



UNIVERZITET U SARAJEVU  
ELEKTROTEHNIČKI FAKULTET  
ODSJEK ZA RAČUNARSTVO I INFORMATIKU

# **Tema: Analiza recenzija putovanja koristeći ML bazu podataka za mašinsko učenje**

**ZAVRŠNI RAD**

-Prvi ciklus studija-

**Mentor:**

doc. dr Emir Buza, dipl. ing. el.

**Student:**

Šejla Pljakić

Sarajevo, 2020.

## Postavka rada

**Teme za završne radove 1. ciklusa za 2019/2020 studijsku godinu**

**Nastavnik:** doc. dr Emir Buza, dipl. ing. el.

**Student:** Šejla Pljakić

**Tema:** Analiza recenzija putovanja koristeći ML bazu podataka za mašinsko učenje

### Cilj:

- Upoznavanje sa mašinskim učenjem i metodama mašinskog učenja
- Korištenje MLDB programa za treniranje modela
- Analiza i rješavanje konkretnog modela

### Opis:

### Okvirni sadržaj rada:

1. Uvod - U okviru uvodnog poglavlja prikazati će se uvod u temu, ciljevi rada, metodologija korištena za izradu rada kao i struktura rada.
2. U ovom poglavlju će se detaljno objasniti sve o pojmu mašinskog učenja
3. Opisati će se koje su to sve metode mašinskog učenja i kako doprinose mašinskom učenju
4. Objasniti će se tehnike mašinskog učenja i prikazati će se primjeri tih tehnika kako bi ih znali upotrijebiti na konkretnom problemu
5. Detaljan opis MLDB baze podataka, kakva je to baza i prikaz mogućih instalacija
6. Opis skupa podataka recenzije putovanja
7. Praktičan rad analize koristeći tehnike klasifikaciju i klastering za dobijanje rezultata
8. Zaključak - U okviru ovog poglavlja rezimirat će se urađeno, dati osvrt na rad i smjernice za buduće istraživanje vezano uz ovu temu, kao i prijedlozi za unaprjeđivanje dobivenih rezultata.

**Očekivani rezultati:** Prikaz mjesta koje je najbolje posjetiti u skladu sa analizom i mašinskim učenjem nakon provedenih tehnika mašinskog učenja.

**Polazna literatura:**

1. \*\*\*Data Science, <https://towardsdatascience.com>
2. \*\*\*Machine learning, <https://machinelearningmastery.com>
3. \*\*\*Machine learning in math, <https://www.mathworks.com>
4. \*\*\*MLDB documentation, <https://docs.mldb.ai>

---

**Prof . dr. Ime Prezime**

Univerzitet u Sarajevu

**Naziv fakulteta/akademije:** Elektrotehnički fakultet u Sarajevu

**Naziv odsjeka i/ili katedre:** Računarstvo i informatika

### **Izjava o autentičnosti radova**

Seminarski rad, završni (diplomski odnosno magistarski) rad za I i II ciklus studija i integrirani studijski program I i II ciklusa studija, magistarski znanstveni rad i doktorska disertacija<sup>1</sup>

**Ime i prezime:** Šejla Pljakić

**Naslov rada:** Analiza recenzija putovanja koristeći ML bazu podataka za mašinsko učenje

**Vrsta rada:** Završni rad 1. ciklusa studija

**Broj stranica:** 50

#### **Potvrđujem:**

- da sam pročitao/la dokumente koji se odnose na plagijarizam, kako je to definirano Statutom Univerziteta u Sarajevu, Etičkim kodeksom Univerziteta u Sarajevu i pravilima studiranja koja se odnose na I i II ciklus studija, integrirani studijski program I i II ciklusa i III ciklus studija na Univerzitetu u Sarajevu, kao i uputama o plagijarizmu navedenim na Web stranici Univerziteta u Sarajevu;
- da sam svjestan/na univerzitetskih disciplinskih pravila koja se tiču plagijarizma;
- da je rad koji predajem potpuno moj, samostalni rad, osim u dijelovima gdje je to naznačeno;
- da rad nije predat, u cjelini ili djelimično, za stjecanje zvanja na Univerzitetu u Sarajevu ili nekoj drugoj visokoškolskoj ustanovi;
- da sam jasno naznačio/la prisustvo citiranog ili parafraziranog materijala i da sam se referirao/la na sve izvore;
- da sam dosljedno naveo/la korištene i citirane izvore ili bibliografiju po nekom od preporučenih stilova citiranja, sa navođenjem potpune reference koja obuhvata potpuni bibliografski opis korištenog i citiranog izvora;
- da sam odgovarajuće naznačio/la svaku pomoć koju sam dobio/la pored pomoći mentora/ice i akademskih tutora/ica.

**Mjesto, datum:**

Sarajevo, mart, 2020. god.

**Potpis:**

\_\_\_\_\_

---

<sup>1</sup> U radu su korišteni slijedeći dokumenti: Izjava autora koju koristi Elektrotehnički fakultet u Sarajevu; Izjava o autentičnosti završnog rada Centra za interdisciplinarne studije – master studij „Evropske studije“, Izjava o plagijarizmu koju koristi Fakultet političkih nauka u Sarajevu.

## Sažetak

Mašinsko učenje danas se često koristi u svakodnevnom životu iako toga niste ni svjesni. Kako bi poboljšali naš život i učinili ga kvalitetnijim od presudnog je značaja saznati i naučiti bitne stvari o mašinskom učenju i kako to znanje iskoristiti u stvarnom svijetu i svakodnevnim problemima. Mnogi ne znaju da se upravo za prepoznavanje lica i prepoznavanje govora koriste algoritmi mašinskog učenja, kao i za automatsko prevođenje. Poznavanjem osobina, metoda i tehnika mašinskog učenja, steći će se bolji uvid u problematiku kao i rješavanje mnogih zadataka posebno zadatka koji je predmet istraživanja ovog završnog rada.

Danas ljudi često putuju u razne krajeve svijeta kako turistički tako i poslovno i problem predstavljaju uvijek aktivnosti na koje žele provesti svoje vrijeme i koliko treba vjerovati recenzijama drugih putnika i njihovim iskustvima.

Nekada nečija dobra recenzija može navesti drugu osobu da posjeti određeno mjesto, što može imati negativne konotacije u smislu nezadovoljstva i bezpotrebnog trošenja resursa kao što su novac i vrijeme. Zato će se u ovom radu tehnikama mašinskog učenja pokušati analizirati recenzije u cilju dobijanja zaključka na koje aktivnosti je najbolje potrošiti resurs poput vremena. U ovom radu se koristi open-source softver MLDB baza podataka i programski jezik Python u cilju analize skupa podataka koji je objavio Tripadvisor.com.[16]

**Ključne riječi:** Mašinsko učenje, prepoznavanje govora, prepoznavanje lica, automatsko prevođenje, tehnike mašinskog učenja, recenzije putovanja, open-source softver, MLDB, Python, Tripadvisor.com

## Abstract

Machine learning is increasingly used in everyday life, although we are not even aware of it. Therefore, to improve our lives and make them better quality, it is crucial to learn how to learn the essentials of machine learning and how to use that knowledge in the real world. Many do not know that machine learning algorithms, as well as automatic translation, are used for face recognition and speech recognition, so machine learning is comprehensive in everyday life. Knowing the methods and techniques of machine learning, you will gain a better understanding of the problem as well as solve many tasks, especially the task that is the subject of research in this bachelor thesis.

Today, people often travel to different parts of the world for both tourism and business, and the problem is always the activity where we want to spend our time in these countries, and how much we have to trust the reviews of other travellers and their experiences with those activities.

Sometimes a good review leads us to visit one place and then we get disappointed and spend resources such as time and money on something that is not worth visiting at all. So machine learning techniques will analyze reviews to help decide on which activities are better to spend time. I will use the open-source software MLDB database and the Python programming language to make the query and train the input dataset published by Tripadvisor.com and downloaded it from the UCI official site. With specially selected machine learning techniques and using the programming languages, we will come to the final results.

**Keywords:** Machine Learning, Speech Recognition, Face Recognition, Automatic Translation, Machine Learning Techniques, Travel Reviews, Open-Source Software, MLDB, Python, Tripadvisor.com

## Sadržaj

Popis slika .....	7
Popis tabela .....	8
1 Uvod .....	9
2 Objašnjenje pojma - mašinsko učenje.....	10
3 Metode mašinskog učenja.....	13
3.1 Supervizirano mašinsko učenje.....	13
3.2 Nesupervizirano mašinsko učenje.....	14
3.3 Učenje ojačanja.....	15
4 Tehnike mašinskog učenja.....	16
4.1 Regresija.....	16
4.2 Klasifikacija.....	17
4.3 Klasterizacija(grupisanje) .....	19
4.4. Redukovanje dimenzija.....	21
4. Metode cjelina.....	22
4.6 Neuronske mreže i duboko učenje.....	23
4.7 Prijenosno učenje.....	24
4.8 Obrada prirodnog jezika.....	25
4.9 Umetanja riječi.....	26
5 Machine learning database - MLDB.....	27
5.1 Arhitektura MLDB-a.....	28
5.2 Podrška za algoritme mašinskog učenja.....	30
5.3 Instaliranje MLDB-a na različitim platformama.....	32
6 Opis skupa podataka recenzije putovanja.....	34
7 Praktični rad koristeći bazu MLDB i programski jezik Python.....	35
Zaključak .....	48
Literatura.....	49

## Popis slika

Slika .2.1 Prikaz rada algoritama mašinskog učenja.....	11
Slika 4.1.1. Prikaz odnosa potrošnje predviđene i posmatrane energije.....	17
Slika 4.2.1 Grafik logističke regresije.....	18
Slika 4.3.1. Grupiranje zgrada u efikasne (zelene) i neučinkovite (crvene) skupine.....	20
Slika 4.3.2. Prikaz grupisanja elemenata.....	20
Slika 4.4.1. Analiza baze podataka MNIST rukom pisanih cifara.....	22
Slika 4.6.1. Neuronske mreže sa skrivenim slojem.....	23
Slika 4.6.2. Neuronske mreže sa više skrivenih slojeva.....	23
Slika 4.9.1. Aritmetika sa vektorima riječi.....	26
Slika 5.1. Životni ciklus MLDBa.....	27
Slika 5.1.1. Prikaz arhitekture MLDBa.....	28
Slika 5.2.2. Tok rada mašinskog učenja.....	30
Slika 5.2.3 Učinkovitost MLDBa.....	32
Slika 5.3.1. Oficijalna stranica MLDB baze podataka.....	33
Slika 7.1. Dio programskog koda učitavanja skupa podataka u bazu.....	35
Slika 7.2. Prikaz broja umjetničkih galerija.....	37
Slika 7.3. Prikaz broja plesnih klubova.....	38
Slika 7.4. Prikaz broja barova.....	38
Slika 7.5. Prikaz broja restorana.....	39
Slika 7.6. Prikaz broja muzeja.....	39
Slika 7.7. Prikaz broja odmarališta.....	40
Slika 7.8. Prikaz broja parkova/piknik mjesta.....	41
Slika 7.9. Prikaz broja plaža.....	42
Slika 7.10. Prikaz broja religijskih institucija.....	42
Slika 7.11. Prikaz broja pozorišta.....	43
Slika 7.12. Isječak koda iz MLDBa.....	43
Slika 7.13. Isječak koda iz MLDBa unos skupa podataka koristeći proceduru.....	44
Slika 7.14. Procedura za treniranje skupa podataka uz pomoć K-means tehnike.....	45
Slika 7.15. Grafici pojedinih kategorija.....	46
Slika 7.16. Grafici parkova/piknik mjesta i religijskih institucija.....	47
Slika 7.17. Isječak koda logističke regresije iz MLDB baze podataka.....	47



## Popis tabela

Tabela 5.2.1. Prikaz tehnika MLDBa.....	30
Tabela 5.3.2. Prikaz verzija instalacije MLDBa.....	33
Tabela 6.1. Karakteristike skupapodataka.....	34
Tabela 7.1. Prikaz tabele skupa podataka.....	36
Tabela 7.2. Tabela prikaza skupa podataka koristeći proceduru.....	44

## 1 UVOD

Mašinsko učenje fokusira se na razvoj algoritama koji mogu učiti iz podataka i na osnovu toga vršiti razne predikcije. Nastalo je u okruženju u kojem su se dostupni podaci, statističke metode i kompjuterska snaga brzo i istovremeno razvijali. Rast podataka zahtijevao je dodatnu računarsku snagu, što je zauzvrat potaknulo razvoj metoda za analizu velikih skupova podataka. Pionir mašinskog učenja Arthur Samuel 1959, definira mašinsko učenje kao "polje učenja koje daje kompjuterima sposobnost da uče bez eksplicitnog programiranja". Dok je radio za IBM razvio je program koji uči igranje dame i vremenom poboljšava svoj način igranja. [7]

Mašinsko učenje se bavi razvojem algoritama koji bi bili korisni da se oslanjaju na kolekciju primjeraka nekog fenomena. Kolekcije mogu poticati iz prirode, biti ručno izrađene od strane ljudi ili generisane od strane drugih algoritama.

U okviru ovog završnog rada objašnjeni su koncepti mašinskog učenja kao i direktna primjena tehnika mašinskog učenja na konkretan problem. U drugom poglavlju su objašnjeni osnovni pojmovi mašinskog učenja, prikaz algoritama mašinskog učenja i prikaz primjera iz stvarnog svijeta koji su riješeni metodama mašinskog učenja.

U trećem poglavlju objašnjene su metode mašinskog učenja. Poseban akcenat je dat metodama mašinskog učenja koje pripadaju kategorijama kao što su supervizirano (nadgledano) mašinsko učenje, nesupervizirano (nenadgledano) učenje i pojačano učenje.

U četvrtom poglavlju opisane su tehnike mašinskog učenja koje su proizašle iz metoda, a potom i navedeni primjeri kako bi se saznalo kada koju tehniku iskoristiti.

U petom poglavlju opisana je baza podataka za mašinsko učenje zajedno sa svim osobinama.

U šestom poglavlju opisan je skup podataka recenzija putovanja zajedno sa svim atributima.

U sedmom poglavlju prikazan je praktični dio analize recenzija putovanja u istočnoj Aziji koristeći MLDB bazu podataka i programski jezik Python, kao i rezultati analize.

Na samom kraju dat je zaključak s osvrtom na čitav završni rad.

## 2 Objašnjenje pojma - mašinsko učenje [1]

Mašinsko učenje je tehnika analitičke obrade podataka koja nastoji usmjeriti računare da rade ono što prirodno dolazi ljudima i životinjama, a to je učenje iz iskustva.

Računari su stroge logike, tako da ako se želi da oni nešto urade moraju im se pružiti detaljne upute šta tačno treba da rade. Mašinsko učenje fokusira se na razvoj računarskih programa koji mogu pristupiti podacima i koristiti ih za poboljšavanje niza aktivnosti.

Algoritmi mašinskog učenja koriste računske metode za „učenje“ informacija direktno iz podataka bez oslanjanja na unaprijed određeni model. Algoritmi adaptivno poboljšavaju svoje performanse kako se povećava broj uzoraka dostupnih za učenje.

Sa porastom rada u okviru velikih podataka (big data- podskup mašinskog učenja)[18], mašinsko učenje postalo je ključna tehnika za rješavanje problema u raznim područjima, kao što su:

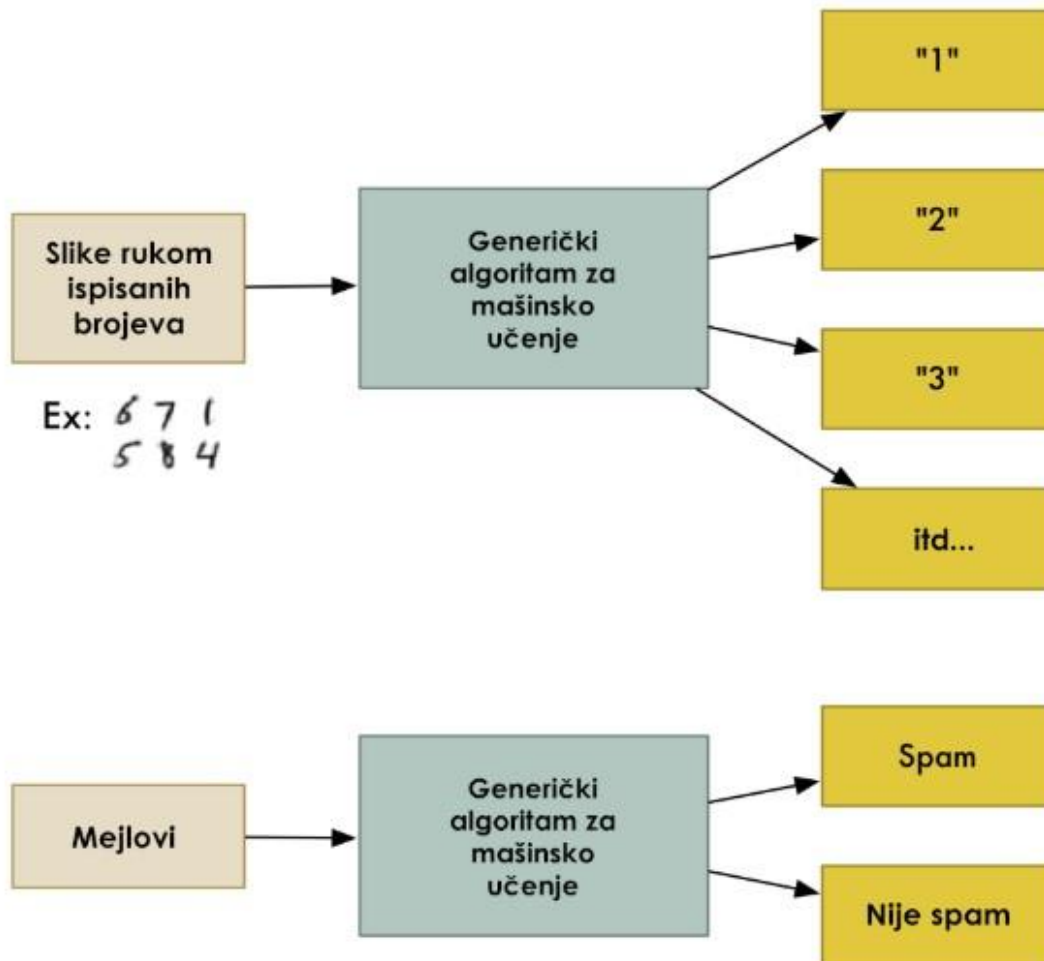
- Računarske finansije za kredite i algoritamsko trgovanje.
- Obrada slike i računarska vizija za prepoznavanje vida i detekciju pokreta
- Računarska biologija za otkrivanje kancera, pronalazak novih lijekova i sekvenciranje DNK
- Automobilska, vazдушna i proizvodna energija za prediktivno održavanje
- Obrada prirodnog jezika za aplikacije za prepoznavanje glasa

Algoritmi mašinskog učenja pronalaze prirodne šablone u podacima i pomažu u donošenju odluka i predviđanja. Algoritmi se svakodnevno koriste za donošenje kritičnih odluka u medicinskoj dijagnozi, trgovanju dionicama i predviđanju energetskog opterećenja. Kao primjer mogu se uzeti medijske stanice koje koriste mašinsko učenje kako bi dali preporuke raznih pjesama ili filmova. Prodavači koriste mašinsko učenje kako bi stekli uvid u ponašanje kupaca prilikom kupovine.

Mašinsko učenje treba koristiti kada postoji složen zadatak ili problem koji uključuje veliku količinu podataka i varijabli, ali nema postojeće formule ili jednačine rješavanja.

Proces učenja započinje opažanjima ili podacima, direktnim iskustvom ili uputama, kako bi se u potrazi za uzorcima donijele bolje odluke u budućnosti na temelju postojećih primjera. Primarni cilj je omogućiti računarima da automatski uče bez ljudske intervencije ili pomoći.

Mašinsko učenje zasniva se na ideji da postoje generički algoritmi koji mogu pokazati nešto interesantno o skupu podataka, a da se pritom ne mora napisati poseban kod za taj problem. Umjesto pisanja koda, ubacuju se podaci u generički algoritam, a on dalje pravi svoju logiku na osnovu podataka. Na primjer, jedna vrsta ovih algoritama je klasifikacioni algoritam koji može da smjesti podatke u različite grupe. Isti klasifikacioni algoritam koji se koristi da prepozna rukom pisane brojeve mogao bi se koristiti i za klasifikaciju mejlova u “spam”/“nije spam”, bez promjene linija koda. Isti algoritam se može koristiti za različite vrste klasifikacija, tj. na osnovu skupa podataka za treniranje moguće je jedan te isti algoritam koristiti za rješavanje više različitih problema.



**Slika 2.1.** Prikaz rada algoritama mašinskog učenja [2]

Algoritam prikazan na slici 2.1 može se zamisliti kao crna kutija s obzirom na to da algoritam „smišlja“ sopstvenu logiku za mašinsko učenje. Mašinsko učenje je krovni termin koji pokriva mnogo ovakvih vrsta klasifikacionih algoritama. Mašinsko učenje danas doživljava eksponencijalni rast, posebno u pogledu računarske vizije. Danas je stopa pogreške kod ljudi samo 3% u računarskoj viziji. To znači da su računari već bolji u prepoznavanju i analiziranju slika od ljudi. Prije više decenija računari su bili komadi mašina veličine sobe, danas oni mogu uočiti svijet oko nas na način za koji se mislilo da nije moguće. Ovo postignuće je omogućeno napredovanjem u mašinskom učenju i nije samo uspjeh kompjuterskih i AI stručnjaka već ova znanja imaju veliku primjenu u stvarnom životu pa tako spašavaju živote mnogih ljudi i svijet čine boljim mjestom.

Na primjer, problem svrstavanja 10 000 slika pasa u odgovarajuće vrste, računar korištenjem mašinskog učenja i posebnog skupa podataka to izvršava za nekoliko minuta, dok bi za takav problem nekom stručnjaku za pse trebalo znatno više vremena.

Primjena računarske vizije korištenjem mašinskog učenja od velikog je značaja posebno za zemlje trećeg svijeta kao i u ruralnim selima u kojima postoji nedostatak ljekara. Ovaj način posmatranja problema kroz mašinsko učenje može se tretirati kao pomoć drugog mišljenja ljekaru, čime se osigurava vjerodostojnost njihove dijagnoze. Tako da je svrha računarske vizije u medicinskom polju umnožavanje stručnosti specijalista i raspoređivanje znanja na onim mjestima gdje ga ljudi najviše trebaju.

Modeli jezika su algoritmi koji pomažu mašinama da razumiju tekst i izvršavaju sve vrste operacija poput prevođenja teksta. Prema Jeffu Deanu, [17] postignut je veliki napredak u jezičkim modelima. Danas računari mogu razumjeti odlomke teksta na mnogo dubljem nivou nego što su mogli prije. Iako nisu na nivou čitanja čitave knjige i ne razumiju je kao i ljudi, sposobnost razumijevanja nekoliko odlomaka teksta temeljna je za stvari poput poboljšanja Google sistema pretraživanja. Model BERT,[17] najnoviji model obrade prirodnog jezika (NLP) [17] koji je Google objavio, koristi se u algoritmima za rangiranje pretraživanja, što je pomoglo u poboljšanju rezultata pretraživanja za mnoštvo različitih vrsta upita koji su ranije bili vrlo teški. Drugim riječima, sistem pretraživanja sada može bolje razumjeti različite vrste pretraživanja koje vrše korisnici i pomoći u pružanju boljih i tačnijih odgovora.

Danas u svijetu mašinskog učenja stručnjaci pronalaze problem koji žele riješiti i usredotočeni su na pronalaženje pravog skupa podataka kako bi obučili model i izvršili određeni zadatak. Tako se problem počinje od nule – inicijaliziraju parametar modela sa slučajnim tačkama, a zatim se pokušavaju saznati svi zadaci iz skupa podataka. Slično je sa svakim novim učenjem koji se pojavljuje u ljudskom svijetu zaboravi se sve ono što se prije naučilo i postavi se u kožu novorođenčeta te iznova uče nove stvari, na taj način iskorištava se puni potencijal novog znanja.

Budućnost mašinskog učenja je u određivanju velikog modela koji će biti multifunkcionalan i koji će činiti više stvari. Na primjer, model računarske vizije koji može dijagnosticirati dijabetičku retinopatiju, klasificirati različite vrste pasa, prepoznati lice i istovremeno se koristiti u automatskim vozilima i dronovima, sve to je moguće uz model koji koristi mašinsko učenje. Taj model djeluje tako što aktivira različite dijelove modela samo kada su potrebni, zato će model većinu vremena biti u praznom hodu (oko 99% vremena), a kada je potrebno zatražit će se pravi fragment za aktivaciju. Izgradnja ovog modela stvorila bi puno zanimljivih računarskih sistema i problema mašinskog učenja kao što su skalabilnost i struktura modela. Glavno pitanje koje se postavlja je kako će model naučiti i usmjeriti različite dijelove sistema na najprikladniji način kako bi se dobilo optimalno rješenje. Kako bi se ovo pitanje riješilo bit će potrebno puno poboljšanja u istraživanju mašinskog učenja. Za napredak u mašinskom učenju ključni faktor je dobra upućenost u algoritme i etičnost posla. [1]

### 3 Metode mašinskog učenja

Osnove mašinskog učenja obuhvataju učenje iz okruženja, zatim primjenu tog učenja za donošenje odluka. Da bi se to učinkovito postiglo, postoje kategorije algoritama mašinskog učenja koje to omogućavaju. [2]

#### 3.1 Supervizirano mašinsko učenje

Algoritam mašinskog učenja pod supervizijom sastoji se od varijable cilja (zavisne varijable) koju treba predvidjeti iz zadanog skupa prediktora (nezavisnih varijabli). Kod superviziranog učenja cilj je osmisliti funkciju mapiranja ( $f$ ) koja će najbolje opisati ulazne podatke ( $x$ ) za zaključivanje izlaznih podataka ( $Y$ ). Prvo je potrebno pronaći funkciju mapiranja ( $f$ ) koja će postići određeni nivo performansi. Zatim je potrebno primijeniti dobijenu funkciju na nove podatke kako bi se potvrdilo da li se dobijaju isti ili slični rezultati. Rezultati treninga koriste se za pronalaženje funkcije  $f$  tako da je  $Y = f(X)$ . Supervizirano učenje najčešće se koristi u aplikacijama gdje podaci iz prošlosti predviđaju vjerovatne buduće događaje. Na primjer, može se predvidjeti kada je vjerovatno da će transakcije s kreditnim karticama biti lažne ili koji će klijent osiguranja najvjerovatnije podnijeti zahtjev. Postoje dvije vrste problema superviziranog mašinskog učenja: klasifikacija i regresija ovisno o vrsti izlazne varijable. Ako je izlazna varijabla kategorična, to je problem s klasifikacijom. (Primjer: Boja može biti crvena, plava, ljubičasta itd.) Ako je izlazna varijabla stvarna vrijednost (brojčana vrijednost), onda je to problem sa regresijom. (Primjer: Visina može biti na skali od 0m do 2m) [2]

Lista algoritama superviziranog mašinskog učenja:

- Linearna regresija
- Podrška vektorskih mašina
- Logistička regresija
- Naivni Bayes
- Linearna diskriminatorska analiza
- Stabla odluka

## 3.2 Nesupervizirano mašinsko učenje

Za razliku od superviziranog mašinskog učenja, nesupervizirano mašinsko učenje ne pretpostavlja tačan skup izlaznih vrijednosti „Y“, nema izlaza tj. ne postoji nijedna varijabla cilja ili ishoda koja bi se mogla predvidjeti. Sistem nije upućen u "pravi odgovor" pa algoritam mora shvatiti šta se prikazuje. Cilj je istražiti podatke i pronaći neku strukturu unutar. Ova metoda učenja dobro funkcioniše s transakcijskim podacima. Na primjer, prepoznavanje kupaca sa sličnim atributima koji mogu biti tretirani na sličan način u marketinškim kampanjama ili se mogu pronaći glavni atributi koji razdvajaju segmente kupca jedan od drugog.

Također, cilj je predstaviti najzanimljiviju strukturu koja dobro opisuje ulazne podatke. Postoje dvije vrste nesuperviziranih problema mašinskog učenja: klasterizacija i udruživanje. Problem s klasterizacijom se javlja kod grupisanja ulaznih podataka u predefinisane grupe ili klastere. (Primjer: grupisanje biračkog ponašanja po spolu).

Udruživanje nastaje kada se otkriju pravila unutar ulaznih podataka. (Primjer: ženske glasačice obično glasaju za kandidatkinje). Isto tako, ovi se algoritmi koriste za segmentaciju tekstualnih tema, preporuku stavki i identifikaciju izdataka podataka. [2]

Lista algoritama mašinskog učenja koji nisu supervizirani:

- Hijerarhijska klasterizacija
- Samoorganizacija karata
- Preslikavanje najbližeg susjeda
- Razlaganje pojedinačne vrijednosti
- K-klasterizacija
- Lokalni vanjski faktor
- Neuronske mreže
- Algoritam očekivanja – maksimizacija
- Analiza glavnih komponenti
- Negativna matrična faktORIZACIJA

### 3.3 Pojačano učenje (engl. Reinforcement learning)

Ukoliko se posmatra miš u labirintu koji pokušava pronaći skrivene komade sira, što je više puta miš izložen labirintu, to će biti uspješniji u pronalaženju sira. U početku se miš može kretati nasumično, ali nakon nekog vremena iskustvo mu pomaže razumjeti sa kakvim radnjama se približava siru.

Proces učenja kod miša odražava ono što je potrebno uraditi s pojačanim učenjem radi treniranja sistema ili igre. Općenito govoreći, ovo je metoda mašinskog učenja koja agentu pomaže da nauči iz iskustva. Memorisanjem radnji i korištenjem pokušaja i pogreške u postavljenom okruženju, ova metoda može maksimizirati kumulativnu nagradu. U ovom primjeru miš je agent, a labirint okoliš. Skup mogućih radnji miša jest pomicanje prednje, stražnje, lijeve ili desne strane, a nagrada je sir.

Prateći korake algoritma potrebno je doći do ukupne nagrade koja u konačnici ima svoj pozitivan ili negativan ishod. Ukupna nagrada predstavlja zbir svih pozitivnih i negativnih nagrada na putu, a cilj je uvijek pronaći najbolji put koji maksimizira nagradu.

Korištenjem ovog algoritma mašina je osposobljena za donošenje određenih odluka. Za razliku od superviziranog i nesuperviziranog mašinskog učenja, pojačano učenje bazirano je na pronalaženju najboljeg puta koji treba proći u nekoj situaciji kako bi se maksimizirala nagrada. Odluka se donosi uzastopno.

Reinforcement learning (RL) se može koristiti ako postoji malo historijskih podataka o nekom problemu, jer on ne traži unaprijed informacije za razliku od tradicionalnih metoda mašinskog učenja. Nije iznenađujuće što je RL posebno uspješan s igrama, tj. sa "savršenim informacijskim" igrama kao što su šah i Go. Pomoću igara, povratne informacije od strane agenta i okoliša dolaze brzo, omogućujući modelu da brzo uči. Nedostatak RL-a je da može proći puno vremena da se uvježba ako je problem složen.

Baš kao što je IBM Deep Blue pobijedio najboljeg šahovskog igrača 1997. godine, AlphaGo, algoritam baziran na RL-u, pobijedio je Go najboljeg igrača 2016. Trenutno su RL pioniri DeepMind timovi u Velikoj Britaniji. [2]

U aprilu 2019. godine ekipa OpenAI Five bila je bila prva ekipa iz područja AI koja je pobijedila Dota 2 tim svjetskog prvaka u e-sportu. U tom momentu nije bilo RL algoritama koji bi mogli izvršiti svoj zadatak na vrijeme, pa je iz tih razloga ekipa OpenAI Five odabrala ovu igru. Isti tim AI koji je pobijedio Dota 2, prvi je razvio i robotsku ruku koja se može usmjeriti u blok. [2]



## 4 Tehnike mašinskog učenja

### 4.1 Regresija

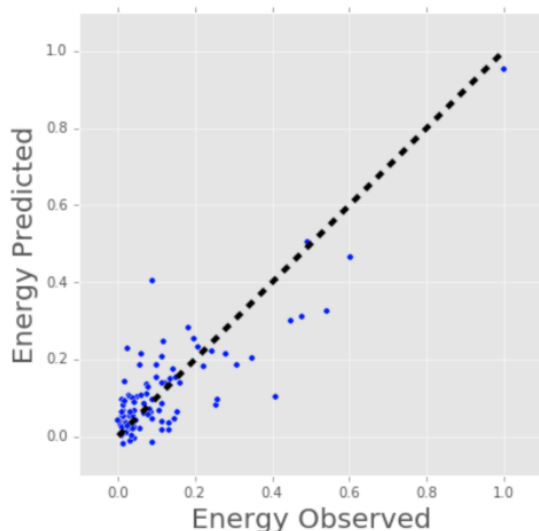
Regresijske tehnike spadaju u kategoriju superviziranih metoda mašinskog učenja. Ova tehnika pomaže u predviđanju (na osnovu brojčanih vrijednosti) budućeg trenda na bazi prethodnih podataka, kao što je na primjer, procjena cijene nekretnine na osnovu prethodnih cijena nekretnina u istoj ili sličnoj kategoriji. U ovoj skupini metoda, najjednostavnija je linearna regresija. Linearna regresija za modeliranje skupa podataka koristi matematičku jednačinu  $y = m * x + b$ . Linearni regresijski model trenira se na osnovu parova podataka (x, y) tako što se izračunava položaj i nagib linije koja minimizira ukupnu udaljenost između svih podataka i linija. Regresijska prava u ovom slučaju predstavlja liniju koja najbolje aproksimira opažanja u podacima. [4]

Primarni cilj regresije je modeliranje odnosa između varijabli koje se iterativno procesiraju pomoću mjere pogreške u predviđanjima koje je napravio model. Regresija je statistički proces pa se može upotrijebiti za određivanje razreda problema i klase algoritma.

Na primjer, ukoliko se linearna regresija koristi za predviđanje potrošnje energije zgrada, tada je potrebno prikupiti podatke kao što su starost zgrade, kvadratno postolje, broj priključene opreme itd., a sve u cilju kako bi se adekvatno mogla predvidjeti buduća potrošnja.

Budući da postoji više ulaza može se koristiti i višestruka varijabilna linearna regresija. Princip je isti kao kod jednostruke linearne regresije, s jednom značajnom razlikom, a to je da regresijska linija u ovom slučaju nije dvodimenzionalna nego višedimenzionalna. Na slici 4.1.1 prikazano je koliko se linearni regresijski model uklapa u stvarnu potrošnju energije zgrade.

Linearna regresija može se koristiti i za procjenu težine svakog faktora koji doprinosi konačnom predviđanju potrošene energije. Koristeći linearnu regresiju, može se lako odrediti da li su najvažnija starost, veličina ili visina.



**Slika 4.1.1** Prikaz odnosa potrošnje predviđene i posmatrane električne energije[4]

Opseg regresijskih tehnika kreće se od jednostavnih (poput linearne regresije) do složenih (poput regulisane linearne regresije, polinomne regresije, stabla odlučivanja, neuronskih mreža i td).

## 4.2 Klasifikacija

Klasifikacija, druga tehnika superviziranog mašinskog učenja predviđa ili objašnjava vrijednost klase. Na primjer, ova tehnika može pomoći kod predviđanja da li će kupac kupiti neki proizvod na online shopu ili neće kupiti. Izlaz može biti da ili ne: (DA) kupac će kupiti proizvod ili (NE) kupac neće kupiti proizvod. Metode klasifikacije ipak nisu ograničene na dvije klase, nego naprotiv broj klasa može biti onoliko koliko ima određenih vrsta u koje je potrebno klasificirati pojedine elemente ili proizvode.

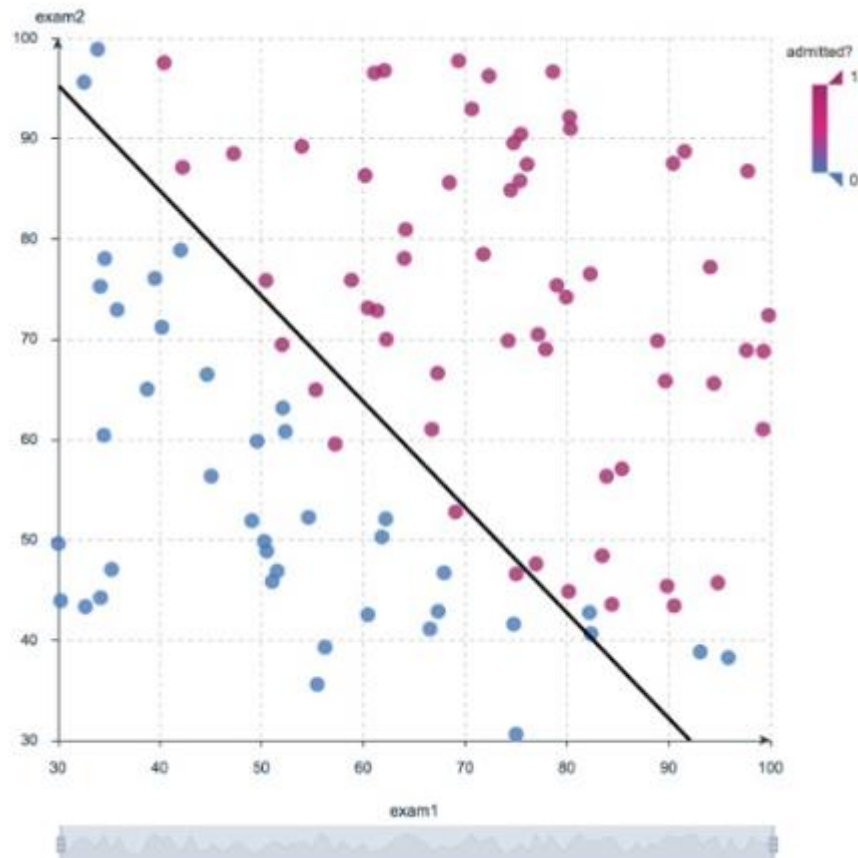
U ovom slučaju, izlaz će imati 3 različite vrijednosti:

- 1) slika sadrži automobil
- 2) slika sadrži kamion
- 3) slika ne sadrži automobil niti kamion.

Dobar primjer pojednostavljenog metoda klasifikacije predstavlja logistička regresija. Pored ove jednostavne metode, u ovu skupinu spadaju i metode kao što su: stabla odlučivanja, vektorske mašine za podršku i neuronske mreže.

Najjednostavniji algoritam klasifikacije je logistička regresija - što se čini kao metoda regresije, ali nije. Logistička regresija procjenjuje vjerovatnoću pojave događaja na bazi jednog ili više ulaza. Logistička regresija može uzeti kao ulaz bodove za studenta kako bi se procijenila vjerovatnoća da će student biti primljen na određeni fakultet.

Za predstavljanje vjerovatnoće nekog događaja koristi se broj između 0 i 1, gdje 1 predstavlja potpunu sigurnost. Ako je procijenjena vjerovatnoća za studenta veća od 0.5, predviđa se da će on ili ona biti primljen/a. Ako je procijenjena vjerovatnoća manja od 0.5, predviđa se da će on ili ona biti odbijeni. Na slici 4.2.1 grafički su prikazani rezultati logističke regresije za zadane rezultate ispita pri prijemu studenata na fakultet. Logistička regresija omogućava crtanje linije koja predstavlja granicu odluke.



**Slika 4.2.1.** Grafik logističke regresije [4]

Tehnike klasifikacije predviđaju diskretne odgovore - na primjer, je li adresa e-pošte originalna ili neželjena pošta ili je li kancer zloćudni ili dobroćudni. Modeli klasifikacije razvrstavaju ulazne podatke u kategorije. Tipične aplikacije uključuju medicinsko snimanje i prepoznavanje govora.

Klasifikacija se može koristiti za slučaj kada je moguće podatke označiti ili razdvojiti u određene grupe ili klase. Aplikacija za prepoznavanje ručno pisanog teksta koristi klasifikaciju za prepoznavanje slova i brojeva. U obradi slike i računarskoj viziji koriste se nesupervizirane tehnike prepoznavanja uzoraka za otkrivanje objekata i za segmentaciju slike.

### 4.3 Klasterizacija (grupisanje)

Metode klasterizacije pripadaju kategoriji nesuperviziranih metoda mašinskog učenja koje imaju za cilj grupisanje ili klasterizaciju podataka (opažanja) sa sličnim karakteristikama u iste ili slične klastere (grupe). Metode klasterizacije ne koriste izlazne informacije za obuku, već umjesto toga algoritam definira izlaz. Kod metoda klasterizacije može se koristiti vizualizacija samo za uvid u kvalitetu rješenja. [9]

Klasterizacija, poput regresije, opisuje klasu problema i klasu metoda.

Postoji više vrsta algoritama klasterizacije. Svi ovi algoritmi mogu se generalno klasificirati u dvije skupine i to na algoritme podjele (k- means, fuzzy c-means, k-median, ...) i hijerarhijske algoritme klasterizacije.

Ove metode koriste se za istraživačke analize podataka kao što je pronalaženje skrivenih obrazaca ili grupisanja u podacima. Aplikacije za klaster analizu uključuju analizu genske sekvence, istraživanje tržišta i prepoznavanje objekata.

Najpopularnija metoda klasterisanja je K-Means, gdje "K" predstavlja broj klastera koje korisnik želi formirati. Postoje različite tehnike za odabir vrijednosti K, poput metode (elbow).

Pseudokod K-means algoritma je dat u nastavku. [19]

K-Means ( $\{x_1, \dots, x_n\}, K$ )

$\{s_1, s_2, \dots, s_k\} \leftarrow \text{OdaberiSlučajno}(\{x_1, \dots, x_n\}, K)$

**for**  $k \leftarrow 1$  **to**  $K$

**do**  $\mu_k \leftarrow s_k$

**while** kriterij zaustavljanja ne bude ispunjen

**do for**  $k \leftarrow 1$  **to**  $K$

**do**  $\mu_k \leftarrow \{\}$

**for**  $n \leftarrow 1$  **to**  $N$

**do**  $j \leftarrow \arg \min_j |\mu_j - x_n|$

$\omega_j \leftarrow \omega_j \cup \{x_n\}$  (preraspodjela vektora)

**for**  $k \leftarrow 1$  **to**  $K$

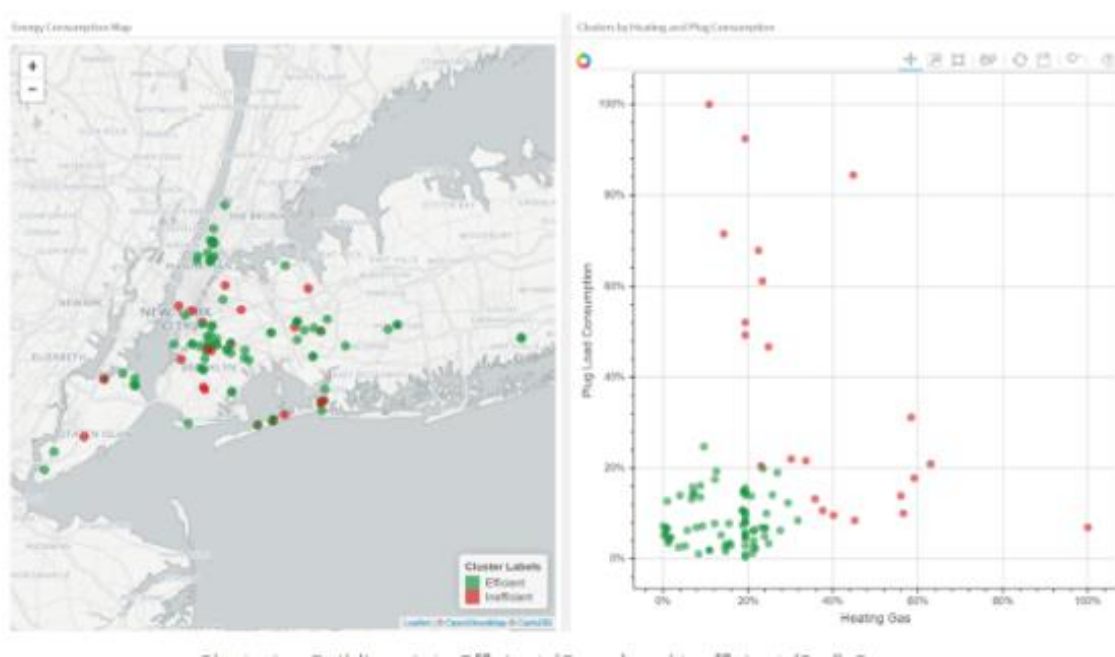
**do**  $\mu_k \leftarrow \frac{1}{\omega_k} \sum_{x \in \omega_k} x$  (ponovno računanje centroida)

**return**  $\{\mu_1, \dots, \mu_k\}$

Na sljedećem primjeru demonstriran je rad k-means algoritma na skupu podataka o zgradama

Skup podataka na osnovu kojeg je izvršena klasterizacija sastojao se od 4 vrste izmjenjenih vrijednosti, kao što su klimatizacija, priključena oprema, kućni plin i plin za grijanje. U ovom slučaju izabrana su samo dva klastera, u nadi da će algoritam izvršiti klasterizaciju podataka u dvije grupe, tj. skupine kako slijedi: grupa podataka koju sadrže energetski učinkovite zgrade, i druga skupina podataka koju čine neučinkovite zgrade.[4][5]

S lijeve strane se može vidjeti položaj zgrada, a s desne strane dvije od četiri dimenzije koje se koriste kao ulazi: priključena oprema i plin za grijanje.



**Slika 4.3.1.**Grupisanje zgrada u učinkovite (zelene) i neučinkovite (crvene) skupine[4]

Postoji više vrsta klastering metoda, kao što je klastering algoritma zasnovan na gustoći zatim mean shift clustering, agglomerative hierarchical clustering, expectation–maximization clustering. [4]



**Slika 4.3.2.** Prikaz grupisanja elemenata [9]

Na primjer, ako neki telekom operater želi optimizirati broj lokacija na kojima se nalaze GSM bazne stanice, tada se mogu koristiti klastering algoritmi za procjenu potrebnog broja baznih stanica, kako bi iste instalirali na optimalan i dobrovoljan način s aspekta pokrivenosti signala za određeno područje regiona. Prepotstavka je da jedan mobilni uređaj može komunicirati samo s jednom bazom stanicom. Algoritam klasteringa treba na osnovu ove informacije i informacije o maksimalnoj pokrivenosti signalom jedne bazne stanice, dati najbolji položaj za svaku baznu stanicu na način da se optimizira prijem signala za grupe korisnika.[4] [5]

#### 4.4. Redukovanje dimenzija

Kao što ime sugerše, koristi se smanjenje dimenzionalnosti kako bi se uklonili najmanje važni podaci (suvišne kolone) iz skupa podataka. U praksi se često nalaze skupovi podataka sa stotinama ili čak hiljadama kolona. Drugim riječima, redukovanje broja dimenzija (broja kolona u ovom slučaju) je veoma važan aspekt za jednostavniji način procesiranja podataka od strane metoda mašinskog učenja. Na primjer, slike mogu sadržavati hiljade piksela, gdje većina tih piksela nije važna s aspekta neke analize u kojoj se traži samo određeni objekat ili predmet koji ima veoma specifične vrijednosti piksela. U tim slučajevima potrebni su algoritmi za redukovanje dimenzija kako bi skup podataka bio pogodan za dalje analize i procesiranja.

Najpopularnija metoda smanjenja dimenzija je Principal Component Analysis (PCA ili analiza glavnih komponenti)[4]. Ta metoda smanjuje dimenziju prostora povećavajući razumljivost podataka, ali istovremeno minimizirajući gubitak informacija. To čini stvaranjem novih varijabli koje sukcesivno maksimiziraju varijansu. U slučaju kada je linearna korelacija značajna, tada je i broj redukovanih dimenzija u skupu podataka značajan, pa je samim tim značajno i smanjen broj dimenzija u skupu podatka bez pretjerano velikog gubitka informacije. Također može se izmjeriti stvarni opseg gubitka podataka i u skladu s tim prilagoditi.

Na slici 4.4.1 prikazna je analiza skupa podataka MNIST[5] rukom pisanih brojeva. MNIST sadrži hiljade uzoraka rukom pisanih brojeva od 0 do 9, koji se koriste za testiranje metoda mašinskog učenja za grupisanje i razvrstavanje. Svaki red skupa podataka predstavlja vektorski zapis izvorne slike (veličina  $28 \times 28 = 784$ ) i s klasom (nula, jedan, dva, tri, ..., devet). Zato se u ovom slučaju vrši redukovanje dimenzija podataka s 784 px na 2, tj. na 2 dimenzije koje su veoma pogodne za vizuelnu prezentaciju.

Druga popularna metoda je t-stohastičko umetanje susjeda (t-SNE), što čini nelinearno redukovanje dimenzionalnosti. Ljudi obično koriste t-SNE za vizuelizaciju podataka, ali može se koristiti i za zadatke mašinskog učenja poput redukovanja prostora i grupisanja.

Redukovanje podataka u dvije dimenzije omogućava da se vizuelno lakše sagledaju podaci.[4] [5]



**Slika 4.4.1.** Analiza baze podataka MNIST rukom pisanih cifara[4]

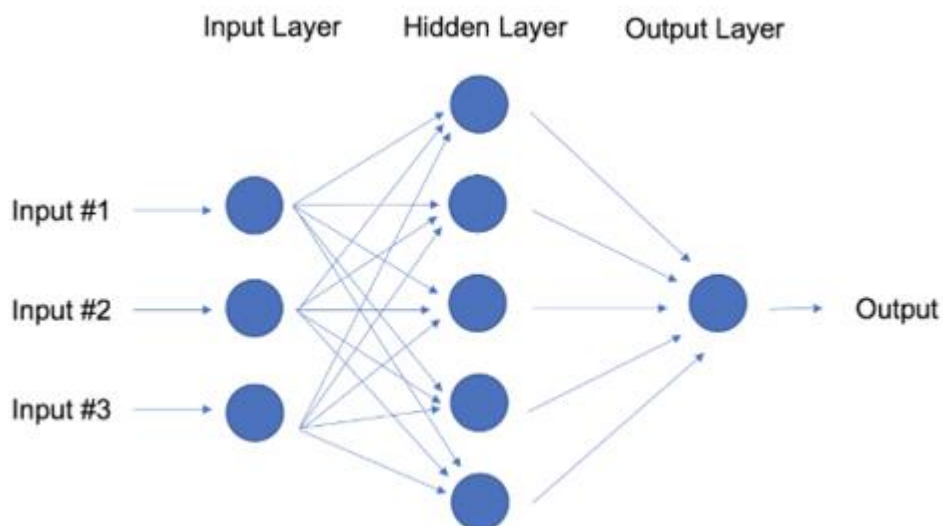
## 4.5 Metode cjelina

Ensemble methods (metode cjelina) pripadaju kategoriji superviziranih metoda mašinskog učenja. Ova metoda je nastala zbog ljudi i njihovih potreba, pa se može objasniti primjerom iz života. Ukoliko neka osoba odluči napraviti bicikl, zato što ta osoba nije zadovoljna ponudom na tržištu, prva stvar koju će učiniti je pronalazak najboljih dijelova koji su potrebni za formiranje bicikla. Jednom kada se sastave svi dijelovi, rezultujući bicikl zasjenit će sve ostale opcije koje su bile ponuđene na tržištu.

Ova metoda koristi istu ideju kombinacije nekoliko predviđenih modela kako bi se dobila kvalitetnija predviđanja nego što svaki od modela može pružiti samostalno. Primjer takvog modela je Random Forest algoritam koji kombinuje mnoštvo stabala odlučivanja istreniranih s različitim uzorcima skupa podataka. Kao rezultat dobijena je veća kvaliteta predviđanja Random Forest algoritmom od kvalitete predviđanja jednim stablom odlučivanja. Ova metoda predstavlja način da se smanji odstupanje i pristranost jednog modela mašinskog učenja. To je važno jer svaki dati model može biti tačan pod određenim uvjetima, ali netačan pod drugim uvjetima, zato je potrebno ispitati više različitih modela. Kombinacijom dva modela kvaliteta prognoza se uravnotežuje. Najpopularniji algoritmi cjelina su Random Forest, XGBoost i LightGBM. [4] [9]

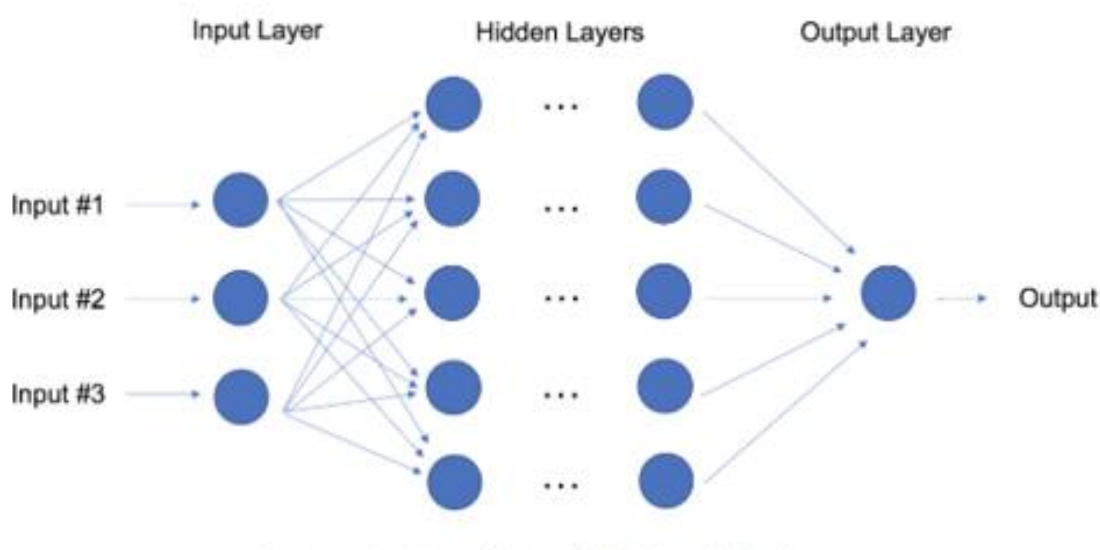
## 4.6 Neuronske mreže i duboko učenje (engl. deep learning[18])

Za razliku od linearnih i logističkih regresija koje su primjer linearnih modela, cilj neuronskih mreža je iskoristiti nelinearne obrasce u podacima dodavanjem slojeva parametara u modelu. Na slici 4.6.1 prikazana je jednostavna neuronska mreža koja ima tri ulaza, jedan skriveni sloj s pet parametara i izlazni sloj.



**Slika 4.6.1.** Neuronske mreže sa skrivenim slojem[4]

Struktura neuronskih mreža je dovoljno fleksibilna da izgradi linearnu i logističku regresiju. Izraz duboko učenje (deep learning) [18] dolazi od neuronske mreže koja sadrži puno skrivenih slojeva. Ta mreža prikazana je na slici 4.6.2 i obuhvata široki raspon arhitektura.



**Slika 4.6.2.** Neuronske mreže sa više skrivenih slojeva[4]



Naročito je teško pratiti razvoj dubokog učenja, dijelom zato što su istraživačke i industrijske zajednice udvostručile svoje napore, svakodnevno stvarajući nove metodologije.

Za najbolju izvedbu, tehnike dubokog učenja, zahtijeva se puno podataka i puno računarske snage jer ova metoda vrši samoinicijalizaciju parametara u velikom broju arhitektura. Vrlo brzo postaje jasno zašto stručnjaci dubokog učenja trebaju vrlo moćne računare poboljšane grafičkim procesorskim jedinicama.

Konkretno, tehnike dubokog učenja izuzetno su uspješne u područjima vida kao što su klasifikacija slika, teksta, zvuka i videa. Najčešći softverski paketi za duboko učenje su Tensorflow i PyTorch. [4] [5]

## 4.7 Prijenosno učenje

Transfer learning ili prijenosno učenje zasniva se na prijenosu znanja sa jednog modela na drugi. Na primjer, naučnik koji mjesecima vježba visokokvalitetni model da klasificira slike u majice, košulje i polo majice i ukoliko taj naučnik želi da izgradi sličan model tako da klasificira slike haljina u jeans, kožne i cvjetne haljine, koristit će znanja iz prvog modela. Uz pomoć prijenosnog učenja znanje koje je ugrađeno u prvi model može se lako primijeniti na drugi model.

Prijenosno učenje odnosi se na ponovno korištenje prethodno istrenirane neuronske mreže i prilagođavanje novom, ali sličnom zadatku. Konkretno, nakon što se dobije trenirana neuronska mreža koristeći podatke za prvi zadatak, može se prenijeti dio treniranih slojeva i kombinovati ih sa novim podacima zadatka. Dodavanjem nekoliko novih slojeva, nova neuronska mreža može se brzo prilagoditi novom zadatku.

Glavna prednost prijenosnog učenja je u tome što treba manje podataka za trening neuronske mreže, a to je posebno važno jer je obuka za algoritme dubokog učenja skupa. Uz prijenosno učenje glavni resursi kao što su vrijeme i novac mogu se sačuvati.

Ukoliko se pretpostavi da se za model košulje koristi neuronska mreža s 20 skrivenih slojeva, eksperimentalnim putem, može se zaključiti da se 18 slojeva modela košulje može kombinovati s jednim novim slojem parametara da bi se dobio model za klasifikaciju slike haljina. Ulazi i izlazi iz ova dva modela su različiti, ali slojevi koji se mogu ponovno upotrijebiti mogu rezimirati informacije koje su relevantne za oba modela.

Prijenosno učenje postalo je sve popularnije i sada je na raspolaganju mnogo solidnih unaprijed istreniranih modela za uobičajene zadatke dubokog učenja poput klasifikacije slika i teksta.[5][6]

## 4.8 Obrada prirodnog jezika

Ogroman postotak svjetskih podataka i znanja nalazi se u nekom obliku ljudskog jezika. Očito je da računari još ne mogu u potpunosti razumjeti ljudski tekst, ali mogu se osposobiti za obavljanje određenih zadataka. Na primjer, telefoni se mogu osposobiti za automatsko dovršavanje tekstualnih poruka ili ispravljanje pogrešno napisanih riječi. Čak se i mašina može osposobiti da jednostavno razgovara sa čovjekom.

Obrada prirodnog jezika (Natural Language Processing-NLP)[9] sama po sebi nije metoda mašinskog učenja, već je široko korištena tehnika pripreme teksta za mašinsko učenje. Postoji tona tekstualnih dokumenata u raznim formatima (riječi, internetski blogovi i td.), većina ovih tekstualnih dokumenata je puna pogrešaka pri upisu, nedostajućih znakova i drugih riječi koje je potrebno filtrirati. Trenutno je najpopularniji paket za obradu teksta NLTK (Natural Language ToolKit), kreiran od strane istraživača na Stanfordu. [4]

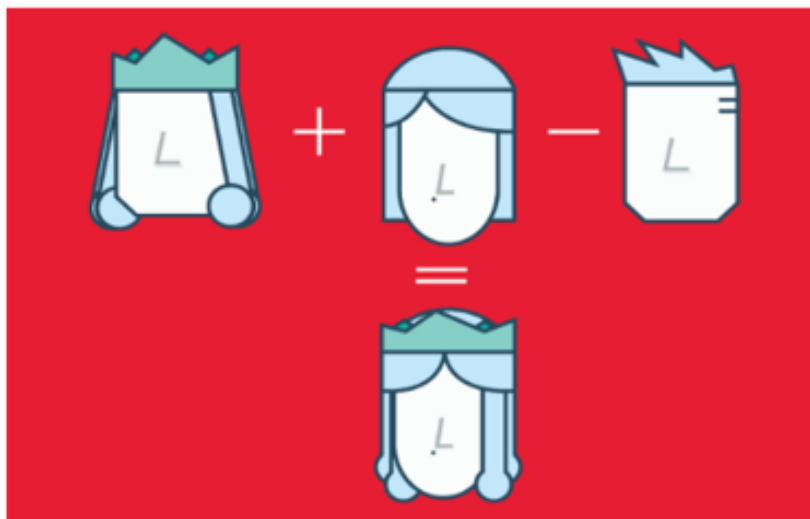
Najjednostavniji način preslikavanja teksta u numerički prikaz je izračunavanje učestalosti svake riječi unutar svakog tekstualnog dokumenta. Dobar primjer je matrica cijelih brojeva gdje svaki red predstavlja tekstualni dokument, a svaka kolona predstavlja riječ. Ova matrična reprezentacija frekvencija riječi uobičajeno se naziva terminsko-frekvencijska matrica (TFM). Također se na osnovu te matrice može formirati još jedan popularni matrični prikaz tekstualnog dokumenta tako što će se svaki zapis na matrici podijeliti s težinom koliko je svaka riječ važna u čitavom kolekciji dokumenata. Ova metoda se naziva metoda obrnute frekvencije dokumenata (TFIDF) i ona obično bolje funkcionise za trening mašinskog učenja.[5] [6] [9]

## 4.9 Umetanja riječi

TFM i TFIDF iz prethodnog poglavlja su numerički prikazi tekstualnih dokumenata koji samo tekstualnu i težinsku frekvenciju predstavljaju kao tekstualne dokumente. Suprotno tome, umetanje riječi može promijeniti kontekst riječi u dokumentu. U kontekstu riječi, ta umetanja mogu kvantificirati sličnost riječi, što zauzvrat omogućava aritmetiku s riječima.

Word2Vec je metoda koja se temelji na neuronskim mrežama i koja preslikava riječi u kolekciju numeričkih vektora. Potom te vektore koristi za pronalaženje sinonima, izvođenje aritmetičkih operacija sa riječima ili za predstavljanje tekstualnih dokumenata. U tom slučaju uzima se srednja vrijednosti svih vektora riječi u dokumentu. Ukoliko se koristi dovoljno velika kolekcija tekstualnih dokumenata za proces umetanja riječi gdje su na primjer ponuđene riječi kralj, kraljica, muškarac i žena – dio te kolekcije, tada se vrši procjena vektora. Vektor ('riječ') je numerički vektor koji predstavlja riječ 'riječ' u procjeni vektora 'žena' može se izvesti aritmetička operacija sa vektorima.

$$\text{vector}('king') + \text{vector}('woman') - \text{vector}('man') \sim \text{vector}('queen')$$



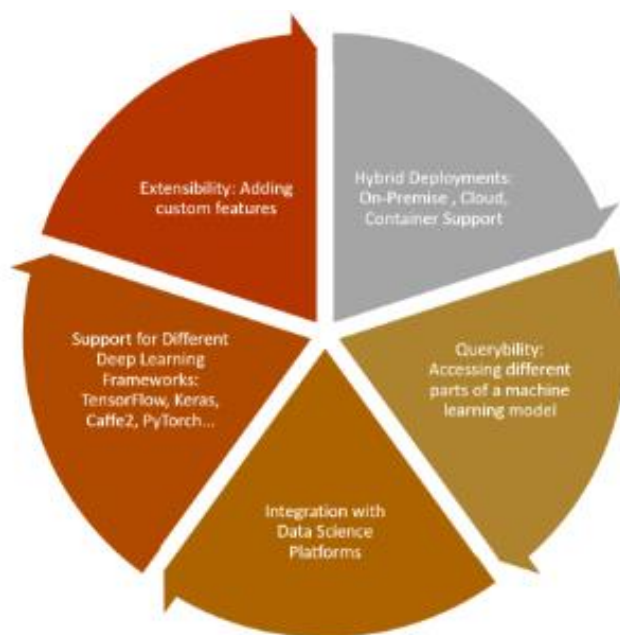
**Slika 4.9.1.** Aritmetika sa vektorima riječi [4]

Reprezentacije riječi omogućavaju pronalaženje sličnosti računanjem kosinusne teoreme između vektorskog predstavljanja dvije riječi. Pomoću kosinusne teoreme mjeri se ugao između dva vektora. Ova tehnika se često koristi prije primjene nekog konkretnog algoritma mašinskog učenja. [4] [5]

## 5 Baza podataka za mašinsko učenje (engl. Machine learning database – MLDB)

Rješenja za mašinsko učenje u stvarnom svijetu rijetko su samo pitanje izgradnje i testiranja modela. Često najteži problem koji se može vidjeti u računarstvu je upravljanje i automatizacija životnog ciklusa modela mašinskog učenja od treniranja do optimizacije. Za kontrolu životnog ciklusa modela, naučnici s podacima moraju ustrajati i ispitivati njegovo stanje. Ovaj bi se problem mogao činiti trivijalnim dok se ne uzme u obzir da bilo koji prosječni model dubokog učenja može uključivati stotine skrivenih slojeva i milione međusobno povezanih čvorova. Pohranjivanje i pristup mnogobrojnim graficima računanja daleko je od trivijalnog. Većinu vremena timovi za nauku o podacima provode pokušavajući prilagoditi NoSQL modele mašinskog učenja bazi podataka. Radi mnogih problema naučnici su došli do zaključka da rješenja za mašinsko učenje trebaju novu vrstu baze podataka. [10]

MLDB je baza podataka dizajnirana za eru mašinskog učenja. Platforma je optimizirana za pohranjivanje, transformisanje i navigaciju grafikona računanja koji predstavljaju strukturu mašinskog učenja poput duboke neuronske mreže. Platforme u cloud mašinskom učenju poput AWS SageMaker ili Azure ML već uključuju modele grafikona mašinskog učenja, ali ipak postoji dosta zahtjeva stvarnih rješenja mašinskog učenja koje mogu imati koristi od stvarne baze podataka. [10]

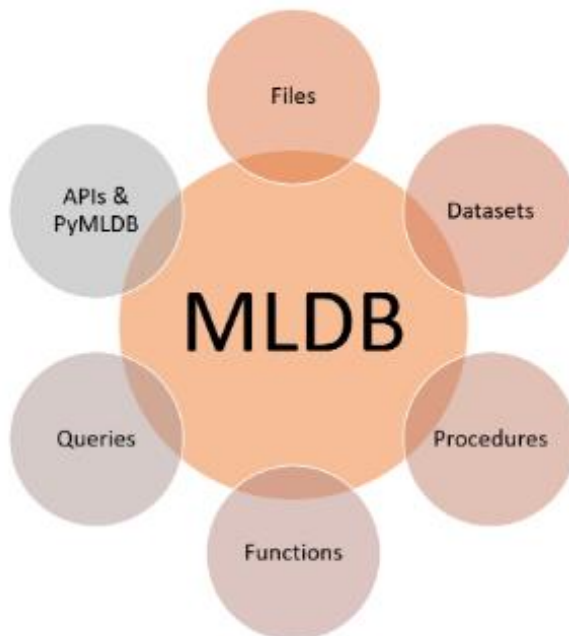


**Slika 5.1.** Životni ciklus MLDBa [10]

MLDB nudi open-source bazu podataka za pohranu i određivanje upita mašinskog učenja. Platforma je prvi put izvedena u sklopu Datacrat-a, a nedavno je kompanija AI powerhouse Elementai dobila relevantnu validaciju čitave baze podataka. MLDB je dostupan u različitim oblicima, kao što su usluge u Cloudu, VirtualBox VM ili Docker instance koji se mogu implementirati na bilo kojoj kontejnerskoj platformi. [10]

## 5.1 Arhitektura MLDB-a

Arhitektura MLDB-a kombinuje različite elemente životnog ciklusa mašinskog učenja kako bi se postigao traženi efekat baze podataka. Tehnički se model MLDB-a može sažeti u šest jednostavnih komponenti: datoteke, skupovi podataka, procedure, funkcije, upiti i API-ji.



**Slika 5.1.1.** Prikaz arhitekture MLDBa [10]

### Datoteke

Datoteke predstavljaju jedinicu apstrakcije u MLDB arhitekturi. U MLDB modelu, datoteke se mogu koristiti za učitavanje podataka za modele, kao parametri za funkciju ili za zadržavanje određenog skupa podataka. MLDB podržava izvornu integraciju s popularnim datotečnim sistemima kao što su HDFS i S3. [10]

### Skupovi podataka

MLDB skupovi podataka predstavljaju glavnu jedinicu podataka koju koriste modeli mašinskog učenja. Strukturno su skupovi podataka bez shema, samo imaju nazive, koji se nalaze u ćelijama na sjecištu redova i kolona. Tačke podataka sastoje se od vrijednosti i vremenske oznake. Svaka tačka koja pripada podacima može se predstaviti kao (red, kolona, vremenska oznaka, vrijednost) zbir, a skupovi podataka mogu se smatrati trodimenzionalnim matricama. Skupovi podataka mogu se formirati putem API-ja MLDB-a, a mogu se također učitati u datoteke ili pohraniti putem procedura. [10]

## Procedure

U MLDB-u se procedure koriste za implementaciju različitih aspekata modela mašinskog učenja poput treninga ili transformacije podataka. S tehničkog stajališta, procedure su imenovani programi za višestruku upotrebu koje se koriste za implementaciju dugotrajnih batch operacija bez povratnih vrijednosti. Procedure općenito nadilaze skupove podataka i mogu se konfigurirati putem SQL izraza. Izlazi iz procedura mogu uključivati skupove podataka i datoteke. [10]

## Funkcije

MLDB funkcije koriste apstraktne podatke radi izračunavanja koja se koriste u procedurama. Funkcije su imenovani programi za višestruku upotrebu koji se koriste za implementiranje izraza koji mogu prihvatiti ulazne i vratiti izlazne vrijednosti. MLDB funkcije objedinjuju SQL izraze koji se bave specifičnim računanjem. [10]

## Upiti [10]

Jedna od glavnih prednosti MLDB-a je ta što koristi SQL kao mehanizam za upis podataka pohranjenih u bazu podataka. Platforma podržava prilično cjelovitu gramatiku zasnovanu na SQL-u, koja uključuje poznate konstrukcije poput SELECT, WHERE, FROM, GROUP BY, ORDER BY i mnogih drugih. Na primjer, u MLDB-u može se koristiti SQL upit za treniranje modela klasifikacije slike:

```
mldb.query("SELECT * FROM images LIMIT 3000") [10]
```

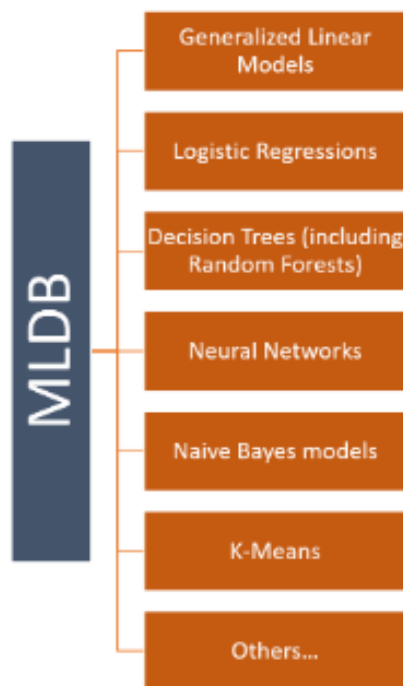
## API-ji i Pymldb [10]

Sve mogućnosti MLDB-a izložene su putem jednostavnog REST API-ja. Platforma također uključuje pymldb, Python biblioteku koja apstraktno prikazuje mogućnosti API-ja koristeći prijateljsku sintaksu. Sljedeći kôd pokazuje kako koristiti pymldb za stvaranje i za upite baze podataka.

```
from pymldb import Connection
mldb = Connection("http://localhost")
mldb.put( "/v1/datasets/demo", {"type": "sparse.mutable"})
mldb.post("/v1/datasets/demo/rows", {"rowName": "first",
"columns": [{"a", 1, 0}, {"b", 2, 0}]})
mldb.post("/v1/datasets/demo/rows", {"rowName": "second",
"columns": [{"a", 3, 0}, {"b", 4, 0}]})
mldb.post("/v1/datasets/demo/commit")
df = mldb.query("select * from demo")
print type(df)
```

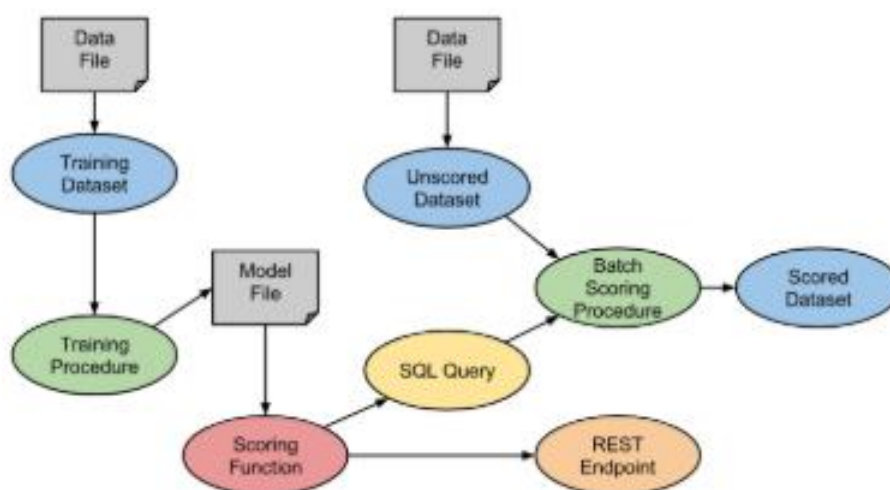
## 5.2 Podrška za algoritme mašinskog učenja

MLDB pruža podršku za veliki broj algoritama koji se mogu koristiti iz procedura i funkcija. Platforma pruža podršku različitim tehnikama za duboko učenje poput TensorFlow. [11]



**Tabela 5.2.1.** Prikaz tehnika MLDBa [10]

Način rada MLDB-a zasniva se na detaljnoj analizi modela i pažljivo odrabnim koracima kao što je to prikazano na slici 5.2.2. [10] [11]



**Slika 5.2.2.** Tok rada mašinskog učenja[10]

- Proces započinje s datotekom u koju se učitava skup trenirajućih podataka

Šejla Pljakić: „Analiza recenzija putovanja koristeći ML bazu podataka za mašinsko učenje“

- Vršiti se procedura za kreiranje modela datoteke
- Datoteka modela koristi se za parametriranje funkcije bodovanja
- Ta funkcija je odmah dostupna putem REST krajnje tačke
- Također ta funkcija je dostupna i putem SQL upita
- Procedura za bodovanje koristi SQL za primjenu funkcije bodovanja na ciljane podatke, nakon čega algoritam završava sa traženim podacima

Ključne karakteristike MLDB-a koje odgovaraju nizu aplikacija su:

**Brzina:** Proces treniranja, modeliranja i pronalaska podataka u MLDB-u zahtjeva dobre performanse. MLDB ima veliku moć obrade u poređenju sa bibliotekama H2O, Scikit-Learn ili Spark MLlib, koje su poznate kao istaknute biblioteke mašinskog učenja. [11]

**Skalabilnost:** MLDB podržava vertikalno skaliranje s većom učinkovitošću, tako da se svi memorijski moduli kao i jezgre mogu istovremeno koristiti bez ikakvih problema s kašnjenjem ili performansama.

**Open-source proizvodi:** Zajedničko izdanje MLDB-a dostupno je i distribuirano u vlasništvu i hostingu GitHub-a.

**SQL podrška:** Ovo čini MLDB vrlo korisnim, zajedno s podrškom za veliku obradu podataka. MLDB može obraditi, istrenirati i predvidjeti podatke pomoću tablica baza podataka koje imaju milione redova, uz istovremenu obradu.

**Mašinsko učenje:** MLDB je razvijen za aplikacije i modele mašinskog učenja visokih performansi. Podržava duboko učenje s graphicima TensorFlow-a koji ga čine superiornim u otkrivanju znanja.

**Jednostavnost implementacije:** Postoje instalacijski paketi za više platformi i programska okruženja, uključujući Jupyter, Docker, JSON, Cloud, Hadoop i mnoge druge.

**Kompatibilnost i integracija:** MLDB omogućava visok stepen kompatibilnosti s različitim aplikacijskim programskim okruženjima (API-ima) i modulima, uključujući JSON, REST i slojeve na bazi Pythona.

**Razvoj i pokretanje:** MLDB se lako pokreće na HTTP portu koji omogućava jednostavno okruženje i brzu implementaciju.

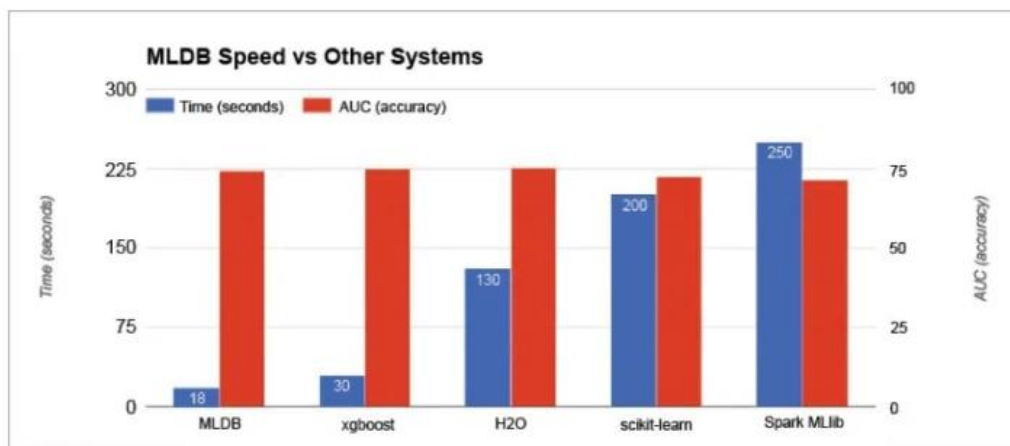
**MongoDB i NoSQL podrška:** Interfejs MongoDB i MLDB se može razviti za podršku MLDB SQL upita. Ovi SQL upiti mogu se izvoditi na MongoDB kolekcijama, koje MLDB-u daju više ovlasti za interakciju sa NoSQL bazama podataka za nestrukturirane i heterogene skupove podataka.

Na slici 5.2.3 prikazane su performanse MLDB-a u poređenju sa drugim bibliotekama. Izvođenje Random Forest pristupa tačnije takvih 100 vrsta vrši se na 1 milion redova s jednim čvorom pomoću MLDB-a i drugih biblioteka. Iz grafičkih rezultata vidljivo je da je MLDB u poređenju sa drugim bibliotekama bolji, treba manje vremena, a njegova se tačnost dobro



Šejla Pljakić: „Analiza recenzija putovanja koristeći ML bazu podataka za mašinsko učenje“

uspoređuje s ostalim bibliotekama mašinskog učenja. Učinkovitost MLDB-a usporediva je s performansama xgboost, H2O, Scikit-Learn i Spark MLlib. [10] [11] [12]

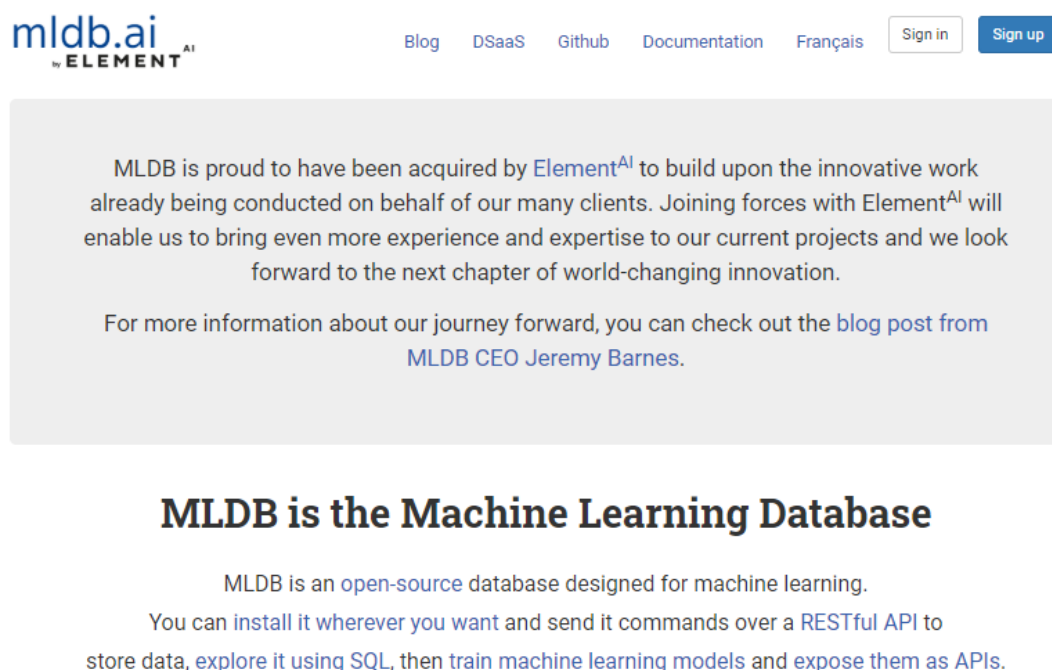


Slika 5.2.3. Učinkovitost MLDBa [11]

### 5.3 Instaliranje MLDB-a na različitim platformama

MLDB nudi internetsko okruženje za najlakšu primjenu i praktično iskustvo. Besplatna sesija MLDB-a može biti aktivna 90 minuta nakon prijave (registracije) na <https://mldb.ai/#signup>. Dostupno je puno demonstracija i dokumentacije tako da se MLDB na Cloudu može koristiti bez instalacije na lokalni sistem. Čak se i samostalno stvoreni podaci mogu učitati u ovu sesiju koja se održava.

Postoje dva izdanja MLDB-a koja su besplatna i distribuirana su kao izdanja zajednice i poduzeća. Za pokretanje MLDB Enterprise Edition verzije, potrebno je unijeti licencni ključ za aktiviranje softvera. Licencni ključ može se napraviti za korisnike prilikom prijave na [https://mldb.ai/#license\\_management](https://mldb.ai/#license_management) i ispunjavanja potrebnih uslova u obrascu za registraciju. Na slici 5.3.1 i 5.3.2 prikazana je zvanična stranica MLDB-a kao i prikaz verzija instalacije. [12]



**Slika 5.3.1.** Oficijalna stranica MLDB baze podataka[12]

	Community edition	Enterprise edition (Free trial)
MLDB	Available	Available
Licensing	Apache License v2.0	Non-commercial
Cost / Pricing	Free	Free
Issues and support	GitHub issues	MLDB support
Download Instructions	<a href="https://github.com/mldbai/mldb/blob/master/Building.md">https://github.com/mldbai/mldb/blob/master/Building.md</a>	<a href="https://docs.mldb.ai/doc/builtin/Running.md.html#packages">https://docs.mldb.ai/doc/builtin/Running.md.html#packages</a>

**Tabela 5.3.2.** Prikaz verzija instalacije MLDBa [11]

MLDB je jedan od prvih primjera baza podataka dizajniran od početka kako bi se omogućila rješenja mašinskog učenja. Platforma se još uvijek može puno poboljšati radi podrške modernim tehnikama mašinskog i dubokog učenja. [12]

## 6 Opis skupa podataka recenzije putovanja

Prije izrade praktičnog dijela u svrhu rješavanja početnog problema, potrebno je navesti nekoliko karakteristika skupa podataka koji će se koristiti u nastavku. Skup podataka koji se koristi su recenzije u 10 kategorija za destinacije unutar istočne Azije. Svaki putnik rangirao je uslugu sljedećim vrijednostima: odlično(4), vrlo dobro(3), prosječno(2), siromašno(1), užasno(0). [16]

<b>Karakteristike seta podataka:</b>	Multivarijabilni, Tekstualni	<b>Broj instanci:</b>	980	<b>Područje:</b>	Istočna Azija
<b>Karakteristike atributa:</b>	realni	<b>Broj atributa:</b>	11	<b>Datum objave</b>	2018-12-19
<b>Tehnike mašinskog učenja:</b>	Klasifikacija, Klastering	<b>Vrijednosti koje nedostaju:</b>	N/A	<b>Broj otvaranja na webu:</b>	59630

**Tabela 6.1.** Karakteristike skupa podataka

Podaci o atributima:

Atribut 1: Jedinstveni korisnički ID

Atribut 2: Prosječna ocjena korisnika o umjetničkim galerijama

Atribut 3: Prosječna ocjena korisnika o plesnim klubovima

Atribut 4: Prosječna ocjena korisnika o barovima za osvježenje

Atribut 5: Prosječna ocjena korisnika o restoranima

Atribut 6: Prosječna ocjena korisnika o muzejima

Atribut 7: Prosječna ocjena korisnika o odmaralištima

Atribut 8: Prosječna ocjena korisnika o parkovima / izletištima

Atribut 9: Prosječna ocjena korisnika o plažama

Atribut 10: Prosječna ocjena korisnika o pozorištima

Atribut 11: Prosječna ocjena korisnika o vjerskim institucijama

## 7 Praktični rad koristeći bazu MLDB i programski jezik Python

Za praktičan rad duži od 90 minuta potrebno je pratiti određene korake.

Korak 1: Instalirati Oracle VM Virtual Box

Korak 2: Importovati mldb.ova fajl unutar virtuelne mašine sa zvanične stranice [20]

Korak 3: Pokrenuti fajl unutar virtuelne mašine i koristeći browser pristupiti MLDB bazi na localhost:8080

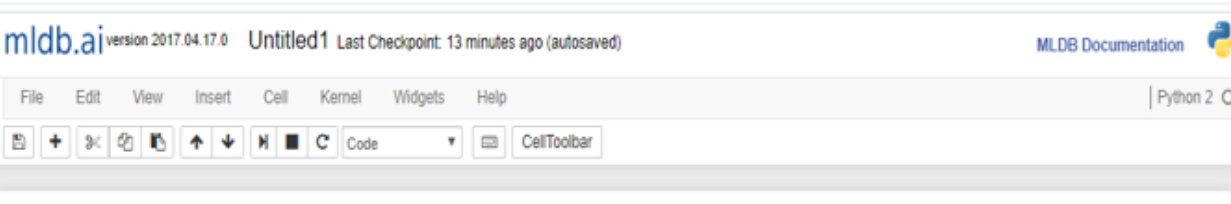
Korak 4: Kreirati novi folder i novi fajl Python2 unutar baze podataka

Korak 5: Izvršiti importovanje svih potrebnih biblioteka kao i učitavanje skupa podataka

Korak 6: Izvršiti učitavanje csv fajla tj. ciljanog skupa podataka koristeći UCI zvaničnu stranicu [16]

Korak 7: Izvršiti čitanje fajla u posebnu varijablu data (slika 7.1 linija 2)

Korak 8: Prikazati čitavu tabelu pozivom data



The screenshot shows the MLDB.ai web interface. At the top, it says 'mldb.ai version 2017.04.17.0' and 'Untitled1 Last Checkpoint: 13 minutes ago (autosaved)'. There is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with icons for file operations and a 'Code' dropdown menu. The main area displays two Python code cells. The first cell, labeled 'In [1]:', contains imports for pandas, numpy, sklearn preprocessing, matplotlib, and seaborn, along with setting the font size and style. The second cell, labeled 'In [2]:', contains code to read a CSV file, drop missing values, and print the shape and columns of the data. The output of the second cell is shown below the code, displaying the shape (980, 11) and a list of column names.

```
In [1]: import pandas as pd
import numpy as np
from sklearn import preprocessing
import matplotlib.pyplot as plt
plt.rc("font", size=14)
from sklearn.linear_model import LogisticRegression
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)

In [2]: data=pd.read_csv('tripadvisor_review.csv',header=0)
data=data.dropna()
print(data.shape)
print(list(data.columns))

(980, 11)
['User ID', 'art galleries', 'dance clubs', 'juice bars', 'restaurants', ' museums', 'resorts', 'parks/picnic spots', 'beache
s', 'theaters', 'religious institutions ']
```

**Slika 7.1.** Dio programskog koda za učitavanje skupa podataka u bazu

In [3]: data

Out[3]:

	User ID	art galleries	dance clubs	juice bars	restaurants	museums	resorts	parks/picnic spots	beaches	theaters	religious institutions
0	User 1	0.93	1.80	2.29	0.62	0.80	2.42	3.19	2.79	1.82	2.42
1	User 2	1.02	2.20	2.66	0.64	1.42	3.18	3.21	2.63	1.86	2.32
2	User 3	1.22	0.80	0.54	0.53	0.24	1.54	3.18	2.80	1.31	2.50
3	User 4	0.45	1.80	0.29	0.57	0.46	1.52	3.18	2.96	1.57	2.86
4	User 5	0.51	1.20	1.18	0.57	1.54	2.02	3.18	2.78	1.18	2.54
5	User 6	0.99	1.28	0.72	0.27	0.74	1.26	3.17	2.89	1.66	3.66
6	User 7	0.90	1.36	0.26	0.32	0.86	1.58	3.17	2.66	1.22	3.22
7	User 8	0.74	1.40	0.22	0.41	0.82	1.50	3.17	2.81	1.54	2.88
8	User 9	1.12	1.76	1.04	0.64	0.82	2.14	3.18	2.79	1.41	2.54
9	User 10	0.70	1.36	0.22	0.26	1.50	1.54	3.17	2.82	2.24	3.12

**Tabela 7.1.** Prikaz tabele skupa podataka

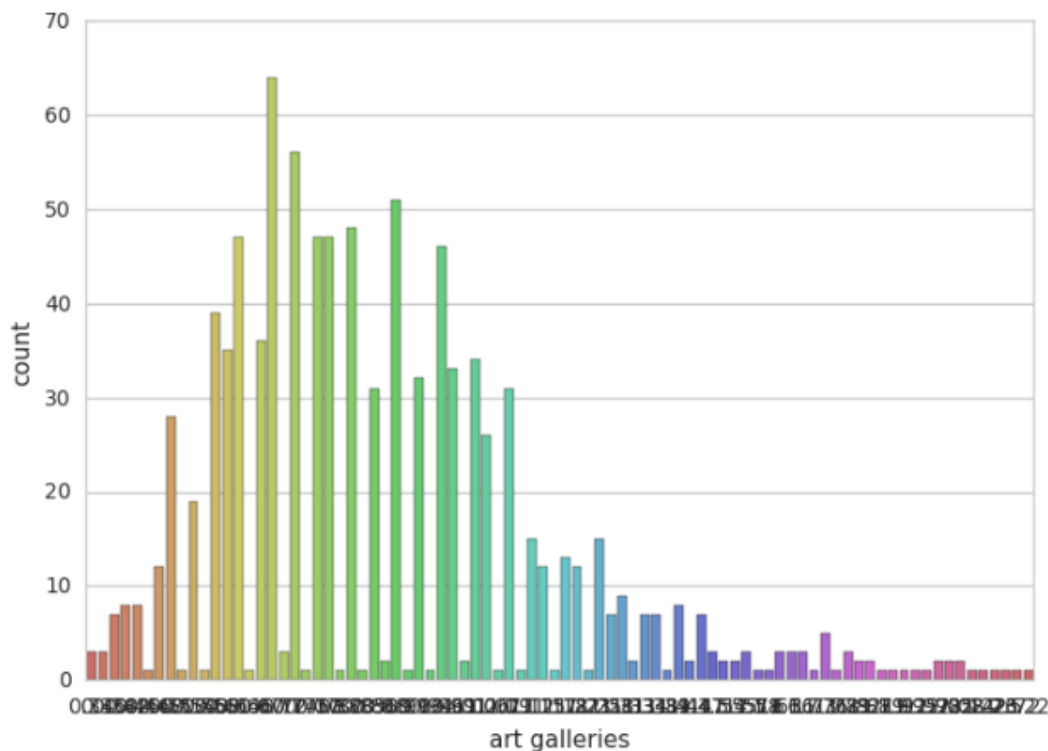
Postoji 10 kategorija mjesta u istočnoj Aziji koji imaju recenzije od 0-4.

Najbolje bi bilo pokušati sa tehnikama mašinskog učenja kao što su klasifikacija i klasterizacija. Grupisanjem aktivnosti će se najbolje procijeniti koja je aktivnost tj. mjesto najpoželjnije za posjetiti u istočnoj Aziji.

S obzirom da se klasifikacija koristi ako se podaci mogu označiti, kategorisati ili razdvojiti u određene grupe ili klase tada postaje jasan odabir ove metode.

Nakon grafičke reprezentacije ovih kategorija izdvojene su kategorije od značaja i moguće rješenje. Na narednim slikama prikazani su pojedinačni grafici ovih kategorija. [12]

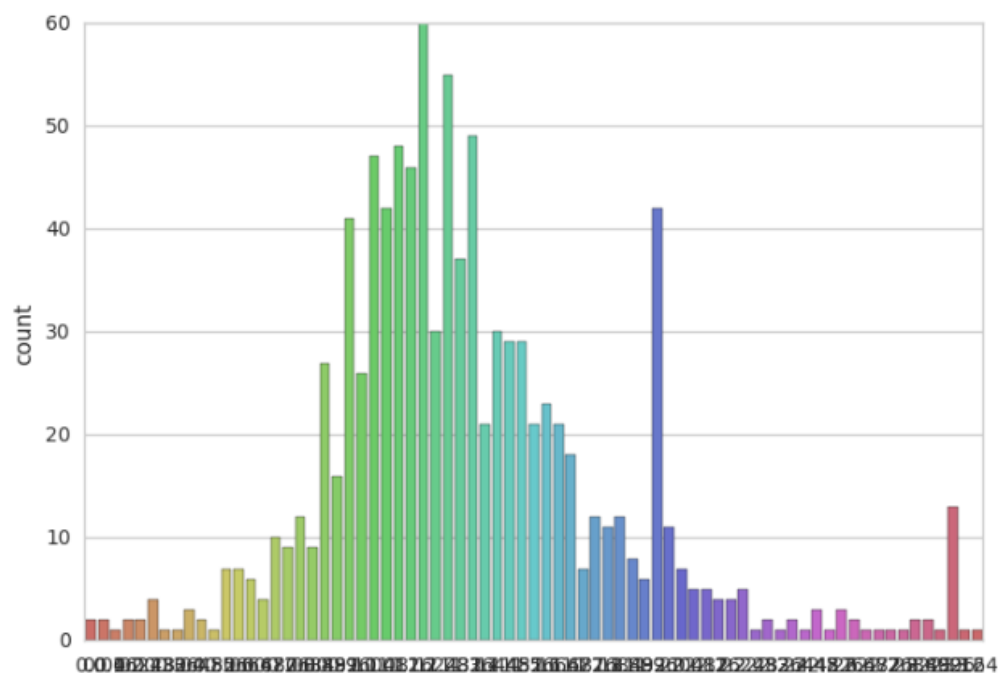
```
In [15]: sns.countplot(x='art galleries', data=data, palette='hls')  
plt.show()  
plt.savefig('count_plotart')
```



**Slika 7.2.** Prikaz broja umjetničkih galerija

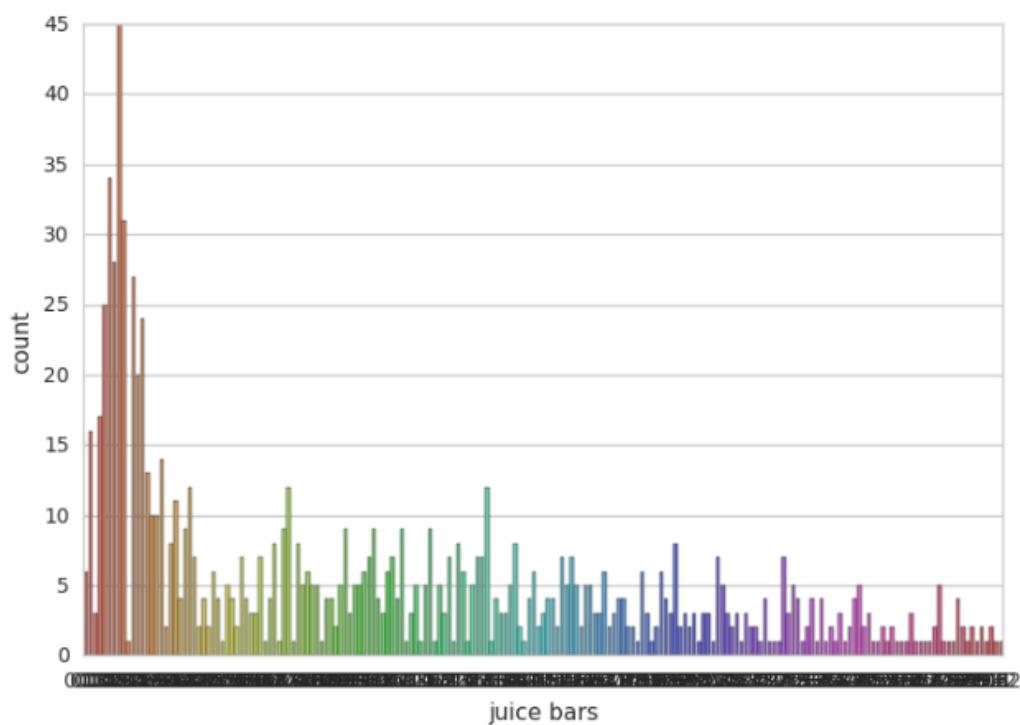
Umjetničke galerije se nalaze u rasponu vrijednosti od 0-3 gdje je veliki broj ljudi ocjenilo ovu aktivnost, ali ipak ima puno oscilacija u recenzijama na ovom grafiku.

```
In [16]: sns.countplot(x='dance clubs', data=data, palette='hls')
plt.show()
plt.savefig('count_plotclubs')
```



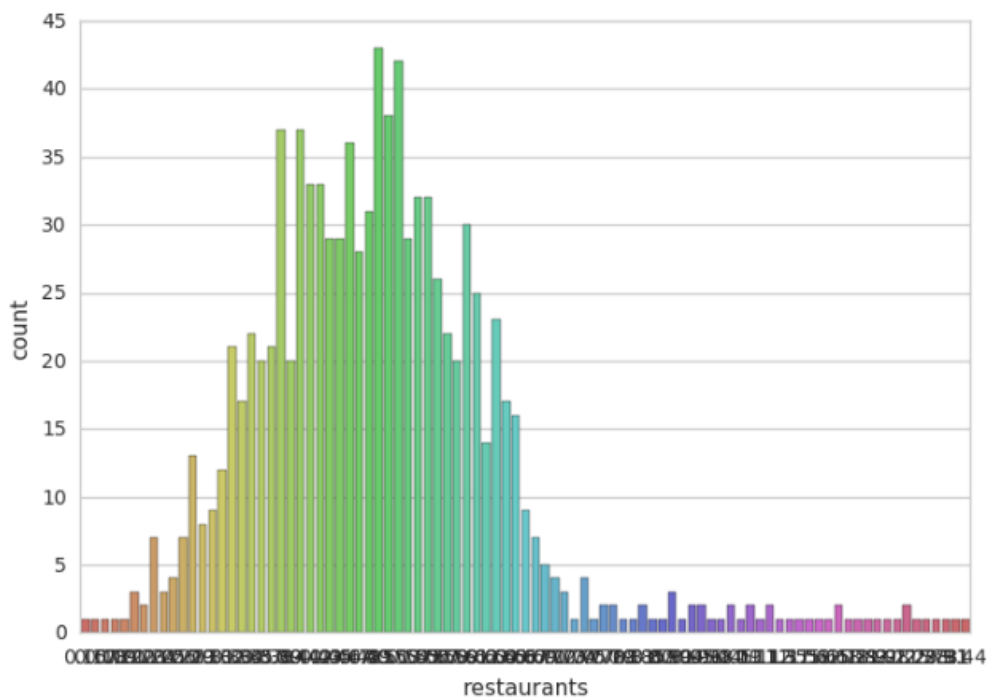
Slika 7.3. Prikaz broja plesnih klubova

```
In [18]: sns.countplot(x='juice bars', data=data, palette='hls')
plt.show()
plt.savefig('count_plotbars')
```



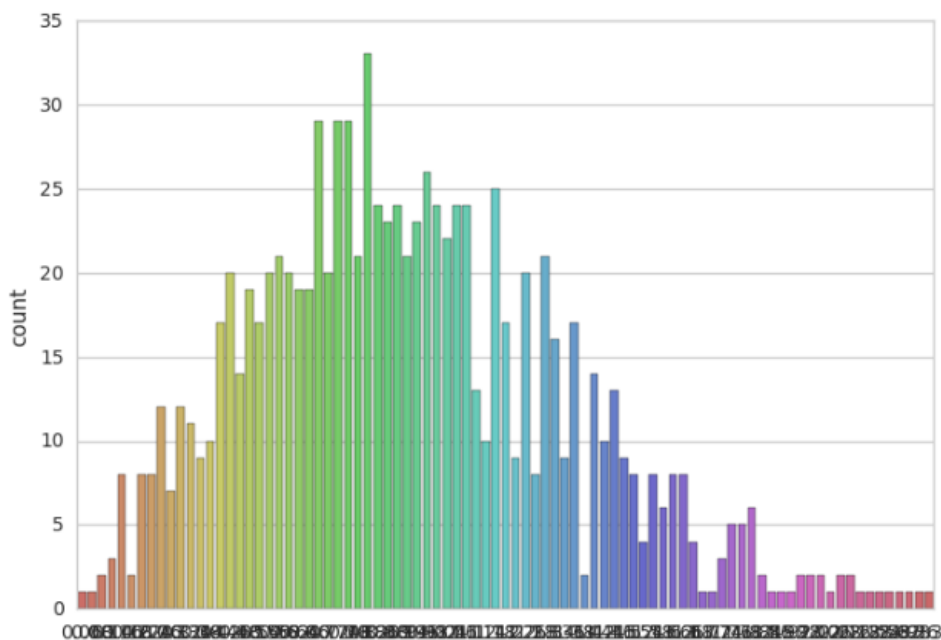
Slika 7.4. Prikaz broja barova

```
In [19]: sns.countplot(x='restaurants', data=data, palette='hls')
plt.show()
plt.savefig('count_plot_restaurants')
```



**Slika 7.5.** Prikaz broja restorana

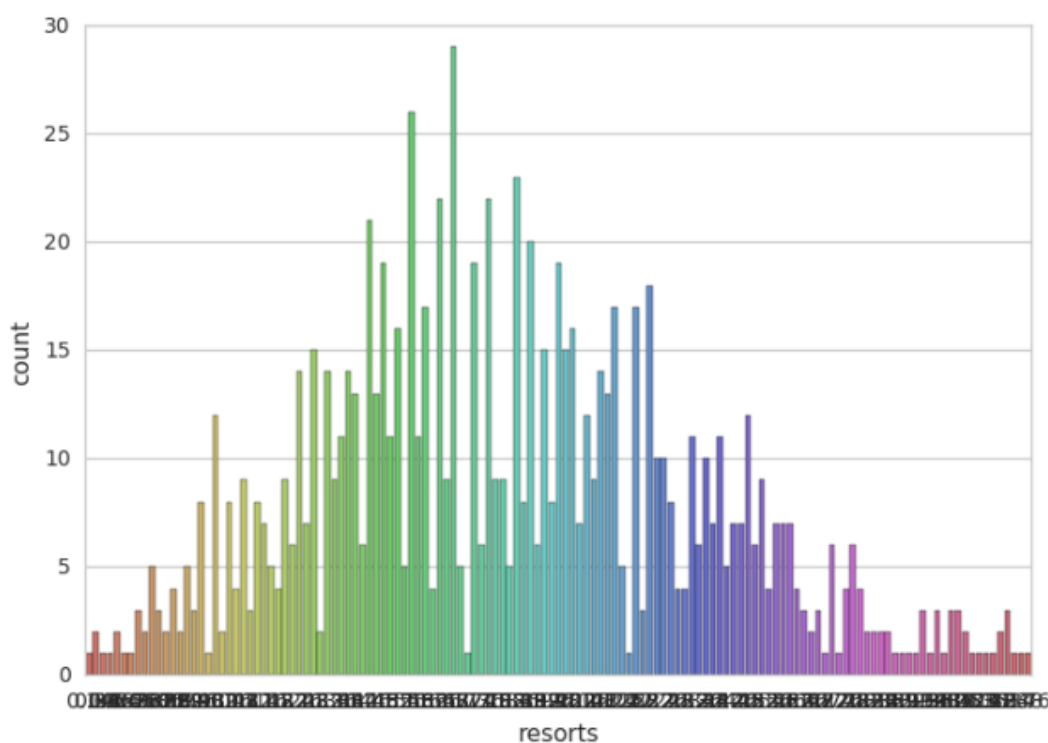
```
In [23]: sns.countplot(x='museums', data=data, palette='hls')
plt.show()
plt.savefig('count_plot_museums')
```



**Slika 7.6.** Prikaz broja muzeja



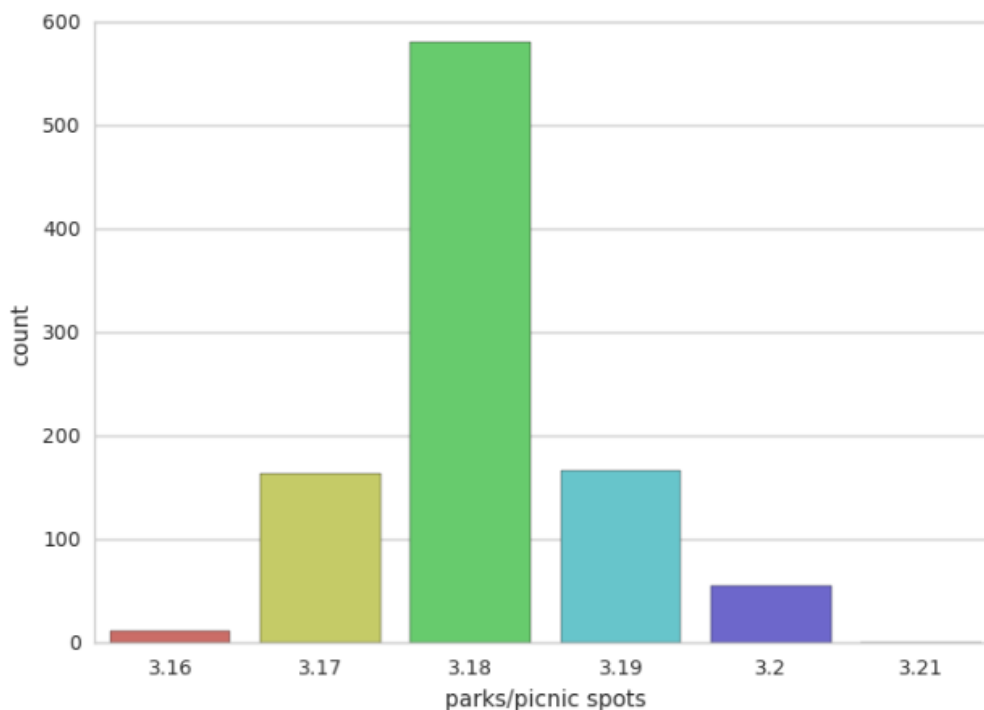
```
In [12]: sns.countplot(x='resorts', data=data, palette='hls')  
plt.show()  
plt.savefig('count_plot')
```



**Slika 7.7.** Prikaz broja odmarališta

Situacija je ista kod svih prethodnih grafika puno oscilacija i veliki broj osoba do 50 ili 70 daje ocjenu u rasponu od 0-3 ili od 2-4. Potraga nastavlja dalje za grafikom koji daje bolju sliku.

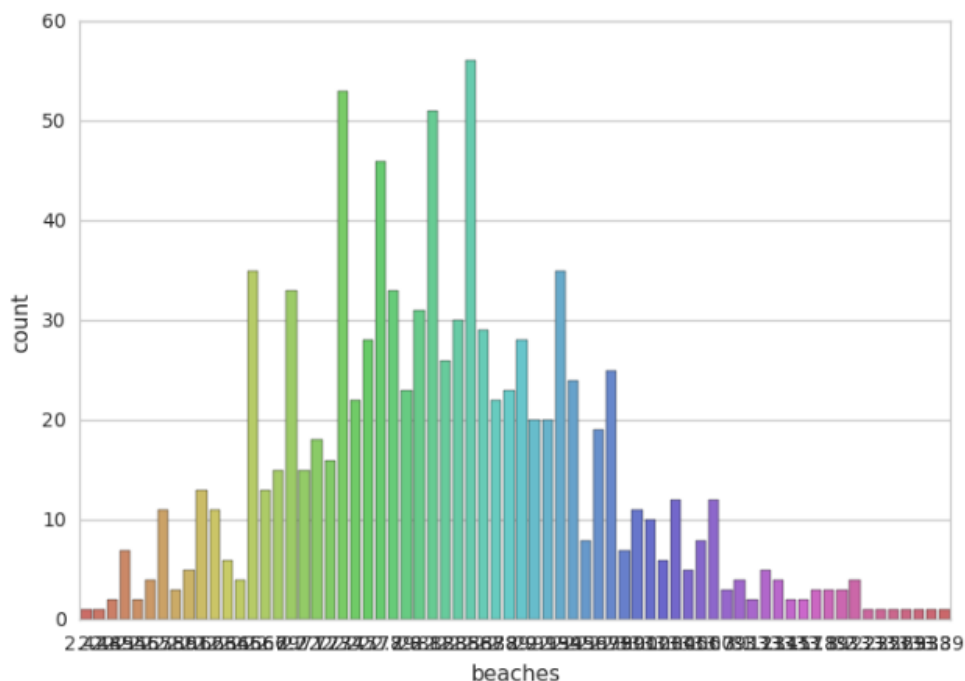
```
In [25]: sns.countplot(x='parks/picnic spots', data=data, palette='hls')  
plt.show()  
plt.savefig('count_plot_parks')
```



**Slika 7.8.** Prikaz broja parkova/piknik mjesta

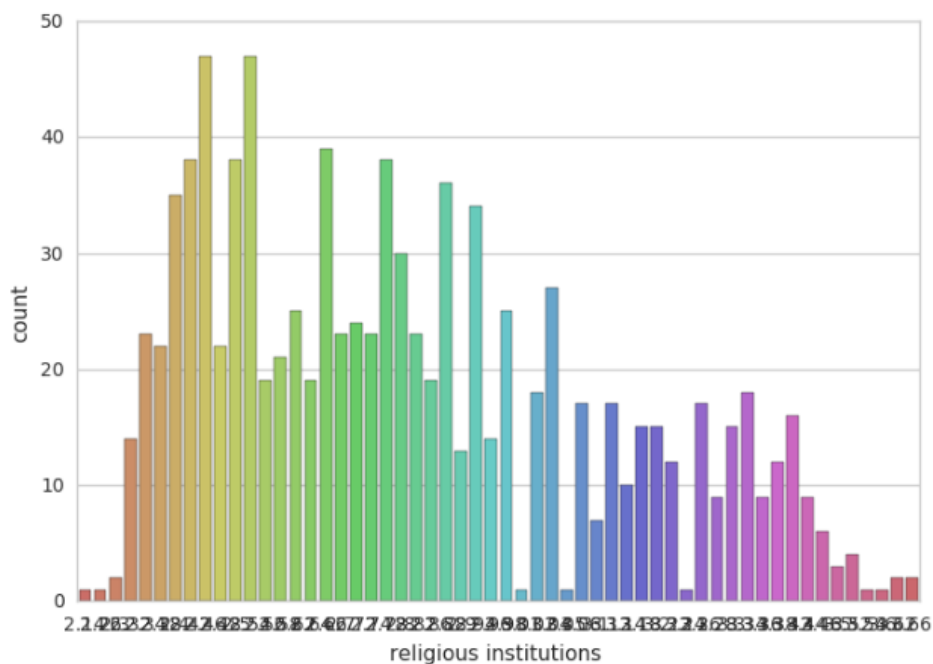
Ovaj grafik parkovi/mjesta za piknik daje najbolju sliku gdje je do 600 osoba dalo ocjenu od 3.16-3.21 što je značajna informacija u odnosu na prethodne grafike. Tako da ova kategorija postaje kategorija od značaja za daljnja istraživanja.

```
In [27]: sns.countplot(x='beaches', data=data, palette='hls')
plt.show()
plt.savefig('count_plot_beaches')
```



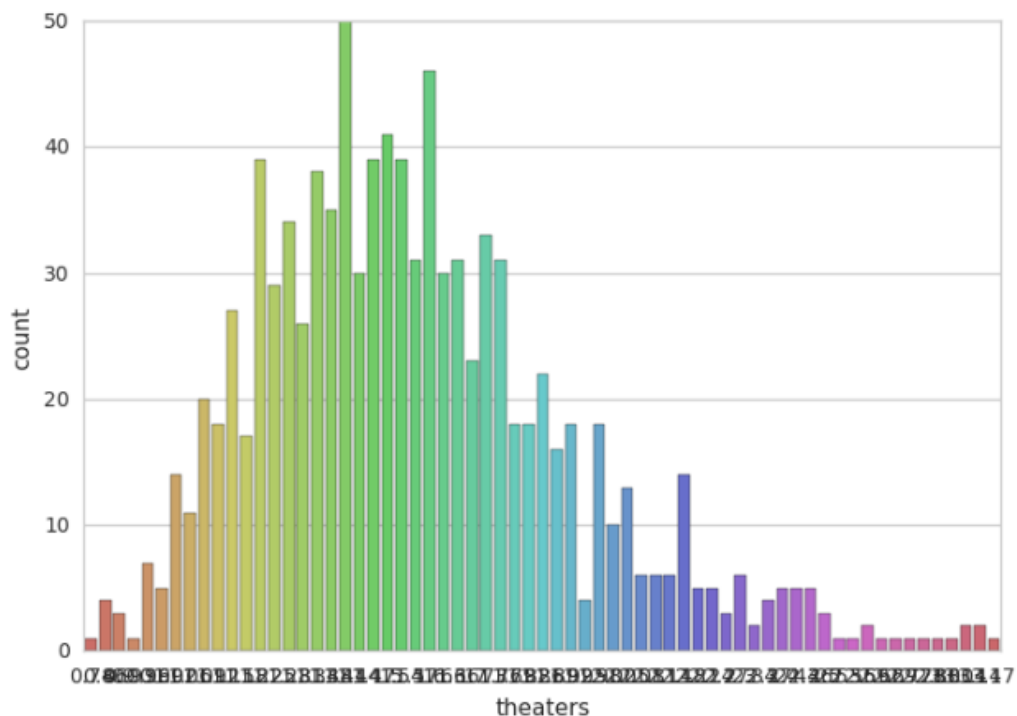
Slika 7.9. Prikaz broja plaža

```
In [16]: sns.countplot(x='religious institutions ', data=data, palette='hls')
plt.show()
plt.savefig('count_plot_religion')
```



Slika 7.10. Prikaz broja religijskih institucija

```
In [29]: sns.countplot(x='theaters', data=data, palette='hls')
plt.show()
plt.savefig('count_plot_theaters')
```



**Slika 7.11.** Prikaz broja pozorišta

Plaže i pozorišta također imaju puno oscilacija što se tiče ocjena koje su davali posjetioči. Religijske institucije imaju nešto bolju sliku, skoro pa konstantan broj ljudi je davao ocjenu od 2-3 što također prikazuje jednu kategoriju od značaja zajedno sa parkovima/piknik mjestima.

Na osnovu prethodnog može se zaključiti da mjesta koje osobe više posjećuju uglavnom predstavljaju ona mjesta koja su interesantna i drugim osobama

Nakon grafičke reprezentacije u radu je korištena k-means klastering metoda da bi se izvršilo grupisanje podataka u 10 kategorija.

Na sljedećoj slici (slika 7.12) prikazan je dio programskog koda koji se odnosi na konekciju baze podataka.

```
from pymldb import Connection
mldb = Connection("http://localhost")
```

**Slika 7.12.** Isječak koda iz MLDBa

```
In [33]: mldb.put('/v1/procedures/import_review', {
    "type": "import.text",
    "params": {
        "dataFileUrl": "https://archive.ics.uci.edu/ml/machine-learning-databases/00484/tripadvisor_review.csv",
        "outputDataset": "review",
        "runOnCreation": True
    }
})

Out[33]: PUT http://localhost/v1/procedures/import_review
201 Created
{
  "status": {
    "firstRun": {
      "runStarted": "2020-03-07T21:49:36.6691425Z",
      "status": {
        "rowCount": 980,
        "numLineErrors": 0
      },
    },
    "runFinished": "2020-03-07T21:49:39.2020142Z",
    "id": "2020-03-07T21:49:36.668315Z-463496b56263af05",
    "state": "finished"
  },
  "config": {
    "params": {
      "outputDataset": "review",
      "runOnCreation": true,
      "dataFileUrl": "https://archive.ics.uci.edu/ml/machine-learning-databases/00484/tripadvisor_review.csv"
    },
    "type": "import.text",
    "id": "import_review"
  },
  "state": "ok",
  "type": "import.text",
  "id": "import_review"
}
```

Slika 7.13. Isječak koda iz MLDBa unos skupa podataka koristeći proceduru [15]

```
In [3]: mldb.query("select * from review")

Out[3]:
```

	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6	Category 7	Category 8	Category 9	Category 10	User ID
_rowName											
2	0.93	1.80	2.29	0.62	0.80	2.42	3.19	2.79	1.82	2.42	User 1
3	1.02	2.20	2.66	0.64	1.42	3.18	3.21	2.63	1.86	2.32	User 2
4	1.22	0.80	0.54	0.53	0.24	1.54	3.18	2.80	1.31	2.50	User 3
5	0.45	1.80	0.29	0.57	0.46	1.52	3.18	2.96	1.57	2.86	User 4
6	0.51	1.20	1.18	0.57	1.54	2.02	3.18	2.78	1.18	2.54	User 5
7	0.99	1.28	0.72	0.27	0.74	1.26	3.17	2.89	1.66	3.66	User 6
8	0.90	1.36	0.26	0.32	0.86	1.58	3.17	2.66	1.22	3.22	User 7
9	0.74	1.40	0.22	0.41	0.82	1.50	3.17	2.81	1.54	2.88	User 8
10	1.12	1.76	1.04	0.64	0.82	2.14	3.18	2.79	1.41	2.54	User 9

Tabela 7.14. Tabela prikaza skupa podataka koristeći proceduru

Šejla Pljakić: „Analiza recenzija putovanja koristeći ML bazu podataka za mašinsko učenje“

Nakon učitavanja skupa podataka potrebno je napraviti proceduru koristeći K-means metodu klasteringa. Na slici 7.14. prikazana je procedura koja razmješta podatke u 10 klastera, a zatim je prikazan grafik pojedinih rezultata.

```
In [16]: mldb.put('/v1/procedures/review_train_kmeans', {
    'type' : 'kmeans.train',
    'params' : {
        'trainingData' : 'select * EXCLUDING("User ID") from review',
        'outputDataset' : 'review_clusters',
        'numClusters' : 10,
        'metric' : 'euclidean',
        'runOnCreation' : True
    }
})

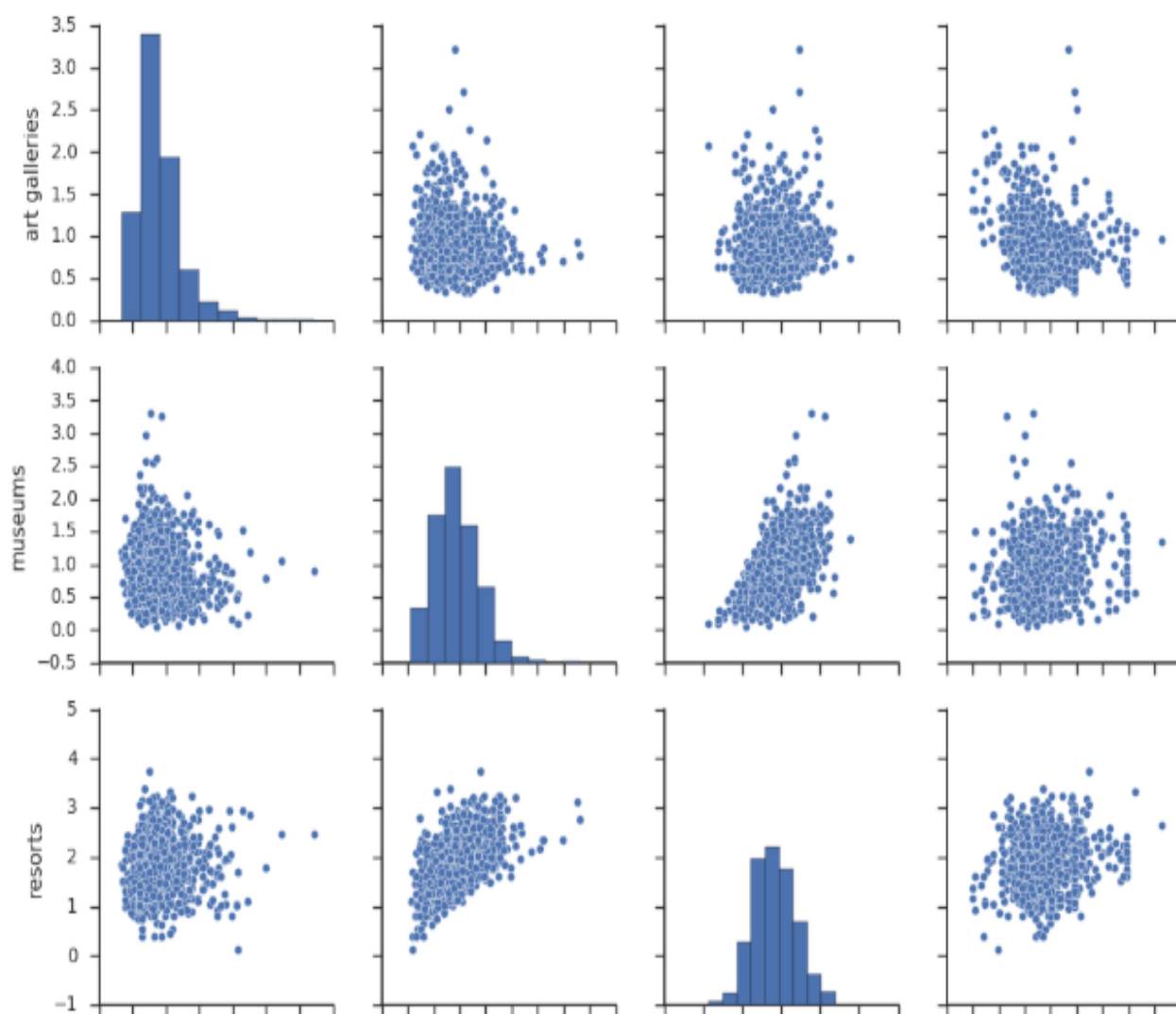
Out[16]: PUT http://localhost/v1/procedures/review_train_kmeans
201 Created
{
  "status": {
    "firstRun": {
      "runStarted": "2020-03-08T19:37:39.745277Z",
      "runFinished": "2020-03-08T19:37:43.3369915Z",
      "id": "2020-03-08T19:37:39.741377Z-463496b56263af05",
      "state": "finished"
    }
  },
  "config": {
    "params": {
      "trainingData": "select * EXCLUDING(\"User ID\") from review",
      "metric": "euclidean",
      "outputDataset": "review_clusters",
      "numClusters": 10,
      "runOnCreation": true
    },
    "type": "kmeans.train",
    "id": "review_train_kmeans"
  },
  "state": "ok",
  "type": "kmeans.train",
  "id": "review_train_kmeans"
}
```

**Slika 7.14.** Procedura za treniranje skupa podataka uz pomoć K-means tehnike

Rezultati se svakako poklapaju sa procjenom pa su glavne 2 kategorije parkovi/piknik mjesta i religijske institucije.

```
15]: g = sns.pairplot(data, vars=["art galleries", "museums", "resorts", "dance clubs"])
plt.show()
plt.savefig('plot_bars_galleries_museums')

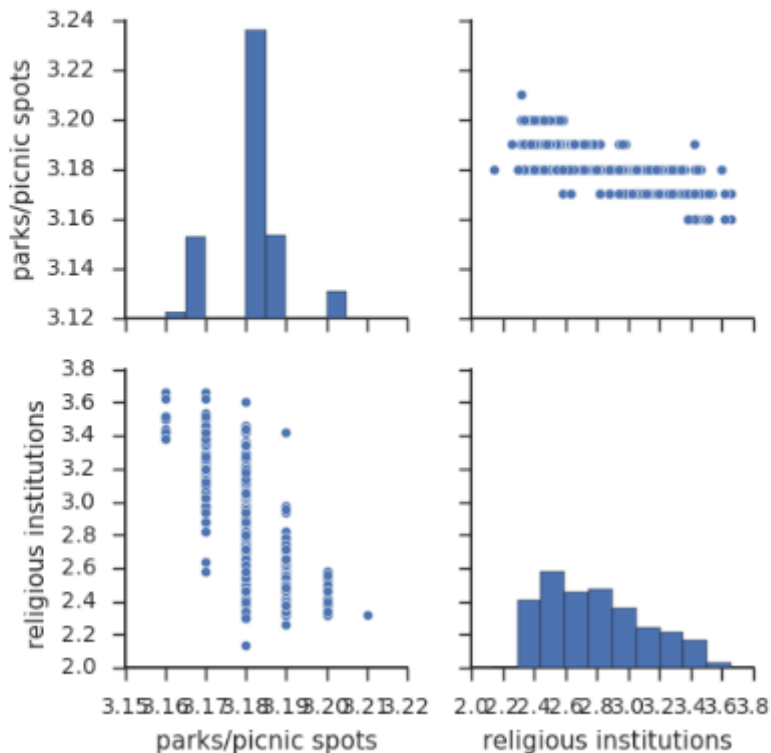
<matplotlib.figure.Figure at 0x7f7ffd3b9710>
```



**Slika 7.15.** Grafici pojedinih kategorija

Za donošenje finalne odluke koja je kategorija najpoželjnija potrebno je upariti na grafiku dvije kategorije od značaja parkove/piknik mjesta i religijske institucije, a nakon toga koristiti i logističku regresiju za procjenu. To je prikazano na slici 7.16. i 7.17.

```
In [21]: g = sns.pairplot(data, vars=["parks/picnic spots", "religious institutions "])
plt.show()
plt.savefig('plot_parks_religion')
<matplotlib.figure.Figure at 0x7f7ffe917090>
```



**Slika 7.16.** Grafici parkova/piknik mjesta i religijskih institucija

```
In [18]: %%time
logit = LogisticRegression(solver='lbfgs', n_jobs=-1, random_state=7)
logit.fit(X_train, y_train)
```

CPU times: user 29.7 ms, sys: 69.7 ms, total: 99.4 ms  
Wall time: 2.82 s

```
In [19]: round(logit.score(X_train, y_train), 3), round(logit.score(X_test, y_test), 3),
Out[19]: (0.981, 0.864)
```

**Slika 7.17** Isječak koda logističke regresije iz MLDB baze podataka

Na osnovu grafika i procjene logističke regresije može se zaključiti da su parkovi ili piknik za oko 98% napoželjnija mjesta za posjetu.



## 8 Zaključak

Mašinsko učenje izgrađeno je na statističkom okviru. Pored toga mašinsko učenje obuhvata i veliki broj drugih područja matematike i računarskih nauka. Statističko modeliranje se bavi pronalaženjem odnosa između varijabli, dok mašinsko učenje u prvi plan stavlja i predviđanje (predikciju).

U radu su kratko pisane mnoge tehnike mašinskog učenja, dok je na konkretnom primjeru prikazana i primjena tih tehnika.

Izvršena analiza na konkretnom primjeru pokazuje da su parkovi/piknik mjesta najbolja za posjetiti u istočnoj Aziji. Pokazuje se da su parkovi posebno interesantna područja za razonodu, odmor i zabavu. Tome ide u prilog što većina ovih parkova ima besplatan ulaz, tj. veoma su pristupačni starijim i mlađim osobama, odnosno generacijama svih starosnih dobi. Stoga su vlasti i organizacije iskoristile svoje resurse da ta mjesta budu i najuređenija samim tim i najpoželjnija.

Ukoliko se ikada nađete u zemljama istočne Azije, ne trošite mnogo svoje resurse poput vremena, najvrjednijeg resursa, na mjesta koja nisu toliko dobra kao što su parkovi/piknik mjesta gdje ćete najviše uživati.

Na kraju se može zaključiti da upotreba metoda mašinskog učenja sve više dobija na značaju. Zapravo, danas ne postoji niti jedna oblast u kojem se tehnike i metode mašinskog učenja ne koriste za rješavanje problema. S obzirom na to da postoji preveliki broj metoda mašinskog učenje, koje se mogu upotrijebiti za bilo koji problem gdje je neophodno izvršiti neki vid klasifikacije ili klasterizacije podataka, sada se postavlja novo pitanje: Kako odabrati odgovarajući metod mašinskog učenja za konkretan problem koji se rješava?

## Literatura

[1]	Šta je mašinsko učenje- Objašnjenje pojma Dostupno na: <a href="https://www.mathworks.com/discovery/machine-learning.html">https://www.mathworks.com/discovery/machine-learning.html</a>
[2]	Metode mašinskog učenja Dostupno na: <a href="https://towardsdatascience.com/ai-machine-learning-deep-learning-explained-simply-7b553da5b960">https://towardsdatascience.com/ai-machine-learning-deep-learning-explained-simply-7b553da5b960</a>
[3]	Metode mašinskog učenja Dostupno na: <a href="https://www.sas.com/en_us/insights/analytics/machine-learning.html">https://www.sas.com/en_us/insights/analytics/machine-learning.html</a>
[4]	Tehnike mašinskog učenja Dostupno na: <a href="https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9">https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9</a>
[5]	Tehnike mašinskog učenja Dostupno na: <a href="https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/">https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/</a>
[6]	Metode i tehnike mašinskog učenja Dostupno na: <a href="https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/">https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/</a>
[7]	Mašinsko učenje Dostupno na: <a href="https://c2.etf.unsa.ba/course/view.php?id=332">https://c2.etf.unsa.ba/course/view.php?id=332</a>
[8]	Metode mašinskog učenja Dostupno na: <a href="https://medium.com/@yannmjl/what-is-machine-learning-in-simple-english-b0aaa251cb60">https://medium.com/@yannmjl/what-is-machine-learning-in-simple-english-b0aaa251cb60</a>
[9]	Tehnike mašinskog učenja Dostupno na: <a href="https://blogs.oracle.com/bigdata/machine-learning-techniques">https://blogs.oracle.com/bigdata/machine-learning-techniques</a>
[10]	MLDB Dostupno na: <a href="https://hackernoon.com/technology-fridays-mldb-is-the-database-every-data-scientist-dreams-of-368b50b5a434">https://hackernoon.com/technology-fridays-mldb-is-the-database-every-data-scientist-dreams-of-368b50b5a434</a>
[11]	MLDB Dostupno na: <a href="https://www.kdnuggets.com/2016/10/mldb-machine-learning-database.html">https://www.kdnuggets.com/2016/10/mldb-machine-learning-database.html</a>
[12]	Praktični rad, MLDB dokumentacija Dostupno na: <a href="https://docs.mldb.ai/">https://docs.mldb.ai/</a>
[13]	Praktično istraživanje Dostupno na: <a href="https://cloud.ibm.com/apidocs/natural-language-classifier">https://cloud.ibm.com/apidocs/natural-language-classifier</a>
[14]	Praktično istraživanje Dostupno na: <a href="https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8">https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8</a>

[15]	<p>Procedure</p> <p>Dostupno na: <a href="http://localhost:8080/ipy/notebooks/_tutorials/_latest/Procedures%20and%20Functions%20Tutorial.ipynb">http://localhost:8080/ipy/notebooks/_tutorials/_latest/Procedures%20and%20Functions%20Tutorial.ipynb</a></p>
[16]	<p>Skup podataka</p> <p>Dostupno na: <a href="https://archive.ics.uci.edu/ml/machine-learning-databases/00484/tripadvisor_review.csv">https://archive.ics.uci.edu/ml/machine-learning-databases/00484/tripadvisor_review.csv</a></p>
[17]	<p>Jeff Dean intervju- mašinsko učenje, NLP i BERT</p> <p>Dostupno na: <a href="https://venturebeat.com/2019/12/13/google-ai-chief-jeff-dean-interview-machine-learning-trends-in-2020/">https://venturebeat.com/2019/12/13/google-ai-chief-jeff-dean-interview-machine-learning-trends-in-2020/</a></p>
[18]	<p>Deep learning</p> <p>Dostupno na: <a href="https://www.investopedia.com/terms/d/deep-learning.asp">https://www.investopedia.com/terms/d/deep-learning.asp</a></p>
[19]	<p>Pseudokod K-means algoritma</p> <p>Dostupno na: <a href="https://rpubs.com/chzelada/285083">https://rpubs.com/chzelada/285083</a></p>
[20]	<p>Potrebni fajl za instalaciju mldb baze podataka koristeći virtuelnu mašinu</p> <p>Dostupno na: <a href="http://public.mldb.ai/mldb.ova">http://public.mldb.ai/mldb.ova</a></p>