve informacije tj. podatci koji su relevantni od kojih će on moći nešto naučiti. Kada bi to sada bio primjer sa jabukama i narančama ti podatci bi bili npr. oblik i boja predmeta prema kojima bi računalo prepoznalo o kojoj se voćki radi. Korak prikupljanja podataka je temelj svega kod strojnog učenja pa je tako jako bitno da se omogući pružanje što više primjera o predmetu, kako bi računalo ili neki stroj naučilo da određeni objekt može imati određene varijacije te kako ne bi došlo do greške pri radu stroja [13].

2. Korak: Priprema podataka

Nakon procesa prikupljanja podataka, drugi korak je priprema podataka za daljnju obradu. Ključni dio ovog koraka jest prepoznavanje i minimalizacija ikakvih pogrešaka u bazi prikupljenih podataka. Još jedan važan dio ovog koraka bio bi podjela informacija u dvije skupine. U prvoj skupini odnosno u većoj skupini nalazilo bi se oko 80% podataka koji bi se koristili za tzv. trening fazu strojnog učenja, a u drugoj skupini bilo bi ostalih 20% podataka koji bi se koristili za fazu ispitivanja odnosno fazu evaluacije. Ovaj dio jako je važan jer je bitno da se ne koriste isti podatci za ove dvije faze jer onda dolazi do mogućnosti nastajanja pogrešaka koje bi nastale zbog automatizma u kojem bi računalo već unaprijed naučilo koje je rješenje na ponuđeni problem i ne bi se znalo adekvatno prilagoditi nekom drugom problemu [13].

3. Korak: Odabir modela

Nakon rješavanja svih koraka koji su orijentirani prema dijelu sa podacima. Na red dolazi dio u kojemu se odabire vrsta modela koja će se koristiti. Jako je važno izabrati pravilan model jer određeni modeli su namijenjeni za korištenje u radu sa tekstualnim podacima dok su neki na primjer bolji za korištenje sa slikama ili videozapisima [13].

4. Korak: Trening

Ovo je korak u kojem se koriste onih 80% podataka i odabrani model u kojem se odvija proces učenja. Trening je korak koji je baza svega u strojnom učenju i u ovom koraku se zahtjeva puno eksperimentiranja i strpljenja jer nije svaki primjer jednostavan kao razlikovanje jabuka od banana te može potrajati i imati dosta neuspjelih pokušaja prije nego li se dođe do određenih rezultata. Također bi bilo poželjno da onaj koji uči to računalo ili stroj bude stručnjak u tom području koje se pokušava naučiti to računalo, kako bi se ukazalo na sve moguće greške koje nebi bile vidljive svakome tko nije u tom poslu svaki dan [13].

5. Korak: Evaluacije

Korak evaluacije odnosno korak ispitivanja obavlja se nakon što model uspješno prođe kroz prijašnji korak treninga i učenja da bi se vidjelo kako će se taj sustav ponašati u okruženju stvarnog svijeta. Ovdje dolazi do korištenja preostalih 20% podataka s kojima se sustav odnosno model nije susretao tijekom faze treninga. Ovaj korak pomaže i pokazuje programerima odnosno cijelom timu

stručnjaka koji je radio na strojnom učenju tog uređaja jesu li uspjeli u postizanju cilja kojega su si zadali prilikom izvođenja svih dosadašnjih koraka [13].

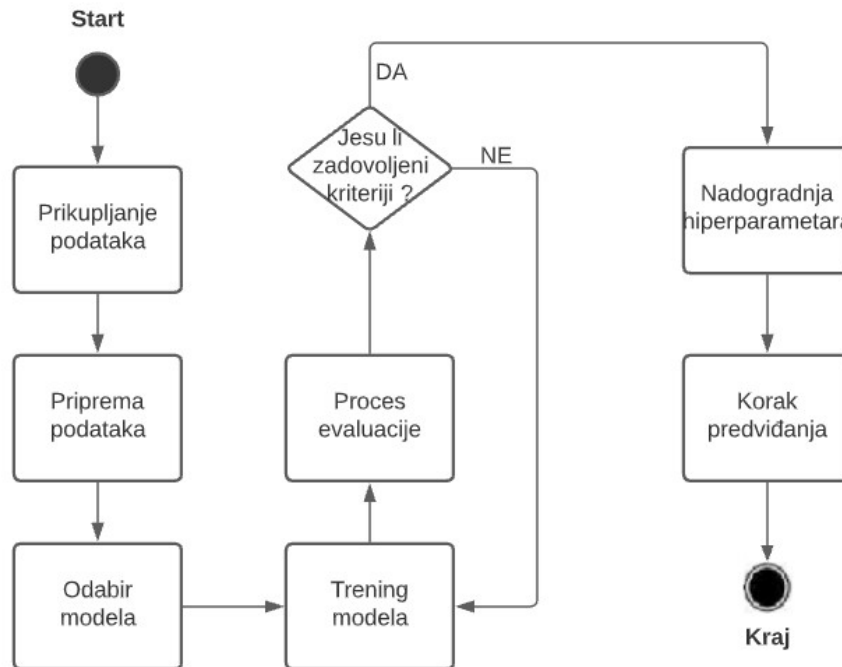
6. Korak: Optimizacija ili nadogradnja hiperparametara

Optimizacija ili nadogradnja hiperparametara je korak koji se izvršava samo u slučaju ako je prethodni korak evaluacije bio uspješan. U ovom koraku se gledaju i proučavaju mogućnosti za dodatnim poboljšavanjem sustava. Ovdje se model vraća nazad u fazu treninga gdje se ispituju dodatne mogućnosti kako bi se sustavu poboljšala preciznost. Ovaj korak se čini iz razloga da se svaki sustav koji je prošao kroz proces strojnog učenja dovede u što bolju optimiziranost odnosno da što brže obavlja svoje zadaće, a da pritom ima što manji trošak i postotak pogreške [13].

7. Korak: Predviđanja

Posljednji korak u procesu strojnog učenja je predviđanje. Ovo faza u kojoj se smatra da je model spreman za praktičnu upotrebu. Ovaj korak je ono što vidi krajnji korisnik modela koji je prošao kroz cijeli postupak strojnog učenja [13].

Na slici 2. nalazi se UML dijagram aktivnosti koji prikazuje sedam koraka u strojnom učenju. Na slici su vidljivi svi navedeni koraci. U petom koraku tj. koraku evaluacije nalazi se i upit jesu li zadovoljeni određeni kriteriji. U slučaju da nije stroj se vraća u trening fazu strojnog učenja, a ako zadovoljava uvijete ide dalje na slijedeći korak odnosno optimizaciju i nadogradnju hiperparametara.



Slika 2. UML dijagram aktivnosti sedam koraka u strojnom učenju

Izvor:[14]

3.2. Vrste i modeli strojnog učenja

Strojno učenje također se može podijeliti na tri glavne vrste. Tri glavne vrste strojnog učenja su: nadzirano učenje, nenadzirano učenje i tzv. podržano/ojačano učenje.

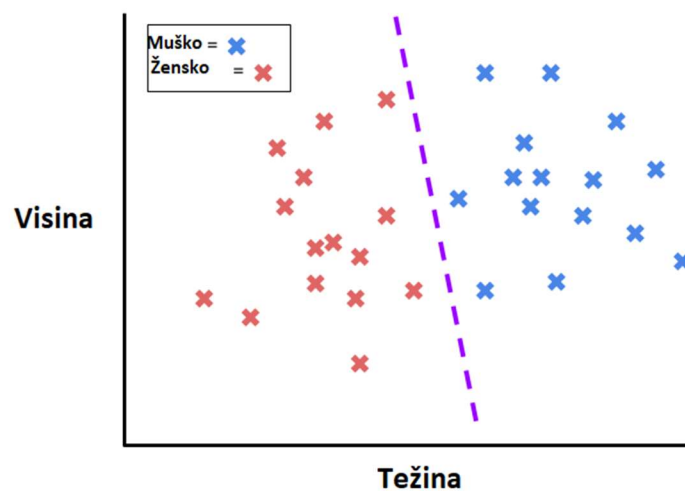
3.2.1. Nadzirano strojno učenje

Nadzirano strojno učenje (engl. *Supervised Machine Learning*) je oblik strojnog učenja koji dosta često koristi. Naziv nadzirano učenje potiče iz ideje da je treniranje ovog tipa algoritma kao da imamo učitelja odnosno profesora koji nadgleda tj. nadzire cijeli proces učenja. Nadzirano učenje radi na principu da dobijemo ulaznu varijablu ili podatak općenito, zadatak je predvidjeti točan podatak na izlazu ili oznaku [15].

Nadzorni algoritmi strojnog učenja napravljeni su da rade na principu učenja po primjerima. Prilikom treniranja ove vrste algoritama važno je da osoba koja upravlja treniranjem (nadzornik) daje određene ulazne podatke sa njihovim točnim izlaznim podacima. Gdje će algoritam tokom treninga tražiti određene uzorke koji se ponavljaju kako bi kasnije sam mogao znati izlazne podatke bez pomoći nadzornika. Nadzirano učenje može se podijeliti na dvije podvrste: klasifikaciju i regresiju [15].

Kod klasifikacijske vrste nadziranog strojnog učenja, klasifikacijski algoritmi dobivaju podatke iz određene skupine. Njihov zadatak da te podatke razvrstaju u određene skupine odnosno klase u koje bi ti podatci spadali na temelju odrađenog treninga. Jedan od najpoznatijih primjera ovog tipa nadziranog strojnog učenja koristi se u elektroničkoj pošti prilikom određivanja dolaznih poruka. Prilikom kojih se te poruke razvrstavaju u određene kategorije kao što su npr. *spam* poruke [15].

Na slici 3. nalazi se grafički prikaz kako klasifikacijsko nadzirano strojno učenje dijeli dobivene podatke u određene skupine. Na ovom konkretnom primjeru ulazni podatci su ljudi odnosno osobe koje imaju svoje osobine kao što su visina i težina itd., a algoritam na temelju odrađene faze treninga ih dijeli u kategorije muških i ženskih osoba.



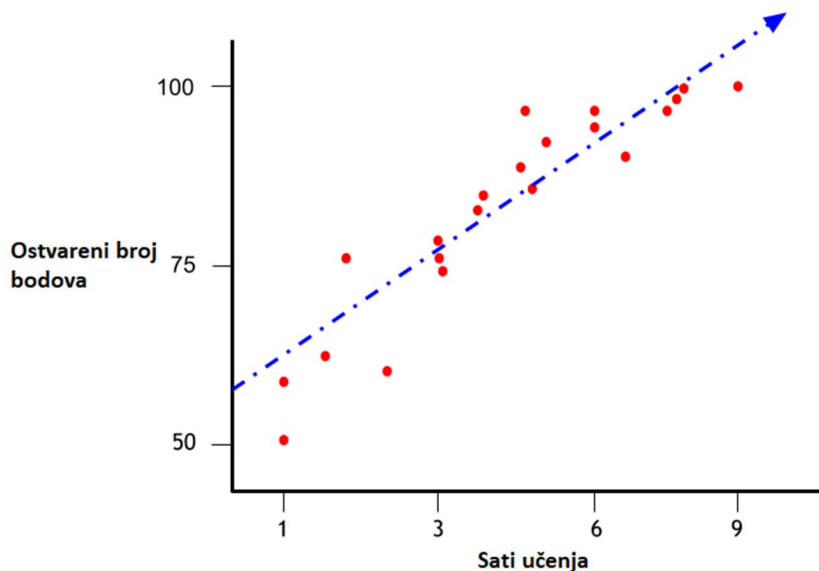
Slika 3. Grafički prikaz klasifikacijskog procesa

Izvor: [15]

Regresija je predvidljivi statistički proces gdje model pokušava pronaći važne poveznice između ovisnih i neovisnih varijabli. Ova vrsta algoritama koristi se u predviđanju brojeva prodaje, ostvarenog prihoda, rezultata ispita ili ispitivanja itd. Tri najčešća tipa algoritama regresije su:

- Linearna regresija,
- Logistička regresija i
- Polinomna regresija [15].

Na slici 4. nalazi se grafički prikaz za linearnu regresiju gdje se za ulazne podatke uzima broj ostvarenih bodova studenata na nekom ispitu te određeni broj sati koje je pojedini student proveo spremajući se za ispit.



Slika 4. Grafički prikaz linearne regresije

Izvor: [15]

3.2.2. Nenadzirano strojno učenje

Kod nenadziranog strojnog učenja (engl. *Unsupervised Machine Learning*) nisu poznate izlazne oznake ili točni izlazni podatci. Zadatak ovog načina strojnog učenja jest otkriti kakve je strukture izlazni podatak [16].

Naziv nenadziranog strojnog učenja proizlazi iz toga što za razliku od nadziranog strojnog učenja kao što je već ranije navedeno, model se ne može istrenirati odnosno ne možemo ga naučiti što će biti izlazna varijabla kada ni sam tim koji radi strojno učenje ovog tipa ne zna što bi to moglo biti pa tako strojevi, odnosno modeli moraju sami tražiti određene skrivene uzorke i informacije

koje najvjerojatnije nisu niti vidljive ljudskom oku kako bi nešto naučili. Ovo je proces u kojem modeli sami uče [16].

Kao što kod nadziranog strojnog učenja postoje dvije vrste algoritama: klasifikacija i regresija koji se u ovoj podvrsti strojnog učenja ne mogu primjenjivati iz razloga što su ulazni podatci poznati, ali ne i izlazne oznake i podatci. Iz tog razloga nenadzirano strojno učenje ima svoju vrstu algoritma koja se zove klasteriranje (grupiranje) (engl. *Clustering*) [16].

Algoritmi klasteriranja mogu se podijeliti na [16]:

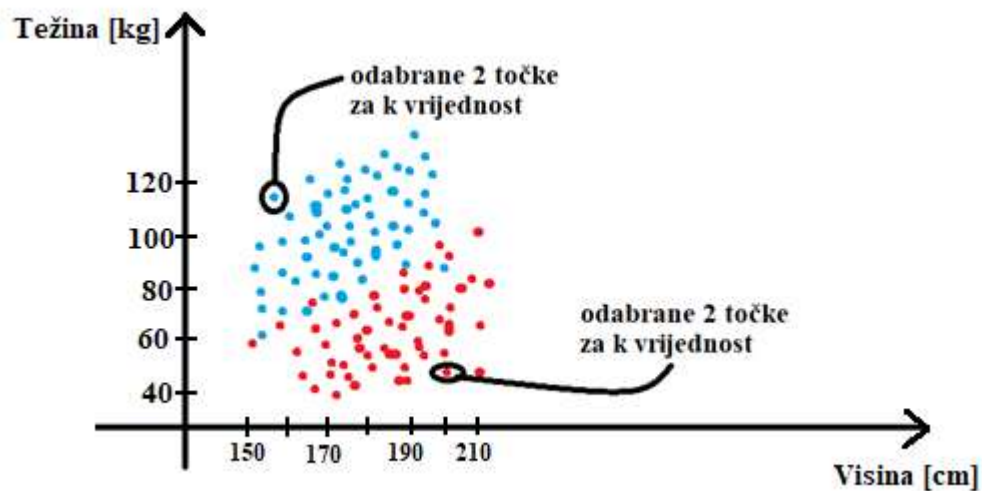
- Ekskluzivno klasteriranje (engl. *Exclusive Clustering*),
- Preklapajuće klasteriranje (engl. *Overlapping Clustering*),
- Hijerarhijsko klasteriranje (engl. *Hierarchical Clustering*) i
- Vjerojatnosno klasteriranje (engl. *Probabilistic Clustering*).

Ekskluzivno klasteriranje radi na principu da ako određeni podatak pripada jednog grupi odnosno jednom klasteru ne može biti uključen kao pripadnik niti jednog drugog klastera. Skroz suprotno ekskluzivnom klasteriranju je preklapajuće klasteriranje kod kojeg podatak može pripadati u dva ili više klastera. Hijerarhijsko klasteriranje temelji se na zajednici između dva susjedna klastera. Dok se posljednja vrsta odnosno vjerojatnosno klasteriranje temelji na čistim matematičkim zakonima vjerojatnosti. Ove četiri algoritma klasteriranja imaju svoje pod vrste. Od ta četiri algoritma klasteriranja najčešće korišteni algoritmi su:

- K-znači (engl. *K-means*) i
- Hijerarhijsko klasteriranje [16].

K-means je jedan od najjednostavnijih algoritama nenadziranog strojnog učenja. Princip rada *K-means* algoritma počinje sa nasumičnim odabiranjem centroida kao početne točke svake grupe podataka. Centroidi odnosno težišta grupa su zamišljene ili stvarne točke koje predstavljaju središte određene grupe podataka na grafičkom prikazu. Sljedeći korak je taj da se računaju udaljenosti između točki podataka i centroida svake grupe podataka. Zatim se svaki podatak dodjeljuje onoj grupi čijem je centroidu taj podatak najbliži. Te se onda centroidi grupa ponovno traže i tako se taj proces ponavlja sve do trenutka se ne postigne da više nema točki podataka koje bi se dodijelile novoj grupi [17].

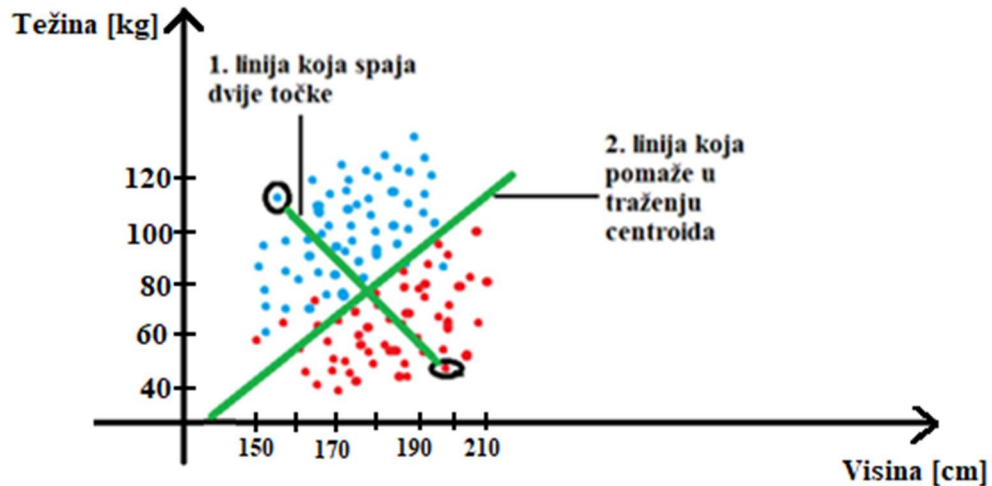
Slika 5. pokazuje prvi korak gdje imamo dvije grupe (klaster) podataka koje su ovdje označene crvenom i plavom bojom. Ovdje se izabire nasumično vrijednost k koja iznosi 2.



Slika 5. Prvi korak u K -means klasteriranju

Izvor: [18]

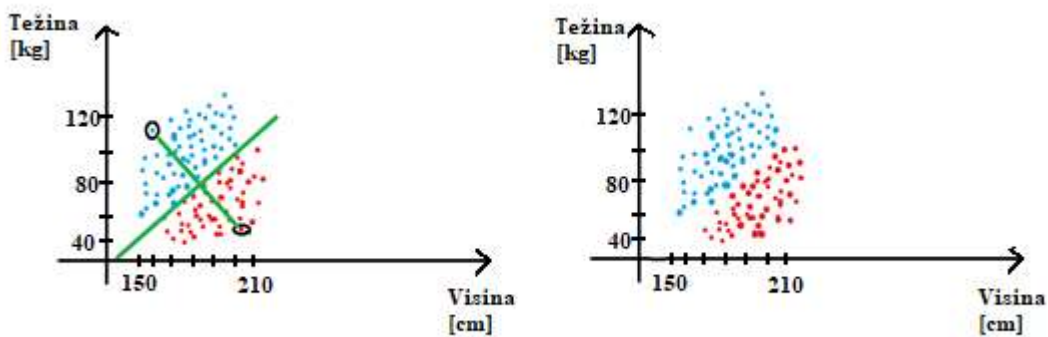
Sljedeći korak je spajanje točaka koje su izabrane u prvom koraku. Kada se te dvije točke spoje dobije se prva linija. Nakon nje povlači se linija koja razdvaja crvene od plavih podataka. Na sjecištu tih dviju točaka dobije se centroid ili središte. Ovaj proces prikazan je na slici 6. Ovaj dio može se i izračunati kako bi se došlo do koordinata centroida, a može se i ovako grafički prikazati.



Slika 6. Drugi korak u *K-means* klasteriranju

Izvor: [18]

Zatim se svi podatci koji se nalaze na „pogrešnoj“ strani linije koja odvaja, u ovom slučaju crvene od plavih podataka, dodjeljuju se u tu vrstu podataka na čijoj se strani nalaze odnosno čijem centroidu su bliži. Onda se ovaj proces sa spajanjem linija i traženjem centorida ponavlja sve dok se podaci koji se nalaze na grafu skroz ne podijele u dvije posebne u grupe. Na slici 7. prikazan je ponovno proces traženja centroida u kojem su se podaci promijenili (lijevo), te završni izgled grafa nakon završenog procesa *K-means* kada su se podaci kompletno razvrstali na plave i crvene (desno).

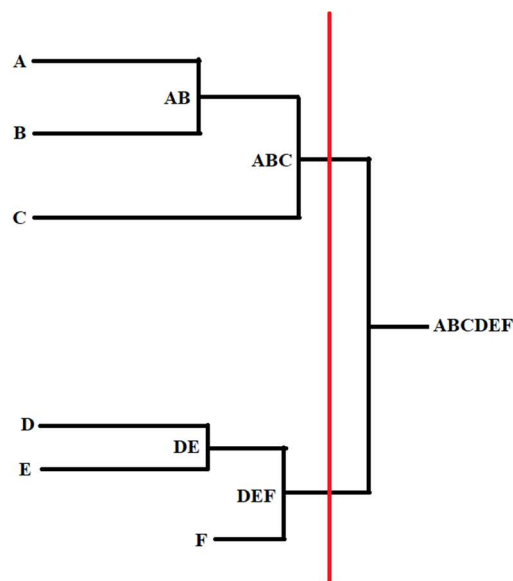


Slika 7. Završni koraci u *K-means* klasteriranju

Izvor: [18]

Hijerarhijsko klasteriranje je algoritam koji se koristi kod nenadziranog strojnog učenja, a koji se temelji na klasteriranju prema hijerarhiji. Ovaj algoritam također ima svoje vrste, a one su: *Agglomerative hierarchical clustering* i *Divisive hierarchical clustering*. Najjednostavnije za objasniti princip rada *Agglomerative hierarchical clustering-a* jest da se kod njega svaki podatak gleda kao posebna grupa podataka. Onda se traži udaljenost između svih tih grupa podataka nakon čega se grupe sa najmanjom udaljenosti spajaju zajedno u novu grupu podataka. Ovaj postupak sa traženjem udaljenosti se zatim ponavlja sa novonastalim grupama podataka te se on ponavlja sve dok se ne dođe do krajnjeg cilja da se svi podatci spoje u jednu novu grupu [19].

Dijagram koji prikazuje ovu vrstu hijerarhijskog klasteriranja zove se Dendogram. On je prikazan na slici 8. Na ovoj slici vidljiva je još i jedna linija koja pokazuje kada se dendogram presiječe s njom na određenom mjestu kako dobivamo različiti broj grupa podataka[19]. Vidimo kako se grupe podataka koje su najbliže spajaju zajedno kao što je to u ovom primjeru grupa podataka A i B se spajaju u grupu podataka AB zatim sa grupom podataka C u grupu podataka ABC itd.



Slika 8. Dendogram

Agglomerative hierarchical clustering može se razlikovati po metodi koja je korištena za određivanje udaljenosti za stvaranje grupa odnosno na kojem principu se računa udaljenost između

grupa podataka. Neke od metoda za određivanje udaljenosti su: pojedinačna veza (engl. *Single link*), potpuna veza (engl. *Complete link*), centroid i prosječni veza (engl. *Average link*) [19].

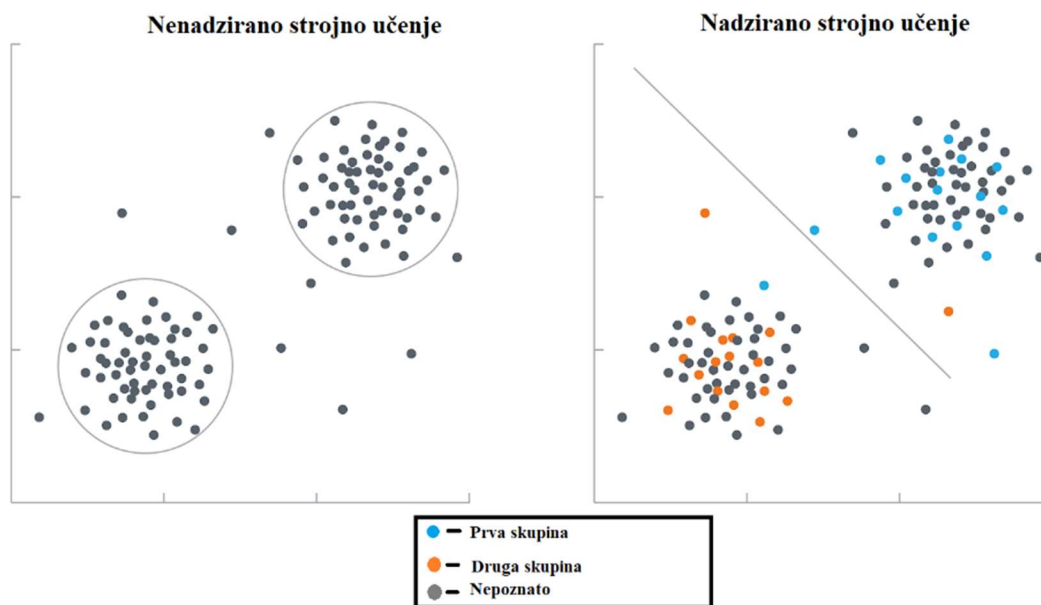
Druga vrsta hijerarhijskog klasteriranja je *Divisive hierarchical clustering* koja se rjeđe koristi od ovih dviju vrsta hijerarhijskog klasteriranja. Njen princip rada je potpuno suprotan *Agglomerative hierarchical clustering* metodi. Kod nje ide sve obrnuto u smislu da se iz jedne velike grupe podatka ide prema što manjoj tj. veliki klaster se dijeli na manje [19].

Razlike u korištenju nadziranog i nenadziranog strojnog učenja su te da se nadzirano učenje koristi za situacije u kojima znamo što točno tražimo u podacima i znamo krajnje rezultate iz kojih možemo naučiti modele kako da rade i kako da pristupaju problemima, dok kod nenadziranog strojnog učenja ne znamo što tražimo i kakav bi rezultat odnosno izlaz mogao biti te zbog toga ne možemo niti učiti modele već ih moramo pustiti da oni na temelju vlastitih algoritama dolaze do nekih zaključaka koji će im pomoći da sami nauče neke stvari. Takvi strojevi rade na temelju vlastitog iskustva kojeg su stekli prilikom procesa istraživanja. Iz tih razloga nenadzirano učenje ima problem mogućeg davanja rezultata sa većom mogućnošću pogreške.

Nenadzirano strojno učenje može se koristiti kod pripreme podataka za nadzirano strojno učenje gdje je zadatak otkriti te izlazne oznake i podatke, također se može koristiti u sustavima preporuke koji pokušavaju predvidjeti određene podatke kao npr. ocjenu korisnika koju je dao nekom proizvodu. Nenadzirano učenje još se može koristiti i kod otkrivanja anomalija tj. otkrivanju nekih rijetkih predmeta, događaja itd. koji imaju određena odstupanja u nekom pogledu od skupine u kojoj se nalaze [20].

Na slici 9. nalazi se grafička usporedba između nadziranog strojnog učenja i strojnog učenja bez nadzora. Iz ovog grafičkog prikaza vidljivo je kako se kod nenadziranog strojnog učenja ulazni podatci klasteriziraju u dvije grupe prema određenim kriterijima i da se u ovom načinu strojnog učenja nalazi samo nepoznata skupina podatka jer se ne zna koje su vrste izlazni podatci dok kod nadziranog strojnog učenja imamo dvije skupine podataka za koje znamo njihove izlazne varijable pa ih je moguće razdvojiti dok se nepoznata skupina podataka dodjeljuje prema sličnostima sa određenom poznatom skupinom podatka. Iz ovog primjera može se zaključiti kako bi najbolje bilo koristiti i jedan i drugi način strojnog učenja. Pa bi se tako iz nenadziranog došlo do informacija o

kakvim nepoznatim izlaznim varijablama je riječ te onda sa tim rezultatima izvrši i nadzirano strojno učenje kako bi se mogućnost za pogreškom što više smanjila prilikom podjele podataka.



Slika 9. Grafička usporedba nenadziiranog i nadziiranog strojnog učenja

Izvor: [21]

3.2.3. Ojačano strojno učenje

Ojačano ili podržano strojno učenje (engl. *Reinforcement Machine Learning*) je vrsta strojnog učenja koja se bavi istraživanjem pitanja kako se inteligentni agenti trebaju ponašati i kako trebaju vršiti neke radnje u nekom okruženju. Te kako bi uz te radnje došli do najboljih mogućih ishoda [22].

Najprije prije početka pojašnjenja principa rada i prednosti ove vrste strojnog učenja trebamo imati potrebno razumijevanje definicija pojmova kao što su: inteligentni agenti, okruženja i nagrade. Agenti su entiteti koji vrše određene radnje kako bi dobili nagradu. Okruženje je određeni scenariji ili događaj u kojemu se agent nalazi, dok je nagrada nešto što agent dobije ako izvrši zapjevanu akciju odnosno zadatak uspješno.

Najjednostavniji primjer kako opisati princip rada ojačanog strojnog učenja bio bi primjer kako naučiti psa da nam da šapu ili prilikom dresure kada pse uče da stane na određenu zapovijed dok

hoda. Pa bi taj proces izgledao ovako: u ovom slučaju pas bi bio agent koji se nalazi u nekom okruženju koje može biti ulica, dvorište, kuća itd. i taj će pas na zapovijed početi hodati, te će isto tako na zapovijed prestati hodati za što će biti nagrađen ukoliko stane i ispuni zadatak ili će biti kažnjen kako bi znao da nije postupio dobro. Na sličnom principu se i strojeve uči pomoću ovog načina strojnog učenja.

Kao i kod prijašnjih vrsta strojnog učenja i ojačano strojno učenje ima svoje algoritme, a oni su [22]:

- Algoritmi temeljeni na vrijednosti (engl. *Value-Based*),
- Algoritmi temeljeni na politikama tj. pravilima (engl. *Policy-based*) i
- Algoritmi temeljeni na modelu (engl. *Model-Based*).

Ojačano strojno učenje se može podijeliti na dvije vrste, a to su: pozitivno ojačano strojno učenje i negativno ojačano strojno učenje.

Pozitivno ojačano strojno učenje je vrsta ojačanog učenja kod koje dolazi do povećanja snage i učestalosti ponašanja koje se povećavaju zbog pojave nekog određenog ponašanja. Prednosti su mu što su maksimizirane performanse i promjene ostaju dulje vrijeme. Dok je negativna strana to što se rezultati mogu umanjiti ako imamo previše tih povećanja [22].

Negativno ojačano strojno učenje se definira kao jačanje ponašanja koje se pojavljuje zbog negativnog stanja koje je trebalo zaustaviti. Negativna strana ove vrste ojačanog strojnog učenja je to što njime može postići samo minimalno ponašanje [22].

Dakle za razliku od nadziranog i nenadziranog strojnog učenja koje imaju određene podatke ili primjere koji su dani da bi se model naučilo kako postupati kod ojačanog strojnog učenja model uči na principu interakcije sa svojim okruženjem i skupljanjem iskustva iz tog postupka.

Ovaj oblik strojnog učenja ima široku primjenu u svijetu pa se tako ojačano strojno učenje može pronaći u robotici, u kontroli semafora (za sada samo u simulacijama), u *gaming* industriji itd. [22].

4. Alati i platforme za primjenu algoritama strojnog učenja

Odabir pravog alata za primjenu strojnog učenja može biti jednako važno kao i izabiranje pravog algoritma za strojno učenje nekog sustava [23]. Ovi alati i platforme se koriste kako bi se brže i preciznije ti sustavi naučili i kako bi im se dala mogućnost učenja i napretka bez previše unosa podataka od strane programera ili bilo koje druge osobe zadužene za proces strojnog učenja.

Razlike u određenim alatima mogu biti u značajkama kao što su: cijena platforme, vrsta jezika kojom se vrši strojno učenje, podržane vrste algoritama i značajki te platforma na kojoj alat radi. Tako su neki od najpoznatijih alata za primjenu algoritama strojnog učenja trenutno: *Scikit Learn*, *PyTorch*, *TensorFlow*, *Knime* itd. [24].

4.1. Scikit Learn

Scikit Learn je jedan od poznatijih alata za strojno učenje. To je alat koji pruža razvijanje strojnog učenja u programskom jeziku Python. To je jednostavan i učinkovit alat koji je pogodan za korištenje u predvidljivim statističkim analizama podatka. Važan podatak je da je to *open source* platforma koja je dostupna svima i besplatna je za korištenje svakome. Algoritmi i značajke koje omogućuje rad Scikit Learn-a su klasifikacija, regresija, klasteriranje, smanjenje dimenzionalnosti, predobrada podataka i odabir modela. Scikit Learn podržava nadzirano strojno učenje i strojno učenje bez nadzora[24],[25]. Programski jezici koji su podržani na ovoj platformi su Python, C++, C i programski jezik Cython. U ovom poglavlju bit će prikazan i primjer *K-means* klasteriranja korištenjem knjižnica iz Scikit Learn-a. Programski jezik korišten u ovom primjeru je Python.

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

%matplotlib inline
```

Slika 10. Uvoz Scikit Learn knjižnica

Izvor: [26]

Na slici 10. prikazan je postupak u kojem se uvoze knjižnice iz Scikit Learn alata. Iz ovog dijela koda vidljivo je da knjižnice koje se koriste su [26]:

- Pandas - koje se koriste za čitanje i pisanje proračunskih podataka.
- Numpy – koje se koriste za izvođenje određenih učinkovitih proračunskih radnji.
- Matplotlib – knjižnica koja se koristi za vizualizaciju podataka.

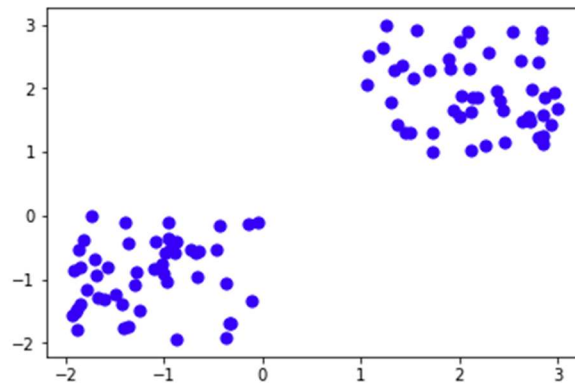
```
X= -2 * np.random.rand(100,2)

X1 = 1 + 2 * np.random.rand(50,2)

X[50:100, :] = X1

plt.scatter(X[ :, 0], X[ :, 1], s = 50, c = 'b')

plt.show()
```



Slika 11. Generiranje i grafički prikaz nasumičnih podataka

Izvor: [26]

Slika 11. pokazuje dio koda u kojem se generiraju nasumični podatci koji se zatim dijele u dvije grupe. Ispod dijela koda prikazan je grafički prikaz generiranih podataka podijeljenih u dvije grupe.

```
from sklearn.cluster import KMeans

Kmean = KMeans(n_clusters=2)

Kmean.fit(X)
```

Slika 12. Dodjeljivanje vrijednosti varijabli k

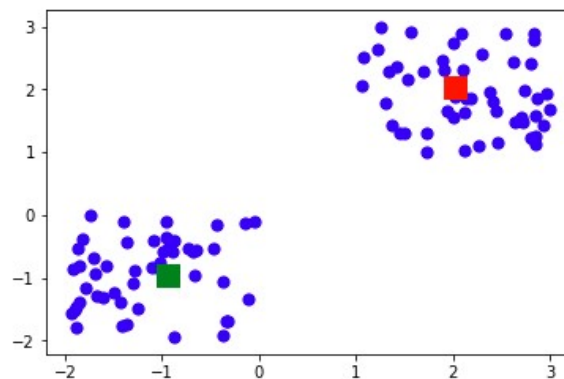
Izvor: [26]

Zatim dolazimo do dijela u kojem koristimo Scikit Learn tj. u ovom dijelu koda koriste se dostupne funkcije koje se nalaze u njegovoj knjižnici. U kodu prikazanom na slici 12. dodijeljena proizvoljna vrijednost 2 za `k(n_clusters)`. [26]

```
Kmean.cluster_centers_  
  
plt.scatter(X[ : , 0], X[ : , 1], s =50, c='b')  
  
plt.scatter(-0.94665068, -0.97138368, s=200, c='g', marker='s')  
  
plt.scatter(2.01559419, 2.02597093, s=200, c='r', marker='s')  
  
plt.show()
```

Slika 13. Kod za ispis grafičkog prikaza centroida

Izvor: [26]



Slika 14. Grafički prikaz dviju grupa sa njihovim centroidima

Izvor: [26]

Korak koji slijedi nakon dodjeljivanja vrijednosti `k` jest određivanje i izrada grafičkog prikaza centroida koja je prikazana na slici 13. Na toj slici je vidljivo da se upisuju dobivene koordinate za centroide kao i boja u kojoj želimo da taj centroid bude prikazan. Grafički prikaz dobivenih centroida prikazan je na slici 14.

Zatim pomoću linije koda „`Kmean.labels_`“ vršimo testiranje klasteriranja u kojem dobivao rezultate koji su prikazani na slici 15. Iz tih rezultata vidljivo je da 50 podataka pripada jednoj grupi dok ostatak pripada drugoj grupi [26].

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1])
```

Slika 15. Ispis rezultata

Izvor: [26]

4.2. Platforma za strojno učenje TensorFlow

TensorFlow je jedna od najpoznatijih platformi za strojno učenje. Prema usporedbi sa PyTorch-om je gotovo ista što se tiče platformi na kojima je moguć njen rad, programskih jezika koje podržava i po tome što je također besplatna *open source* platforma. Ova platforma je pogodna jer nudi jednostavan način izrade modela za fazu treninga u strojnom učenju jer ima veliki broj opcija prilikom izrade modela s čime omogućuje korisniku izradu točno onakvog modela kakav mu je potreban. Također TensorFlow nudi JavaScript knjižnicu za programiranje protoka podataka [24],[27].

Kao i kod Scikit Learn-a ovdje je također prikazan postupak *K-means* klasteriranja samo ovaj puta korištenjem TensorFlow knjižnice.

Kao i kod prvog slučaja uvoze se knjižnice matplotlib i numpy te se ne koristi knjižnica pandas što je prikazano na slici 16. Također je prikazan broj grupa koji iznosi 3, količina podataka 200 te broj ponavljanja 100 [28].

```
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf

points_n = 200
clusters_n = 3
iteration_n = 100
```

Slika 16. Uvoz knjižnica u TensorFlow-u

Izvor: [28]

Nakon uvoza potrebnih knjižnica, sljedeći korak je izrada modela za klasteriranje. Najprije se generiraju nasumični podatci iz kojih se nasumično uzimaju točke za početne centroide. Ove linije koda prikazane su na slici 17.

```
points = tf.constant(np.random.uniform(0, 10, (points_n, 2)))  
centroids = tf.Variable(tf.slice(tf.random_shuffle(points), [0, 0], [clusters_n, -1]))
```

Slika 17. Generiranje nasumičnih podataka i odabir centroida

Izvor: [28]

Za sljedeći korak koji slijedi točke podataka i centroide moraju se proširiti sa dvije dimenzije u tri dimenzije što će omogućiti operacije oduzimanja pomoću kojih se računaju udaljenosti između točaka podataka i centroida. Koraci proširenja centroida i točaka podataka i računanja udaljenosti prikazani su na slici 18 [28].

```
points_expanded = tf.expand_dims(points, 0)  
centroids_expanded = tf.expand_dims(centroids, 1)  
  
distances = tf.reduce_sum(tf.square(tf.subtract(points_expanded, centroids_expanded)), 2)  
assignments = tf.argmin(distances, 0)
```

Slika 18. Proširivanje u tri dimenzije i računanje udaljenosti između podataka i centroida

Izvor: [28]

Na slici 19. nalazi se dio koda u kojem se točke podataka dodjeljuju svakoj grupi podataka i računaju se srednje vrijednosti pomoću kojih se računaju nove vrijednosti odnosno koordinate centroida [28].

```

means = []
for c in range(clusters_n):
    means.append(tf.reduce_mean(
        tf.gather(points,
            tf.reshape(
                tf.where(
                    tf.equal(assignments, c)
                ), [1, -1])
            ), reduction_indices=[1]))

new_centroids = tf.concat(means, 0)
update_centroids = tf.assign(centroids, new_centroids)

```

Slika 19. Dodjeljivanje podataka određenoj grupi podataka i računanje novih centroida

Izvor: [28]

Krajnja faza *K-means* klasteriranja u TensorFlow-u kreće sa izradom grafičkog prikaza. Na slici 20. prikazan je ta faza kreiranja grafa u kojem se najprije ažuriraju novi centroidi iz prethodnog koraka i vraćaju im se vrijednosti zajedno sa vrijednostima klastera [28].

```

with tf.Session() as sess:
    sess.run(init)
    for step in xrange(iteration_n):
        [_, centroid_values, points_values, assignment_values] = sess.run([update_centroids,
            centroids, points, assignments])

```

Slika 20. Vraćanje vrijednosti grupama podataka i ažuriranje centroida

Izvor: [28]

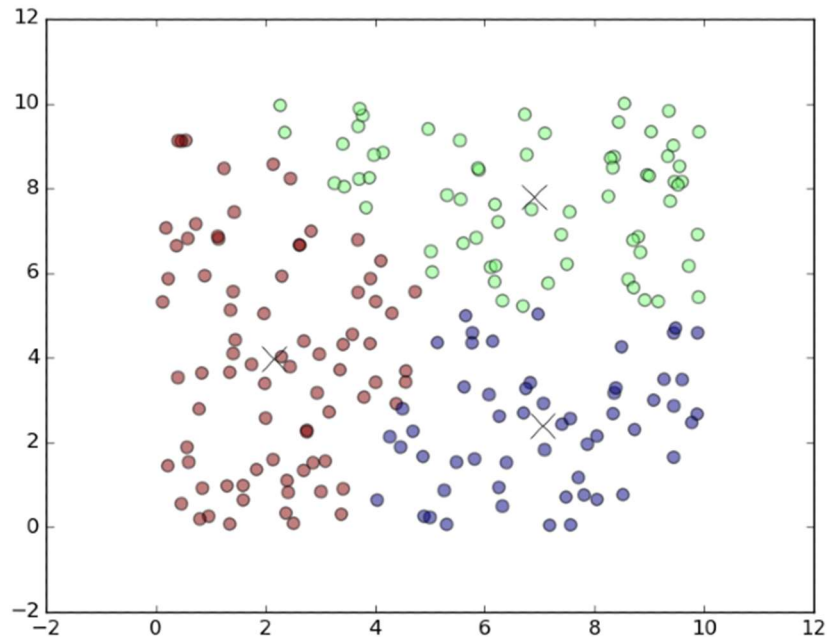
Na kraju piše se dio koda koji se odnosi na grafički ispis rezultata. Ovdje se prikazuju koordinate centroida pojedine grupe podataka [28]. Na slici 21. prikazan je kod za izradu i prikaz grafa te ispod koda prikazano je ispis tj. traženi grafički prikaz.

```

print("centroids", centroid_values)

plt.scatter(points_values[:, 0], points_values[:, 1], c=assignment_values, s=50, alpha=0.5)
plt.plot(centroid_values[:, 0], centroid_values[:, 1], 'kx', markersize=15)
plt.show()

```



Slika 21. Kod za prikaz grafičkog rješenja i grafičko rješenje

Izvor: [28]

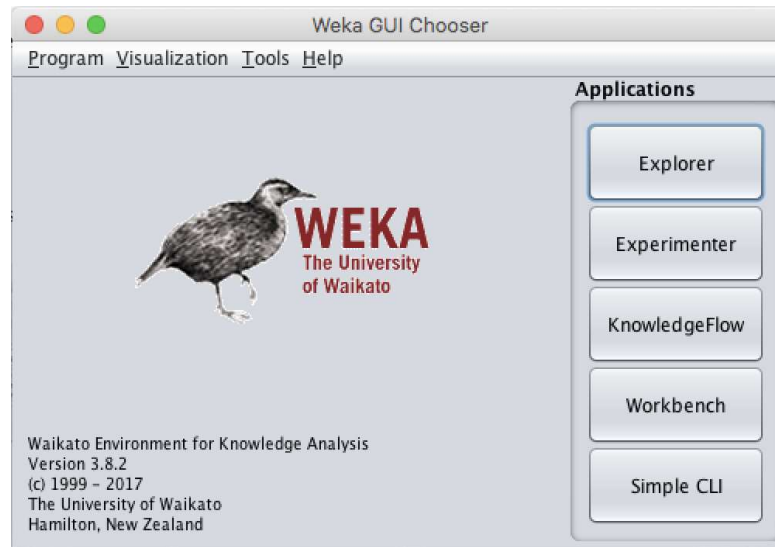
4.3. PyTorch

PyTorch je također besplatan alat za primjenu algoritama strojnog učenja. On se od Scikit Learn-a razlikuje po tome što su programski jezici koji se kod ovog alata koriste: Python, C++ i računalna platforma te model sučelja za programiranje pod nazivom CUDA. PyTorch kao i Scikit radi na Linux, MacOS i Windows operativnim sustavima. PyTorch je također dostupan i ima potporu za rad na *cloud* servisima. PyTorch je program koji je više pogodan za duboko učenje nego što je za strojno učenje, ali je također moguće napraviti i proces strojnog učenja na ovoj platformi [24],[29].

4.4. Weka

Weka je isto open source softver koji pruža alate za pripremu podataka i korištenje nekolicine algoritama strojnog učenja. Uz alate za pripremu podataka sadrži još i alate za rad s algoritmima regresije, klasterizacije, vizualizacije i klasifikacije. Weka je također besplatna platforma koja radi na Windows, Linux i MacOS operativnim sustavima. Weka je softver koji se najviše koristi za rudarenje podataka (engl. *Data mining*). To je oblik analitike podataka u kojem se velike količine podataka izvlače bitni podatci. Za rudarenje podataka koriste se razni algoritmi strojnog učenja koji se u programu Weka mogu koristiti direktno na podacima ili putem Java koda. Za razliku od ostalih alata Weka podržava samo JavaScript programski jezik [30],[31]. Ono što Weku razlikuje od ostalih alata za strojno učenje je to što je alat koji koristi grafičko korisničko sučelje (engl. *Graphical user interface - GUI*) koje omogućuje učitavanje skupa podataka, pokretanje i rad s algoritmima [32]

Na slici 22. prikazana je početna stranica Weka alata za primjenu strojnog učenja na kojoj se nalazi izbornik na kojem korisnik može odabrati željenu aplikaciju koju Weka nudi. Nakon odabira aplikacije unutar Weka-e otvara mu se GUI za odabranu aplikaciju.



Slika 22. Početna stranica Weka programskog alata

Izvor: [33]

4.5. RapidMiner

Ovaj alat daje mogućnost za strojno učenje, duboko učenje, prikupljanje podataka, predvidljivu analitiku itd. Također može se koristiti za istraživanje, razvijanje aplikacija te obrazovanje. Koristi isključivo Javu kao programski jezik, može raditi na raznim operativnim sustavima. Koristi se za pripremu podataka, vizualizaciju te provjerava i optimizira modele. Razlika između njega i ostalih navedenih alata je u tome što on uz svoju besplatnu verziju također nudi i verzije tj. proširenja alata koje se plaćaju [24]. RapidMiner isto kao i Weka koristi GUI s kojim pomaže pri implementaciji i dizajniranju u analitičkim dijelovima rada. RapidMiner nudi 30-dnevni besplatni probni period nakon čega se prelazi na odabrani plan plaćanja usluge. RapidMiner nudi mogućnost rada u svom proširenju na internetskom pretraživaču što je danas sve popularnije. Ta usluga košta 10 USD (američki dolar) mjesečno, a omogućuje rad njihovog programa RapidMiner Go putem nekog od internetskih pretraživača kao što su Google Chrome, Safari itd. bez obaveze skidanja posebnog programa na svoje računalo. Druga mogućnost je preuzimanja kompletnog programa RapidMiner Studio koji pruža puno više mogućnosti i usluga nego RapidMiner Go verzija koja je zapravo napravljena za one koji se žele učiti strojno učenje i ostale mogućnosti koje pruža RapidMiner Go [34]

5. Primjena strojnog učenja u telekomunikacijama

Strojno učenje i umjetna inteligencija u području telekomunikacija imaju široki obujam primjene između ostalog koriste se u procesu zaštite računalnih mreža od neautoriziranih pristupa mreži, klasifikaciji IoT uređaja. Također strojno učenje može se koristiti u upravljanju mrežnim prometom i predviđanju rezultata analiza [35].

5.1. Uloga strojnog učenja u sigurnosti mrežnog prometa

Sve češće mrežni promet je na udaru ilegalnih napada i pokušaja krađe podataka. Tako imamo DDoS napade (engl. *Distributed Denial of Service*) kao jednu vrstu od najčešćih prijetnji mrežnom prometu. DDoS je vrsta napada na određenu internetsku stranicu. Cilj DDoS napada je preplaviti promet na toj stranici više nego što poslužitelj ili mreža to može podnijeti. To se radi na način da se promet koji se šalje u mrežu sastoji od konstantnih dolaznih poruka, zahtjeva za spajanjem, lažnih paketa itd. Sa DDoS napadima hakeri postižu da stranica ili poslužitelj kojeg su napali postaje nekorisna ili nedostupna [36].

Prilikom dolaznog prometa on se dijeli na dvije skupine stvarni promet i nestvarni promet. U nestvarni odnosno lažni promet spadaju gore navedene vrste podataka koje se šalju kako bi se mreža preplavila. U tu vrstu prometa spadaju podvrste DDoS prometa. DDoS promet razlikuje se po vrsti protokola koji je korišten kako bi se izvršio napad pa tako imamo npr. DNS DDoS napade koji koriste DNS protokol, UDP DDoS napade, CharGen DDoS napade i tako dalje [37].

UDP protokol se zbog svoje jednostavnosti često koristi u DDoS napadima. Uređaj koji prima UDP pakete nema dovoljno veliki kapacitet za preuzimanje tolike količine dolaznog prometa, a uz to još pokušava odgovoriti sa velikim brojem ICMP (engl. *Internet Control Message Protocol*) „*destination host unreachable*“ poruka koje stvaraju dodatno zagušenje mreže [37].

DNS protokol je zadnjih godina jedan od vodećih protokola za pojačavanje DDoS napada. Kod ove vrste DDoS napada uz standardnu opremu koriste se i pojačivači koji su uređaji koji se nalaze van botnet mreže koji pružaju odgovore na upite. Računala unutar botnet mreže šalju upite i lažiraju IP adresu izvora tj. IP adresu poslužitelja kojeg se napada te zbog toga dolazi do slanja odgovora od pojačivača do IP adrese napadnutog poslužitelja koji sada ima puno veći broj zahtjeva koje njegov kapacitet mreže ne može podnijeti te dolazi do preplavlivanja [37].

CharGen je protokol koji kreira pakete sadržaja od 0-512 slučajnih znakova kao odgovor na zahtjev poslan na UDP ili TCP port 19. Prilikom slanja TCP zahtjeva na koji podržava CharGen protokol server kreće slati slučajne znakove sve dok se veza sa poslužiteljem ne prekine. Ako je poslan UDP zahtjev poslužitelj također odgovara sa slučajno odabranim znakovima svaki puta kada dobije UDP datagram [37].

Kod DDoS napada umjetna inteligencija se koristi za njihovo sprječavanje. Umjetne neuronske mreže koriste se za prepoznavanje lažnog prometa koji se stvara na ulazu u mrežu. Da bi se ANN mogle koristiti najprije se moraju odraditi postupci prikupljanja podataka koji sadrže zapise o prometu u mreži. Prikupljeni podatci prolaze kroz proces normalizacije parametara vrijednosti kako bi se mogli koristiti u ANN. Sljedeći korak je izrada i razvoj ANN modela u kojem se uključuju procesi učenja, proces evaluacije i proces testiranja. Na cjelokupnog procesa izvršava se analiza dobivenih rezultata [37]. Strojno učenje ovdje se može koristiti prilikom prikupljanja podataka za korištenje u umjetnoj neuronskoj mreži i prilikom faze njenog razvoja.

Strojno učenje se također može koristiti kod detektiranja DDoS napada koje generiraju IoT uređaji. Internet stvari je princip koji prikazuje mrežnu infrastrukturu u kojoj je bilo koji uređaj spojen na internet i s drugim uređajima u mrežu. U toj mreži svi uređaji međusobno dijele podatke o sebi i o okruženju u kojem se nalaze. Najčešći primjer IoT mreža nalazi se u pametnim kućama. IoT uređaji korišteni u pametnim kućama poznati su još i kao SHIoT (engl. *Smart Home Internet of Things*) uređaji. Uređaji sa ugrađenim sensorima u sebi spajaju se na IoT mrežu koja prikuplja podatke od drugih uređaja te se onda vrši analiza tih prikupljenih podataka da bi se onda najvrijednije informacije mogle slati na aplikacije koje onda izvršavaju određenu zadaću na temelju tih rezultata analize [38].

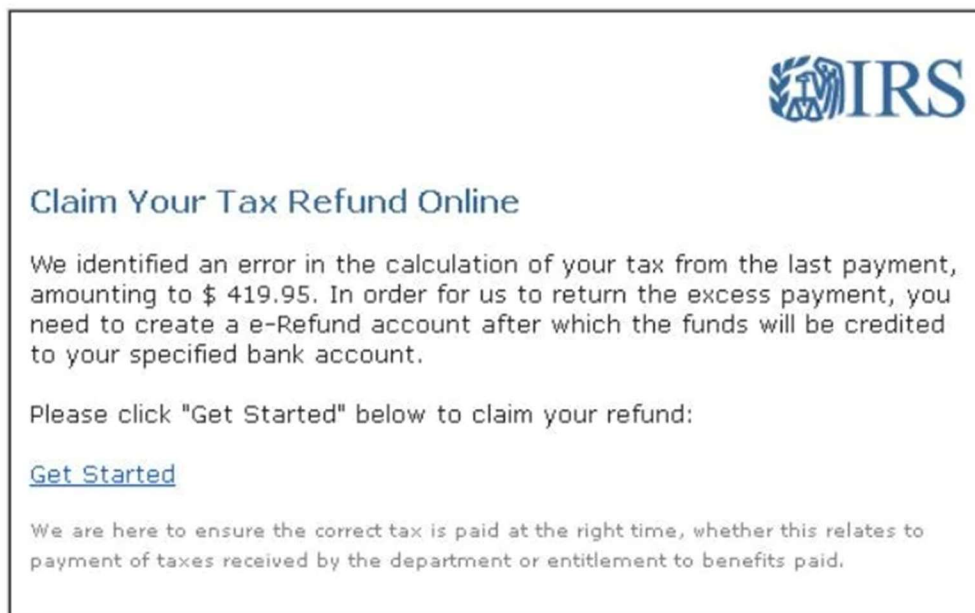
Velika većina tih SHIoT uređaja su jeftini uređaji koji nemaju značajnije mogućnosti te ih je zbog te njihove jednostavnosti lagano kompromirati kako bi se izvršile zloćudne radnje što uključuje i generiranje DDoS napada. Da bi se razvio model temeljen na strojnom učenju koji će detektirati bilo kakvu pojavu anomalija u mreži mora postojati skup podataka koji će predstavljati stvarni i nestvarni promet. Za razvoj takvog modela može se koristiti metoda „stablo logističkih modela“ (engl. *Logistic Model Tree, LMT*). LMT je klasifikacijski model nadziranog strojnog učenja u koji kombinira logističku regresiju i učenje na stablu odluka (engl. *Decision Tree Learning*). Prilikom razvoja ovakvih modela SHIoT uređaji se dijele u više klasa. Za svaku od tih

klasa se onda razvijaju LMT modeli koji detektiraju anomalije te se za svaku klasu model razvija zasebno [39].

Razvijeni model radi u dvije faze. Prva faza je preduvjet za detekciju DDoS generiranog prometa u drugoj fazi. Također prva faza uključuje i klasifikaciju SHIoT uređaja na temelju generiranog prometa [39].

Napad krađom identiteta (engl. *Phishing*) su vrste prijevara koje se najčešće događaju putem elektroničke pošte, ali su česte i putem SMS poruka ili telefonskih poziva. To su napadi koji za cilj imaju na nelegitiman način ukrasti podatke od uglavnom veće skupine korisnika [40]. Slično *phishingu* su spam poruke koje također mogu biti štetne za računalo i mrežu, ali su uglavnom namijenjene za promocije ili reklame. Dok *phishing* poruke koje dolaze uvijek imaju određene kriminalne namjene [41].

Na slici 22. nalazi se slika e-pošte kao primjer za *phishing* poruku u kojoj se korisniku nalaže kako je nastala određena greška prilikom plaćanja poreza te kako bi mu se vratio njegov novac da je potrebno da on napravi profil preko linka koji se nalazi u e-pošti.



Slika 23. *Phishing* poruka

Izvor: [42]

Umjetna inteligencija u ovom primjeru sa *phishing* porukama može se koristiti da se na temelju algoritama nadziranog i nenadziranog strojnog učenja dolazne poruke dijele u kategorije spam poruka i *phishing* poruka.

Problem kod razlikovanja *phishing* poruka od *spam* poruka je taj što su one u principu podvrsta *spam* poruka pa ih je zbog toga teško razlikovati. Njihovo razlikovanje može se napraviti uz pomoć korištenja algoritma hijerarhijskog klasteriranja. Taj proces izgleda ovako: Dolazna pošta dijeli se u tri skupine. Pošto nema označenih podataka koji bi bili dostupni za treniranje od korisnika se traži da označi koja dolazne poruke u kategoriju u koju spadaju poruke iz te prve od tri skupine. Kada korisnik izvrši taj zadatak algoritam za klasteriranje ima podatke prema kojima može odraditi učenje. Te se ostale poruke iz druge dvije skupine dijele prema algoritmu koji je prošao kroz fazu treninga temeljenu na korisnikovom odabiru. Na kraju svake skupine koristi se klasifikator iz prethodne skupine za fazu treninga algoritma. Dok se za prvu skupinu koriste izlazni podatci iz algoritma klasteriranja za trening [41]. Na ovaj način se sustav nauči kako prepoznati *phishing* poruku o kojoj onda kasnije može obavijestiti korisnika.

5.2. Strojno učenje u području klasifikacije IoT uređaja

Vrsta strojnog učenja koja se koristi za razvijanje klasifikacijskog modela je logistički regresijski model sa logitboost-om. Logitboost je algoritam koji se koristi za dodatno pojačavanje klasifikacijskog algoritma [43].

Prvi korak u razvijanju klasifikacijskog modela je odabir značajki koje će biti relevantne u konstruiranju modela. Ispravan odabir značajki donosi veću preciznost klasifikacijskog modela, veću brzinu klasifikacije te može smanjiti pojavu prekomjernog opterećenja. Cilj je izabrati što manju količinu značajki koje će dati najbolje rezultate klasifikacijskog modela kako bi se smanjilo vrijeme predviđanja klase uređaja i kako bi se smanjila kompleksnost općenito [43].

Sljedeći korak je prikupljanje skupa podataka koji služe za razvijanje klasifikacijskog modela. Cilj modela jest odrediti kojoj klasi pripada taj IoT uređaj. To određivanje temelji može se temeljiti na karakteristikama količine prometa koju generira u određenom vremenskom periodu mjerenja. Karakteristike ovise o značajkama IoT uređaja koji proizvodi taj promet [43].

Zatim ide proces strojnog učenja u kojem se primjenjuje algoritam logističke regresije. Algoritam logističke regresije prikazuje uvjetnu vjerojatnost da određeni uređaj koji se prati

pripada određenoj klasi. Zatim se uvodi logitboost algoritam temeljen na logističkoj regresiji koji se koristi radi optimizacije. Korištenje ove vrste algoritma strojnog učenja za izradu klasifikacijskog modela pokazalo se iznimno točno sa preciznošću od 99.79% prema [43] istraživanju.

5.3. Ostale mogućnosti primjene strojnog učenja u telekomunikacijama

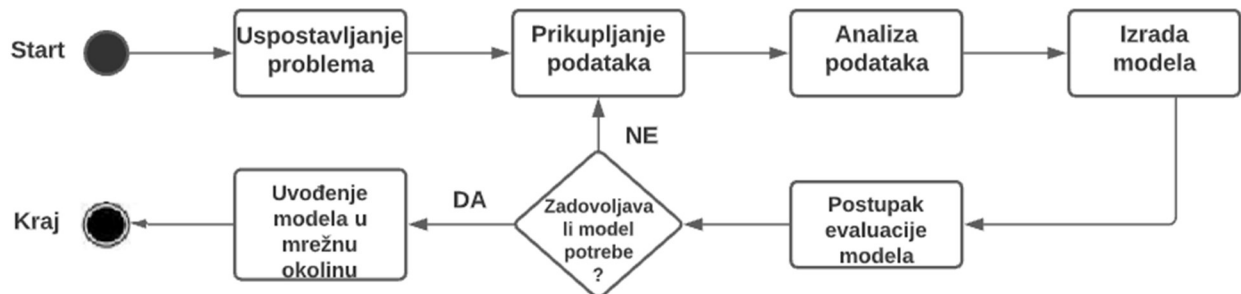
Neke od ostalih područja primjene umjetne inteligencije i strojnog učenja u telekomunikacijama su: predviđanje mrežnog prometa, prediktivno održavanje uređaja u mreži, itd.

Predviđanje mrežnog prometa jako je bitno u današnje vrijeme pogotovo ako je riječ o predviđanju mobilnog mrežnog prometa jer u današnje vrijeme svakim danom sve je više korisnika mobilnih mreža čije upravljanje iz dana u dan postaje sve zahtjevnije. Za predviđanje mobilnog mrežnog prometa može se koristiti više metoda strojnog učenja [44]. Jedna od tih metoda je metoda potpornih vektora (engl. *Support Vector Machine- SVM*). SVM je statistička metoda strojnog učenja čiji je glavni cilj mapiranje nelinearnih podataka u viši dimenzionalni linearni prostor gdje će ti podatci biti klasificirani po hiperravnini (engl. *Hyperplane*) [45]. Hiperravan su granice odlučivanja koje pomažu pri klasifikaciji podataka. Ova metoda može biti jako korisna za izradu predviđanja zagušenja prometa u mreži.

Predviđanje količine generiranog mrežnog prometa vrši se s kombinacijom algoritama strojnog učenja, dubokog učenja i/ili umjetnih neuronskih mreža. Strojno učenje u procesu predviđanja generiranog mrežnog prometa koristi se u fazama prikupljanja podataka, rudarenja podacima i obrade podataka te izradi modela na kojem se onda vrši daljnje istraživanje. Algoritmi ML koji se mogu koristiti u ovim postupcima su *K-means* i algoritam hijerarhijskog klasteriranja iz vrste nenadziranog strojnog učenja. Moguće je koristiti i druge algoritme ML, odabir algoritma ovisi o tome koja je namjena tj. što se pokušava postići [46].

Prediktivno održavanje uređaja u mreži je održavanje uređaja koje se obavlja na način da se prati u kakvom je stanju uređaj koji se održava. Najčešće se njegovo stanje prati putem bežičnih senzora koji se nalaze na uređaju i koji šalju izvještaje centralnom računalu zaduženom za daljnju analizu stanja uređaja i odlučivanje treba li se uređaj zamijeniti ili popravljati. Uz korištenje regresijskog algoritma nadziranog strojnog učenja može se saznati koliko je uređaju ostalo još sati

korisnog rada. Također uz korištenje klasifikacijskog algoritma nadziranog strojnog učenja može se predvidjeti kvar u danom vremenskom periodu [47].



Slika 24. UML dijagram aktivnosti strojnog učenja u području telekomunikacija

Izvor: [48]

Na slici 24. prikazan je UML dijagram aktivnosti rada algoritama strojnog učenja u telekomunikacijama. Prvi korak je uspostavljanje problema iz razloga što je faza treniranja unutar strojnog učenja uvijek dugotrajna i često vrlo skupa jako je važno da se točno i precizno odredi problematika zadanog zadatka. Problematika se može razvrstati na način da se odmah zna koja će se vrsta algoritma koristiti za rješavanje problema. Drugi korak jest prikupljanje podataka. Kao i u prije razrađenim poglavljima, potrebno je prikupiti što veću količinu podataka koji opisuju mrežu. Ti podatci npr. mogu biti vremena trajanja vremenskih poziva. U trećem koraku radi se analiza podataka pri kojoj se pokušavaju otkriti vrijednosti i značajke mreže da bi se mogao što vjerodostojnije izraditi model. Sljedeći korak je upravo izrada modela koja uključuje odabir modela, trening i doradu tj. nadogradnju modela. U ovom koraku još se mora paziti da se odabire ispravan algoritam i ispravna metoda za treniranje modela. Preposljednji korak u ovom dijagramu je proces evaluacije u kojemu se radi analiza nastalih mogućih grešaka i unakrsna provjera ispravnosti. Unakrsna provjera ispravnosti se izvršava kako bi se vidjelo radi li model dovoljno dobro. Zadnji korak je korak uvođenja učenog modela u funkcionalnu mrežnu okolinu [48].

6. Zaključak

Strojno učenje je jako koristan proces u današnje vrijeme jer donosi veću preciznost rada sustava, olakšava posao ljudima koji su donedavno morali izvršavati te poslove samostalno, a sada uz pomoć strojnog učenja mogu napraviti modele i naučiti uređaje koji će im pomagati u izvršavanju zadatka. Iako je dosta koristan postupak dosta je i kompliciran za izvršiti ga od početka do kraja. Od prvog koraka pa sve do posljednjeg, svaki korak je jednako važan u procesu strojnog učenja jer ako se izvrši i sitna pogreška kod bilo kojeg koraka dolazi do mogućnosti za neispravnim radom naučenog stroja.

Umjetna inteligencija kao znanost i tehnologija je čovjekova budućnost. Sve više danas se čovjek oslanja na svoje pametne uređaje koje koristi i koji ga okružuju, a koji se svi temelje na umjetnoj inteligenciji. Strojno učenje koje je podsustav u umjetnoj inteligenciji može biti izvedeno u tri vrste. Najbolji rezultati se naravno mogu očekivati kada bi uređaj mogao proći kroz sve tri vrste strojnog učenja, ali naravno to u velikom broju slučajeva nije moguće niti potrebno. U praksi se najčešće koristi nenadzirano strojno učenje jer u velikoj većini slučajeva se ne znaju podatci koji se očekuju na izlazu te je zbog toga ova vrsta strojnog učenja češća. Strojno učenje ima razne alate i platforme u kojima može biti izvedeno, a za izradu koda strojnog učenja koriste se poznati programski jezici kao što su Python, C++, Java itd.

Kada se govori o mogućnostima primjene metoda strojnog učenja ona je opširna. Strojno učenje može se koristiti gotovo u svakom području. Pa se tako strojno učenje koristi u medicini, farmaciji, vojnoj industriji, *gaming* industriji i između ostalog i u telekomunikacijskoj industriji. U informacijsko-telekomunikacijskom području strojno učenje može se koristiti za sve vrste prikupljanja i grupiranja podataka, za klasifikaciju IoT uređaja, za razvrstavanje elektroničke pošte, zaštitu od kibernetičkih napada, predviđanje gustoće prometa u mreži itd.

Do sve češćeg upotrebe strojnog učenja došlo je zbog toga što se povećao broj podataka koji je dostupan u današnje vrijeme. Ta količina podataka je previše za običnog čovjeka pa čak i za tim ljudi te zbog toga sve su više popularni sustavi koji omogućuju obradu podataka tih količina. Kao što napreduje sve ostalo u svijetu tako je za očekivati kako će s vremenom doći i nove metode strojnog učenja koje će možda biti jednostavnije i efikasnije od ovih koje danas postoje.

Literatura

- [1] Europska komisija. Što je je to AI (umjetna inteligencija) i trebamo li je se bojati? Preuzeto sa: https://ec.europa.eu/croatia/basic/what_is_artificial_intelligence_hr [Pristupljeno: travanj 2021.]
- [2] javaTpoint. History of Artificial Intelligence. Preuzeto sa: <https://www.javatpoint.com/history-of-artificial-intelligence> [Pristupljeno: travanj 2021.]
- [3] Dartmouth. Artificial Intelligence (AI) Coined at Dartmouth. Preuzeto sa: <https://250.dartmouth.edu/highlights/artificial-intelligence-ai-coined-dartmouth> [Pristupljeno: travanj 2021.]
- [4] Software Testing Help. What is Artificial Intelligence: Definition & Sub-Fields Of AI. Preuzeto sa: <https://www.softwaretestinghelp.com/what-is-artificial-intelligence/> [Pristupljeno: travanj 2021.]
- [5] Towards data science. What is Deep Learning and How does it work ? Preuzeto sa: <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac> [Pristupljeno: travanj 2021.]
- [6] Asis&t.Natural language processing. Preuzeto sa: <https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/aris.1440370103> [Pristupljeno: travanj 2021.]
- [7] javaTpoint. What is an Expert System?. Preuzeto sa: <https://www.javatpoint.com/expert-systems-in-artificial-intelligence> [Pristupljeno: travanj 2021.]
- [8] Elements od AI. Kako se izrađuju neuronske mreže. Preuzelo sa: <https://course.elementsofai.com/hr/5/2> [Pristupljeno: lipanj 2021.]
- [9] Forbes. The 10 Best Examples of How Ai Is Already Used In Our Everyday Life. Preuzeto sa: <https://www.forbes.com/sites/bernardmarr/2019/12/16/the-10-best-examples-of-how-ai-is-already-used-in-our-everyday-life/?sh=71ab59a51171> [Pristupljeno: travanj 2021.]
- [10] Make use of. 5 Simple Tips to Manage What You Watch on Netflix. Preuzeto sa: <https://www.makeuseof.com/tag/tips-manage-watch-netflix/> [Pristupljeno: travanj 2021.]

- [11] IBM. Machine Learning. Preuzeto sa: <https://www.ibm.com/cloud/learn/machine-learning>
[Pristupljeno: travanj 2021.]
- [12] PCChip. Što je to „machine learning“ ili strojno učenje ?. Preuzeto sa:
<https://pcchip.hr/helpdesk/sto-je-to-machine-learning-ili-strojno-ucenje/>
[Pristupljeno: travanj 2021.]
- [13]Live Code Stream. 7 Steps of Machine Learning. Preuzeto sa:
<https://livecodestream.dev/post/7-steps-of-machine-learning/> [Pristupljeno: travanj 2021.]
- [14]Great Learning. What is Machine Learning? How much Machine Learning Works and future of it ?. Preuzeto sa: <https://www.mygreatlearning.com/blog/what-is-machine-learning/> [Pristupljeno: travanj 2021.]
- [15] Towards data science. A brief Introduction to Supervised Learning. Preuzeto sa:
<https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590>
[Pristupljeno: travanj 2021.]
- [16] Towards data science. Unsupervised Learning and Data Clustering. Preuzeto sa:
<https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a>
[Pristupljeno: travanj 2021.]
- [17] Cebecci Z, Yildiz F. Comparison of K-Means and Fuzzy C-Means Algorithms on Different Cluster Structures. Journal of Agricultural Informatics. 2015; 6(3):13-23 Preuzeto sa:
[https://www.researchgate.net/publication/282861550_Comparison_of_K-Means_and_Fuzzy_C-Means_Algorithms_on_Different_Cluster_Structures] [Pristupljeno: srpanj 2021]
- [18]Analytics Vidhya. A Simple Explanation of K-Means Clustering. Preuzeto sa:
<https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/>
[Pristupljeno: srpanj 2021.]
- [19] Data Analytics. Hierarchical Clustering Explained with Python Example. Preuzeto sa:
https://vitalflux-com.cdn.ampproject.org/v/s/vitalflux.com/hierarchical-clustering-explained-with-python-example/amp/?amp_gsa=1&_js_v=a6&usqp=mq331AQKKAFAQrABIACAw%3D%3D#amp_tf=Izvor%3A%20%251%24s&aoh=16260895352381&referrer=https%3A%2F%2Fwww.go

[ogle.com&share=https%3A%2F%2Fvitalflux.com%2Fhierarchical-clustering-explained-with-python-example%2F](https://www.vitalflux.com/hierarchical-clustering-explained-with-python-example/) [Pristupljeno: srpanj 2021.]

[20] altexsoft. Unsupervised Learning: Algorithms and Examples. Preuzeto sa: <https://www.altexsoft.com/blog/unsupervised-machine-learning/> [Pristupljeno: travanj 2021.]

[21]Lawtomated. Supervised Learning vs Unsupervised Learning. Which is better ?. Preuzeto sa: <https://lawtomated.com/supervised-vs-unsupervised-learning-which-is-better/> [Pristupljeno: travanj 2021.]

[22]Guru. Reinforcement Learning Tutorial. Preuzeto sa: <https://www.guru99.com/reinforcement-learning-tutorial.html> [Pristupljeno: travanj 2021.]

[23] Machine Learning Mastery. Machine Learning Tools. Preuzeto sa: <https://machinelearningmastery.com/machine-learning-tools/> [Pristupljeno: srpanj 2021.]

[24] My server name. 11 Najpopularnijih softverskih alata za strojno učenje 2021. godine. Preuzeto sa: <https://hr.myservername.com/11-most-popular-machine-learning-software-tools-2021> [Pristupljeno: srpanj 2021.]

[25]Scikit-learn. Preuzeto sa: <https://scikit-learn.org/stable/> [Pristupljeno: srpanj 2021.]

[26]Towards data science. Understanding K-means Clustering in Machine Learning. Preuzeto sa: <https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1> [Pristupljeno: srpanj 2021.]

[27] TensorFlow. Preuzeto sa: <https://www.tensorflow.org/> [Pristupljeno: srpanj 2021.]

[28] Altoros. Implementing k-means Clustering with TensorFlow. Preuzeto sa: <https://www.altoros.com/blog/using-k-means-clustering-in-tensorflow/> [Pristupljeno: srpanj 2021.]

[29] PyTorch. Preuzeto sa: <https://pytorch.org/> [Pristupljeno: srpanj 2021.]

[30] TutorialsPoint. Preuzeto sa: https://www.tutorialspoint.com/weka/what_is_weka.htm [Pristupljeno: kolovoz 2021.]

[31] Weka 3. Preuzeto sa: <https://www.cs.waikato.ac.nz/ml/weka/> [Pristupljeno: srpanj 2021.]

- [32] Machine Learning Mastery. How to Run Your First Classifier in Weka ?. Preuzeto sa: <https://machinelearningmastery.com/how-to-run-your-first-classifier-in-weka/>
[Pristupljeno: kolovoz 2021.]
- [33] freeCodeCamp. Weka Tutorial – GUI-based Machine Learning with Java. Preuzeto sa: <https://www.freecodecamp.org/news/machine-learning-using-weka/>
[Pristupljeno: kolovoz 2021.]
- [34] RapidMiner. Preuzeto sa: <https://rapidminer.com/products/> [Pristupljeno: kolovoz 2021.]
- [35] SearchNetworkig.TechTarget. What is the role of machine learning in networking ?. Preuzeto sa: <https://searchnetworking.techtarget.com/answer/What-is-the-role-of-machine-learning-in-networking> [Pristupljeno: srpanj 2021]
- [36] Norton. What is disistributed denial of service attack (DDoS) and what can you do about them ?. Preuzeto sa: <https://us.norton.com/internetsecurity-emerging-threats-what-is-a-ddos-attack-30sectech-by-norton.html> [Pristupljeno: srpanj 2021]
- [37] Peraković D, Periša M, Cvitić I, Husnjak S. Model for Detection and Classification of DDoS Traffic Bassed on Artificial Neural Network. Telfor Journal. 2017; 9(1): 26-31
- [38] IBM. What is the Internet of Things (IoT)?. Preuzeto sa : <https://www.ibm.com/blogs/internet-of-things/what-is-the-iot/> [Prisutupljeno: srpanj 2021.]
- [39] Cvitić I, Peraković D, Gupta B, Cho K.K.R. Boosting-based DDoS Detection in Internet of Things Systems. IEEE Internet of Things Journal. 2021.
- [40] Phishing.org. What Is Phishing ?. Preuzeto sa: <https://www.phishing.org/what-is-phishing>
[Pristupljeno: srpanj 2021]
- [41] Gupta B.B, Tewari A, Cvitić I, Peraković D, Chang X. Artificial intelligence empowered emails classifier for Internet of Things based systems in industry 4.0. 2021.
- [42] Norton. Phishing email examples to help you identify phishing scams. Preuzeto sa: <https://us.norton.com/internetsecurity-online-scams-phishing-email-examples.html> [Pristupljeno: srpanj 2021]

[43] Cvitić I, Peraković D, Periša M, Gupta B. Ensemble machine learning approach for classification of IoT devices in smart home. International Journal of Machine Learning and Cybernetics. 2020

[44] Jamshidi S. The Applications of Machine Learning Techniques in Networking. 2019.

Preuzeto sa: <https://www.cs.uoregon.edu/Reports/AREA-201902-Jamshidi.pdf>

[Pristupljeno: srpanj 2021.]

[45] Akhtar M, Maridpour S. A Review od Traffic Congestion Prediction Using Artificial Intelligence. Journal of Advanced Transportation. 2021. Preuzeto sa:

<https://www.hindawi.com/journals/jat/2021/8878011/#abstract> [Pristupljeno: srpanj 2021.]

[46] Manish J R, Theyazn H H. A Review of Network Traffic Analysis and Prediction Tehniques.

Preuzeto sa: <https://arxiv.org/ftp/arxiv/papers/1507/1507.05722.pdf> [Pristupljeno: kolovoz 2021.]

[47] Towards data science. How to Implement Machine Learning For Predictive Maintenance.

Preuzeto sa: <https://towardsdatascience.com/how-to-implement-machine-learning-for-predictive-maintenance-4633cdbc4860> [Pristupljeno: srpanj 2021.]

[48] Wang M, Cui Y, Wang X, Xiao S, Jiang J. Machine Learning for Networking:

Workflow, Advances and Opportunities. IEEE Network. 2017. Preuzeto sa:

<https://arxiv.org/ftp/arxiv/papers/1709/1709.08339.pdf> [Pristupljeno: srpanj 2021.]

Popis kratica

ML	Machine Learning
AI	Artificial Inteligence
IoT	Internet of Things
NLP	Natural language processing
ANN	Artificial Neuron Network
GUI	Graphical User Interface
DDoS	Distributed Denial of Service
UDP	User Datagram Protocol
DNS	Domain Name System
ICMP	Internet Control Message Protocol
IP	Internet Protocol
SHIoT	Smart Home Internet of Things
LMT	Logistic Model Tree

Popis slika

Slika 1. Prikaz izračunate vrijednosti sviđanja videozapisa nekom korisniku na platformi "Netflix"	6
Slika 2. UML dijagram aktivnosti sedam koraka u strojnom učenju	10
Slika 3. Grafički prikaz klasifikacijskog procesa	11
Slika 4. Grafički prikaz linearne regresije	12
Slika 5. Prvi korak u <i>K-means</i> klasteriranju	14
Slika 6. Drugi korak u <i>K-means</i> klasteriranju.....	15
Slika 7. Završni koraci u <i>K-means</i> klasteriranju.....	15
Slika 8. Dendogram	16
Slika 9. Grafička usporedba nenadziranog i nadziranog strojnog učenja.....	18
Slika 10. Uvoz Scikit Learn knjižnica	20
Slika 11. Generiranje i grafički prikaz nasumičnih podataka	21
Slika 12. Dodjeljivanje vrijednosti varijabli k	21
Slika 13. Kod za ispis grafičkog prikaza centroida.....	22
Slika 14. Grafički prikaz dviju grupa sa njihovim centroidima.....	22
Slika 15. Ispis rezultata	23
Slika 16. Uvoz knjižnica u TensorFlow-u	23
Slika 17. Generiranje nasumičnih podataka i odabir centroida	24
Slika 18. Proširivanje u tri dimenzije i računanje udaljenosti između podataka i centroida	24
Slika 19. Dodjeljivanje podataka određenoj grupi podataka i računanje novih centroida.....	25
Slika 20. Vraćanje vrijednosti grupama podataka i ažuriranje centroida	25
Slika 21. Kod za prikaz grafičkog rješenja i grafičko rješenje	26
Slika 22. Početna stranica Weka programskog alata	27
Slika 23. <i>Phishing</i> poruka.....	31
Slika 24. UML dijagram aktivnosti strojnog učenja u području telekomunikacija	34



Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ završni rad
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na
objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz
necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

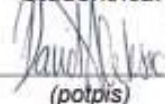
Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj
visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu _____ završnog rada
pod naslovom **Mogućnosti primjene metoda strojnog učenja u području**
telekomunikacija

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom
repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, _____ 20.8.2021 _____

Student/ica:


(potpis)