

Interoperabilnost servisa strojnog učenja različitih pružatelja usluga u oblacima

Tikvica, Andrea

Master's thesis / Diplomski rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:968901>

Rights / Prava: [Attribution-NoDerivs 3.0 Unported/Imenovanje-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2024-07-15**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



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

Andrea Tikvica

**INTEROPERABILNOST SERVISA
STROJNOG UČENJA RAZLIČITIH
PRUŽATELJA USLUGA U OBLACIMA**

DIPLOMSKI RAD

Varaždin, 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Andrea Tikvica

Matični broj: 46357/17–R

Studij: Informacijsko i programsko inženjerstvo

INTEROPERABILNOST SERVISA STROJNOG UČENJA
RAZLIČITIH PRUŽATELJA USLUGA U OBLACIMA
DIPLOMSKI RAD

Mentor:

Doc. dr. sc. Darko Andročec

Varaždin, rujan 2019.

Andrea Tikvica

Izjava o izvornosti

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

Autorica potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Glavna teza ovog rada je mogućnost postizanja interoperabilnosti između odabranih pružatelja sustava u oblaku za strojno učenje. Tri vodeća pružatelja sustava u oblaku za strojno učenje su Google, Amazon i Microsoft. Navedeni sustavi predstavljaju platforme od kojih svaka platforma nudi veliki broj usluga koje su uglavnom zasnovane na primjeni umjetne inteligencije. Platforma kao usluga jedan je od modela usluga u oblacima među kojima se nalaze softver kao usluga te infrastruktura kao usluga. Kako bismo znali postoji li mogućnost interoperabilnosti treba shvatiti na koji način radi strojno učenje te njegovu svrhovitost. U okviru rada navedene su usluge strojnog učenja spomenutih platformi te je na praktičnom dijelu rada prikazana njihova interoperabilnost. Rezultat rada je prikaz suradnje servisa te nove razine korištenja strojnog učenja.

Ključne riječi: strojno učenje; platforma kao usluga; interoperabilnost; servis; oblak; Google Cloud Platform; Microsoft Azure;

Sadržaj

1. Uvod	1
2. Strojno učenje.....	2
2.1. Vrste strojnog učenja	3
2.1.1. Nadzirano strojno učenje.....	4
2.1.2. Nenadzirano strojno učenje.....	5
2.1.3. Podržano strojno učenje.....	6
2.2. Slučajevi korištenja strojnog učenja	6
2.2.1. Prepoznavanje glasa i zvuka.....	6
2.2.2. Prepoznavanje slike ili video zapisa	8
2.2.3. Prepoznavanje teksta.....	8
2.2.4. Otkrivanje prijevara	9
3. Primjena strojnog učenja	10
3.1. Neuronske mreže.....	10
3.1.1. Oblici umjetnih neuronskih mreža.....	11
3.1.1.1. Konvolucijske neuronske mreže	11
3.1.1.2. Ponavljajuće neuronske mreže.....	12
3.1.1.3. Višeslojne neuronske mreže.....	12
3.2. Regresija.....	12
3.2.1. Linearna regresija	13
3.2.2. Logistička regresija	13
3.3. Stablo odlučivanja.....	14
3.4. Metoda potpornih vektora (SVM).....	15
3.5. Algoritam za povećanje gradijenta (GBM)	16
3.6. Programski okviri dubinskog učenja	16
3.6.1. TensorFlow	17
3.6.2. Caffe	18
3.6.3. Keras	19
3.6.4. DeepLearning4J.....	19
3.6.5. PyTorch.....	20
4. Servisi strojnog učenja.....	21
4.1. Usluge u oblaku	21
4.2. Modeli u oblaku.....	22
4.3. Pružatelji usluga strojnog učenja.....	23

4.3.1. Google Cloud Platform	24
4.3.2. Microsoft Azure	25
4.3.3. Amazon Web Services	26
4.3.4. Oracle	27
4.3.5. IBM	27
5. Interoperabilnost	28
6. Praktični dio rada	30
6.1. Korištene tehnologije.....	30
6.1.1. Java	30
6.1.2. NetBeans	31
6.1.3. Apache Tomcat	31
6.1.4. REST API.....	32
6.2. Konfiguracija korištenih pružatelja usluga	32
6.3. Struktura programskog rješenja	34
6.4. Detaljni opis funkcionalnosti programa.....	35
6.5. Kratki osvrt.....	39
7. Zaključak	40
Popis literature	41
Popis slika	45
Popis tablica.....	46

1. Uvod

Informatizacija je uvelike pridonijela boljoj organizaciji rada i boljem poslovanju mnogih privatnih i javnih tvrtki, ali i ustanovama kod kojih je došlo do smanjenja obujma posla (pogotovo administrativnog). Unatoč tome svakim su se danom podaci nastavili količinski povećavati te su počeli zauzimati znatnu količinu memorije. Kao rezultat toga nerijetko se počela javljati otežana brzina obrada podataka. Stoga se godinama nastojalo doći do novih rješenja kojima bi se taj problem riješio ili barem minimizirao. Jedan od načina rješenja spomenutog problema nastojao se riješiti uvođenjem strojnog učenja. Ono je uglavnom imalo zadaću samostalno odrađivati unaprijed definirani posao na temelju ranije zadanih podataka. Pri tome bi strojno učenje također imalo i mogućnosti samostalnog unosa novih podataka za vrijeme rada.

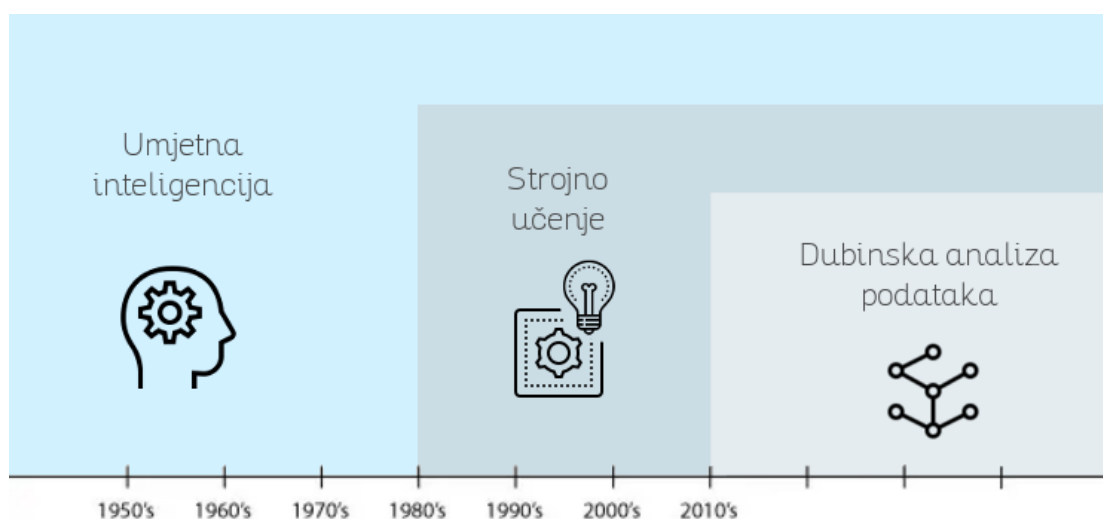
Danas je strojno učenje izrazito aktualna tematika upravo zbog velikih mogućnosti njegove primjene. Primjerice, strojno učenje se primjenjuje prilikom prepoznavanja lica, dubinskih analiza podataka i predviđanja, računalne lingvistike, itd. Nadalje, danas se sve veći broj uspješnih poduzeća u okviru svog poslovanja služi strojnim učenjem kao uslugom primarno namijenjenom njihovim korisnicima. Primjerice, Netflix koristi strojno učenje kao sustav preporuke filmova, Google kao sustav pretraživanja itd. No, Google (Google Cloud Platform) spada i među tri vodeća sustava za strojno učenje u oblaku. Osim njega tu se još ubrajaju i Microsoft (Azure Machine Learning) i Amazon (Amazon Web Services). Navedeni servisi strojnog učenja nude MLaaS (engl. Machine Learning as a Service) platforme pomoću kojih je moguće trenirati modele na temelju malo ili nimalo stručnosti o području u kojem rade. U ovom radu pobliže će se prikazati interoperabilnost između Google i Microsoft sustava na temelju odabranih usluga. Također, time će se prikazati suradnja između sustava s istreniranim modelima.

Na samom početku rada opisano je strojno učenje te su navedene i objašnjene vrste strojnog učenja. Daljnji sadržaj rada opisuje primjenjivost strojnog učenja pri čemu je također navedeno par osnovnih algoritama. U nastavku rada ukratko su opisane usluge u oblaku, odnosno definirane su osnovne vrste usluga te je obrazložena njihova važnost. Potom je definiran pojam interoperabilnosti kao i njegova primjenjivost u svrhu postizanja suradnje između odabranih platformi. Osim teorijskih osnova, u svrhu jasnijeg prikaza mogućnosti interoperabilnosti odabranih usluga u oblaku, u okviru rada napravljen je i praktični dio rada. Rad se na kraju zaokružuje kratkim osvrtom na kreiranu aplikaciju te zaključkom kojim se ističe važnost primjene strojnog učenja kao i mogućnosti njegove nadogradnje pomoću interoperabilnosti.

2. Strojno učenje

Strojno učenje predstavlja dio umjetne inteligencije (*engl. artificial intelligence*) koje radi s algoritmima potrebni za donošenje određenih odluka. Pri tom su strojevi dizajnirani tako da obrađuju veliki broj naučenih podataka, informacija i algoritama u svrhu obavljanja postavljenih zadataka. Strojevi s implementiranom umjetnom inteligencijom moraju imati sposobnost prilagođavanja okolini u kojoj se nalaze te sposobnost učenja kako bi se mogli što bolje prilagoditi u novoj okolini. S obzirom na to da se u nekim slučajevima strojno učenje pokazalo sklono pogreškama rezultiralo je nastankom dubinskog učenja (*engl. deep learning*) kao podskupa strojnog učenja. Slika 1. prikazuje kronološki prikaz nastanka dijelova umjetne inteligencije. Dubinsko učenje poznato je po otkrivanju znanja u skupovima podataka u kojem se strojno učenje primjenjuje na djelomično strukturirane baze podataka. Također ono se primjenjuje u različitim djelatnostima: financijama (kreditna sposobnost, zlouporaba kartice), trgovini (CRM), medicini (dijagnoza), itd. Nadalje, pomoću dubinskog učenja postižu se rezultati koji su slični onima koje bi čovjek dobio. Takvi rezultati se temelje na logičkoj strukturi i slojevitoj strukturi računalnih jedinica koje se nazivaju umjetne neuronske mreže (*engl. artificial neural networks*) [1].

Postoje tri vrste strojnog učenja prema stilu učenja: nadzirano (*engl. supervised*), nenadzirano (*engl. unsupervised*) i podržano strojno učenje (*engl. reinforcement learning*). Svaka vrsta ima svoj algoritam za rješavanje problema te je potrebno prepoznati koja vrsta strojnog učenja odgovara određenoj situaciji. Drugi način kategoriziranja algoritama za strojno učenje je prema sličnosti oblika funkcije. U nastavku rada ti oblici su također nabrojani te ukratko objašnjeni [2].



Slika 1: Kronološki prikaz nastanka dijelova umjetne inteligencije (Izvor: Patel, 2018)

Kako bi se pospješila učinkovitost te daljnja aplikativnost i razvoj strojnog učenja potrebno ga je primijeniti u okviru aplikacije gdje bi ono bilo izrazito korisno. U okviru poslovanja to se prvenstveno odnosi na aplikacije za predviđanje, marketing, otkrivanje prijevara, upravljanje zalihama te nadzor stroja Internet stvari [3].

Strojno učenje se pokazalo kao moćan alat za poboljšanje sustava. Međutim, unatoč svim svojim prednostima ono ima i nekoliko nedostataka. U Tablici 1. prikazana je njihova usporedba [4].

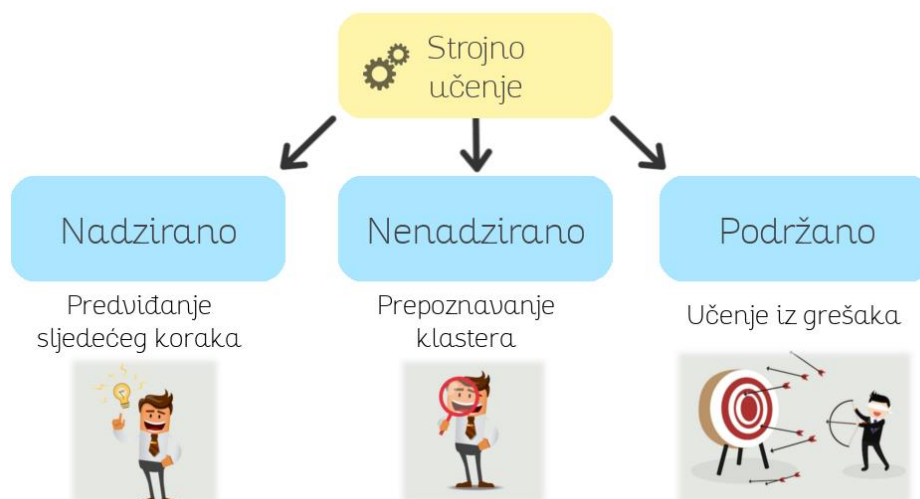
Tablica 1: Prikaz prednosti i nedostataka kod strojnog učenja

Prednosti	Nedostaci
<ul style="list-style-type: none"> • Jeftin za rad. Plaća se onoliko koliko i koristi. • U suradnji s javnim oblacima pruža se jeftino pohranjivanje podataka. • Pruža SDK i API usluge pomoću kojih se ugrađuju dodatne funkcionalnosti u aplikacije u raznim programskim jezicima. • Programi temeljeni na strojnom učenju su automatizirani, točnije nije potrebna ljudska intervencija. • Osigurava se kontinuirano poboljšanje jer algoritmi s vremenom steknu nova iskustva pa tako pospešuju točnost i učinkovitost. • U dinamičnom okruženju pomoću strojnog učenja može se raditi s višedimenzionalnim skupovima podataka. • Ima mogućnost pružanja boljeg korisničkog iskustva. 	<ul style="list-style-type: none"> • Zahtjeva veliki skup podataka kojima pokreće treninge a ti podaci trebaju biti dobre kvalitete. • Ponekad može zahtijevati dodatna poboljšanja na snazi računala kako bi se algoritmi naučili i razvili u predviđenom vremenu. • Tumačenje rezultata može biti izazov. • Osjetljivo na pogreške. Ako se radi s dovoljno malo podataka tada istrenirani model može davati neobične rezultate. U tom slučaju može doći do pogrešaka u radu u programu koje je teško otkriti ali i ispraviti.

(Izvor: DataFlair, 2018)

2.1. Vrste strojnog učenja

Strojevi na temelju vrste raspoloživih podataka i trenutnog istraživačkog pitanja donose odluku o treniranju modela uz pomoć odabranog modela učenja s algoritmima. S obzirom na to da postoji velik broj različitih slučajeva odnosno modela za analiziranje grupirane su vrste strojnog učenja (Slika 2.) pri čemu je svaka vrsta strojnog učenja pogodna za rješavanje određenog problema. U nastavku rada slijedi kratki opis svake vrste strojnog učenja.

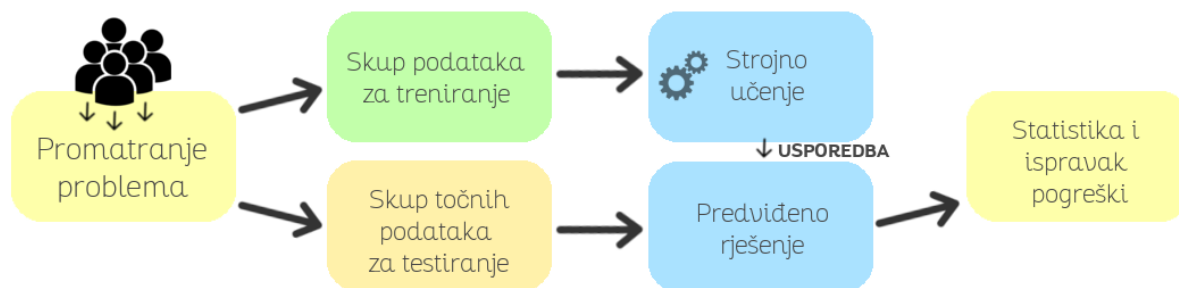


Slika 2: Vrste strojnog učenja (Izvor: Heidenreich, 2018)

2.1.1. Nadzirano strojno učenje

Većina modela zasniva se na primjeni nadziranog strojnog učenja kao oblika učenja u kojem algoritam uči na temelju obilježenog skupa podataka. Dostupan mu je ključ odgovora kako bi mogao usporediti dobivene rezultate te tako procijeniti svoju točnost u obuci. Dakle, za primjenu ove vrste strojnog učenja potrebno je poznavati vrijednosti ulaznih i izlaznih podataka. Podatke uglavnom određuje čovjek kako bi što preciznije opisao model kojem pripada podatak. Tijekom nadziranog strojnog učenja stroj uči povezivati probleme s predviđenim odgovorima i sve točnije pružati odgovarajuća rješenja. Nakon uspješne obuke stroj ima sposobnost promatranja novog modela kojeg prije nije obrađivao i davati odgovarajuća rješenja za isti (Slika 3.) [5].

Funkcija koja spaja ulazne (X) i izlazne podatke (Y) ima sljedeći oblik: $Y = f(X)$. Iz funkcije je vidljivo da je za svaki izlaz poznat njegov ulaz. Funkcija se iterativno ponavlja i predviđa podatke te dobiva povratne informacije o točnosti predviđanja. Učenje završava kada algoritam postigne prihvatljivu razinu performansi. Na Slici 3. prikazan je slijed izvođenja nadziranog strojnog učenja.

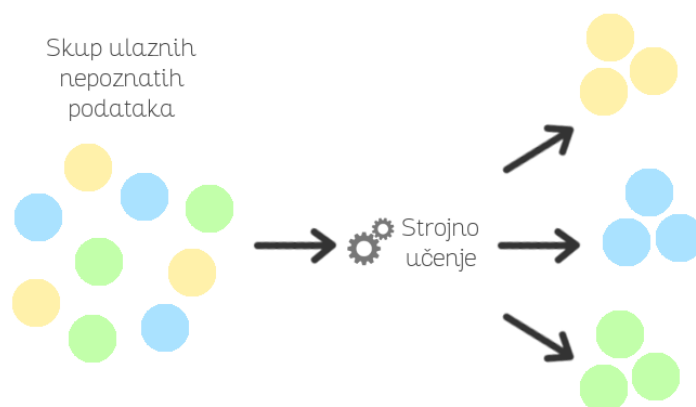


Slika 3: Nadzirano strojno učenje (Izvor: vlastita izrada, 2019)

Algoritmi nadziranog strojnog učenja moguće je dalje grupirati na temelju dva osnovna problema: klasifikacijskog i regresijskog problema. Klasifikacija se odnosi na izlazne varijable koje imaju diskretnu odnosno nebrojčanu vrijednost poput kategorije. Primjer klasifikacijskog problema može biti kategoriziranje bolesnih i zdravih osoba u bolnici. Kod regresije izlazna varijabla predstavlja brojčanu i kontinuiranu vrijednost poput težina osobe izražena u kilogramima, godišnja plaća izražena u kunama, i slično [2].

2.1.2. Nenadzirano strojno učenje

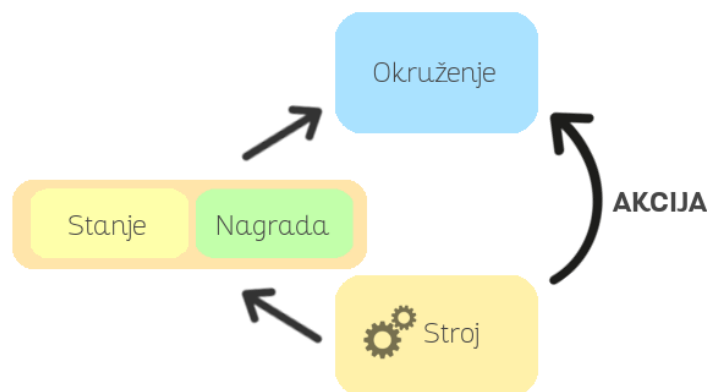
Suprotno nadziranom, kod nenadziranog strojnog učenja dostupan je neobilježen skup podataka koje algoritam pokušava riješiti grananjem obrazaca i značajki. Stroj prilikom nenadziranog učenja nema predviđen model već od ulaznih podataka treba provjeravati njihovu valjanost. Točni podaci nisu poznati te je algoritmima prepušteno vlastito otkrivanje gdje oni sami označavaju strukturu ulaznih podataka. Algoritmima su poznati podaci bez ciljne vrijednosti zbog čega dolazi do problema grupiranja podataka (*engl. clustering*). Problem povezivanja također je prisutan prilikom udruživanja velike količine podataka sličnih značajki [2]. Primjer grupiranja podataka prikazan je na Slici 4.



Slika 4: Nenadzirano strojno učenje (Izvor: vlastita izrada, 2019)

2.1.3. Podržano strojno učenje

Sredina između nadziranog i nenadziranog strojnog učenja predstavlja podržano strojno učenje u kojem se koristi mala količina obilježenih podataka te također podržava veći skup neobilježenih podataka. U ovom obliku strojnog učenja proces analize podataka stvara izlazne varijable koje su naučene iz grešaka (Slika 5.). Rezultat pokretanja algoritma podržanog strojnog učenja u novom okruženju jesu mnogobrojne greške. Međutim, davanjem signala o ispravnosti danog rješenja algoritam može s vremenom prepoznavati razliku između dobrih i loših signala. Na takav način algoritam uči ne ponavljati pogreške na temelju prepoznavanja dobrih signala [5].



Slika 5: Podržano strojno učenje (Izvor: vlastita izrada, 2019)

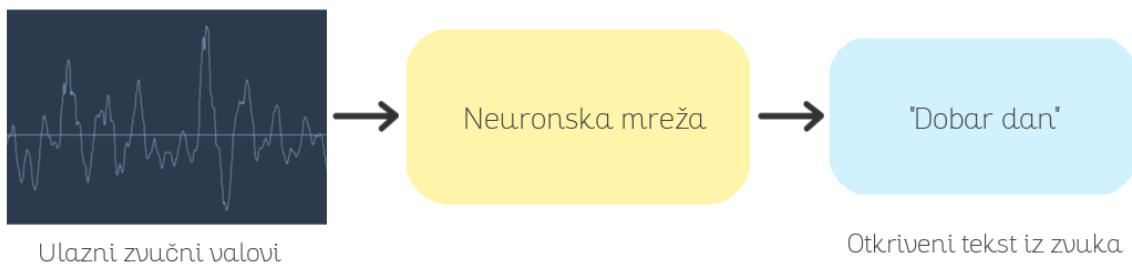
2.2. Slučajevi korištenja strojnog učenja

Implementacija strojnog učenja u aplikacije u današnjici nije nepoznanica. Mnoge tvrtke uvođenjem strojnog učenja najčešće steknu prednost nad trenutnom konkurencijom na tržištu. Stoga postoji mnoštvo slučajeva korištenja strojnog učenja pomoću kojih tvrtke mogu poboljšati poslovanje i znatno smanjiti troškove. Analitika ponašanja korisnika, obrada prirodnog jezika i otkrivanje prijevara neke su od rješenja koje mogu unaprijediti aplikaciju ili cjelokupno poslovanje. U nastavku rada slijede najčešći slučajevi korištenja strojnog učenja s popratnim primjerima.

2.2.1. Prepoznavanje glasa i zvuka

Prepoznavanje glasa i zvuka s godinama ima drastičan napredak u razvoju te predstavlja najpopularniji oblik dubokog učenja. Aplikacije s uslugom prepoznavanja glasa i

zvuka također imaju mogućnost analize osjećaja, otkrivanje grešaka, razumijevanje jezika i slično. Prvi korak prepoznavanja zvuka jest pohrana zvučnih valova u zapis razumljiv računalu. Zapis je u brojčanom obliku te svaki broj označava visinu zvučnog vala u tom trenutku. Zatim se zapis obrađuje u neuronskoj mreži gdje se analizira svaka milisekunda pokušavajući otkriti svako slovo koji odgovara tonu (Slika 6.) [6, str. 6].



Slika 6: Proces obrade prepoznatog zvuka (Izvor: Geitgey, 2016)

Postoje različite primjene prepoznavanja zvuka primjerice glasovno pretraživanje (Google tražilica), virtualna asistencija (Siri, Alexa, Google asistent (Slika 7.), Cortana), pametni zvučnici (Google home) te aplikacije koje se aktiviraju na glas (Uber). Kod virtualnih asistencija uključene su višestruke funkcionalnosti jer sustav, osim što treba prepoznati tekst govora, treba obraditi zatraženi zahtjev te dati odgovor na isti. Od virtualnog asistenta također možemo tražiti da odradi određene akcije poput zakazivanje sastanka, uspostavljanje poziva, postavljanje alarma, pretraživanje na internetu, itd. Svi zvučni zapisi ostaju zauvijek pohranjeni kako bi korisniku mogao u budućnosti davati točnije odgovore.



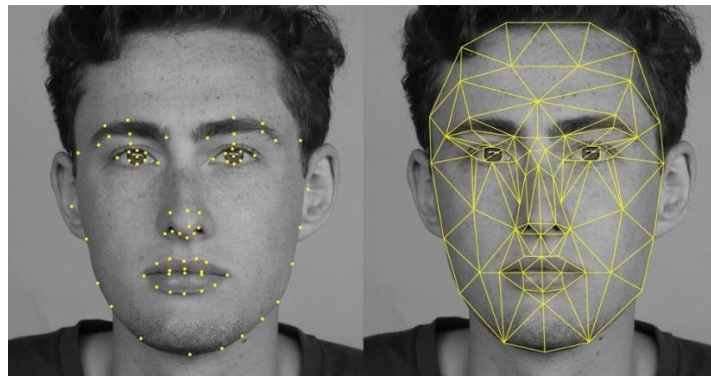
Slika 7: Google asistent aplikacija (Izvor: Google Assistant Guide, 2018)

2.2.2. Prepoznavanje slike ili video zapisa

Prepoznavanje elemenata sa slike ili video zapisa je reprezentativan primjer gdje se strojno učenje primjenjuje kao biometrijsko rješenje koje mjeri jedinstvene značajke otkrivenih objekata ili lica osobe sa slike ili iz video zapisa. Uz prepoznavanje slike nastale su i druge usluge poput prepoznavanje pisanog ili tiskanog teksta, otkrivanje logotipa, itd. U slučaju detekcije video zapisa ono se dijeli u kadrove u kojima se analizira svaka digitalna slika [7].

Danas se sve češće javlja potreba da se automatski prepoznaju osobe na fotografiji ili na video zapisu. Mnogo je razloga zbog kojih mnoge privatne i javne ustanove automatski žele prepoznati osobu na fotografiji ili na video zapisu. Primjerice prepoznavanje osobe sa slike ili video zapisa se sve više počinje koristiti kod prijave na let odnosno u okviru međunarodnih graničnih prijelaza na aerodromima. Također, prepoznavanje osobe sa slike se sve više koristi kada se želi omogućiti pristup povjerljivim poslovnim ili sigurnosnim informacijama samo određenim osobama. Međutim, prepoznavanje osobe sa slike ili video zapisa nije isključivo dostupno određenim skupinama ljudi već ono danas postaje sve više pristupačnije ljudima.

U tom kontekstu može se navesti da se danas korisnici Facebook-a, možda i neprimjetno, susreću s mogućnošću automatskog prepoznavanja osobe sa slike prilikom objavljivanja slika. Slika 8. prikazuje primjer prepoznavanja osobe na temelju slike iz putovnice [8].



Slika 8: Primjer prepoznavanja lica osobe s putovnice (Izvor: Murray, 2017)

2.2.3. Prepoznavanje teksta

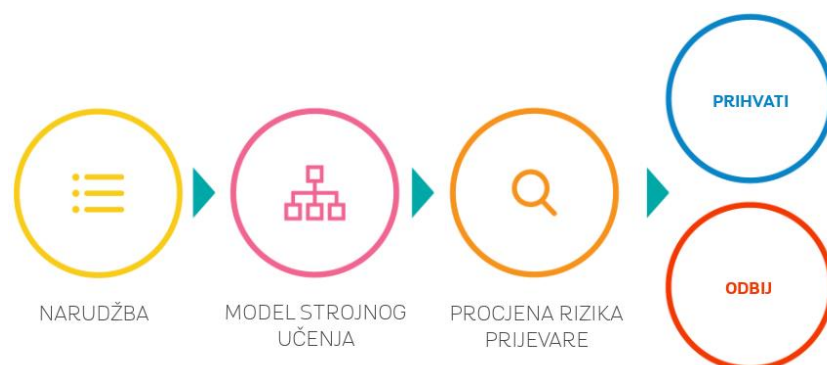
U posljednjih nekoliko godina istodobna dostupnost sve jeftinijih i moćnijih računala, moćnih algoritama i velikih baza podataka doprinijela je brzom napretku u području prepoznavanja teksta. Pri tome su naročito upečatljivi rezultati postignuti na području prepoznavanja rukopisa. Pored navedenih, dodatni razlog strmovitog razvoja mogućnosti

prepoznavanja teksta i rukopisa vezan je uz razvoj dubinskog učenja. Zahvaljujući upravo primjeni dubinskog učenja algoritmi za prepoznavanje rukopisa danas mogu prepoznati bilo koji rukopis, znamenke ili stil pisanja neovisno o korištenom pismu. Stoga se danas prepoznavanje teksta najčešće primjenjuje prilikom analiziranja povijesnih dokumenata, pregleda arhiva ili prilikom digitalizacije ručno ispunjene dokumentacije [9].

Prepoznavanje teksta također omogućuje uslugu otkrivanja jezika te predstavlja najkorišteniju uslugu u aplikacijama koje rade s tekstualnim zapisima. Google Prevoditelj je najpoznatija aplikacija za prevođenje unesenog, otkrivenog ili govorenog teksta. S obzirom na to da Google u svojoj bazi ima preko sto jezika ono također omogućuje otkrivanje na kojem je jeziku napisan tekst. Google je također uz korištenje prepoznavanja teksta razvio novu uslugu sekvenca-za-sekvencu (*engl. sequence-to-sequence – S2S*) te se koristi za rezimiranje teksta. Ova usluga strojnog učenja najčešće služi za generiranje naslova novinskog članka.

2.2.4. Otkrivanje prijevara

Otkrivanje prijevara čest je slučaj korištenja te je prisutan u skoro svim primjerima korištenja strojnog učenja. Neke tvrtke koje već imaju sustav za otkrivanje prijevara uglavnom rade na temelju poznatih obrazaca te dolazi do problema ako se dogodi prijevarena nepoznatog obrasca tada sustav u pravilu ne zna kako reagirati. Mogućnost prijevare prisutna je u svakom sustavu. Stoga se pojavila potreba za razvijanjem sustava za otkrivanje i sprječavanje prijevare uz pomoć strojnog učenja. Da bi otkrivanje prijevare bilo efikasno modeli strojnog učenja moraju analizirati akcije u realnom vremenu te ih onda klasificirati kao ispravnu ili lažnu (Slika 9.) [7]. Uslugu otkrivanja prijevare najčešće koriste tvrtke za financije poput PayPal i MasterCard. Sustavi tijekom provedbe transakcije, izvršene putem mobilnog bankarstva ili kartičnim plaćanjem, provode procjenu rizika prijevare na temelju prošlih kupnji kupca. Lokacija kupca, vrijeme i iznos računa bitni su elementi za analizu prema kojima se također može otkriti prijevarena [10].



Slika 9: Proces procjene i sprječavanje prijevare (Izvor: Kartman, 2019)

3. Primjena strojnog učenja

Strojno učenje radi na temelju algoritama pomoću kojih se obrađuje velik broj podataka. Takvi algoritmi tijekom svog učenja mijenjaju svoj oblik kao što bi čovjek stjecao novo znanje. Pri tome algoritmi imaju mogućnost obrade velikih količina nestrukturiranih podataka pri čemu ih oni oblikuju u smislene podatke i te im daju smislenost. Također, algoritmi imaju mogućnost prilagođavanja svojih parametara s obzirom na povratne informacije zadnje akcije. Uzimajući u obzir navedene prednosti danas postoji mnogo algoritama strojnog učenja za obuku različitih modela predviđanja [11]. Google, Amazon i YouTube neke su od tvrtki koje primjenjuju strojno učenje kako bi predvidjele kakav bi sadržaj dalje mogao zanimati korisnika njihove usluge, ali i za prikazivanje određenih reklamnih oglasa sukladno interesima korisnika usluge. U nastavku rada slijedi opis slučajeva korištenja te opis osnovnih algoritama strojnog učenja. Također su navedeni programski okviri (*engl. framework*) za strojno učenje koji imaju cilj smanjiti složenost implementacije velikih modela dubokog učenja primjerice konvolucijske neuronske mreže.

3.1. Neuronske mreže

Umjetne neuronske mreže predstavljaju skup algoritama koji konstruiraju modele iterativnim postupkom te su oni slični strukturi bioloških neuronskih mreža. Dizajnirani su da prepoznaju obrasce numeričkog oblika u kojima su prevedeni svi podaci iz stvarnog svijeta. Pri tom stroj uz pomoć neuronskih mreža može grupirati i razvrstavati nepoznate ulazne varijable. Dakle, nepoznati podaci se grupiraju prema sličnosti dok se ostali podaci pripremaju za treniranje i obradu. Umjetne neuronske mreže često se koriste kod regresijskih i klasifikacijskih problema. Prilikom rješavanja problema toleriraju se pogreške te stroj može doći do zadovoljavajućeg rješenja i u slučaju obrade nepotpunih podataka. Čvor unutar neuronske mreže, koji se aktivira po potrebi, predstavlja mjesto u okviru kojeg se odvijaju računske operacije [12].

Neuronske mreže implementirane u strojnom učenju imaju ulogu modeliranja podataka koji koriste neurone strojnog učenja. Aplikacije koje su razvijene na temelju neuronske mreže primjenjuju se u različitim područjima, poput zdravstva, financija, stanja u skladištu, maloprodaje i sličnog. Takve aplikacije najčešće imaju funkcionalnosti poput prepoznavanje rukopisa, opisivanja stanja i analize fotografije na temelju vizualnih elemenata, procesiranje računalnog vida itd. K tome umjetne neuronske mreže rade samostalno te tijekom obrade podataka dodjeljuju težine neuronima kako bi izlaz neuronske mreže bio ispravan.

3.1.1. Oblici umjetnih neuronskih mreža

S obzirom na to da se dubinsko učenje značajno razvilo, s vremenom su se umjetne neuronske mreže podijelile u nekoliko oblika. Konvolucijske, ponavljajuće i višeslojne neuronske mreže tri su oblika umjetne neuronske mreže. Tijekom svog dugogodišnjeg djelovanja spomenuti oblici su se pokazali veoma fleksibilnim i korisnim u mnogim slučajevima rješavanja problema. Osnovna razlika između prva dva navedena oblika je u njihovoj primjeni. Kod konvolucijske neuronske mreže obrađuju se slike i videozapisi (poput očiju za prepoznavanje objekata), a kod ponavljajuće neuronske mreže obrađuju se jezični obrasci (poput ušiju i usana za analizu teksta i govora). Višeslojne neuronske mreže poznate su kao višeslojna mreža perceptrona te su dio strojnog učenja pod nadzorom koji se koriste u svrhu rješavanja problema bez povratnih veza [13]. U nastavku rada slijedi detaljni opis spomenutih oblika umjetnih neuronskih mreža.

3.1.1.1. Konvolucijske neuronske mreže

Konvolucijske neuronske mreže (*engl. Convolutional Neural Networks – CNN*) dio su umjetne neuronske mreže koja se bavi prepoznavanjem i označavanjem slika i videozapisa uz pomoć računalnog vida. Dakle, prepoznati elementi sa slikovnih podataka preslikavaju se u izlazne podatke koji se mogu prikazati u tablici ili matrici. Time se rješavaju svi problemi koji imaju ulaze slikovnih parametara i kada je potrebno provesti određeno predviđanje. Primjer korištenja konvolucijskih neuronskih mreža jest automatsko označavanje osoba na slici na Facebook-u (Slika 10.) [14], [15].



Slika 10: Primjer automatskog označavanja osoba korištenjem CNN-a
(Izvor: Facebook, 2019)

Kako bi generalizacija izlaznih podataka bila što točnija primjenjuju se svojstva konvolucijske neuronske mreže. Prvo svojstvo je dijeljenje težina neurona pomoću kojeg mreža može naučiti kompatibilne i različite značajke. Tako se omogućava pristup različitim

podacima unutar klase čime se povećava sposobnost mreže u smislu generalizacije. Sljedeće svojstvo odnosi se na raspršenu povezanost bez koje sve mape mogu poprimati sve vrijednosti iz prethodnih mapa. Stoga je uz pomoć ovog svojstva ostvareno učenje različitih značajki gdje mape poprimaju vrijednosti od samo dvije mape iz prethodnog sloja [16].

3.1.1.2. Ponavljajuće neuronske mreže

Ponavljajuće neuronske mreže (*engl. Recurrent Neural Networks – RNN*) koriste se za obradu problema predviđanja redoslijeda. Najčešće se koristi za obradu prirodnog teksta što uključuje nizove teksta ili snimke govora koje su u obliku vremenskog slijeda. Dakle, kod ovog oblika umjetnih neuronskih mreža podaci ne mogu biti prikazani tablično, kao na primjer u CSV datoteci. S obzirom na to da ovaj oblik neuronske mreže ima mogućnost stvaranje glazbe Google je istu iskoristio za svoj sustav „Google Magenta“. U tom sustavu korisnici mogu stvarati glazbu i umjetnost uz pomoć strojnog učenja koji se temelji na TensorFlow programskom okviru [14].

3.1.1.3. Višeslojne neuronske mreže

Višeslojna neuronska mreža (*engl. Multilayer Perceptrons – MLP*) je oblik umjetne neuronske mreže koji se sastoji od jednog ili više slojeva neurona. Najčešće se koristi na strojnom učenju pod nadzorom jer se nastoji naučiti i modelirati veza između ulaznih i izlaznih varijabli. Model predviđanja ima skriveni i vidljivi sloj. Skrivenih slojeva može biti jedan ili više te oni pružaju razine apstrakcije, dok vidljivi sloj prikazuje izlazne rezultate. Višeslojna neuronska mreža se primjenjuje u slučaju rješavanja problema za predviđanje regresije na temelju predviđanja stvarne vrijednosti. Višeslojna neuronska mreža, kao i ponavljajuće neuronske mreže, također može obrađivati podatke koji nisu u tabličnom obliku kao na primjer tekstualni dokument, slika i vremenski slijed. Nudi visoku fleksibilnost u smislu moguće primjene na različite vrste i oblike podataka [15].

3.2. Regresija

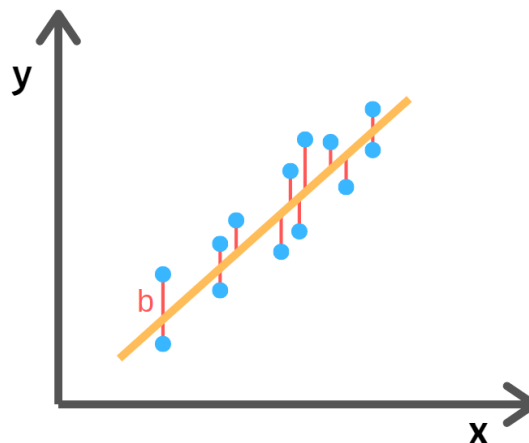
Regresija predstavlja modeliranje odnosa između jedna ovisne varijable y i niza drugih varijabli koje mijenjaju svoj oblik kroz iteracije koristeći sve mjere pogreške u predviđanjima koje je oblikovao model. Proces regresije točnije radi kao metoda statistike te se koristi za statističko strojno učenje [11]. Postoje dvije osnovne vrste regresije koje se razlikuju prema ograničenjima i primjeni. Prilikom modeliranja regresije potrebno je uzeti u obzir sljedeće značajke:

- izbor ovisnih i neovisnih varijabli i njihovog odnosa,
- izostavljene varijable,

- stabilnost regresije te
- ocjena prilagodbe [17].

3.2.1. Linearna regresija

Kao što je već rečeno, postoji veza između ulaznih i izlaznih varijabli prilikom rada strojnog učenja. U algoritmu linearne regresije (*engl. linear regression*) ta veza predstavlja funkciju koja rezultira stvarnim vrijednostima na temelju kontinuiranih varijabli. Pomoću linearne regresije rezultati su prikazani u savršenoj liniji te se postiže sljedećom funkcijom: $y = ax + b$. Varijabla y predstavlja predviđenu vrijednost koju algoritam izračunava ovisno o zadanoj ulaznoj varijabli x . Koeficijent a predstavlja nagib linije dok koeficijent b označava presretanje linije s osi y . Cilj algoritma je uklopiti liniju koja je najbliža točkama (Slika 11.). Tako se smanjuje udaljenost između točaka i regresijske linije što znači da se smanjuje mogućnost pojavljivanja pogreške. Postoje dvije vrste linearne regresije: jednostavna i višestruka linearna regresija. Za razliku od jednostavne linearne regresije gdje postoji jedna ulazna varijabla, kod višestruke linearne regresije postoji više ulaznih varijabli [18].

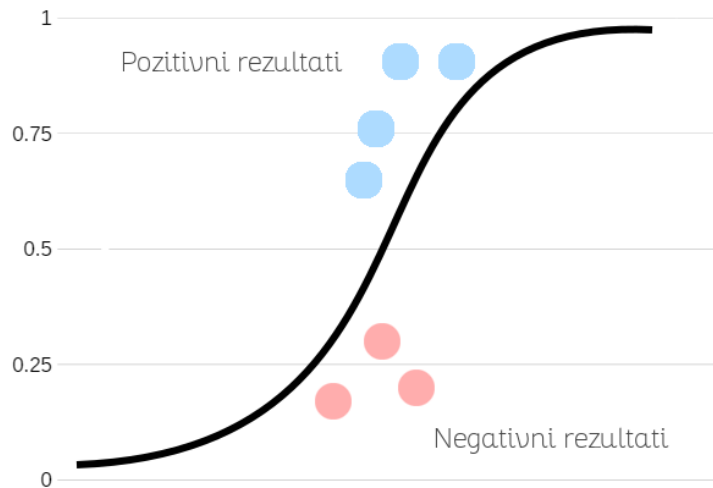


Slika 11: Grafički primjer linearne regresije (Izvor: Shaw, 2019)

3.2.2. Logistička regresija

Drugi oblik regresije je logistička regresija (*engl. logistic regression*) u kojoj predviđanja, nakon oblikovanja modela, imaju diskretne vrijednosti. Vrijednosti su nenumeričkog oblika te su najprikladnije za binarnu klasifikaciju tipa 0 ili 1. Dakle, za logističku regresiju postoji predviđanje za neki događaj koji se može dogoditi (oznaka 1) ili ne može dogoditi (oznaka 0). Primjer logističke regresije je prekidač za svjetlo koji može imati jedno od dva stanja, a to su upaljeno ili ugašeno svjetlo. Pomoću logističke regresije rezultati su prikazani kao krivulja u obliku slova „S“ koja predstavlja kumulativnu logističku distribuciju, a postiže se sljedećom

funkcijom: $h(x) = 1/(1 + e^{-x})$. Točnije, navedenom funkcijom mjeri se odnos između kategorički ovisne varijable te jedne ili više neovisnih varijabli. S obzirom na to da se radi o vjerojatnosti logističkom regresijom, izlazni podaci imaju vrijednosti koje se kreću u rasponu od 0 do 1 (Slika 12.) [18].



Slika 12: Grafički primjer logističke regresije (Izvor: Shaw, 2019)

3.3. Stablo odlučivanja

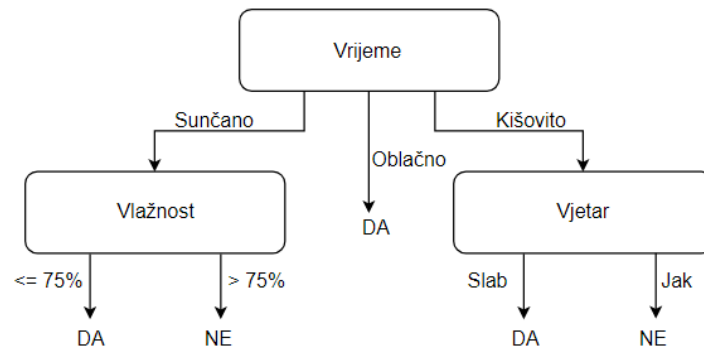
Stablo odlučivanja (*engl. decision tree*) je algoritam nadziranog učenja koji rješava problem klasifikacije. Uz pomoć algoritma stabla odlučivanja konstruira se model odluka na temelju stvarnih podataka. Stablo odlučivanja se sastoji od niza čvorova poput hijerarhije koji tvore stablo kao graf usmjerenja (Slika 8.). Baza grafa započinje jednim čvorom koji se dalje nastavlja po drugim čvorovima koji predstavljaju kategorije potrebne za klasifikaciju stabla. Stablo odlučivanja se pokazao kao praktičan alat za podršku odlučivanju. Razlog tome jesu njegova brzina i točnost [11].

Postoji nekoliko razloga zašto koristiti stablo odlučivanja:

- tumačenje rezultata ne zahtijeva statističko, matematičko ili analitičko znanje,
- omogućava identificiranje značajnih podataka i odnose između njih,
- stabla odlučivanja su otporna na nepotpune vrijednosti,
- ulazne varijable mogu biti u kategorijskom i numeričkom obliku [19].

Danas se stablo odlučivanja najčešće koristi u operativnim istraživanjima, posebno u analizi odluka kojima bi bilo moguće ostvariti strategije koje imaju najveću vjerojatnost

ostvarivanja odnosno postizanja cilja. Zbog navedenog te zbog svoje jednostavnosti, stablo odlučivanja primjenjuje se u različitim industrijama, uključujući građevinarstvo, energetiku, financije, zdravstvo, obrazovanje, pravo itd. Primjer stabla odlučivanja za odlazak na koncert prikazan je na Slici 13.



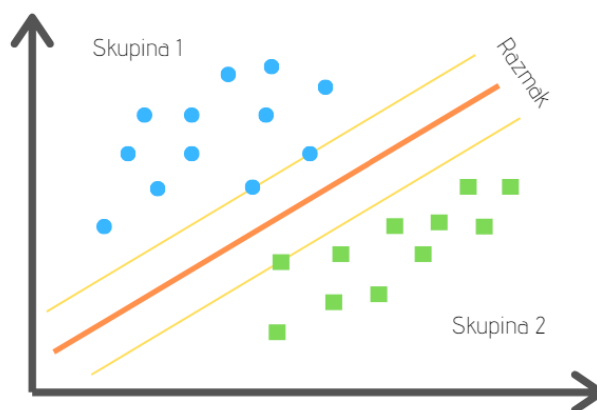
Slika 13: Primjer stabla odlučivanja

3.4. Metoda potpornih vektora (SVM)

Metoda potpornih vektora (*engl. Support Vector Machines – SVM*) je skup algoritama koji se koristi za rješavanje klasifikacijskih i regresijskih problema u okviru strojnog učenja. U svom procesu rada pronalazi liniju koja razdvaja skupove podataka. Pri tome nastoji povećati udaljenost između skupova podataka te se tako onda lakše identificiraju nevidljivi podaci odnosno povećava se vjerojatnost generalizacije (Slika 14.) [20].

SVM algoritmi mogu se podijeliti u dvije skupine: linearni i nelinearni SVM algoritmi. Kod linearnog SVM-a odvojivi podaci imaju čvrstu granicu, dok kod nelinearnog SVM-a odvojivi podaci imaju malu slabu granicu. Prednost SVM algoritma jest zajamčena optimalnost gdje će rješenje uvijek biti globalni maksimum, a ne lokalni minimum. SVM algoritam spada pod algoritme podržanog strojnog učenja što znači da se može primjenjivati i na podacima koji su neoznačeni [21].

Primjena SVM algoritma pokazala se vrlo učinkovitom u području genomike raka. Za klasifikaciju raka moguće je bolest podijeliti na obrasce funkcijskih značajki koji mogu predstavljati tipove raka, dijagnozu, predviđanje koristi od lijekova, itd. U tom slučaju SVM algoritam može biti koristan kako bi se obrasci mogli automatski prepoznavati [22].



Slika 14: Grafički primjer SVM algoritma (Gour, 2019)

3.5. Algoritam za povećanje gradijenta (GBM)

Algoritam za povećanje gradijenta (*engl. Gradient Boosting Machines – GBM*) se koristi u regresijskim i klasifikacijskim problemima te povećava točnost predviđenog modela. Neki od GBM algoritama su sljedeći: XGBoost, AdaBoost, nježno pojačavanje, itd. Pomoću GBM algoritma se stvaraju modeli koji su sastavljeni od više slabijih modela koji imaju samostalnu obuku. Tako se kombiniraju predviđanja kako bi se dobilo sveukupno predviđanje s nekom vrijednosti. Stoga GBM algoritam ponavlja uzorke i jača modele slabog predviđanja te ih pokušava ojačati i poboljšati. Jednom kada algoritam dođe do kraja svoje obrade i kada obrasci nemaju više mogućnost modeliranja, proces se zaustavlja te se uz pomoć algoritma smanjuje gubitnička funkcija te se gubitci svode na minimum [23].

3.6. Programski okviri dubinskog učenja

Tijekom razvoja aplikacije često je poželjno koristiti softverske okvire pomoću kojih se pojednostavljuje implementacija i smanjuje vjerojatnost pojavljivanja pogreške. U svijetu duboke analize podataka postoji mnogo programskih okvira koji omogućuju lakšu implementaciju složenih modela. Pri tome je prilikom odabira okvira potrebno biti oprezan te je potrebno analizirati postavljene prioritete. Neki od najpoznatijih programskih okvira za dubinsko učenje su: TensorFlow, Caffe, Keras, DeepLearning4J i PyTorch [24]. U nastavku rada slijedi kratki opis spomenutih programskih okvira dok je njihova usporedba prikazana Tablicom 2.

Tablica 2: Usporedba programskih okvira za dubinsko učenje

	Podržani programski jezici	Kreator	Značajke
TensorFlow	Python i C++	Tim Google Brain-a	Poznat kao najbolji programski okvir za dubinsko učenje. Koristi se za prepoznavanje i obradu govora, slike i teksta, predviđanje, označavanje.
Caffe	Python i C++	Berkeley AI istraživanje	Koristi se za brzu analizu slike pomoću računalnog vida na temelju konvolucijskih neuronskih mreža.
Keras	Python	Google developer, Francois Chollet	Omogućuje uvoz modela neuronskih mreža putem Karas programskog okvira.
DeepLearning4J	Java	SkyMind, Adam Gibson	Omogućuje obradu velike količine podataka bez utjecaja na brzinu rada. Može obrađivati modele kreirani u Keras-u.
PyTorch	Python	Facebook AI grupa istraživača	Služi kao podrška strojnom učenju te se koristi za brzu i učinkovitu obuku modela dubinskog učenja.

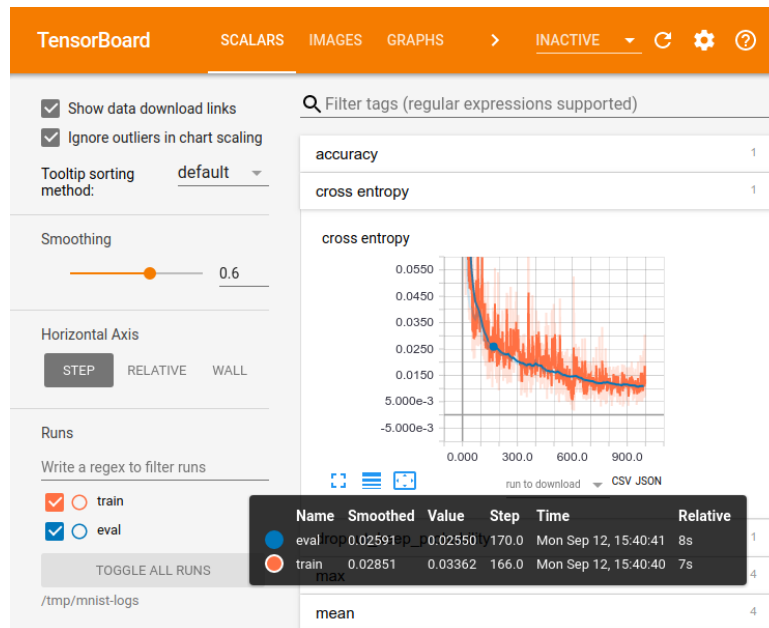
(Izvor: Opala, 2019)

3.6.1. TensorFlow

TensorFlow je programski okvir otvorenog koda kojeg je razvio Google te podržava Python i C++ programske jezike. Koristi se za rješavanje različitih zadataka koji uključuju prepoznavanje govora i elemenata na slici, obradu teksta i predviđanje. Osim toga veoma je koristan za analizu podataka koja uključuje strojno učenje i dubinsku analizu podataka. TensorFlow podržava umjetne neuronske mreže, regresiju i klasifikaciju stoga nudi mogućnost rješavanja složenih problema. TensorFlow je najkorišteniji programski okvir te ga koriste mnoge poznate tvrtke: Uber, Twitter, AirBnB, Dropbox, Intel, DeepMind, itd. Najpoznatiji slučajevi korištenja je u Google prevoditelju, Gmail-u, Google pretraživanju i ostalim Google-ovim proizvodima. U svojoj ponudi ima dva alata: TensorBoard (pruža učinkovitu vizualizaciju i usluge strojnog učenja) i TensorFlow Serving (omogućuje integraciju i implementaciju novih funkcionalnosti i algoritama uz pomoć API-a). Alat TensorBoard daje korisne grafičke vizualizacije modela i mogućnost usporedbe svakog treninga. Primjer provedenog predviđanja s vizualnim prikazom u TensorBoard-u nalazi se na Slici 15.

Dakle, TensorFlow nudi mogućnost integracije te ga to uvelike razlikuje od drugih programskih okvira. Tako se postiže proširenje usluga strojnog učenja u kolaboraciji s drugim programskim okvirima poput Keras-a koji pruža jednostavnije sučelje za učinkovitu izgradnju

neuronskih mreža velikih razmjera s puno slojeva. Također se može izvoditi na raznim poslužiteljima, uređajima, procesorima i grafičkim karticama [24].



Slika 15: Sučelje TensorBoard-a s vizualizacijom rezultata predviđanja

3.6.2. Caffe

Caffe (*engl. Convolutional Architecture for Fast Feature Embedding*) je programski okvir kojeg je razvio Facebook te se koristi za klasifikaciju računalnog vida uz pomoć konvolucijskih neuronskih mreža. Za razliku od TensorFlow-a, koji se bavi razvojem i istraživanjem, Caffe je prikladniji za izradu aplikacija i izgradnju modela temeljeni na slikovnim podacima. Podržava mnoge programske jezike poput C, C++, Python i MathLab. Caffe koristi svoje duboko mrežno skladište pod nazivom „Caffe Model Zoo“ u kojem su dostupne mreže s već provedenom obukom i spremnim podacima za obradu [25].

U tijeku je razvoj nove generacije Caffe programskog okvira pod nazivom Caffe2. Caffe2 je nastao kao integracija dvaju programskih okvira: Caffe i PyTorch. Interoperabilnost između spomenutih okvira rezultirala je modularnim programskim okvirom koji nudi širok raspon usluga umjetne inteligencije. Dakle, rezultat će fleksibilnim korisničkim sučeljem koje pruža PyTorch i primjenom Caffe proizvodne platforme. S obzirom na to da je korist Caffe programskog okvira fokusirana na izradu aplikacija ono se također može integrirati na iOS i Android uređaje [26].

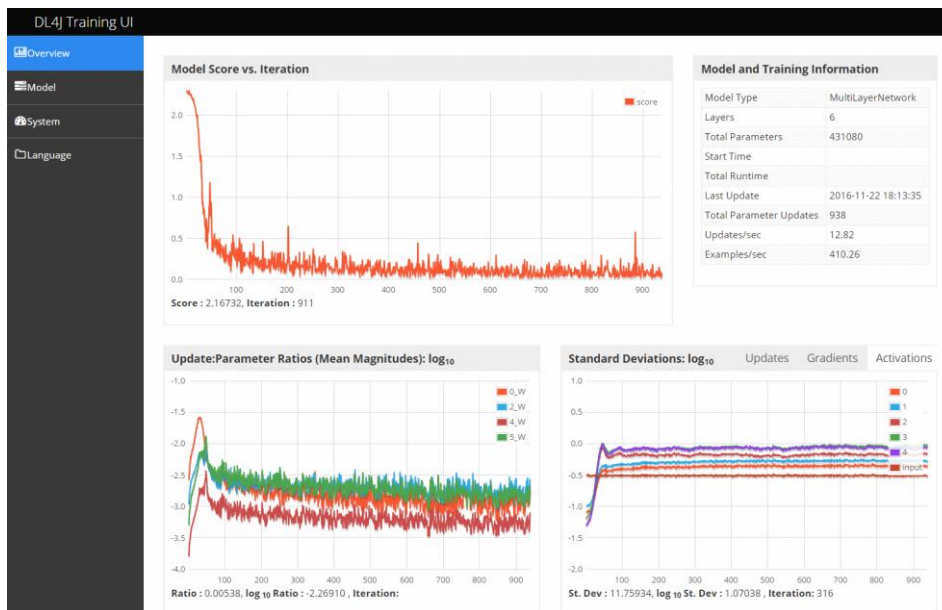
3.6.3. Keras

Keras je programski okvir u Pythonu kojeg je razvio Googleov inženjer kako bi postigao brže i olakšano eksperimentiranje. Njegova primjena poznata je u raznim slučajevima korištenjima primjerice kod prepoznavanja teksta, govora, slike, video zapisa i slično. Keras podržava oblike umjetne neuronske mreže (konvolucijske i ponavljajuće neuronske mreže) koje se mogu pokretati na TensorFlow programskom okviru. Točnije, Keras je moguće koristiti kao modul TensorFlow-a pomoću kojeg se pospješuje razvoj modela dubokog učenja na više slojeva. Time se smanjila razlika između programskih okvira te se postigla učinkovita konstrukcija neuronskih mreža. Implementacija Keras programskog okvira u TensorFlow-u također je za izradu modela dubokog učenja poboljšao API koji je jednostavniji i intuitivan za korištenje. Ako promatramo razinu apstrakcije na kojoj djeluje Keras možemo ga razlikovati s djelovanjem PyTorch programskog okvira. Keras u osnovi ima visok nivo zaštite slojeva dubokog učenja, dok PyTorch korisnicima nudi slobodu mijenjanja slojeva i pristup podacima istih [24], [27]. Interoperabilnost Keras-a moguća je i s DeepLearning4J programskim okvirom čija je implementacija objašnjena u nastavku rada.

3.6.4. DeepLearning4J

Od svih navedenih programskih okvira DeepLearning4J je jedini okvir koji podržava Java programski jezik i nije dostupan u Pythonu. DeepLearning4J je programski okvir otvorenog koda koja predstavlja Java biblioteku za dubinsko učenje. Velika prednost ovog programskog okvira jest sposobnost obrade velike količine podataka bez utjecaja na brzinu rada. Kao i većina programskih okvira, DeepLearning4J također nudi podršku za različite vrste umjetnih neuronskih mreža. Među svojim uslugama uključuje prepoznavanje slika, obrada teksta i govora, te otkrivanje prijevara [24]. Za vizualizaciju obrađenih podataka koristi se DL4J Training UI sučelje (Slika 16.).

DeepLearning4J može se upravljati putem Apache Hadoop-a ili Spark-a koji omogućuju obradu i distribuciju velikih skupova podataka. S obzirom na to da nije dostupan u Python programskom jeziku, integrirana je interoperabilnost s Keras programskim okvirom. Keras konfigurira i trenira modele neuronskih mreža u Pythonu te ih je moguće uvoziti u DeepLearning4J na temelju kojih se provodi analiza. Razlog ove integracije je potreba isporuke modela u proizvodnju nakon završetka projekta unutar Keras-a [28].



Slika 16: Sučelje za vizualizaciju rezultata u DeepLearning4J programskom okviru

3.6.5. PyTorch

Veliki konkurent TensorFlow-u predstavlja programski okvir PyTorch koji je s vremenom, zbog svoje intuitivnosti, dobio veliku pažnju od strane znanstvenika i programera. Kreirao ga je Facebook, a također ga koriste Twitter, Salesforce i mnogi drugi. PyTorch razvijen je u Python i C programskim jezicima, a koristi se za brzu i efikasnu obuku modela strojnog učenja. Među spomenutim programskim okvirima PyTorch se pokazao najjednostavnijim zbog svoje arhitekture koja je dosta pridonijela transparentnom i jednostavnom procesu modeliranja. Kao i mnogi programski okviri PyTorch također podržava konvolucijske i ponavljajuće neuronske mreže. U usporedbi s TensorFlow-om, PyTorch korisnicima nudi slobodu pisanja prilagođenih slojeva i pristup podacima za optimizacijske zadatke. Amazon Web Services u svom alatu SageMaker ima mogućnost integracije s PyTorch okvirom u namjeri da se kreira prilagođeno radno okruženje [24], [27]. PyTorch službeno nema alat za vizualizaciju modela stoga je integracija s drugim sustavima neizbježna.

4. Servisi strojnog učenja

Iako se posljednjih godina sve češće spominje pojam strojnog učenja (*engl. machine learning*) ono je sa svojim razvojem počelo još prije 50 godina. S napretkom tehnologije raste i količina podataka koja se svakodnevno obrađuje, analizira i sortira te se time počinje stvarati ozbiljan problem za performanse rada računala. Uz pomoć strojnog učenja računalo ima sposobnost učenja te tako samostalno poboljšava programe bez dodatnog programiranja. Mnoga poduzeća nude usluge strojnog učenja, a neka od njih su Microsoft, Amazon, Google, Oracle, IBM, itd. [6].

4.1. Usluge u oblaku

Organizacija podataka u tvrtkama predstavlja važan segment za njeno poslovanje. Stoga je bitno osigurati pristup podacima na jednostavan i siguran način. Tradicionalni pristup podacima postupno se smanjuje upravo zbog svoje nefleksibilnosti jer su sadržaji i programi instalirani lokalno na uređajima korisnika odnosno zaposlenika. Osim toga nadogradnja računalne opreme se pokazala veoma skupom i složenom. Također razmjena podataka bila je gotovo nemoguća s obzirom na to da se radi o lokalno spremljenim podacima.

Zbog navedenih razloga počelo se težiti k centralizaciji podataka gdje se svi sadržaji, odnosno podaci, nalaze na jednom mjestu. Tu mogućnost upravo nam osigurava oblak (*engl. cloud*) gdje je prisutan kolaborativan rad nad dokumentima i podacima u realnom vremenu. To znači da se programima, podacima i evidencijama može pristupiti s različitih uređaja, u bilo koje vrijeme i s bilo koje lokacije, a sve što je potrebno je upravo Internet. Ono što je također važno napomenuti jest da su prisutni kontrolirani korisnički pristup i sigurnost podataka unutar oblaka. Točnije, bilježi se i kontrolira svaka promjena datoteke. Rizik od gubitka podataka i datoteka je vrlo mali, a po želji korisnik usluge može zatražiti njihov izvoz.

S obzirom na ograničavanje pristupa sustavu infrastruktura oblaka se ne kupuje već iznajmljuje. Točnije, prisutna je skalabilnost što znači da korisnik usluge plaća iznos u ovisnosti o broju usluga koje koristi. Pri tome je najčešće primjenjiv model poslovanja koji se temelji na davanju licenci za korištenje određenih usluga. Ono što je također potrebno uzeti u obzir jest korisničko iskustvo koje se pokazalo veoma značajnim. Tako se dobije točan dojam o jednostavnosti usluge ili programa za svakodnevno korištenje odnosno koliko su oni zapravo korisni [29].

4.2. Modeli u oblaku

Kako bi se zadovoljile različite potrebe korisnika s vremenom su se pojavili različiti modeli i strategije njihovog uvođenja. Pri tome je svaki model različit po upravljivosti, načinu korištenja i funkcionalnosti. Pružatelji usluga u oblaku nude tri osnovna modela prema Nacionalnom institutu za standarde i tehnologiju (*engl. National Institute of Standards and Technology – NIST*):

- Softver kao usluga (*engl. Software as a Service – SaaS*),
- Platforma kao usluga (*engl. Platform as a Service – PaaS*) i
- Infrastruktura kao usluga (*engl. Infrastructure as a Service – IaaS*).

SaaS se koristi ako je potrebna usluga koja se pokreće na infrastrukturi oblaka nad kojom nije moguća kontrola poput mreže, operacijskih sustava i sličnog. Kontrola pojedinačnih dijelova aplikacije ovisi o korisničkim konfiguracijskim postavkama.

Za implementaciju aplikacije (SaaS proizvoda) koja je razvijena od strane programera pomoću prikladnih biblioteka i alata, a koje podržava davatelj usluga, koristi se PaaS. U ovom slučaju korisnik također nema pristup temeljnoj infrastrukturi oblaka već samo konfiguracijskim postavkama potrebne za implementaciju aplikacije [30]. PaaS koristi usluge IaaS-a tako što je izgrađen na njemu te koristi njegove alate za implementaciju i razvoj. Korisnici PaaS-a svoje programe razvijaju bez instaliranja alata na računalo već to rade online na platformi. Osim što dijele jednu razvojnu platformu također su uključene provjera autentičnosti i autorizacija [31].

Oblik računalstva u oblaku koji pruža usluge virtualizacije računalnih resursa je IaaS. Pružatelj IaaS-a nudi niz usluga koje su vezane za infrastrukturne komponente uključujući servere, mrežni hardver i hardver za pohranu te slojeve virtualizacije. Korisnici na IaaS-u mogu implementirati veće razine automatizacije (npr. virtualni stroj) dok resursima pristupaju putem Interneta [32].

Osim navedenih modela postoje i drugi oblici rješenja:

- Strojno učenje kao usluga (*engl. Machine Learning as a Service – MLaaS*),
- Podaci kao usluga (*engl. Data as a Service – DaaS*),
- Mobilni backend kao usluga (*engl. Mobile Backend as a Service – MBaaS*),
- Mreža kao usluga (*engl. Net as a Service – NaaS*),
- Sigurnost kao usluga (*engl. Security as a Service – SECaaS*),
- Funkcija kao usluga (*engl. Function as a Service – FaaS*), itd. [29].

Navedeni oblici svojom funkcionalnošću nadopunjuju osnovne modele. Na primjer, DaaS ima ulogu distribucije informacija te podatkovne datoteke čini dostupnima korisnicima putem Interneta i zalaže se za SaaS. Točnije ono pruža brži rad s velikom količinom podataka što uključuje integraciju, analizu i objavu istih u stvarnom vremenu koristeći RESTful API web servis (sučelje aplikacijskog programa koji koristi HTML zahtjeve za rad s podacima). Zatim pomoću MBaaS razvojni inženjeri svoje mobilne aplikacije povezuju s podlogom u oblaku također koristeći API. Uzimajući u obzir da već danas postoji velik broj modela oblaka može se zaključiti te da će se broj istih povećati s obzirom na brzinu njegovog razvoja [33].

4.3. Pružatelji usluga strojnog učenja

Kao što je ranije spomenuto, u računalnom sustavu strojno učenje je dio umjetne inteligencije u kojem se obrađuje velika količina podataka. Za obradu tih podataka koriste se različiti pružatelji usluga strojnog učenja. Strojno učenje kao usluga kod svih pružatelja na početku nudi besplatno korištenje kako bi korisnici mogli isprobati njihove performanse te tako procijeniti korisnost usluge prije odluke o pretplati. S obzirom na to da se radi o modelu u oblaku ovakav sustav nije potrebno instalirati na računalo niti pokretati vlastite poslužitelje. K tome tvrtke koje žele primjenjivati strojno učenje često zadovoljava činjenica da mogu imati uspostavljenog pružatelja usluga pomoću kojeg se minimiziraju prijelazna pitanja što pruža određenu sigurnost kod donošenja odluka [34].

S obzirom na to da je većini poduzeća često dobrodošla primjena novih tehnologija, među njima ne postoji bojazan te se danas one najčešće odlučuju na prelazak na rad u oblaku gdje mogu razvijati vlastite aplikacije. Stoga poduzeća svoju infrastrukturu s vlastitim poslužiteljem mijenjaju s konfiguracijama javnog oblaka. Stalna dostupnost podataka, konstantna ažurnost i bolja sigurnost neke su od prednosti koje predstavljaju vrijednost za poduzeća koja posluju na infrastrukturi oblaka. Razvojni timovi tako mogu ostati usredotočeni na razvoj aplikacije, a ne trošiti vrijeme radeći na infrastrukturi za podršku aplikaciji koju je potrebno kontinuirano ažurirati i održavati.

Na tržištu su trenutno tri vodeća pružatelja usluga u platformi oblaka, a to su: Amazon Web Services, Google Cloud Platform i Microsoft Azure. Osim navedenih sustava postoje i ostali mnogobrojni sustavi koji nude usluge strojnog učenja, kao na primjer: IBM, Oracle, itd. Kako bi poduzeće donijelo uspješnu odluku o prelasku na jednu od spomenutih platformi potrebno je uzeti u obzir veličinu i ciljeve tvrtke, te očekivanu stopu rasta i proračun. Razlog tome jest različitost sustava točnije platforme oblaka koje ne rade na isti način. Spomenuta tri pružatelja usluga u platformi oblaka u svojoj ponudi imaju različite značajke koje treba provjeriti i tako potvrditi kompatibilnost s poslovnim modelima unutar poduzeća. K tome Amazon Web

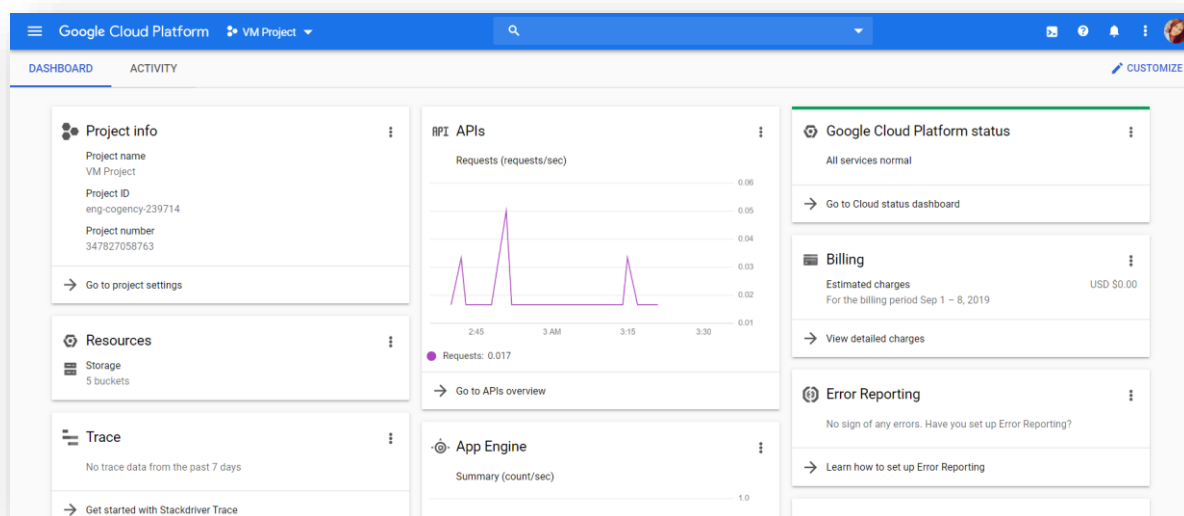
Services je već dulji niz godina vodeća platforma dok Google Cloud Platform i Microsoft Azure neprestano rade na razvoju i rastu. U nastavku rada slijedi opis spomenutih pružatelja usluga strojnog učenja te se navode njihove različitosti.

4.3.1. Google Cloud Platform

Google Cloud Platform (Slika 17.) u svojoj ponudi ima velik broj usluga za programere te se za razvoj poslovnih aplikacija koristi App Engine. Kao i kod ostalih sustava programeri u Google Cloud-u ne moraju brinuti o poslužitelju te se mogu posvetiti provedbi projekta. U Google Cloud Platform-i također postoji mogućnost pohrane i pristupa bazi podataka, računanja, umrežavanja i obrade istih. Google Cloud ima znatno manje usluga nego njena konkurencija, međutim izrazito je popularna te se ona uglavnom koriste za razvoj mobilnih aplikacija [35].

Google Cloud Platform-a prvenstveno služi za primjenu i skaliranje aplikacija i usluga na dostupnoj infrastrukturi koju razvija Google. Prema tome Google Cloud Platform-a pruža računalno okruženje bez poslužitelja te IaaS i PaaS modele oblaka. Nadalje, Google nudi prvu godinu besplatnog korištenja poput ostalih sustava kako bi korisnik mogao kvalitetno naučiti o sustavu te, ovisno o djelatnosti kojom se bavi, odlučiti je li želi nastaviti koristiti uslugu i plaćati pretplatu [36].

Nedostatak Google Cloud Platform-e jest da još uvijek nema mogućnosti sigurnosne kopije, dok ostali sustavi uglavnom imaju tu funkcionalnost. Zato je sigurnosna zaštita podataka visoko razvijena od šifriranja podataka do više provjera autentičnosti i provjere treće strane [35].



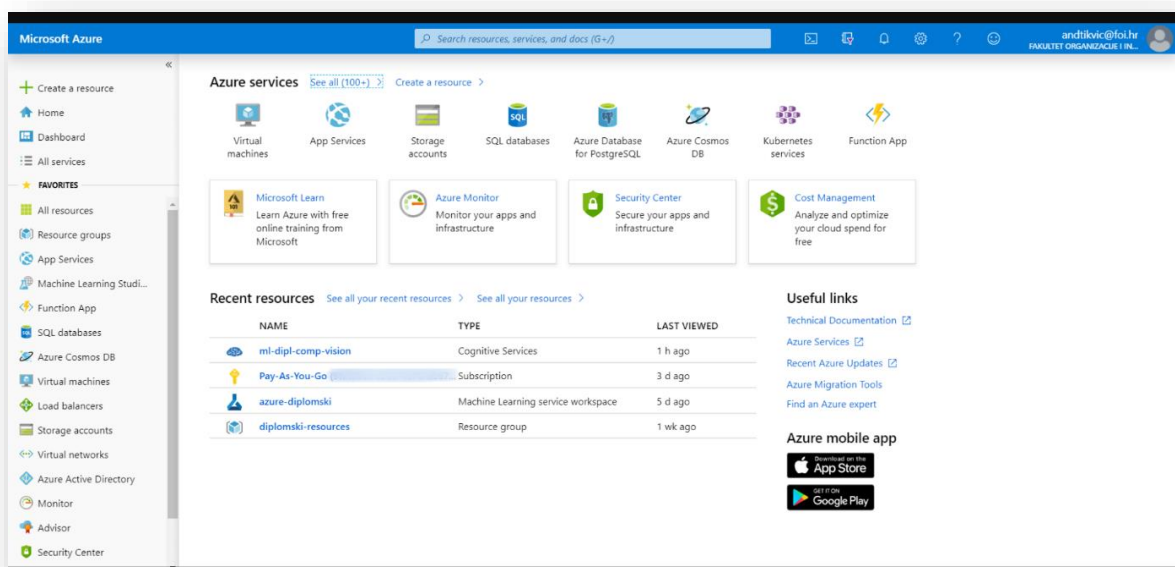
Slika 17: Početna stranica Google Cloud Platform konzole

4.3.2. Microsoft Azure

Kao veliki konkurent javlja se Microsoft Azure koji predstavlja čvrstu integriranu platformu za tvrtke koje su bazirane na Windows operacijskim sustavima. Microsoft Azure nudi široku ponudu usluga za potrebe razvoja aplikacija od strane programera, slično kao kod Amazon Web Service-a. Kod Azure-a veliki dio usluga odnosi se na upravljanje virtualnog stroja koji ima mogućnost brzog izračunavanja i obrade visokog kapaciteta podataka u bilo kojem trenutku. Integriranjem Azure-a u postojeći poslovni model i sustav poduzeća donosi veće i bolje rezultate u vidu razvoja proizvoda [35]. Iako je Azure uvijek funkcionirao kao infrastruktura kao usluga (IaaS) ono također podržava pokretanje poslužiteljskih programa u raznim programskim jezicima poput: Java, PHP, Python, Node.js i .NET. Ovaj sustav je također poznat kao najjednostavniji sustav u oblaku u kontekstu konfiguracije i rada zajedno s uslugama pomoću kojih se poboljšava produktivnost prateći najnoviju tehnologiju [37].

Ako razmotrimo na cijenu korištenja Microsoft Azure-a ona se uvelike razlikuje od primjerice cijena korištenja Amazon Web Service-a. Kod ove platforme cijena ovisi o vrsti proizvoda kojeg je potrebno koristiti te se onda izračunava prema satu korištenja.

Azure je po svemu sudeći platforma s najbrže dostupnim rješenjima u oblaku. Navedeno se odnosi prvenstveno na brzinu implementacije i kvalitetu rada. Također spada među sustave koji nude veći broj sigurnosnih rješenja uključujući i arhivsku pohranu. Uzimajući u obzir sve navedeno može se zaključiti da tvrtkama kojima brzina predstavlja glavnu značajku Azure predstavlja najbolji izbor za integraciju u postojeći poslovni sustav [35]. U nastavku rada Slika 18. prikazuje početnu stranicu Microsoft Azure konzole.

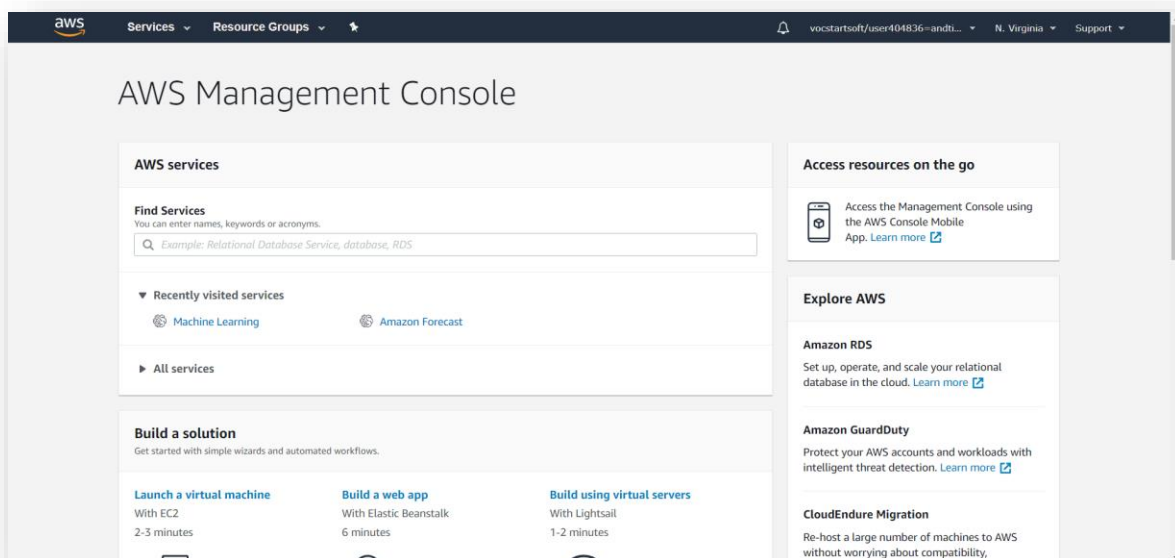


Slika 18: Početna stranica Microsoft Azure konzole

4.3.3. Amazon Web Services

Amazon Web Services (AWS) predstavlja platformu za web hosting u oblaku koju razvoja Amazon. AWS nudi sve značajke u industriji računalstva u oblaku gdje je prisutna kombinacija modela oblaka: infrastruktura kao usluga (IaaS), platforma kao usluga (PaaS) i softver kao usluga (SaaS). AWS posjeduje mnoge usluge koje su potrebne programerima za razvoj aplikacije kao što je lak pristup računalnoj snazi, pohrani podataka i slično. Isto tako posjeduje usluge vezane za mobilne i aplikativne servise te alate za upravljanje.

Cijena korištenja AWS-a vrlo je konkurentna s cijenama ostalih poslužitelja usluga oblaka. Od samog početka rada (2006. godine) AWS je postupno prilagođavao (smanjivao) cijene zbog konkurencije i potražnje na tržištu. Svaki korisnik ima pravo na godinu dana besplatnog korištenja kako bi mogao kvalitetno testirati sustav te donijeti odluku o daljnjem korištenju odnosno o pretplati. Cijena se izračunava na temelju vremena korištenja u sekundama, a ne prema satu. Tako se dobiva razumna cijena korištenja infrastrukture koja je dosta niža od cijene kada biste ulagali u vlastitu infrastrukturu. Slika 19. prikazuje početnu stranicu Amazon Web Services konzole.



Slika 19: Početna stranica Amazon Web Services konzole

Nova funkcionalnost Amazon Web Services-a jest SageMaker koji predstavlja nadogradnju usluge strojnog učenja. SageMaker omogućava razvojnom timu brzu instalaciju krajnje točke (*engl. endpoint*) koja je odmah spremna za rad. Programerima tako ne oduzima vrijeme za izgradnju i održavanje složenih podataka već se mogu usredotočiti na razvoj projekta na kojem rade.

Dok Google Cloud nema niti jedno sigurnosno rješenje, Microsoft Azure više njih, AWS ima jedno pod imenom „*Glacier*“ te uspješno radi prilikom oporavka podataka. Prednost AWS je također konstantno poboljšanje platforme gdje i sami razvojni timovi mogu nadograđivati iskustva i znanje [35].

4.3.4. Oracle

Oracle-ov sustav „Oracle Machine Learning“ nije razvijen koliko i prethodne platforme no u svojoj ponudi ima nekoliko usluga strojnog učenja. Svoje usluge nudi u različitim programskim jezicima te je u tijeku uključivanje novih jezika (trenutno je to Python). Mnogi znanstvenici, programeri i analitičari su bili u mogućnosti bržeg rješavanja projekata koristeći usluge Oracle strojnog učenja. Obradom podataka minimizira se kretanje podataka odnosno povećava se skalabilnost, ubrzava implementacija i održava sigurnost [38].

4.3.5. IBM

IBM nudi potpunu uslugu oblaka u svom sustavu pod nazivom „IBM Watson Machine Learning“. U katalogu usluga nude izgradnju modela i neuronskih mreža na temelju vlastitih podataka te ih se može upotrebljavati i implementirati u aplikaciju. S obzirom na to da se radi o uslugama strojnog učenja nudi se izgradnja i automatizirana obuka modela kao i kod prethodno spomenutih platformi. Proces konstruiranja modela provodi se u nekoliko koraka: definiranje ciljeva projekta, priprema podataka, odabir alata, obuka modela te puštanje modela u rad. IBM-ov sustav za strojno učenje u svojoj ponudi ima nekoliko alata za obrađivanje podataka. Jedan od alata naziva se „AutoAI“ u kojem se podaci unaprijed automatski obrađuju i procjenjuju. Ostali alati rade na sličan princip uz moguće grafičke prikaze koji pomažu kod bolje interpretacije rezultata obuke modela [39].

5. Interoperabilnost

Računarstvo u oblacima postalo je danas sveprisutnija tema, kako u okviru akademskih krugova, tako i u industriji [40]. Prvenstveno zbog mogućnosti financijskih ušteda, sve veći broj poslovnih organizacija danas razmatra korištenje računarstvo u oblacima. Međutim, trenutno postoje problemi koji otežavaju širu primjenjivost računarstva u oblacima. U tom kontekstu Tikvica i Andročec [41] navode da je jedan od najznačajnijih problema računarstva u oblacima nedostatak interoperabilnosti. Glavni problemi interoperabilnosti platforme kao servisa detaljnije su opisani u radu "*Methodology for Detection of Cloud Interoperability Problems*" [42]. U okviru navedenog rada problemi interoperabilnosti su klasificirani korištenjem ontologije u sljedeće kategorije: problemi interoperabilnosti aplikacijskih programskih sučelja, problemi interoperabilnosti na aplikacijskoj razini, pravni problem interoperabilnosti, organizacijski problem interoperabilnosti te problemi interoperabilnosti spremišta podataka u oblacima.

Interoperabilnost se može opisati kao mjera do koje različiti sustavi mogu međusobno komunicirati te uspješno surađivati (Slika 20.). U tom kontekstu međunarodne organizacije za standardizaciju IEEE i ISO definiraju interoperabilnost kao sposobnost dva ili više sustava ili aplikacija da razmjenjuju informacije te ih dalje primjenjuju. Nadalje, u kontekstu računarstva u oblaku, interoperabilnost bi se trebala smatrati i kao sposobnost javnih, privatnih i ostalih vrsta oblaka unutar poduzeća da razumiju aplikacijsko okruženje, njegovo sučelje, konfiguraciju, itd.



Slika 20: Slikoviti prikaz interoperabilnosti sustava (Izvor: vlastita izrada, 2019)

Prilikom razmatranja interoperabilnosti između dva sustava korisno je imati model razmjene podataka. Razlog tome jest činjenica da ima više mogućih aspekata razmjene koji se tiču povezivanja sustava. Sukladno tome postoji i više modela interoperabilnosti od kojih

ćemo izdvojiti model koji je usko povezan s korisničkim korištenjem usluga oblaka. Spomenuti model je podijeljen na četiri razine navedene u okviru Tablice 3. te u nastavku ukratko pojašnjene [43].

Tablica 3: Razine interoperabilnosti

Razina	Svrha	Objekt	Rješenje
Tehnička	Siguran prijenos podataka	Signali	Protokol prijena podataka
Sintaktička	Obrada primljenih podataka	Podatak	Standardizirani formati prijena podataka
Semantička	Obrada i tumačenje primljenih podataka	Informacija	Zajednički direktoriji, podatkovni ključevi
Organizacijska	Automatsko povezivanje procesa između različitih sustava	Procesi (tijek rada)	Standardizirani procesni elementi

(Izvor: Kubicek i Cimander, 2019)

Pri tome se tehnička interoperabilnost bavi uspostavljenim standardom kao što su TCP/IP i REST HTTP protokol odnosno protokolima za razmjenu informacija. Nadalje, prilikom prijena podataka bitno je obratiti pažnju na njihov format. Primjerice, on može biti XML strukture podataka ili JSON podatkovni tok, no tome se brine sintaktička interoperabilnost. Kod semantičke interoperabilnosti, osim što obrađuje primljene podatke, ono im također dodjeljuje značenje i strukturu. Posljednja organizacijska interoperabilnost se odnosi na odgovarajuće standarde prema kojima sustav slanja očekuje da će sustav primatelja razmijeniti podatke. Također, potrebno je navesti da se sustavi se mogu razlikovati na razini sintaktičke interoperabilnosti tako što jedan sustav primjerice koristi XML, a drugi sustav JSON datoteke [43]. Pri tome bi idealni scenariji bio kada bi dva ili više sustava ostvarivali uspješnu komunikaciju na temelju sve četiri navedene razine interoperabilnosti.

Prilikom razvoja aplikacije koja koristi strojno učenje potrebno je odabrati odgovarajućeg pružatelja usluga strojnog učenja u oblaku. No postoje slučajevi kada jedan sustav ne nudi sve potrebne usluge koje bi zadovoljile želje korisnika. Prema tome, ako želimo koristiti usluge različitih sustava potrebno je implementirati interoperabilnost između njih. Sustavi u oblaku omogućuju suradnju jer u svojoj ponudi imaju API-e koji koriste iste standarde. Na taj način dobivamo aplikaciju u kojoj se sustavi upotpunjuju i uspješno surađuju.

6. Praktični dio rada

U okviru rada razvijena je Java web aplikacija koja implementira usluge strojnog učenja dvaju sustava: Google Cloud Platform i Microsoft Azure. Ideja praktičnog dijela jest prikazati mogućnost interoperabilnosti spomenutih platformi. Aplikacija je razvijena u Java programskom jeziku u NetBeans alatu koristeći tehnologije poput: Java Server Faces, REST API i Apache Tomcat. Kao što je ranije navedeno obje izabrane platforme nude širok katalog usluga u domeni strojnog učenja. Za potrebe razvoja aplikacije u okviru praktičnog dijela rada koriste se usluge detekcije teksta uz pomoć Google Cloud Platform-e te usluge prijevoda teksta uz pomoć Microsoft Azure-a.

6.1. Korištene tehnologije

Za potrebe razvoja programskog rješenja korišten je Java programski jezik u NetBeans alatu. U okviru Java programskog jezika program je implementiran Java Server Faces (JSF) programskim okvirom zbog toga što radi na principu slanja i primanja HTTP zahtjeva/odgovora te kako bi se primijenio MVC (*Model-View-Controller*) uzorak. Kako bismo mogli pokretati program koristi se Apache Tomcat koji predstavlja poslužitelj web aplikacija. S obzirom na to da se u programu radi sa sustavima u oblaku jasno je da se njihova implementacija vrši putem njihovih aplikacijsko programskih sučelja (*engl. Application Program Interface – API*) te u ovom slučaju putem REST API-ja. U nastavku rada slijedi kratki opis spomenutih te primijenjenih tehnologija.

6.1.1. Java

Java je objektno-orijentirani programski jezik koji je svoju popularnost postigao zbog svoje suvremene koncepcije i univerzalnosti što podrazumijeva da ga se može pisati u bilo kojem operacijskom sustavu. Ovaj programski jezik kreiran je od strane tima iz tvrtke Sun Microsystems čiji je glavni član James Gosling početkom 90-ih a prva verzija je izašla u javnost 1996. godine [44].

Programeri veliku korisnost Jave pronalaze zbog njenih popularnih razvojnih okruženja, kao na primjer Java Server Faces koji koristi Java Servlet za implementaciju kontrolera (*engl. controller*) koji upravlja centralizacijom zahtjeva. Uz pomoć Jave može se postići velika raznolikost programa otvorenog koda, programskih platformi i API-ja. Oblak daje mogućnost rada u Java okruženju gdje programeri mogu razvijati svoje aplikacije sukladno skalabilnosti te ih nadzirati u platformama kao što je to na primjer Google Cloud. Danas je Java izrazito

snažna na tržištu pametnih uređaja gdje više od polovice mobilnih telefona u svijetu radi na Android sustavu [45].

Nadalje, postoji mnogo ostalih značajki zbog kojeg je programski jezik Java uspio i postao popularan. Neke od tih značajki su sljedeće:

- objektno-orijentiran programski jezik,
- pouzdan,
- jednostavan za korištenje,
- sigurnost podataka,
- jednostavan za naučiti,
- fleksibilan.

Kao prva značajka ponovno se spominje objektno-orijentirano programiranje pomoću kojeg se smanjila vjerojatnost pojave pogrešaka od strane programera što ga čini pouzdanim. Java je nastala kao pojednostavljeni C++ programskog jezika jer je bio složen te su funkcionalnosti bile nedovoljne za Java zahtjeve. Tako su se iskoristile ideje C++ programskog jezika te je Java tako jačala i postajala jednostavnija za korištenje i učenje. Izvorno je Java bila namijenjena za male uređaje odnosno mobilne uređaje koji bi prenosili podatke putem mreže zbog čega je bila uključena visoka razina sigurnosti. Također nudi fleksibilnost kod web preglednika jer se Java Applet-i izvršavaju na klijentskoj strani kako bi se brzo pokretao [44].

6.1.2. NetBeans

NetBeans predstavlja integrirano razvojno okruženje (*engl. Integrated Development Environment – IDE*) koji je besplatan za korištenje i baziran je na Java programskom jeziku. Iako je NetBeans IDE nastao za Java development, ono također podržava PHP, C++, HTML5, Groovy te JavaScript i JavaFX. Također, NetBeans se može pokretati na bilo kojem operacijskom sustavu koji podržavaju Java virtualne strojeve (*engl. Java Virtual Machine – JVM*) [46].

6.1.3. Apache Tomcat

S obzirom na to da se radi o web aplikaciji za njeno pokretanje potreban je server. Pomoću alata Apache Tomcat možemo pokretati web poslužitelje kojeg je razvio Apache Software Foundation (ASF). U Tomcat-u je prisutna implementacija specifikacija Java Servlet-a gdje se izvršavaju servleti te čini web stranice koje uključuju kodiranje u Java Server Pages (JSP) o čemu uglavnom ovise mnoge web aplikacije [47].

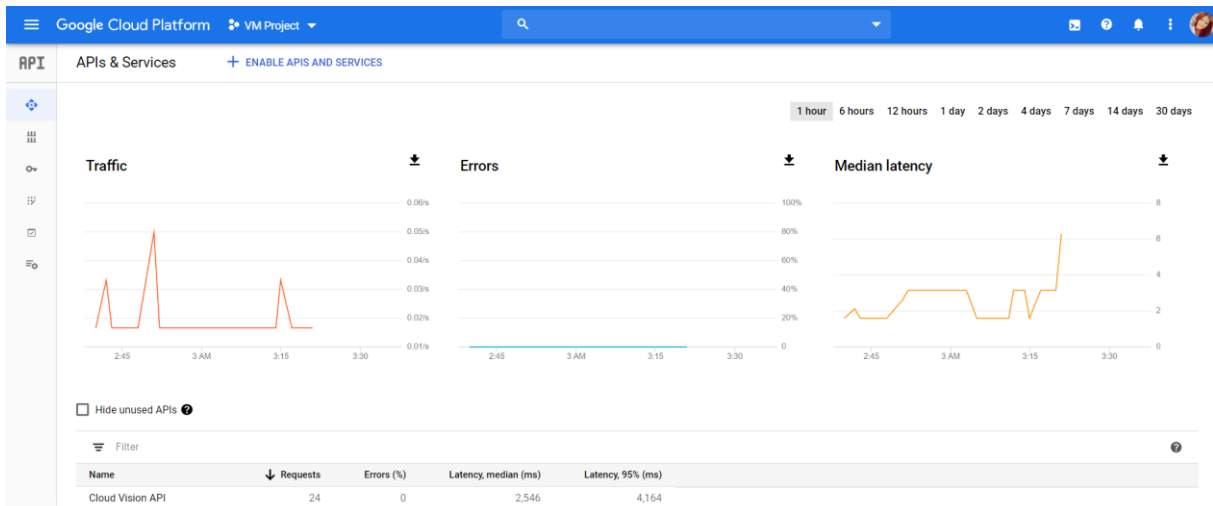
6.1.4. REST API

REST API predstavlja API koji provodi HTTP zahtjeve i prima odgovore (GET, PUT, POST, DELETE). U prijevodu REST je predstavnički prijenos podataka (*engl. Representational state transfer*) čiji je pojam uveo i definirao Roy Fielding 2000. godine u svojoj doktorskoj disertaciji za izradu HTTP 1.1 i URI-a (*engl. Uniform Resource Identifiers*). Zbog svoje jednostavnosti i funkcionalnosti REST je koristan sustavima i aplikacijama u oblaku koje imaju API čime se omogućava povezivanje s uslugama u oblaku. Ponovno korištenje komponenata jedna je od vrlina REST-a što ga čini veoma brzim i pouzdanim [48].

6.2. Konfiguracija korištenih pružatelja usluga

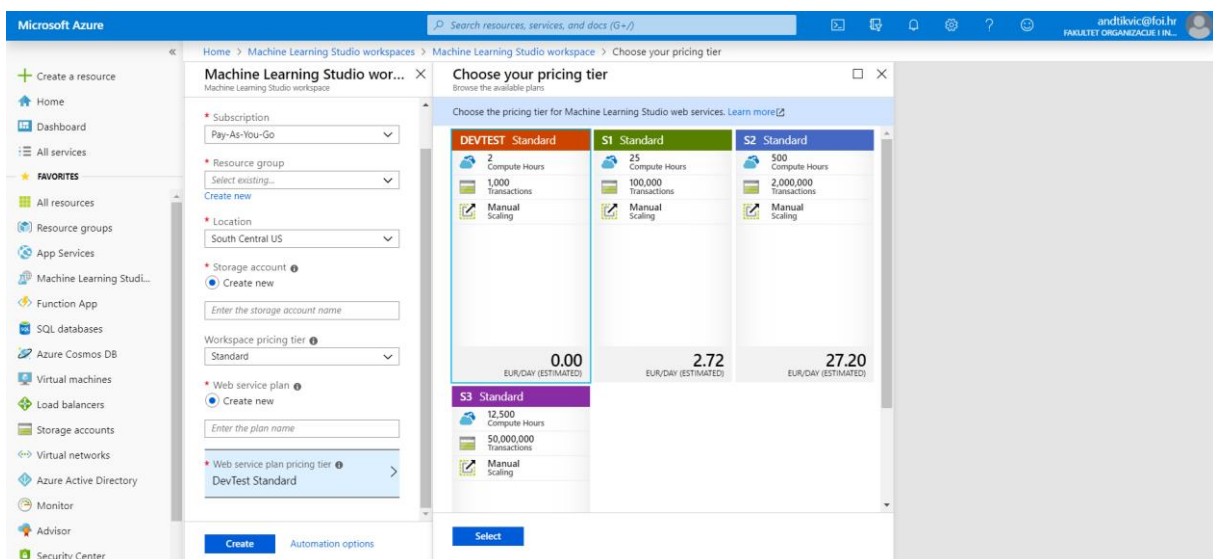
Za potrebe razvoja programa bilo je potrebno postaviti konfiguracije korištenih sustava. Kao što je već spomenuto implementirane su usluge strojnog učenja od Microsoft Azure i Google Cloud Platform sustava. Svaki sustav odnosno platforma ima svoju web stranicu s katalogom usluga koje je moguće integrirati u aplikacije različitih programskih jezika. Neke od usluga koje je moguće integrirati i koristiti su: otkrivanje anomalija, prepoznavanje lica, analiza slike, prognoziranje, predviđanje, prevoditelj, itd. Odabrane usluge u praktičnom dijelu ovog rada jesu detekcija teksta sa slike iz URL-a i prijevod istog u odabranom jeziku.

Postavljanje konfiguracija za Google Cloud Platform-e započinje u Google Cloud konzoli omogućavanjem pristupa Cloud Vision API-u. Zatim je potrebno generirati API ključ putem akreditiva (*engl. credentials*) i pohraniti ga u konfiguracijsku datoteku aplikacije. Tako imamo direktan pristup usluzi iz Google Cloud Platform-e koja je također povezana na račun korisnika gdje nije potrebno unositi korisničko ime i lozinku. Cloud Vision API nudi veliki broj mogućnosti za analizu slike: detekcija lica, detekcija teksta, detekcija logotipa, svojstva slike, itd. Za potrebe ovog rada s Google Cloud Platform-e koristi se detekcija teksta. Google također ima Cloud Storage u kojem je moguće pohranjivati datoteke u spremnike (*engl. buckets*). U ovom slučaju se usluga Cloud Storage-a koristi ako želimo analizirati vlastitu sliku. Na Slici 21. prikazan je promet API zahtjeva koji su se pozivali putem aplikacije.



Slika 21: Promet API zahtjeva u Google Cloud konzoli

Kod Microsoft Azure-a prije svega je potrebno kreirati radni prostor u kojem je određen model plaćanja odnosno odabire se paket (s besplatnih 5000 zahtjeva). Nakon što je kreiran radni prostor, čiji proces možemo vidjeti na Slici 22., potrebno je aktivirati kognitivnu uslugu (*engl. Cognitive Service*). Nadalje, Microsoft Azure ima širok katalog kognitivnih usluga: pretvorba govora u tekst, davanje automatskih prijedloga, prijevod teksta, detekcija lica, provjera gramatike, itd. Za potrebe razvoja aplikacije koristi se usluga za prevođenje teksta točnije *TextTranslator*. Aktivacijom dobivamo pristup cijeloj usluzi s odgovarajućim pretplatničkim ključem koji je potreban za spajanje na uslugu putem aplikacije.



Slika 22: Kreiranje radnog prostora u Microsoft Azure-u

6.3. Struktura programskog rješenja

U NetBeans alatu projekt je kreiran u obliku 'Maven Web application' kako bi se kreirala *pom.xml* datoteka za uključivanje ovisnosti (*engl. dependencies*). Ono je bilo ključno za uključivanje biblioteke za rad s Google Cloud-om ali i za uključivanje ostalih biblioteka potrebne za rad aplikacije. S obzirom na to da se radi o web aplikaciji prilikom kreiranja projekta bilo je potrebno odabrati željeni server. U ovom slučaju to je bio Apache Tomcat za Java EE 7 Web. U postavkama projekta također je bilo potrebno uključiti Java Server Faces radno okruženje. U okviru Tablice 4. prikazan je popis klasa u aplikaciji te u nastavku rada slijedi opis njihove funkcionalnosti.

Tablica 4: Popis klasa u aplikaciji

Naziv klase
AppListener.java
MainInitiator.java
GoogleML.java
AzureML.java
Translation.java

S obzirom na to da se radi o web aplikaciji putem koje je potrebno slati i primiti REST zahtjeve i odgovore bilo je nužno implementirati slušatelja u aplikaciji koji inicijalizira i poništava kontekst aplikacije. Uloga web slušatelja realizirana je u klasi *AppListener.java* koja implementira *ServletContextListener* sučelje. Prilikom pokretanja aplikacije unutar klase pokreće se metoda *contextInitialized* koja dohvaća kontekst aplikacije te sprema atribut za pristup WEB-INF direktoriju. Taj korak bio je nužan zbog pristupa konfiguracijskoj datoteci u WEB-IND direktoriju za dohvaćanje API i pretplatničkog ključa odabranih sustava. Metoda *contextDestroyed* pokreće se prilikom prekidanja odnosno zatvaranja aplikacije.

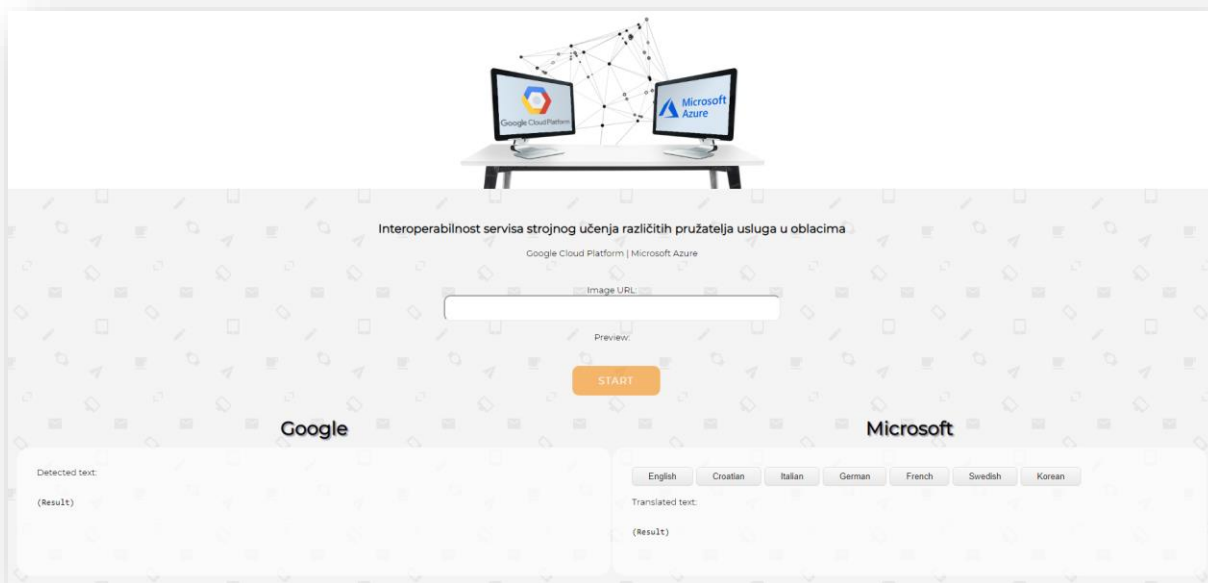
Klasa *MainInitiator.java* predstavlja glavnu klasu od koje se pokreće spajanje i komunikacija s Google i Microsoft sustavima. Prije samog spajanja dohvaća konfiguracijske postavke putem servlet konteksta. Zatim se pripremaju JSON-i za pojedini sustav i šalju se zahtjevi. Nakon što dođe odgovor od Google Cloud Platform-e isti se obrađuje te se dohvaćaju podaci o prepoznatim jezicima u otkrivenom tekstu te sami tekst. Otkriveni tekst se šalje na prevođenje na Microsoft sustav te se vraća odgovor o uspješnosti odnosno prevedeni tekst u odabranim jezicima.

GoogleML.java pokreće se metodom *analyze* te joj se proslijeđuju API ključ i JSON zahtjev. Putem odgovarajućeg URI-a za pokretanje Cloud Vision API-a šalju se zahtjevi s parametrom API ključa i JSON-om kao tijelo zahtjeva. Odgovor sa sustava također dolazi u JSON obliku gdje su navedeni svi detalji o detekciji teksta na slici. Pod tim misli se na xy pozicije svake riječi na slici, detektiran cjelokupan tekst te postotak prisutnosti svih otkrivenih jezika.

Nadalje se pokreće *AzureML.java* pomoću metode *translate* a proslijeđuju joj se pretplatnički ključ i JSON zahtjev. Radi na istom principu kao i Google uz pomoć REST zahtjeva slanjem JSON zahtjeva u kojem se nalazi detektiran tekst za prevođenje. Šalje se zahtjev s parametrima API verzije i odabranih jezika za prijevod. Servis šalje odgovor u JSON obliku prijevoda teksta u odabranim jezicima. Svaki prijevod, s odgovarajućim kodom jezika, sprema se u listu prijevoda tipa *Translation*.

6.4. Detaljni opis funkcionalnosti programa

Kreirana aplikacija otvara se u web pregledniku po želji. Na Slici 23. prikazana je otvorena stranica u kojoj je potrebno unijeti URL odabrane slike s tekstem. Nakon unesenog URL-a te klikom na gumb „START“ pokreće se analiza slike na Google platformi. S obzirom na to da se radi o API-ju za pristup usluzi potreban je API ključ koji je pohranjen u konfiguracijskog datoteci. Datoteka se nalazi u paketu aplikacije te joj se pristupa putem Servlet konteksta (*engl. Servlet Context*). Ispod unesenog linka, u „Preview“ odjeljku, prikazuje odabrana slika. Aplikacija se spaja na sustav te šalje REST zahtjev u JSON obliku čiji su detalji prikazani u Tablici 5. U samom zahtjevu, osim API ključa, potrebno je navesti URI slike te koja vrsta funkcionalnosti nam je potrebna. Odgovor sa sustava također dolazi u JSON obliku te ispisuje uspješnost obrađenog zahtjeva.

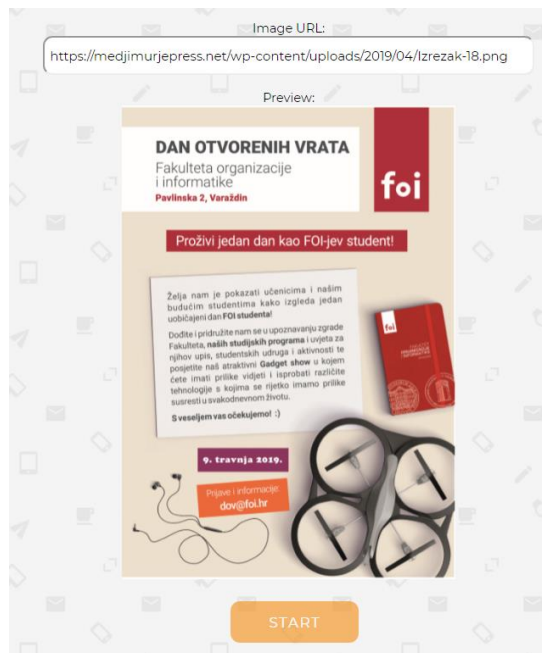


Slika 23: Web stranica aplikacije

Tablica 5: Detalji REST zahtjeva za Google Cloud Platform-u

URI zahtjeva	<code>https://vision.googleapis.com/v1/images:annotate</code>
Zaglavlja	<code>Content-type = application/json</code>
Parametri	<code>api = [API key]</code>
Tijelo zahtjeva	<pre> {"requests":[{" "image":{" "source": {"imageUri":"\" + image_url + "\"} }, "features":[{" "type\":\"TEXT_DETECTION\", "maxResults":1 }] }]}; </pre>

Nakon obrade slike na strani sustava u odjeljku Google-a prikazuje se detektirani tekst odnosno ispis JSON odgovora ako dođe do pogreške. Najčešći razlozi pogreške su da ne postoji tekst na slici ili da sustav trenutno nije u mogućnosti provesti zahtjev. Osim detektiranog teksta prikazuje se i postotak svih prepoznatih jezika. Na Slici 24. prikazana je odabrana slika za detekciju, te je na Slici 25. prikazan odgovor s Google Platform-e.



Slika 24: Odabrana slika za detekciju i prijevod teksta (Izvor: Međimurje Press, 2019)



Slika 25: Otkriveni tekst s odabrane slike

Nakon što je došao odgovor od strane Google-a nadalje se aplikacija spaja na Microsoft Azure platformu te također šalje REST zahtjev. Slično kao kod Google-a potreban je ključ za spajanje na sustav stoga se iz konfiguracijske datoteke čita pretplatnički ključ (*engl. Subscription Key*). U zahtjevu se također šalje JSON zahtjev u kojem se nalazi tekst kojeg je potrebno prevesti u sljedeće jezike: engleski, hrvatski, talijanski, njemački, francuski, švedski i korejski. U okviru Tablice 6. prikazani su detalji REST zahtjeva na Microsoft Azure sustav.

Tablica 6: Detalji REST zahtjeva za Microsoft Azure

URI zahtjeva	<code>https://api.cognitive.microsofttranslator.com/translate</code>
Zaglavlja	<code>Content-type = application/json Ocp-Apim-Subscription-Key = [Subscription Key]</code>
Parametri	<code>api-version = 3.0 to = de,it,hr,en,es,fr,sv,ko</code>
Tijelo zahtjeva	<code>"[{"Text\": \"\" + some_text + \"\"}]\"</code>

Na Slici 26. prikazan je prijevod u odabranom jeziku u Microsoft odjeljku. Prijevod teksta spremljen je u listu prijevoda s odgovarajućim kodom jezika. Prilikom pritiska na gumb odabranog jezika čita se odgovarajući prijevod iz liste te isti prikazuje na web stranicu.



Slika 26: Prijevod otkrivenog teksta sa slike

6.5. Kratki osvrt

Uz pomoć interoperabilnosti može se postići velika raznolikost aplikacija što daje zanimljivu notu daljnjeg razvoja računalstva u oblaku. Ponekad je interoperabilnost nužna ako jedan sustav želimo upotpuniti funkcionalnostima drugog sustava te tako omogućiti pristup željenim funkcionalnostima na jednom mjestu. Jedan slučaj interoperabilnosti implementirao se između Microsoft-a i ONNX (*engl. Open Neural Network Exchange*) što je rezultiralo novim razvojnim okruženjima, kompajlerima i ostalim alatima koji su promovirali strojno učenje [49]. Tijekom razvijanja ove aplikacije svakako sam dobivala nove ideje u cilju ponovnog iskorištavanja sustava za strojno učenje. Primjerice mobilna aplikacija koja bi imala istu funkcionalnost kao i web aplikacija te bi otkriveni tekst s kamere koja snima uživo spremala na Google Drive.

Google Cloud Platform i Microsoft Azure pokazali su se sukladnim sustavima za međusobnu komunikaciju i suradnju. Zbog svoje široke ponude usluga postoje mnoge mogućnosti za druge zanimljive kombinacije i kolaboracije funkcionalnosti. Iako je godina dana besplatnog korištenja sustava zvuči odlično, cijena korištenja ipak nije tako skromna. Jednostavnost implementacije bila je prisutna kod oba sustava popraćene potrebnim dokumentacijama i opširnim primjerima.

7. Zaključak

Strojno učenje pokazalo se kao snažna tehnologija za rješavanje problema koje je teško riješiti klasičnim putem. Glavni izazov za strojeve predstavlja pretvaranje velikog skupa podataka u znanje prema kojem se mogu donositi zaključci i predviđati odgovarajući modeli. Ranije su spomenute vrste strojnog učenja prema kojima su grupirani pristupi rješavanju problema, a to su nadzirano, nenadzirano i podržano strojno učenje.

Danas postoje mnogobrojni sustavi koji potiču razvoj strojnog učenja te korištenje istog. Snažne tvrtke poput Google, Microsoft i Amazon shvaćaju važnost prisutnosti strojnog učenja te svoje sustave svakodnevno nadograđuju. Svojim korisnicima nude godinu dana besplatnog korištenja usluga umjetne inteligencije te u tom vremenu mogu točno procijeniti koju vrstu strojnog učenja njihovo razvojno okruženje zahtjeva. Korištenjem usluga s platformi spomenutih tvrtki programeri nemaju odgovornost održavanja istog već se mogu posvetiti razvoju aplikacije na klijentskoj strani.

Kod razvoja aplikacije, koja koristi strojno učenje, programerima je omogućena primjena raznolikosti funkcionalnosti te iste mogu prilagođavati prema svojim potrebama. Pomoću interoperabilnosti između različitih sustava može doći do nove razine aplikacije u kojoj su pridružene usluge različitih platformi. Dakle, dolazi do produktivnijeg pristupa uslugama strojnog učenja.

Popis literature

- [1] L. Patel, „Machine Learning, Deep Learning & the Wisdom of the Crowd“, *Jumio*, 29-ožu-2018. [Na internetu]. Dostupno na: <https://www.jumio.com/deep-learning-online-identity-verification/>. [Pristupljeno: 09-lip-2019].
- [2] J. Brownlee, „Supervised and Unsupervised Machine Learning Algorithms“, *Machine Learning Mastery*, 12-kol-2019. .
- [3] D. Linthicum, „Machine learning in the cloud: How it can help you right now“, *TechBeacon*, 2018. [Na internetu]. Dostupno na: <https://techbeacon.com/enterprise-it/machine-learning-cloud-how-it-can-help-you-right-now>. [Pristupljeno: 02-lip-2019].
- [4] „Advantages and Disadvantages of Machine Learning Language“, *DataFlair*, 06-velj-2018. [Na internetu]. Dostupno na: <https://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/>. [Pristupljeno: 15-lip-2019].
- [5] H. Heidenreich, „What are the types of machine learning?“, *Medium*, 04-pros-2018. [Na internetu]. Dostupno na: <https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f>. [Pristupljeno: 11-ruj-2019].
- [6] A. Geitgey, „Machine Learning is Fun Part 6: How to do Speech Recognition with Deep Learning“, *Medium*, 07-stu-2018. [Na internetu]. Dostupno na: <https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a>. [Pristupljeno: 18-ruj-2019].
- [7] N. Kartman, „The Top 9 Machine Learning Use Cases for Business Leaders“, *Squadex*, 13-ožu-2019. .
- [8] C. Murray, „Building a Facial Recognition Pipeline with Deep Learning in Tensorflow“, 02-srp-2017. [Na internetu]. Dostupno na: <https://hackernoon.com/building-a-facial-recognition-pipeline-with-deep-learning-in-tensorflow-66e7645015b8>. [Pristupljeno: 18-ruj-2019].
- [9] R. Wiles, „Have we solved the problem of handwriting recognition?“, *Medium*, 27-lip-2019. [Na internetu]. Dostupno na: <https://towardsdatascience.com/https-medium-com-rachelwiles-have-we-solved-the-problem-of-handwriting-recognition-712e279f373b>. [Pristupljeno: 18-ruj-2019].
- [10] A. Nesmiyanova, „The Best Machine Learning Use Cases“, 2019. [Na internetu]. Dostupno na: <https://steelkiwi.com/blog/best-machine-learning-use-cases/>. [Pristupljeno: 18-ruj-2019].
- [11] J. Brownlee, „A Tour of Machine Learning Algorithms“, *Machine Learning Mastery*, 11-kol-2019. .

- [12]C. Nicholson, „A Beginner’s Guide to Neural Networks and Deep Learning“, *Skymind*, 2019. [Na internetu]. Dostupno na: <http://skymind.ai/wiki/neural-network>. [Pristupljeno: 16-ruj-2019].
- [13]S. Hussain, „Use cases of Different Machine Learning Algorithms“, *Medium*, 15-sij-2019. [Na internetu]. Dostupno na: <https://blog.usejournal.com/machine-learning-algorithms-use-cases-72646df1245f>. [Pristupljeno: 16-ruj-2019].
- [14]„Top 10 Machine Learning Algorithms And Its Use Cases“, *Medium*, 16-velj-2019. [Na internetu]. Dostupno na: <https://medium.com/@imaginorlabs/top-10-machine-learning-algorithms-and-its-use-cases-fc303daa2003>. [Pristupljeno: 16-ruj-2019].
- [15]J. Brownlee, „When to Use MLP, CNN, and RNN Neural Networks“, *Machine Learning Mastery*, 23-srp-2018. .
- [16]U. Karn, „An Intuitive Explanation of Convolutional Neural Networks“, *Data science blog*, 11-kol-2016. .
- [17]S. Chatterjee, „A Comprehensive Study of Linear vs Logistic Regression to refresh the Basics“, *Medium*, 22-srp-2019. [Na internetu]. Dostupno na: <https://towardsdatascience.com/a-comprehensive-study-of-linear-vs-logistic-regression-to-refresh-the-basics-7e526c1d3ebe>. [Pristupljeno: 15-ruj-2019].
- [18]R. Shaw, „The Top 10 Machine Learning Algorithms for ML Beginners“, *Dataquest*, 26-lip-2019. .
- [19]C. Nicholson, „Machine Learning Algorithms“, *Skymind*, 2019. [Na internetu]. Dostupno na: <http://skymind.ai/wiki/machine-learning-algorithms>. [Pristupljeno: 16-ruj-2019].
- [20]R. Gour, „Top Machine Learning Algorithms You Should Know to Become a Data Scientist“, *Medium*, 09-tra-2019. [Na internetu]. Dostupno na: <https://becominghuman.ai/top-machine-learning-algorithms-you-should-know-to-become-a-data-scientist-17b16bc85077>. [Pristupljeno: 13-ruj-2019].
- [21]„Support Vector Machines Tutorial - Learn to implement SVM in Python“, *DataFlair*, 04-kol-2017. [Na internetu]. Dostupno na: <https://data-flair.training/blogs/svm-support-vector-machine-tutorial/>. [Pristupljeno: 16-ruj-2019].
- [22]S. Huang, N. Cai, P. P. Pacheco, S. Narandes, Y. Wang, i W. Xu, „Applications of Support Vector Machine (SVM) Learning in Cancer Genomics“, *Cancer Genomics Proteomics*, sv. 15, izd. 1, str. 41–51, pros. 2017.
- [23]„Gradient Boosting Algorithm - Working and Improvements“, *DataFlair*, 20-velj-2018. [Na internetu]. Dostupno na: <https://data-flair.training/blogs/gradient-boosting-algorithm/>. [Pristupljeno: 16-ruj-2019].
- [24]M. Opala, „Deep Learning Frameworks Comparison – Tensorflow, PyTorch, Keras, MXNet, The Microsoft Cognitive Toolkit, Caffe, Deeplearning4j, Chainer“, 10-ruj-2019.

- [Na internetu]. Dostupno na: <https://www.netguru.com/blog/deep-learning-frameworks-comparison>. [Pristupljeno: 17-ruj-2019].
- [25] „Top 8 Deep Learning Frameworks“, *Maruti Techlabs*, 06-ožu-2018. .
- [26] „Caffe2 and PyTorch join forces to create a Research + Production platform PyTorch 1.0“, *Caffe2*, 02-svi-2018. [Na internetu]. Dostupno na: http://caffe2.ai/blog/2018/05/02/Caffe2_PyTorch_1_0.html. [Pristupljeno: 17-ruj-2019].
- [27] P. Migdal i R. Jakubanis, „Keras or PyTorch as your first deep learning framework“, *deepsense.ai*, 26-lip-2018. .
- [28] „DeepLearning4j: Keras model import“, *DeepLearning4J*, 2019. [Na internetu]. Dostupno na: <https://deeplearning4j.org/docs/latest/keras-import-overview>. [Pristupljeno: 17-ruj-2019].
- [29] N. Rodošek, „Cloud Computing ili tzv. usluge u ‚Oblaku‘ za poslovne korisnike“, *CRM sustav po Vašoj mjeri*, 25-sij-2017. .
- [30] P. Mell i T. Grance, „The NIST Definition of Cloud Computing“, str. 7, ruj. 2011.
- [31] M. Boniface i ostali, „Platform-as-a-Service Architecture for Real-Time Quality of Service Management in Clouds“, u *2010 Fifth International Conference on Internet and Web Applications and Services*, 2010, str. 155–160.
- [32] M. Rouse, „What is Infrastructure as a Service (IaaS)?“, *SearchCloudComputing*, ruj-2018. [Na internetu]. Dostupno na: <https://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS>. [Pristupljeno: 16-lip-2019].
- [33] M. Rouse, „What is Data as a Service (DaaS)?“, *SearchDataManagement*, svi-2019. [Na internetu]. Dostupno na: <https://searchdatamanagement.techtarget.com/definition/data-as-a-service>. [Pristupljeno: 06-lip-2019].
- [34] E. Jane, „Top 5 Machine Learning-as-a-Service providers“, *JAXenter*, 12-velj-2018. .
- [35] B. Anar, „Comparing AWS vs Azure vs Google Cloud Platforms For Enterprise App Development“, *Medium*, 14-kol-2018. [Na internetu]. Dostupno na: <https://medium.com/@distillerytech/comparing-aws-vs-azure-vs-google-cloud-platforms-for-enterprise-app-development-28ccf827381e>. [Pristupljeno: 02-ruj-2019].
- [36] „GCP Free Tier“, *Google Cloud*, 20-kol-2019. [Na internetu]. Dostupno na: <https://cloud.google.com/free/docs/gcp-free-tier>. [Pristupljeno: 10-ruj-2019].
- [37] J. Petters, „AWS vs Azure vs Google: Cloud Services Comparison - Varonis“, *Varonis*, 06-kol-2019. [Na internetu]. Dostupno na: <https://www.varonis.com/blog/aws-vs-azure-vs-google/>. [Pristupljeno: 03-ruj-2019].
- [38] C. Berger, „OpenWorld 2018 Developing Predictive Applications with Oracle’s Machine Learning [DEV5758]“, str. 59, 2018.

- [39] „Machine Learning - IBM Watson“, 29-kol-2019. [Na internetu]. Dostupno na: <https://dataplatfrom.cloud.ibm.com/docs/content/wsj/analyze-data/ml-overview.html>. [Pristupljeno: 16-ruj-2019].
- [40] A. Ranabahu i A. Sheth, „Semantics Centric Solutions for Application and Data Portability in Cloud Computing“, u *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, 2010, str. 234–241.
- [41] A. Tikvica i D. Andročec, „Primjena interoperabilnosti računalnog oblaka u pohrani podataka“, *Zb. Rad. Veleuč. U Šiben.*, izd. 3-4/2018, str. 161–175, pros. 2018.
- [42] D. Andročec i N. Vrček, „Methodology for Detection of Cloud Interoperability Problems“, *Int. J. Electr. Comput. Eng. Syst.*, sv. 7., izd. 2., str. 53–59, pros. 2016.
- [43] H. Kubicek i R. Cimander, „Three dimensions of organizational interoperability“, 2009.
- [44] P. Leahy, „Wondering What Java Is? Here’s the Answer“, *ThoughtCo*, 03-srp-2019. [Na internetu]. Dostupno na: <https://www.thoughtco.com/what-is-java-2034117>. [Pristupljeno: 09-ruj-2019].
- [45] M. Rouse, „What is Java?“, *TheServerSide.com*, tra-2019. [Na internetu]. Dostupno na: <https://www.theserverside.com/definition/Java>. [Pristupljeno: 09-ruj-2019].
- [46] M. Rouse, „What is NetBeans?“, *TheServerSide.com*, ožu-2014. [Na internetu]. Dostupno na: <https://www.theserverside.com/definition/NetBeans>. [Pristupljeno: 09-ruj-2019].
- [47] „Is Tomcat An Application Server Or A Web Server?“, *JavaPipe*, 08-srp-2019. [Na internetu]. Dostupno na: <https://javapipe.com/blog/tomcat-application-server/>. [Pristupljeno: 10-ruj-2019].
- [48] D. Booth *i ostali*, „Web Services Architecture“, *W3C Working Group Note*, 2004. [Na internetu]. Dostupno na: <https://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#relwwwrest>. [Pristupljeno: 11-ruj-2019].
- [49] K. Johnson, „AI Weekly: Microsoft, machine learning framework interoperability, and ONNX“, *VentureBeat*, 03-svi-2019. .

Popis slika

Slika 1: Kronološki prikaz nastanka dijelova umjetne inteligencije (Izvor: Patel, 2018).....	2
Slika 2: Vrste strojnog učenja (Izvor: Heidenreich, 2018).....	4
Slika 3: Nadzirano strojno učenje (Izvor: vlastita izrada, 2019).....	5
Slika 4: Nenadzirano strojno učenje (Izvor: vlastita izrada, 2019).....	5
Slika 5: Podržano strojno učenje (Izvor: vlastita izrada, 2019).....	6
Slika 6: Proces obrade prepoznatog zvuka (Izvor: Geitgey, 2016).....	7
Slika 7: Google asistent aplikacija (Izvor: Google Assistant Guide, 2018).....	7
Slika 8: Primjer prepoznavanja lica osobe s putovnice (Izvor: Murray, 2017).....	8
Slika 9: Proces procjene i sprječavanje prijave (Izvor: Kartman, 2019).....	9
Slika 10: Primjer automatskog označavanja osoba korištenjem CNN-a (Izvor: Facebook, 2019).....	11
Slika 11: Grafički primjer linearne regresije (Izvor: Shaw, 2019).....	13
Slika 12: Grafički primjer logističke regresije (Izvor: Shaw, 2019).....	14
Slika 13: Primjer stabla odlučivanja.....	15
Slika 14: Grafički primjer SVM algoritma (Gour, 2019).....	16
Slika 15: Sučelje TensorBoard-a s vizualizacijom rezultata predviđanja.....	18
Slika 16: Sučelje za vizualizaciju rezultata u DeepLearning4J programskom okviru.....	20
Slika 17: Početna stranica Google Cloud Platform konzole.....	24
Slika 18: Početna stranica Microsoft Azure konzole.....	25
Slika 19: Početna stranica Amazon Web Services konzole.....	26
Slika 20: Slikoviti prikaz interoperabilnosti sustava (Izvor: vlastita izrada, 2019).....	28
Slika 21: Promet API zahtjeva u Google Cloud konzoli.....	33
Slika 22: Kreiranje radnog prostora u Microsoft Azure-u.....	33
Slika 23: Web stranica aplikacije.....	36
Slika 24: Odabrana slika za detekciju i prijevod teksta (Izvor: Međimurje Press, 2019).....	37
Slika 25: Otkriveni tekst s odabrane slike.....	37
Slika 26: Prijevod otkrivenog teksta sa slike.....	38

Popis tablica

Tablica 1: Prikaz prednosti i nedostataka kod strojnog učenja	3
Tablica 2: Usporedba programskih okvira za dubinsko učenje	17
Tablica 3: Razine interoperabilnosti.....	29
Tablica 4: Popis klasa u aplikaciji	34
Tablica 5: Detalji REST zahtjeva za Google Cloud Platform-u.....	36
Tablica 6: Detalji REST zahtjeva za Microsoft Azure	38