



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, mart, 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 i prezime**

**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:** 52

### **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 češće koristi u svakodnevnom životu iako toga nismo 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 želimo 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, a 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.....	18
4.3 Klasterizacija(grupisanje) .....	20
4.4. Smanjenje dimenzija.....	22
4.5 Metode ansambla.....	23
4.6 Neuronske mreže i duboko učenje.....	24
4.7 Prijenosno učenje.....	26
4.8 Obrada prirodnog jezika.....	27
4.9 Umetanja riječi.....	28
5 Machine learning database - MLDB.....	29
5.1 Arhitektura MLDB-a.....	30
5.2 Podrška za algoritme mašinskog učenja.....	32
5.3 Instaliranje MLDB-a na različitim platformama.....	34
6 Opis skupa podataka recenzije putovanja.....	36
7 Praktični rad koristeći bazu MLDB i programski jezik Python.....	37
Zaključak .....	50
Literatura.....	51

## Popis slika

Slika 2.1 Prikaz rada algoritama mašinskog učenja

Slika 4.1.1. Prikaz odnosa predviđene i posmatrane energije

Slika 4.2.1 Grafik logističke regresije

Slika 4.3.1. Grupiranje zgrada u efikasne (zelene) i neučinkovite (crvene) skupine

Slika 4.3.2. Prikaz grupisanja elemenata

Slika 4.4.1. Analiza baze podataka MNIST rukom pisanih cifara

Slika 4.6.1. Neuronske mreže sa skrivenim slojem

Slika 4.6.2. Neuronske mreže sa više skrivenih slojeva

Slika 4.9.1. Umetanje riječi

Slika 5.1. Životni ciklus MLDBa

Slika 5.1.1. Prikaz arhitekture MLDBa

Slika 5.2.2. Tok rada mašinskog učenja

Slika 5.2.3 Učinkovitost MLDBa

Slika 5.3.1. Oficijalna stranica MLDB baze podataka

Slika 7.1. Kod učitavanja skupa podataka u bazu

Slika 7.2. Prikaz broja umjetničkih galerija

Slika 7.3. Prikaz broja plesnih klubova

Slika 7.4. Prikaz broja barova

Slika 7.5. Prikaz broja restorana

Slika 7.6. Prikaz broja muzeja

Slika 7.7. Prikaz broja odmarališta

Slika 7.8. Prikaz broja parkova/piknik mjesta

Slika 7.9. Prikaz broja plaža

Slika 7.10. Prikaz broja religijskih institucija

Slika 7.11. Prikaz broja pozorišta

Slika 7.12. Isječak koda iz MLDBa

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

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

Slika 7.15. Grafici pojedinih kategorija

Slika 7.16. Grafici parkova/piknik mjesta i religijskih institucija

Slika 7.17. Isječak koda logističke regresije iz MLDB baze podataka



## **Popis tabela**

Tabela 5.2.1. Prikaz tehnika MLDBa

Tabela 5.3.2. Prikaz verzija instalacije MLDBa

Tabela 6.1. Karakteristike skupa podataka

Tabela 7.1. Prikaz tabele skupa podataka

Tabela 7.2. Tabela prikaza skupa podataka koristeći proceduru

## 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 supervizijsko (nadgledano) mašinsko učenje, nesupervizijsko(nenadgledano) učenje i učenje ojačanja.

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 tumora, pronalazak novih lijekova i sekvenciranje DNK
- Automobilska, vazdušna 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.

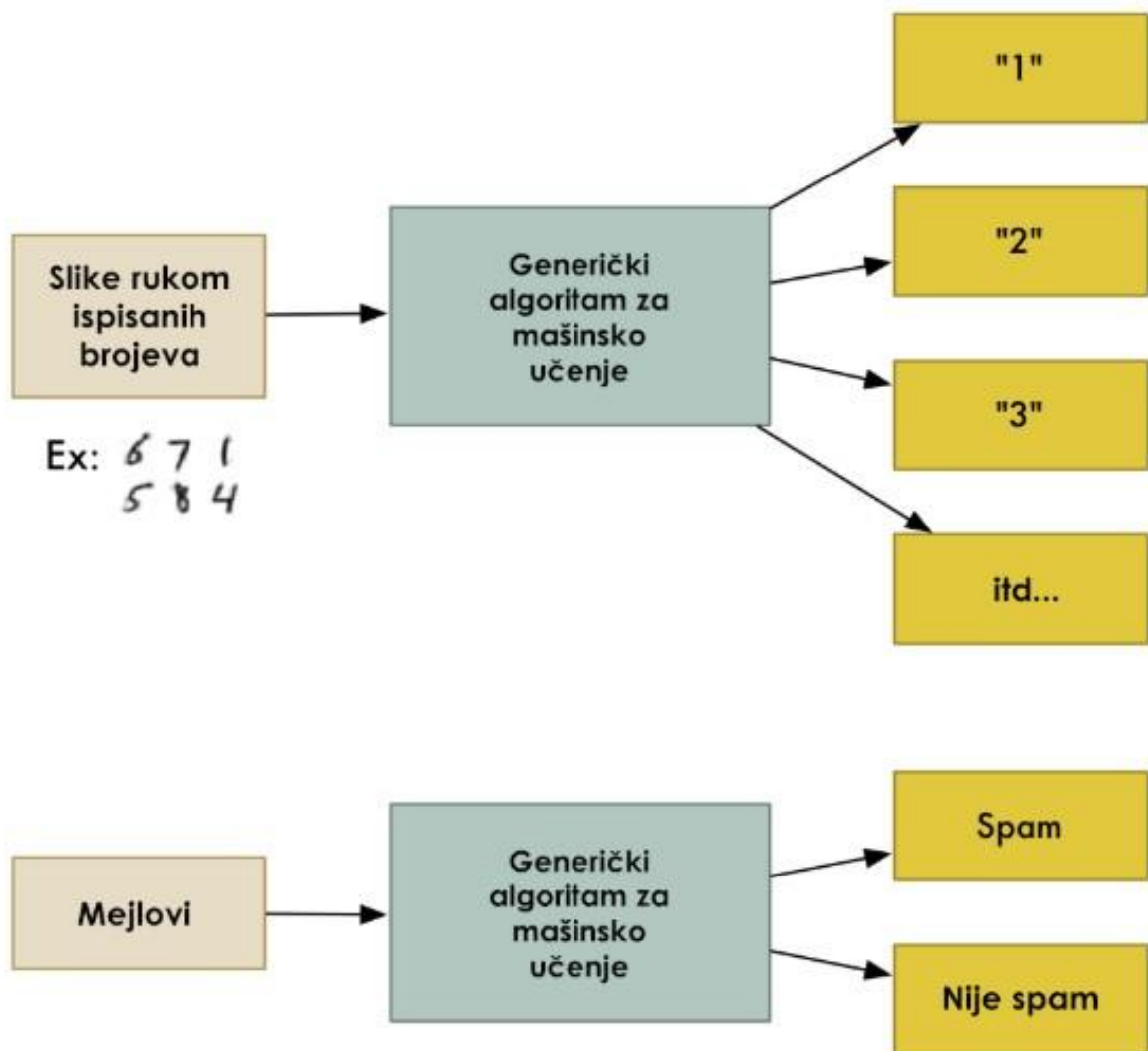
Mašinsko učenje treba koristiti kada postoji složen zadatak ili problem koji uključuje veliku količinu podataka i varijabli, ali nema postojeće formule ili jednačine rješavanja.

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 se zasniva 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. On 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

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 moguć. 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 svojim 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 kao i u matematici. 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 koji 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 nadziranog 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 uključuje:

- Hijerarhijsko klasteriranje
- Samoorganiziranje karata
- Preslikavanje najbližeg susjeda
- Razlaganje pojedinačne vrijednosti.
- K-klasteriranje
- Modeli smjese DBSCAN
- Lokalni vanjski faktor
- Neuralne mreže
- Algoritam očekivanja-maksimizacija
- Analiza glavnih komponenti
- Negativna matična faktorizacija

### 3.3 Učenje ojačanja(Reinforcement learning)

Ukoliko se posmatra miš u labirintu koji pokušava pronaći skrivene komade sira, što je više puta miš izložen labirintu, to će mu biti bolje 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 miša odražava ono što radimo s učenjem ojačanja radi obučavanja sistema ili igre. Općenito govoreći, ovo je metoda mašinskog učenja koja agentu pomaže da nauči iz iskustva. Snimanjem 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, nagrada je sir.

Prateći korake algoritma potrebno je doći do ukupne nagrade do koje ćemo imati pozitivnu ili negativnu nagradu. 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, učenje ojačanja bazirano je na pronalaženju najboljeg puta koji treba proći u nekoj situaciji kako bi se maksimizirala nagrada u situaciji. Odluka se donosi uzastopno.

RL se može koristiti ako imamo malo historijskih podataka o nekom problemu, jer one ne trebaju informacije unaprijed (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 temeljen 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 prva AI koja je pobijedila Dota 2 tim svjetskog prvaka u e-sportu, vrlo složenu video igru koju je odabrala ekipa OpenAI Five jer nije bilo RL algoritama koji bi ih mogli osvojiti na vrijeme. Isti tim AI koji je pobijedio Dota 2, prvak je razvio i robotsku ruku koja se može preusmjeriti u blok. [2]



## 4 Tehnike mašinskog učenja

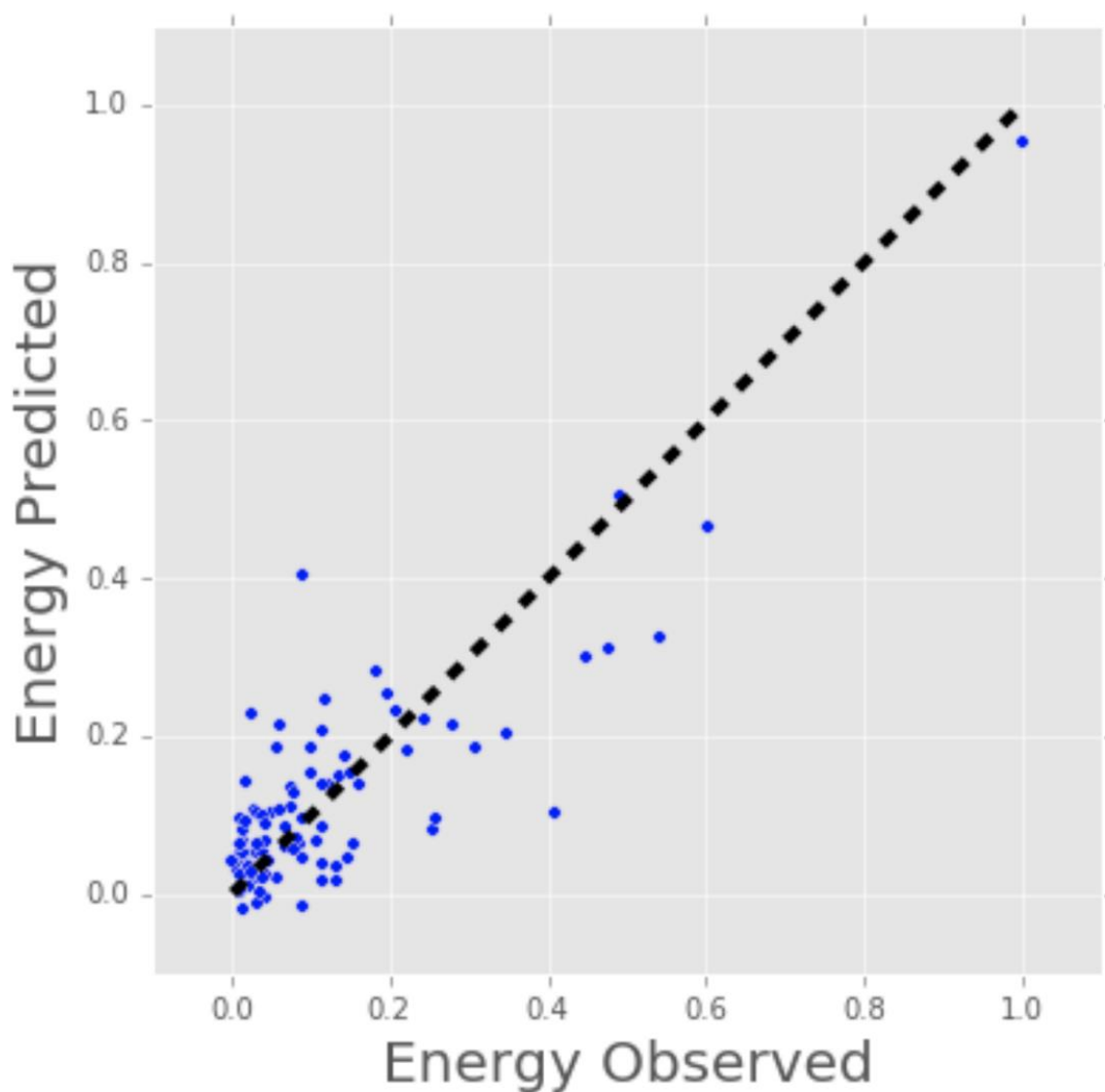
### 4.1 Regresija

Regresijske tehnike spadaju u kategoriju superviziranih metoda mašinskog učenja. Pomoći će predvidjeti ili objasniti određenu brojčanu vrijednost na temelju niza prethodnih podataka kao npr. predviđanje cijene nekretnine na temelju prethodnih podataka o cijenama za slične nekretnine. Najjednostavnija metoda je linearna regresija gdje za modeliranje skupa podataka koristimo matematičku jednačinu pravca ( $y = m * x + b$ ). Trenira se linearni regresijski model s mnogim parovima podataka ( $x, y$ ) tako što se izračunava položaj i nagib linije koja umanjuje ukupnu udaljenost između svih podataka i linija. Tako da za liniju koja najbolje aproksimira opažanja u podacima izračunava se nagib ( $m$ ) i  $y$ -presretanje ( $b$ ). [4]

Regresija se bavi modeliranjem 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 klasu algoritma.

Ukoliko se linearna regresija koristi za predviđanje potrošnje energije (u kWh) određenih zgrada tada se sakupe podaci kao što su starost zgrade, kvadratno postolje i broj priključne zidne opreme. Budući da postoji više ulaza koristi se višestruka varijabilna linearna regresija. Princip je isti kao kod jednostruke linearne regresije, ali u ovom slučaju „linija“ koja se pojavila dogodila se u višedimenzionalnom prostoru na osnovu broja varijabli. Slika 4.1.1 u nastavku pokazuje koliko se linearni regresijski model uklapa u stvarnu potrošnju energije zgrade.

U ovom slučaju može se pomoću ugrađene linije približiti potrošnji energije određene 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 predviđene i posmatrane energije

Regresijske tehnike imaju opseg 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. Može pomoći u predviđanju hoće li internetski kupac kupiti ili ne proizvod. Izlaz može biti da ili ne: da kupac kupi proizvod ili da ne kupi. Metode klasifikacije ipak nisu ograničene na dvije klase. One mogu pomoći u procjeni sadrži li određena slika automobil ili kamion.

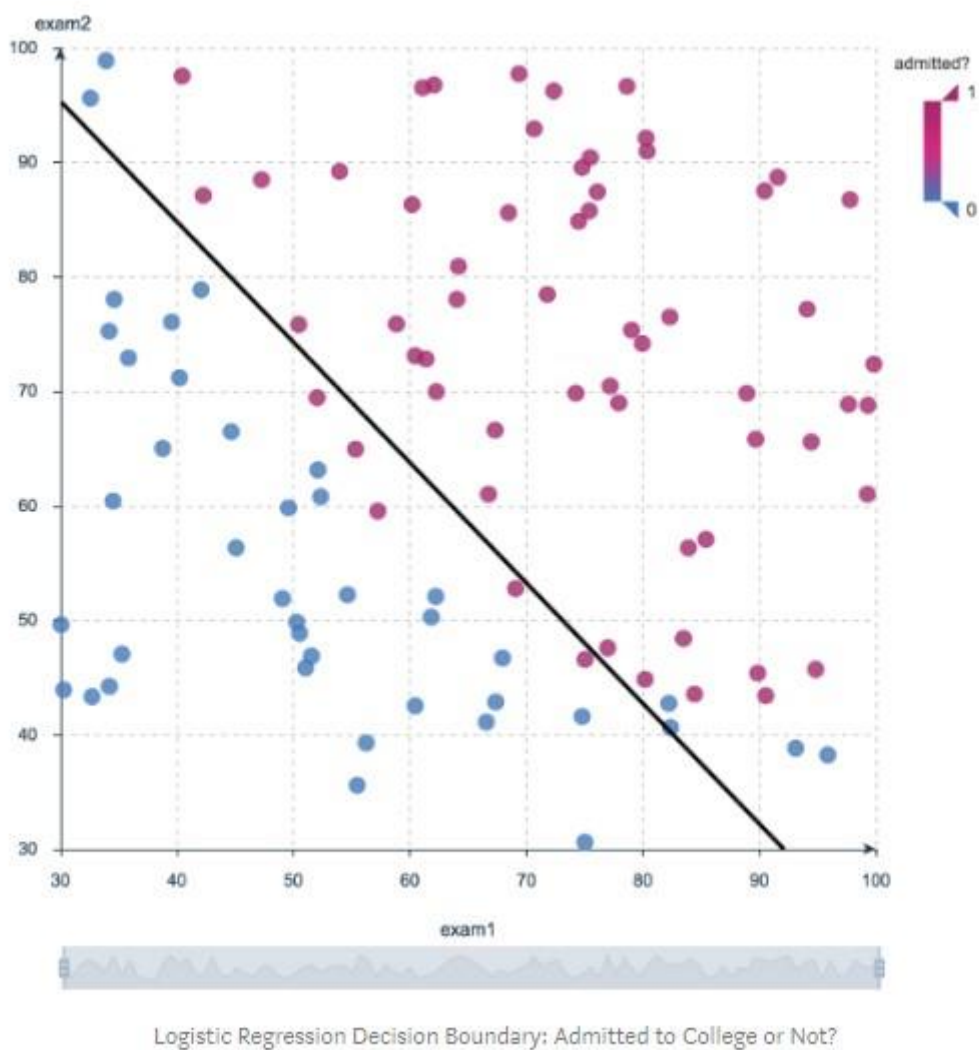
U ovom slučaju, izlaz će biti 3 različite vrijednosti:

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

Kako je logistička regresija najjednostavniji model klasifikacije, tako predstavlja dobar početak klasifikacije. Kako se dalje napreduje, može se otići i u nelinearne klasifikatore poput stabala odlučivanja, slučajnih šuma, vektorskih mašina za podršku i neuralnih mreža.

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

Budući da je procjena vjerojatnoća, izlaz je broj između 0 i 1, gdje 1 predstavlja potpunu sigurnost. Ako je procijenjena vjerovatnoća za studenta veća od 0.5, predviđamo da će on ili ona biti primljen/a. Ako je procijenjena vjerovatnoća manja od 0.5, predviđamo da će on ili ona biti odbijeni. Slika 4.2.1. prikazuje rezultate prethodnih učenika, zajedno s tim da li su primljeni. Logistička regresija omogućava crtanje linije koja predstavlja granicu odluke.



**Slika 4.2.1.** Grafik logističke regresije

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

Klasifikacija se koristi ako se podaci mogu označiti, kategorizirati ili razdvojiti u određene grupe ili klase. Aplikacije za prepoznavanje ručnog pisanja koriste 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)

Metodama klasterizacije prelazi se u kategoriju nesuperviziranih metoda mašinskog učenja jer im je cilj grupisati ili klasterisati opažanja koja imaju slične karakteristike. Metode klasterisanja ne koriste izlazne informacije za obuku, već umjesto toga algoritam definira izlaz. Kod metoda klasterisanja može se koristiti vizualizacija samo za uvid u kvalitetu rješenja.[9]

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

Metode klasterisanja obično se organizuje pristupima modeliranja kao što su centroidni i hijerarhijski. Sve su metode uključene u korištenje inherentnih struktura u podacima kako bi se podaci najbolje organizovali u skupine maksimuma.

Koristi se za istraživačke analize podataka za pronalaženje skrivenih obrazaca ili grupisanja u podacima. Aplikacije za klaster analizu uključuju analizu genske sekvence, istraživanje tržišta i prepoznavanje objekata.

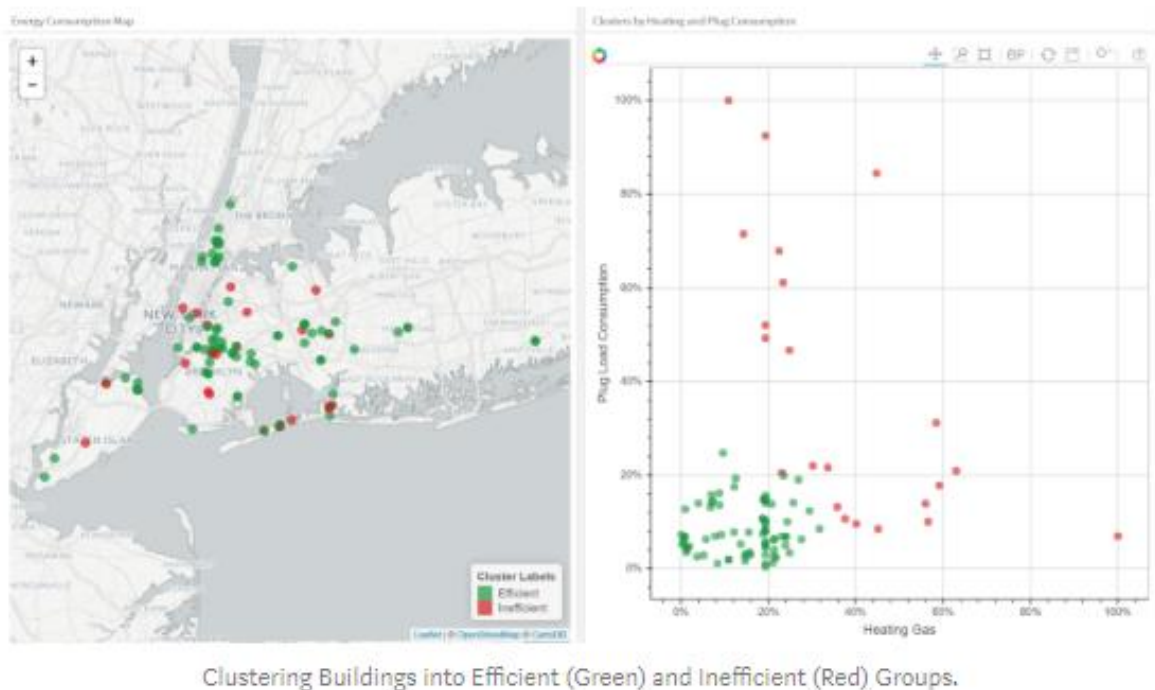
Najpopularnija metoda klasterisanja je K-Means, gdje "K" predstavlja broj klastera koje korisnik želi stvoriti. Postoje različite tehnike za odabir vrijednosti K, poput metode lakta.

Otpribliže, K-means se izvršava na sljedeći način:

1. Nasumično odabere K centre unutar podataka.
2. Svako podatkovnoj tački dodjeljuje najbliže nasumično stvorene centre.
3. Ponovno izračunava središte svakog klastera.
4. Ako se centri ne mijenjaju (ili se vrlo malo mijenjaju), postupak je završen. U protivnom se vraća algoritam na korak 2. (Da se spriječi završavanje u beskonačnoj petlji ako se centri nastave mijenjati, unaprijed se postavi maksimalan broj iteracija.)

Sljedeći plan primjenjuje K-Means na skup podataka o zgradama. Svaki red u parceli ukazuje na učinkovitost svake zgrade. Četiri mjerenja odnose se na klimatizaciju, priključnu opremu (mikrovalne pećnice, hladnjake itd.), kućni plin i plin za grijanje. Za klasterisanje se izabere  $K = 2$ , što olakšava interpretaciju jednog od klastera kao grupe učinkovitih zgrada, a drugog kao grupe neučinkovitih zgrada.[4][5]

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



**Slika 4.3.1.** Grupisanje zgrada u efikasne (zelene) i neučinkovite (crvene) skupine.

Klasterisanje ima jako korisne algoritme, poput prostornog klasterisanja aplikacija zasnovanih na gustoći (DBSCAN), srednjeg klastera pomaka, aglomerativnog hijerarhijskog klasterisanja i td.



Figure 2. Clustering finds hidden patterns in your data.

**Slika 4.3.2.** Prikaz grupisanja elemenata

Ako kompanija koja se bavi mobitelima želi optimizirati lokacije na kojima grade kule za mobitele, mogu koristiti mašinsko učenje za procjenu broja skupina ljudi koji se oslanjaju na svoje kule. Mobitelom se može razgovarati samo s jednim tornjem, tako da tim koristi algoritme grupisanja da bi osmislio najbolji položaj kula sa ćelijama kako bi optimizirao prijem signala za grupe svojih kupaca.

Uobičajeni algoritmi za izvođenje klastera uključuju k-sredstva i k-medoide, hijerarhijsko grupisanje, Gaussove modele smjesa, skrivene Markov modele, samoorganizirajuće karte, nerazumljivo c-značenje grupisanja i subtraktivno grupisanje.[4] [5]

## 4.4. Smanjenje dimenzija

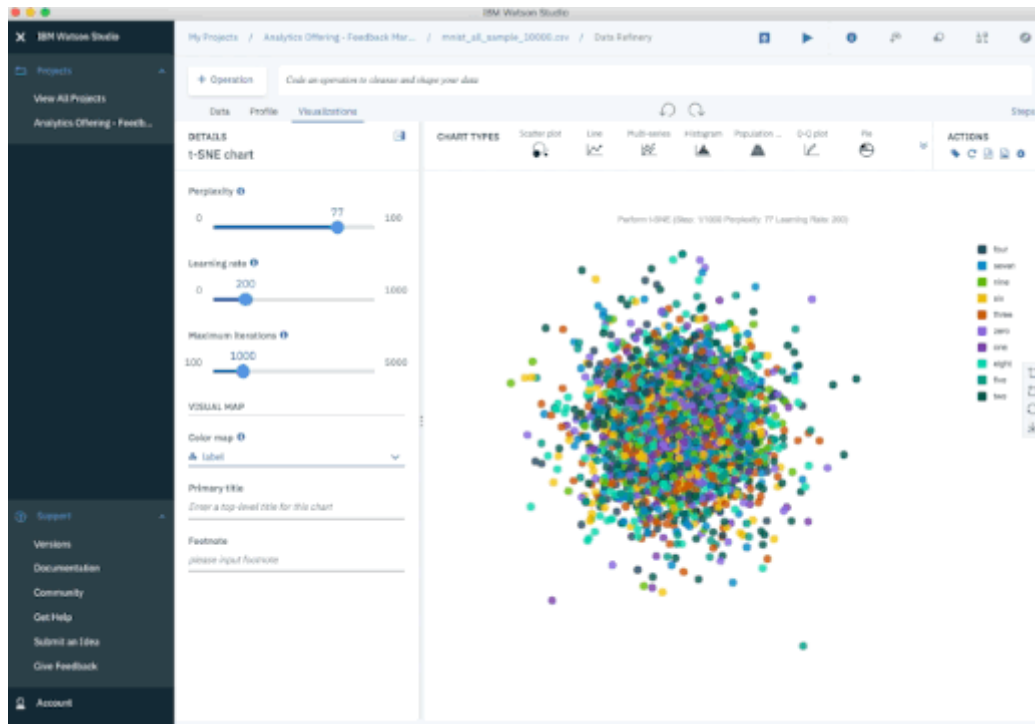
Kao što ime sugerise, koristi se smanjenje dimenzionalnosti kako bi se uklonili najmanje važni podaci (nekada suvišne kolone) iz skupa podataka. U praksi često se nalaze skupovi podataka sa stotinama ili čak hiljadama kolona tako da je smanjenje ukupnog broja od vitalnog značaja. Slike mogu sadržavati hiljade piksela, a svi ti pikseli nisu važni za analizu. Također, prilikom testiranja mikročipova u procesu proizvodnje, možda će biti puno mjerenja i testova primijenjenih na svaki čip, od kojih mnogi daju suvišne informacije. U tim slučajevima potrebni su algoritmi za smanjenje dimenzija kako bi skup podataka bio upravljiv.

Najpopularnija metoda smanjenja dimenzija je analiza glavnih komponenti (PCA),[4] koja smanjuje dimenziju obilježja prostora pronalaženjem novih vektora koji maksimiziraju linearnu varijaciju podataka. PCA može drastično smanjiti dimenziju podataka i bez gubitka previše informacija kada su linearne korelacije podataka jake. Također može se izmjeriti stvarni opseg gubitka podataka i u skladu s tim prilagoditi.

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

Na slici 4.4.1 se prikazuje analiza baze podataka MNIST[5] rukom pisanih cifara. MNIST sadrži hiljade slika cifara od 0 do 9, koje istraživači koriste za testiranje svojih algoritama za grupisanje i razvrstavanje. Svaki red skupa podataka vektorizirana je verzija izvorne slike (veličina  $28 \times 28 = 784$ ) i oznaka za svaku sliku (nula, jedan, dva, tri, ..., devet), zato smanjujemo dimenziju sa 784 (piksela) na 2 (dimenzije u vizuelizaciji).

Projektiranje u dvije dimenzije omogućava nam vizuelizaciju izvornog skupa podataka s velikim dimenzijama.[4] [5]



**Slika 4.4.1.** Analiza baze podataka MNIST rukom pisanih cifara.

## 4.5 Metode ansambla

Ukoliko se odluči napraviti bicikl, koji je drugačiji od onih koji su dostupni u trgovinama i na internetu, može se početi s pronalaženjem najboljih dijelova koji su potrebni. Jednom kada se sastave svi dijelovi, rezultirajući bicikl zasjenit će sve ostale opcije.

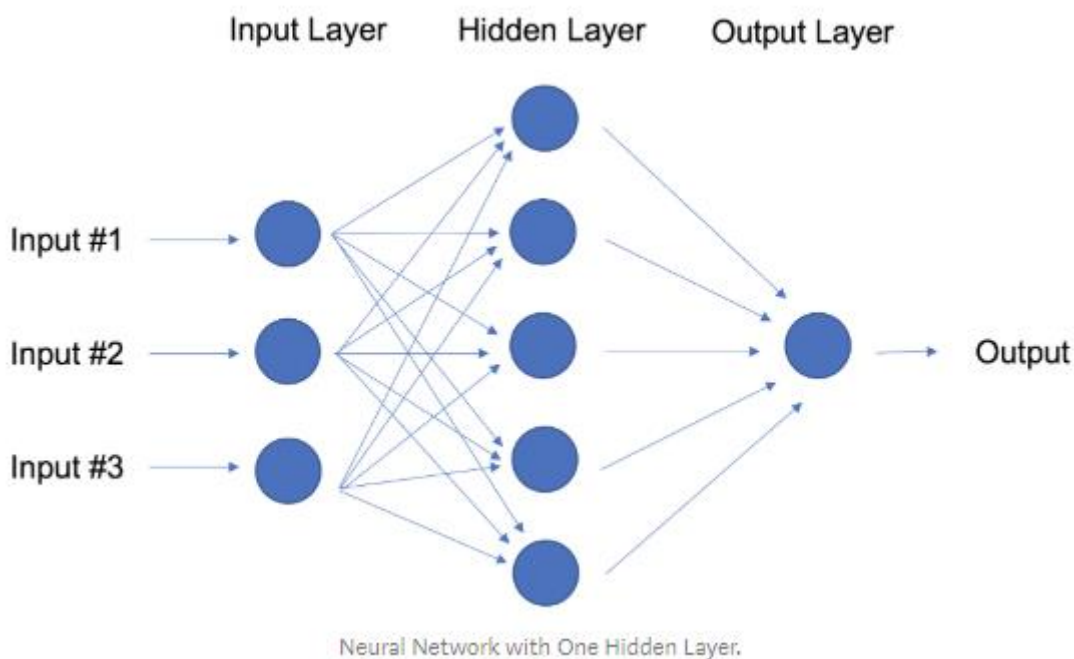
Ansampli metode koriste istu ideju kombinacije nekoliko modela predviđanja (supervizirano mašinsko učenje) kako bi se dobila kvalitetnija predviđanja nego što je svaki od modela mogao pružiti samostalno. Takav je Random Forest algoritam koji je metoda koja kombinira mnoštvo stabala odlučivanja istreniranih s različitim uzorcima skupova podataka, kao rezultat, kvalitete predviđanja šume veća je od kvalitete predviđanja procijenjenih jednim stablom odluka. [4]

Metode ansambla su tu kao 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, s drugim modelom, relativna tačnost može biti obrnuta. Kombinacijom dva modela kvaliteta prognoze se uravnotežuje. Velika većina najboljih pobjednika natjecanja u Kaggleu koristi svojevrzne ansambl metode. Najpopularniji algoritmi ansambla su Random Forest, XGBoost i LightGBM. [4] [9]



## 4.6 Neuronske mreže i duboko učenje

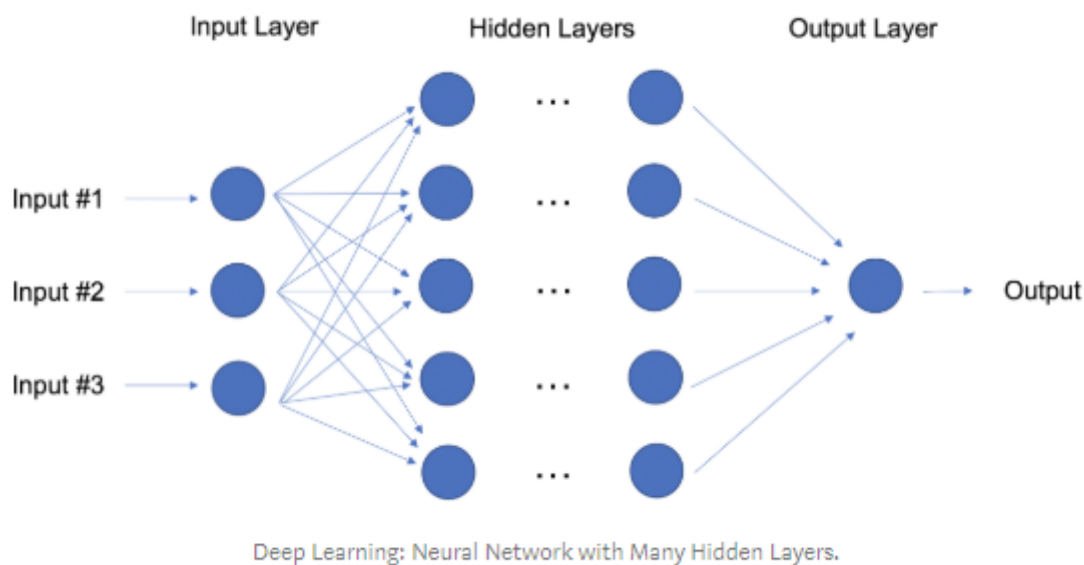
Za razliku od linearnih i logističkih regresija koje se smatraju linearnim modelima, cilj neuronskih mreža je iskoristiti nelinearne obrasce u podatke dodavanjem slojeva parametara u model. Na slici 4.6.1 jednostavna neuronska mreža ima tri ulaza, jedan skriveni sloj s pet parametara i izlazni sloj.



**Slika 4.6.1.** Neuronske mreže sa skrivenim slojem

Struktura neuronskih mreža je dovoljno fleksibilna da izgradi dobro poznatu linearnu i logističku regresiju. Izraz duboko učenje (deep learning) [18] dolazi od neuronske mreže s mnogim skrivenim slojevima (vidi sliku 4.6.2) i obuhvata široku paletu arhitektura.

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



**Slika 4.6.2.** Neuronske mreže sa više skrivenih slojeva

Za najbolju izvedbu, tehnike dubokog učenja, zahtijeva se puno podataka i puno računске snage, jer metoda samoinicijalizira mnoge parametre u ogromnim arhitekturama. Vrlo brzo postaje jasno zašto praktikantima dubokog učenja trebaju vrlo moćni računari poboljšani GPU-ovima (grafičkim procesorskim jedinicama).

Konkretno, tehnike dubokog učenja bile su izuzetno 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

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, samo uz pomoć Transfer Learning-a( prijenosnog učenja) se znanje koje je ugrađeno u prvi model može primijeniti na drugi model.

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

Glavna prednost transfernog učenja je u tome što treba manje podataka za trening neuronske mreže, što je posebno važno jer je obuka za algoritme dubokog učenja skupa i u vremenu i u novcu – i često je vrlo teško pronaći dovoljno označenih podataka za trening.

Ukoliko se pretpostavi da se za model košulje koristi neuronska mreža s 20 skrivenih slojeva, nakon što se pokrenu eksperimenti, postane očigledno da se može prenijeti 18 slojeva modela košulje i kombinirati ih s jednim novim slojem parametara da bi uvježbavali slike pantolona. Model pantolona bi tada imao 19 skrivenih slojeva. Ulazi i izlazi iz dvaju zadataka različiti su, ali slojevi koji se mogu ponovno upotrijebiti mogu rezimirati informacije koje su relevantne za oba.

Prijenosno učenje postajalo je sve popularnije i sada je na raspolaganju mnogo solidnih unaprijed obučениh 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 na 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, mogu se osposobiti telefoni za automatsko dovršavanje tekstualnih poruka ili ispravljanje pogrešno napisanih riječi. Čak se može naučiti mašina da jednostavno razgovara s 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 (riječ, internetski blogovi,...), većina ovih tekstualnih dokumenata bit će 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.

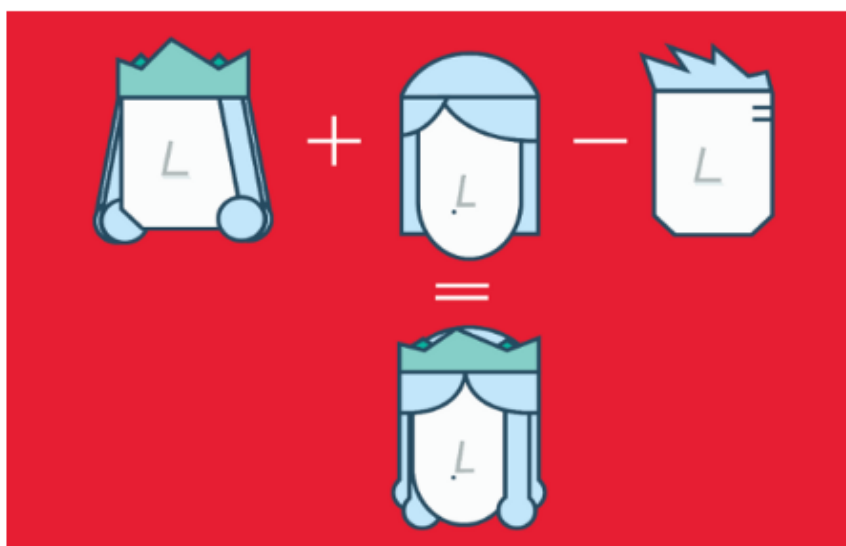
Najjednostavniji način preslikavanja teksta u numerički prikaz je izračunavanje učestalosti svake riječi unutar svakog tekstualnog dokumenta. Dobar primjer je matrica cijelih brojeva gdje svaki red predstavlja tekstualni dokument, a svaka kolona predstavlja riječ. Ova matrična reprezentacija frekvencija riječi uobičajeno se naziva terminska frekvencijska matrica (TFM). Odatle se može stvoriti 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 korpusu dokumenata. Ova metoda se naziva metoda obrnute frekvencije dokumenata (TFIDF) i ona obično bolje funkcionira za trening mašinskog učenja. [5] [6] [9]

## 4.9 Umetanja riječi

TFM i TFIDF su numerički prikazi tekstualnih dokumenata koji samo tekstualnu i težinsku frekvenciju predstavljaju kao tekstualne dokumente. Suprotno tome, ugrađivanje riječi može obuhvatiti kontekst riječi u dokumentu. U kontekstu riječi, ugrađenja mogu kvantificirati sličnost riječi, što zauzvrat omogućava aritmetiku s riječima.

Word2Vec je metoda koja se temelji na neuronskim mrežama koje preslikavaju riječi u korpus numeričkih vektora. Potom te vektore koristi za pronalaženje sinonima, izvođenje aritmetičkih operacija riječima ili za predstavljanje tekstualnih dokumenata (uzima se srednja vrijednost svih vektora riječi u dokumentu). Ukoliko se koristi dovoljno veliki korpus tekstualnih dokumenata za procjenu umetanja riječi gdje su riječi kralj, kraljica, muškarac i žena, dio korpusa, 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.

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



Arithmetic with Word (Vectors) Embeddings.

**Slika 4.9.1.** Umetanje riječi

Reprezentacije riječi omogućavaju pronalaženje sličnosti između riječi računanjem kosinusne sličnosti između vektorskog predstavljanja dvije riječi, sličnost kosinusa mjeri ugao između dva vektora. Izračunava se ugradnja riječi pomoću metoda mašinskog učenja, što je često korak prije primjene algoritma mašinskog učenja. 4] [5]

## 5 Machine learning database - MLDB

Rješenja za mašinsko učenje u stvarnom svijetu rijetko su samo pitanje izgradnje i testiranja modela. Upravljanje i automatizacija životnog ciklusa modela mašinskog učenja od obuke do optimizacije daleko je najteži problem koji se može riješiti u računarskim rješenjima. 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 puno vremena pokušavajući prilagoditi bazi podataka NOSQL modelima mašinskog učenja prije nego što dođu do ne baš očitog zaključka: 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 predstavlja strukturu mašinskog učenja poput duboke neuronske mreže. Platforme u cloud mašinskom učenju poput AWS SageMaker ili Azure ML već uključuju modele grafikona mašinskog učenja, pa zašto nam onda treba drugo rješenje? Ipak postoji dosta zahtjeva stvarnih rješenja mašinskog učenja koje mogu imati koristi od stvarne baze podataka: [10]

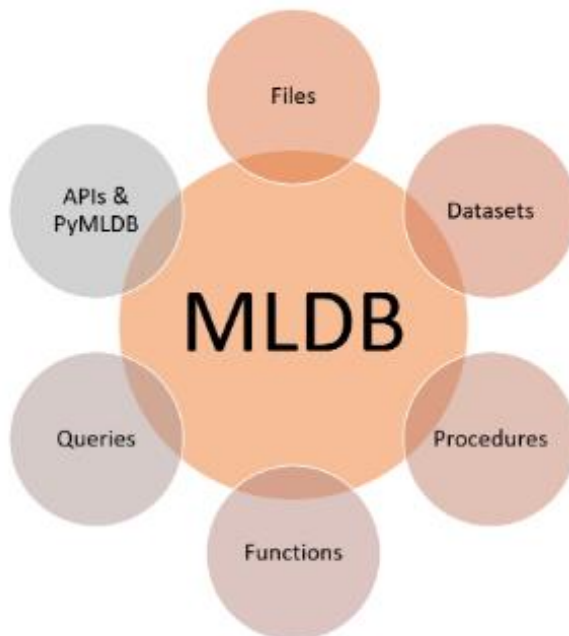


**Slika 5.1.** Životni ciklus MLDBa

MLDB nudi open-source bazu podataka za pohranu i određivanje upita mašinskog učenja. Platforma je prvi put inkubirana u sklopu Datacrat-a, a nedavno ju je stekla kompanija AI powerhouse Elementai kao potvrda važnosti motora baze podataka u modernim projektima mašinskog učenja. 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 kombinira različite artefakte koji apstrahuju različite elemente životnog ciklusa mašinskog učenja. Tehnički se model MLDB može sažeti u šest jednostavnih komponenti: datoteke, skupovi podataka, procedure, funkcije, upiti i API-ji. [11]



**Slika 5.1.1.** Prikaz arhitekture MLDBa

Datoteke [10]

Datoteke predstavljaju uobičajenu 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 [10]

MLDB skupovi podataka predstavljaju glavnu jedinicu podataka koju koriste postupci i modeli mašinskog učenja. Strukturno su skupovi podataka bez shema, samo imaju nazive skupova podataka, koji se nalaze u ćelijama koje su na sjecištu redova i kolona. Tačke podataka sastoje se od vrijednosti i vremenske oznake. Svaka podatkovna tačka može se tako predstaviti kao (red, kolona, vremenska oznaka, vrijednost) zbir, a skupovi podataka mogu se smatrati trodimenzionalnim matricama. Skupovi podataka mogu se stvoriti i podaci im se mogu dodati putem RLD API-ja MLDB-a, a mogu se također učitati u datoteke ili ih pohraniti putem procedura.

## Procedure [10]

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 koji 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.

## Funkcije [10]

MLDB funkcije koriste apstraktne podatke radi izračunavanja koji se koriste u procedurama. Funkcije su imenovani programi za višestruku upotrebu koji se koriste za implementiranje izraza koji mogu prihvatiti ulazne vrijednosti i vratiti izlazne vrijednosti. MLDB funkcije objedinjuju SQL izraze koji izražavaju specifično računanje.

## 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 pripremanje skupa podataka za treniranje modela klasifikacije slike:

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

## API-ji i Pymldb [10]

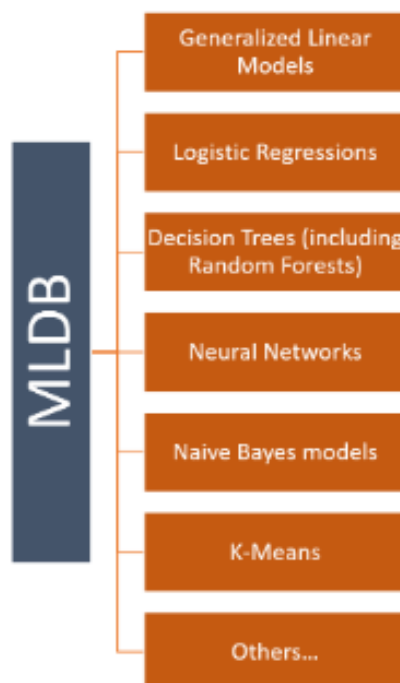
Sve mogućnosti MLDB-a izložene su putem jednostavnog REST API-ja. Platforma također uključuje pymldb, Python biblioteku koja apstraktno prikazuje mogućnosti API-ja u prijateljskoj sintaksi. Sljedeći kôd pokazuje kako koristiti pymldb za stvaranje i za upite baze podataka.

```
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)
```



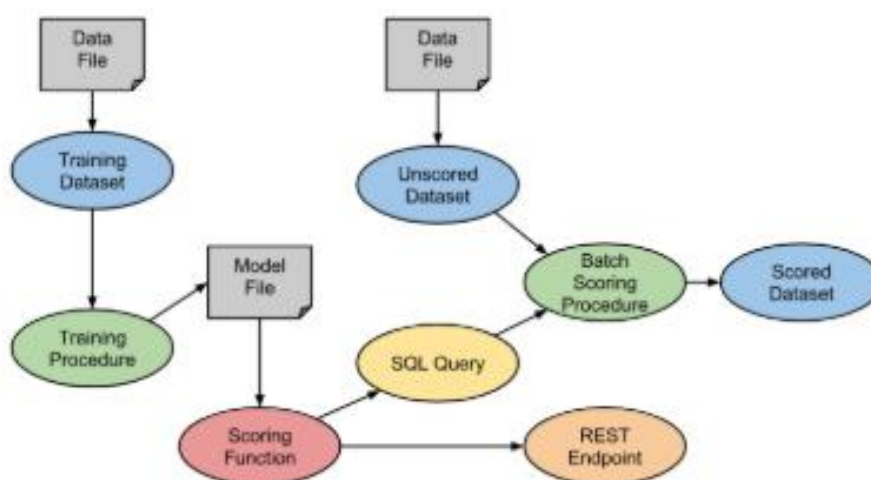
## 5.2 Podrška za algoritme mašinskog učenja

MLDB pruža podršku za veliki broj algoritama koji se mogu koristiti iz postupaka i funkcija. Platforma također izvorno podržava računarski grafik različitih tehnika za duboko učenje poput TensorFlow. [10]



**Tabela 5.2.1.** Prikaz tehnika MLDBa

Uzme se zajednički tok rada u rješenjima za mašinsko učenje kao što su trening i bodovanje modela. Slika 5.2.2 prikazuje kako će se izvesti to u MLDB. [10] [11]



**Slika 5.2.2.** Tok rada mašinskog učenja

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

- Proces započinje s datotekom o treningu, koja se učitava u skup podataka o treningu.
- Vršiti se trening postupak za proizvodnju modela datoteke
- Datoteka modela koristi se za parametrisiranje funkcije bodovanja
- Ova funkcija bodovanja je odmah dostupna putem REST krajnje tačke za pregled u stvarnom vremenu
- Funkcija bodovanja je odmah dostupna putem SQL upita
- Postupak bodovanja koristi SQL za primjenu funkcije bodovanja na neodlučeni skup podataka u paketu, stvarajući ocjenjeni skup podataka

Ključne karakteristike MLDB-a koje odgovaraju nizu aplikacija su:

**Brzina:** Proces treninga, modeliranja i pronalaska podataka u MLDB-u zahtjeva dobre performanse. Ima veliku moć obrade u poređenju s H2O, Scikit-Learn ili Spark MLlib, koje su poznate kao istaknute biblioteke mašinskog učenja.

**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 grafikonima TensorFlow-a koji ga čine superiornim u otkrivanju znanja.

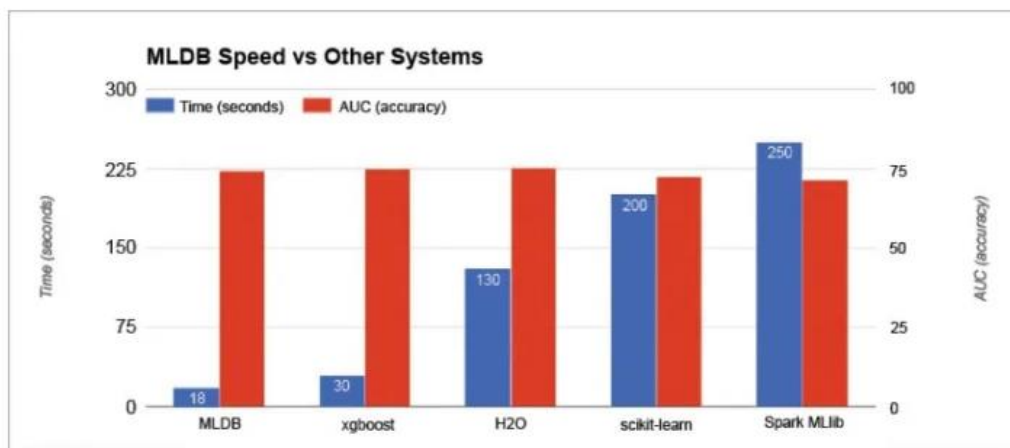
**Jednostavnost implementacije:** Postoje instalacijski paketi za više platformi i programska okruženja, uključujući Jupyter, Docker, JSON, Cloud, Hadoop i mnoge druge.

**Kompatibilnost i integracija:** MLDB omogućuje veći 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 raspoređuje na HTTP portu koji omogućuje 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 s NoSQL bazama podataka za nestrukturirane i heterogene skupove podataka.

Slika 5.2.3 prikazuje performanse MLDB-a u poređenju sa drugim bibliotekama. Izvođenje 100 Tree Random Forest pristupa vrši se na 1 milion redova s jednim čvorom pomoću MLDB 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 uspoređuje s ostalim bibliotekama mašinskog učenja. Učinkovitost MLDB-a usporediva je s performansama xgboost, H2O, Scikit-Learn i Spark MLlib. [10] [11] [12]

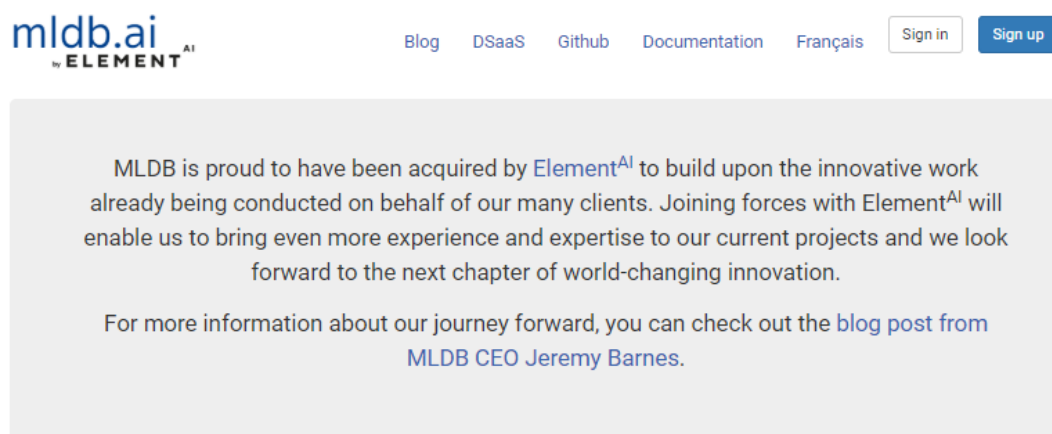


Slika 5.2.3. Učinkovitost MLDBa

### 5.3 Instaliranje MLDB-a na različitim platformama

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

Postoje dva izdanja MLDB-a koja su besplatna i distribuirana su kao izdanja zajednice i poduzeća. Da biste pokrenuli MLDB Enterprise Edition, morate unijeti licencni ključ za aktiviranje softvera. Licencni ključ može se napraviti za prve korisnike prilikom prijave na [https://mldb.ai/#license\\_management](https://mldb.ai/#license_management) i ispunjavanja potrebnih uslova u obrascu za registraciju. [12]



## MLDB is the Machine Learning Database

MLDB is an [open-source](#) database designed for machine learning.  
You can [install it wherever you want](#) and send it commands over a [RESTful API](#) to store data, [explore it using SQL](#), then [train machine learning models](#) and [expose them as APIs](#).

**Slika 5.3.1.** Oficijalna stranica MLDB baze podataka

	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

MLDB je jedan od prvih primjera baza podataka dizajniran od temelja kako bi se omogućila rješenja mašinskog učenja. Platforma se još uvijek može puno poboljšati radi podrške modernim tehnikama mašinskog i dubokog učenja, ali upravo njena fleksibilnost i proširivost čini je prvom iteracijom u novom vremenu. [12]

## 6 Opis skupa podataka recenzije putovanja

Skup podataka koji koristim 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 sa sokovima

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

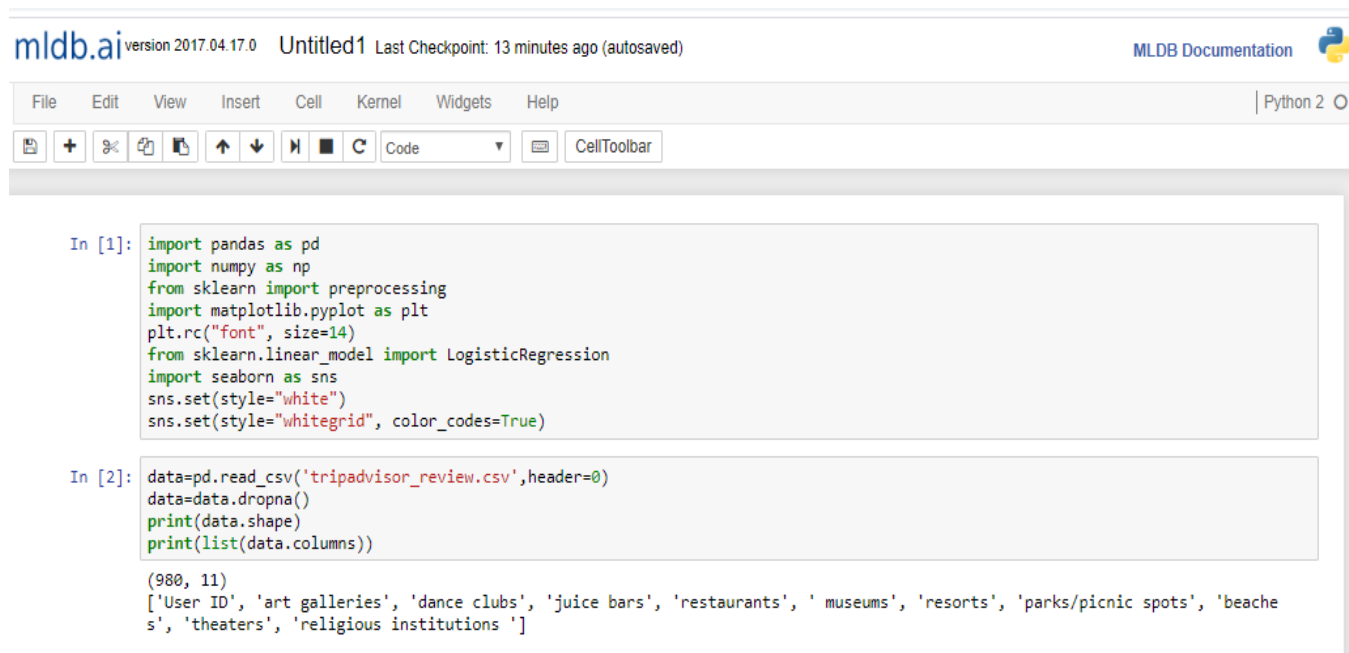
## 7 Praktični rad koristeći bazu MLDB i programski jezik Python

Kako bi koristili MLDB lokalno na računaru za te potrebe instaliran je Oracle VM Virtual Box. Nakon importovanja mldb.ova fajla pronađenog na stranici <http://public.mldb.ai/mldb.ova> i pokretanjem u virtuelnoj mašini na localhost:8080 izvrši se pokretanje mldb.ai baze podataka. [12]

Nakon pokretanja kreiran je novi folder te unutar foldera fajl Python 2.

Izvršeno je importovanje svih potrebnih biblioteka kao i učitavanje skupa podataka.

Prvo je izvršen upload csv fajla koji je pronađen na UCI oficijelnoj stranici. Nakon toga izvršeno je čitanje fajla u posebnu varijablu data (linija 2). Prikaz čitave tabele učinjeno je pozivom data i dobijen je prikaz tabele od 980 redova i 11 kolona [16]

The image is a screenshot of the mldb.ai web interface. At the top, it shows the mldb.ai logo, version 2017.04.17.0, and a document titled 'Untitled1' with a last checkpoint of 13 minutes ago. There are links for 'MLDB Documentation' and a Python logo. Below this is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. A toolbar contains icons for file operations, a dropdown menu set to 'Code', and a 'CellToolbar' button. The main area displays two code cells. The first cell, labeled 'In [1]:', contains imports for pandas, numpy, sklearn preprocessing, matplotlib, seaborn, and LogisticRegression, along with styling settings for seaborn. The second cell, labeled 'In [2]:', contains code to read a CSV file, drop the first column, and print the data's shape and columns. The output of the second cell is shown below the code: a tuple (980, 11) and a list of column names including 'User ID', 'art galleries', 'dance clubs', 'juice bars', 'restaurants', 'museums', 'resorts', 'parks/picnic spots', 'beaches', 'theaters', and 'religious institutions'.

Slika 7.1. Kod učitavanja 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

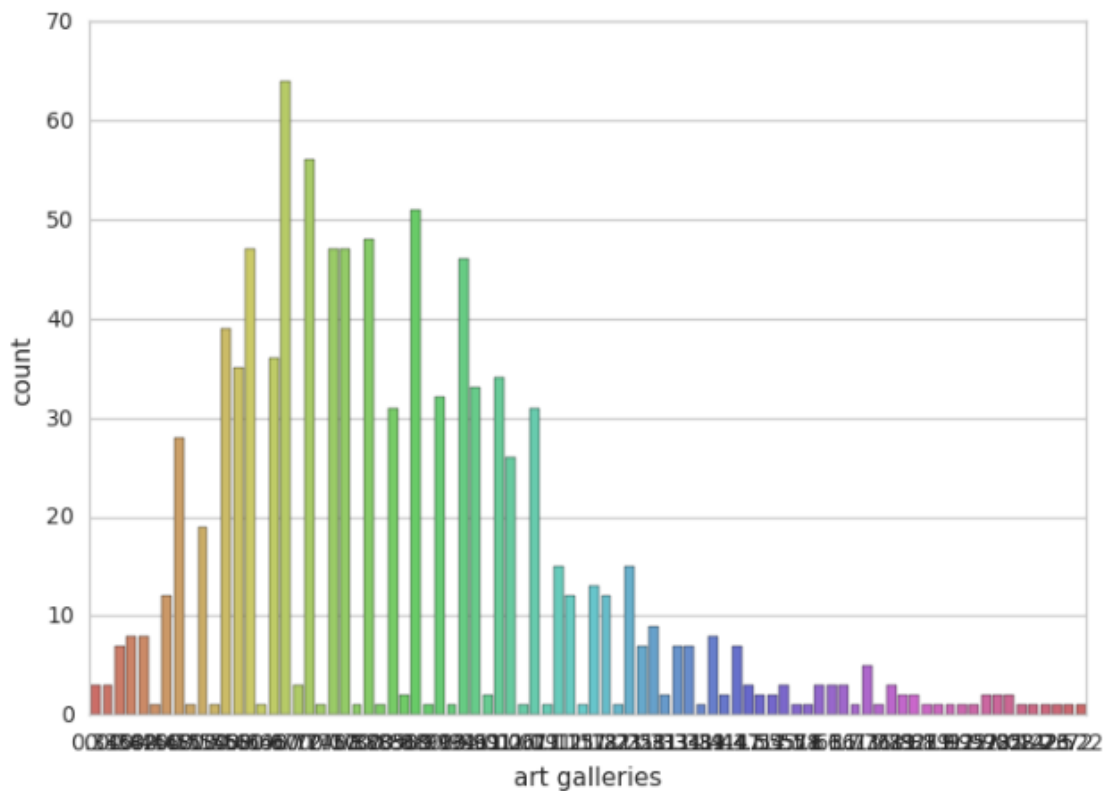
Opažanjem tabele zaključuju se određene činjenice:

Postoji 10 kategorija mjesta u istočnoj Aziji koji imaju recenzije od 0-4. Najbolja tehnika mašinskog učenja za primjenu je klasifikacija i klasterisanje jer 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 jasno zašto sam odabrala ove metode.

Pedstavimo prvo grafički sve ove kategorije kako bi na osnovu grafika došli do procjene mogućeg rješenja i kategorija od značaja [12]

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

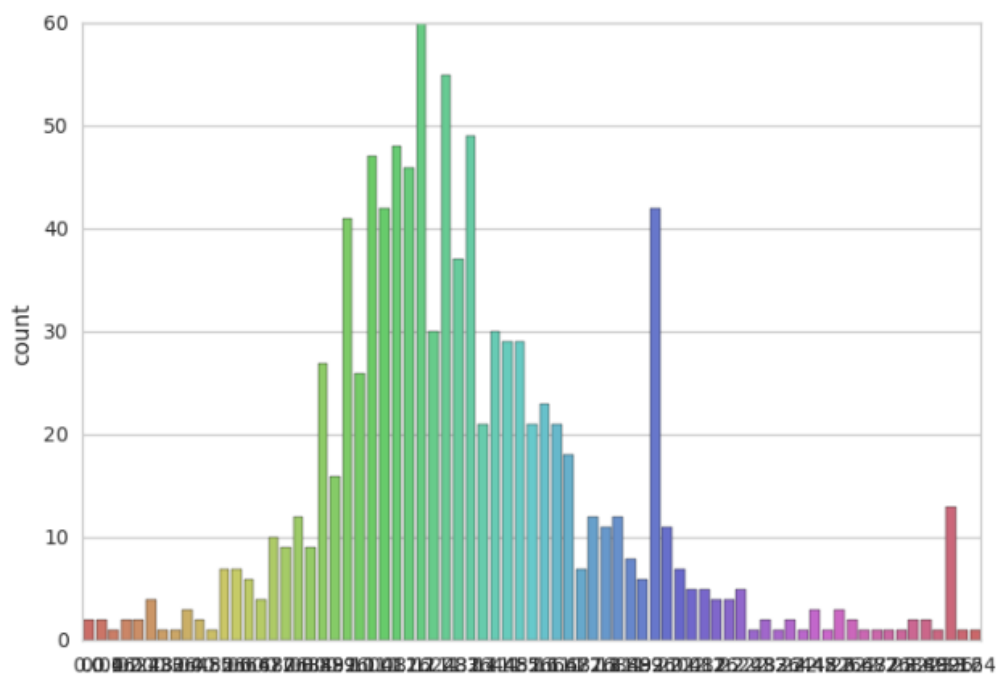


**Slika 7.2.** Prikaz broja umjetničkih galerija

Umjetničke galerije su u rasponu od 0-3 gdje je veliki broj ljudi dalo ocjenu, ali ipak ima puno oscilacija u recenzijama.

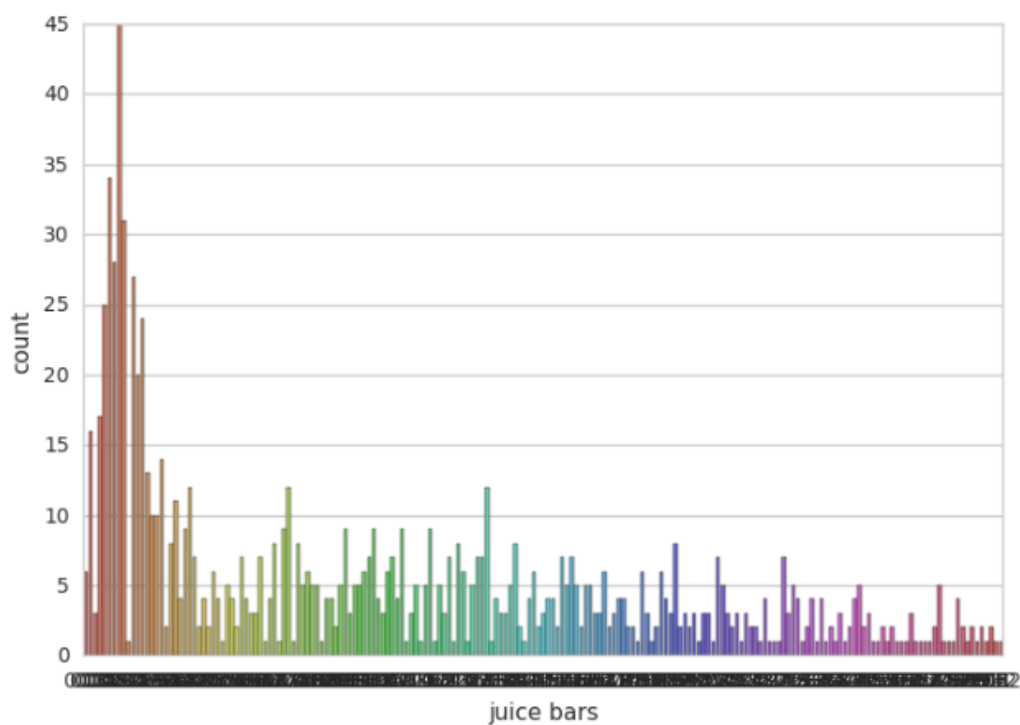


```
In [16]: sns.countplot(x='dance clubs', data=data, palette='hls')
plt.show()
plt.savefig('count_plotclubs')
```



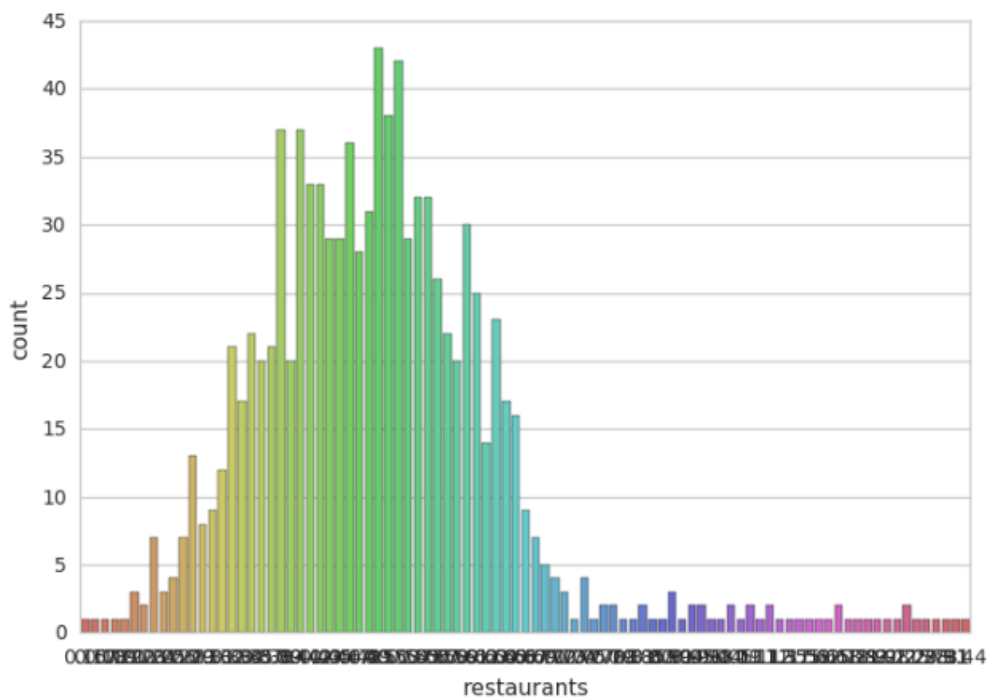
**Slika 7.3.** Prikaz broja plesnih klubova

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



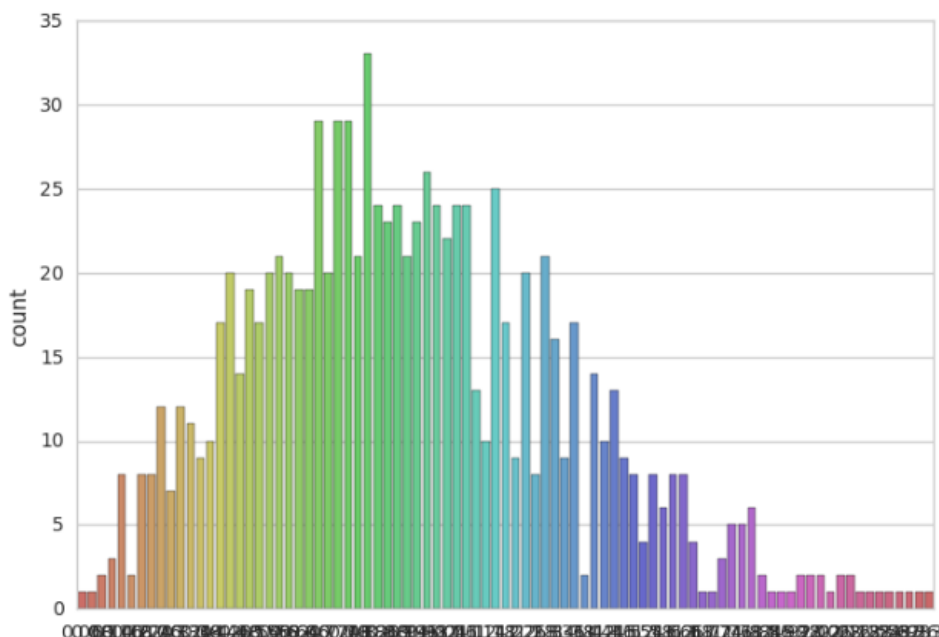
**Slika 7.4.** Prikaz broja barova

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



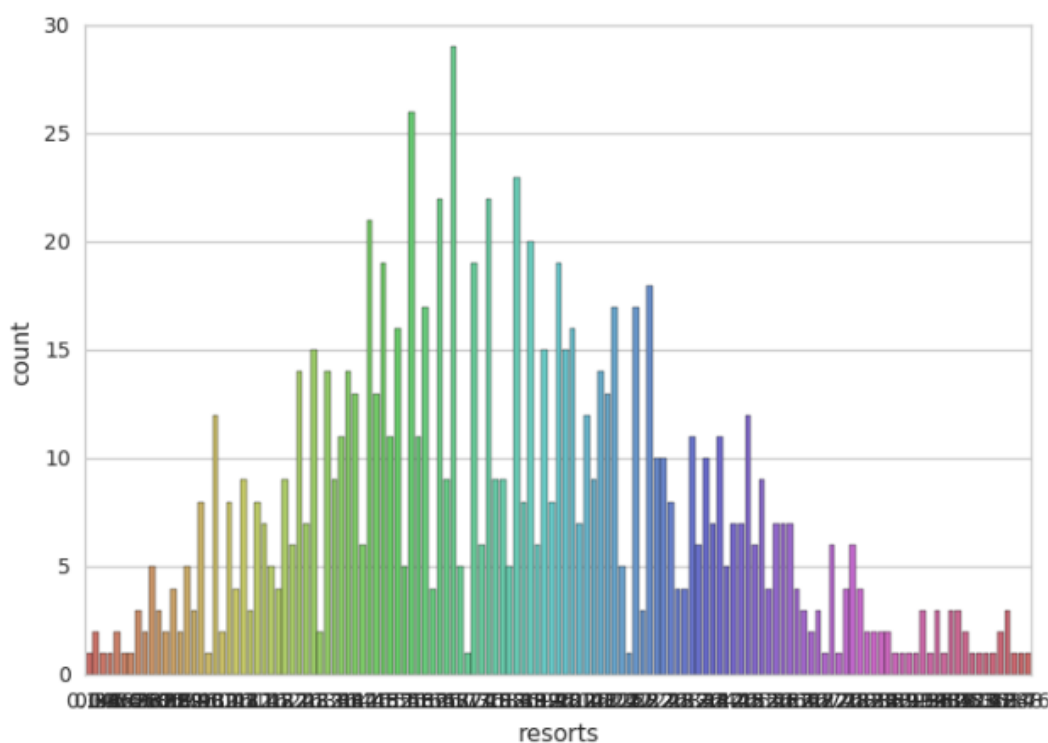
Slika 7.5. Prikaz broja restorana

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



Slika 7.6. Prikaz broja muzeja

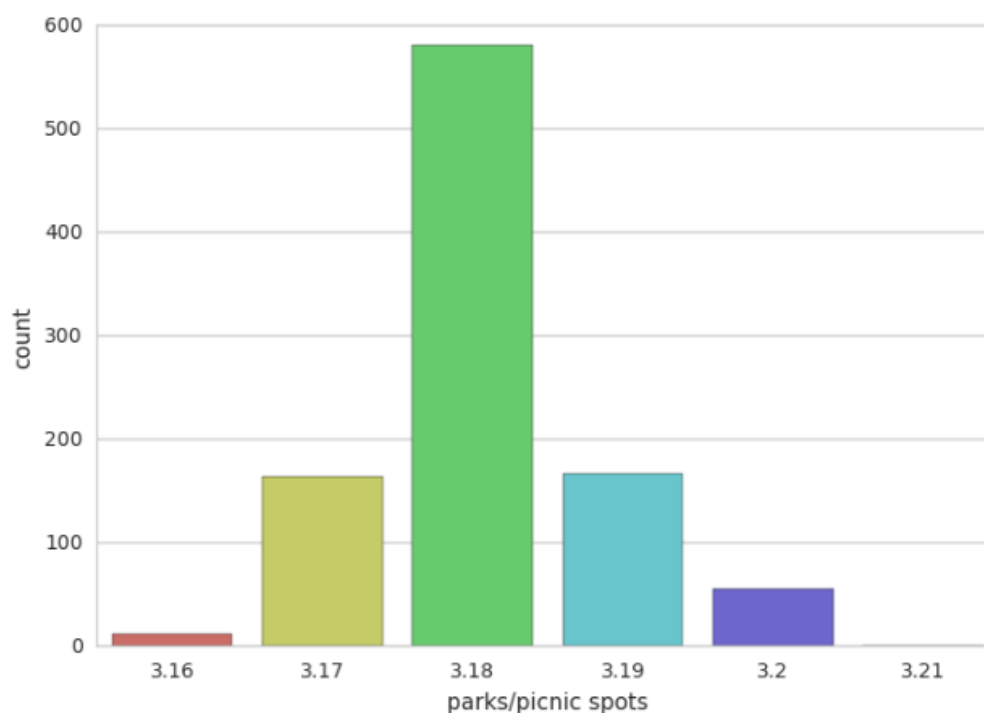
```
In [12]: sns.countplot(x='resorts', data=data, palette='hls')  
plt.show()  
plt.savefig('count_plot')
```



**Slika 7.7.** Prikaz broja odmarališta

Situacija je ista kod svih prethodnih grafika puno oscilacija i veliki broj ljudi do 50 ili 70 daje ocjenu u rasponu od 0-3 ili od 2-4 u potrazi smo za boljim rješenjem tj. za grafikom koji daje bolju sliku.

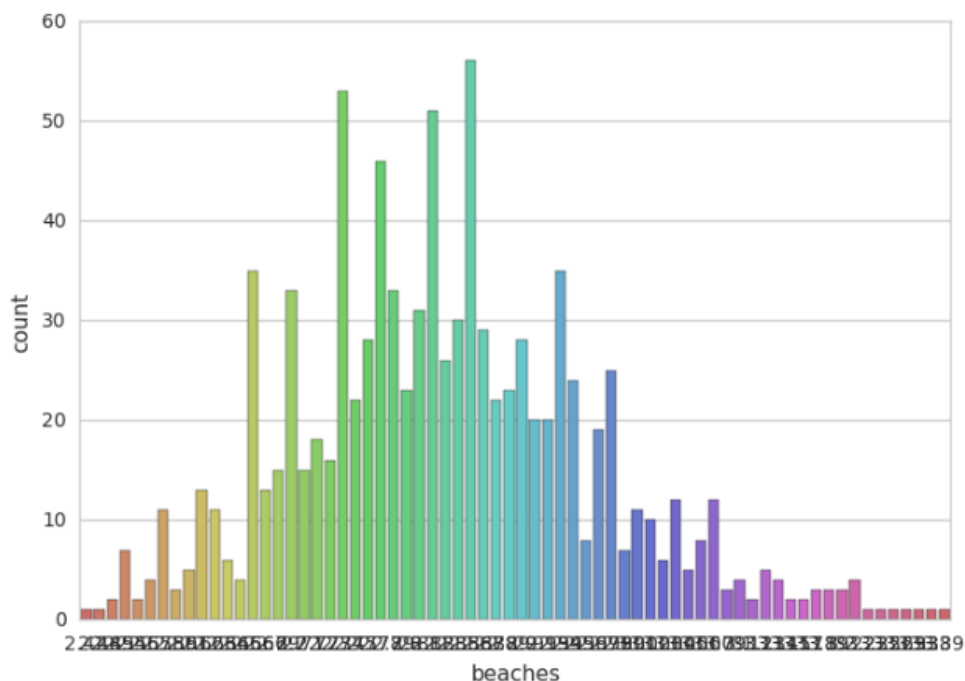
```
In [25]: sns.countplot(x='parks/picnic spots', data=data, palette='hls')  
plt.show()  
plt.savefig('count_plot_parks')
```



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

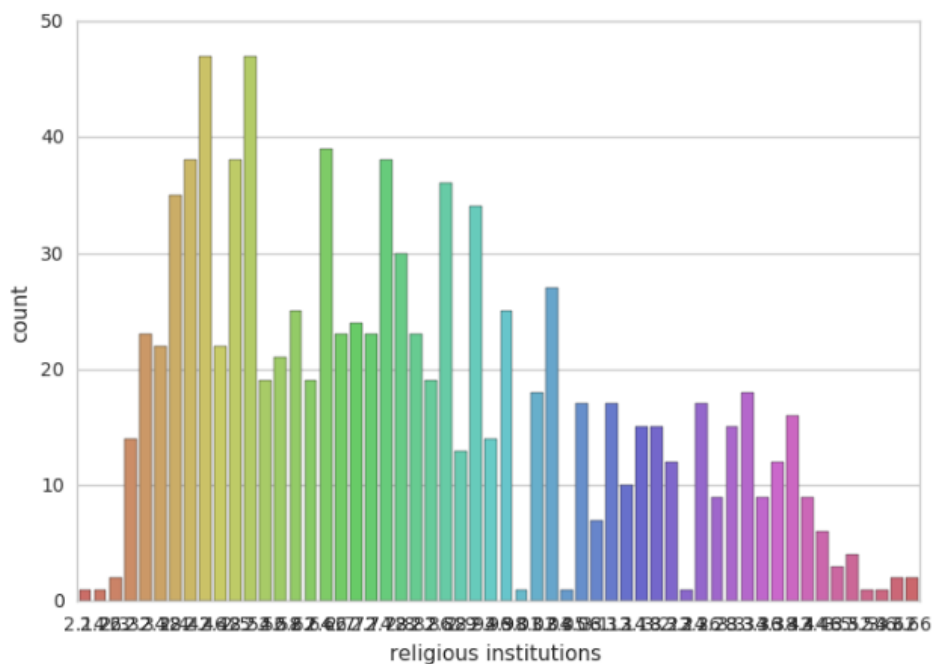
Ovaj grafik parkovi/mjesta za piknik daje najbolju sliku gdje je do 600 ljudi dalo ocjenu od 3.16-3.21 što je značajna informacija u odnosu na prethodne grafike.

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



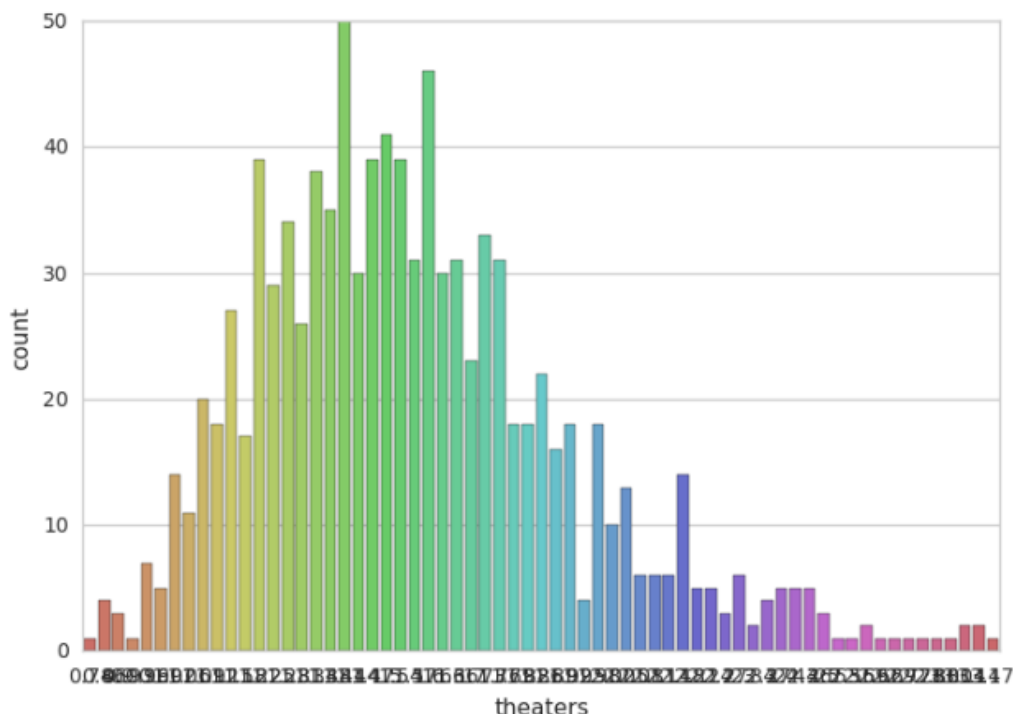
Slika 7.9. Prikaz broja plaža

```
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.

Zaključujemo da mjesta koja su ljudima najmanje skupa, najviše se i koriste i ta mjesta često budu i najbolja tako da imamo dvije kategorije od značaja parkovi/piknik mjesta i religijske institucije kao 2 najpoželjnija mjesta.

Pogledajmo sada metodu klasteringa tačnije K-means i raspodijelimo kategorije u 10 klastera.

Za ove potrebe korisiti se procedura pa je potrebno otvoriti mldb konekciju.

```
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")
```

	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“

Napravimo proceduru za treniranje koristeći K-means metodu klasteringa.

```
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": "2020-03-08T19:37:43.336991Z",
      "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

Vidimo da je statusni kod 201 te je mreža treniranja kreirana.

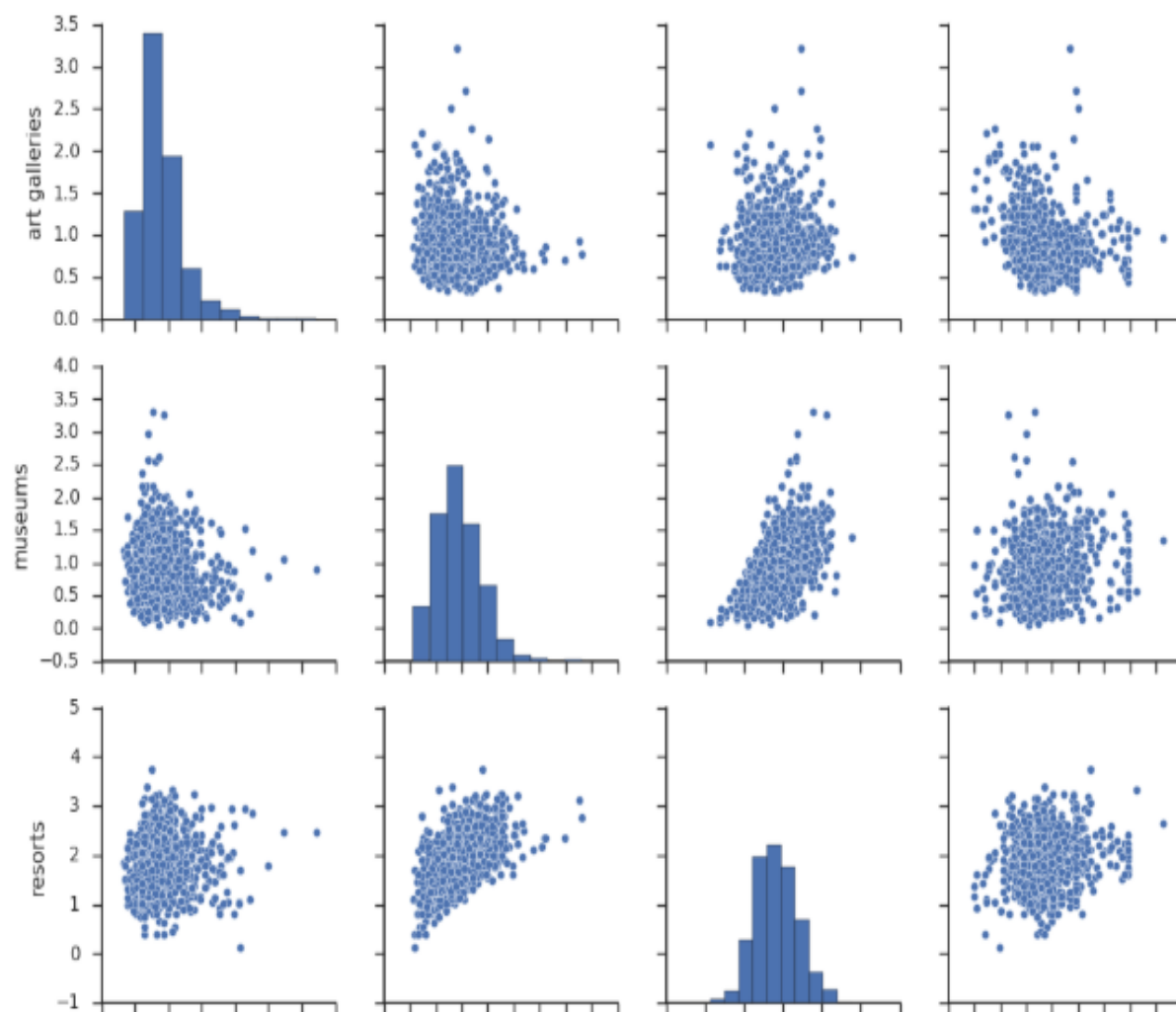
Rezultati se svakako poklapaju sa procjenom pa su glavne 2 kategorije parkovi/piknik mjesta i religijske institucije.

Grafici nekih od rezultata.



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

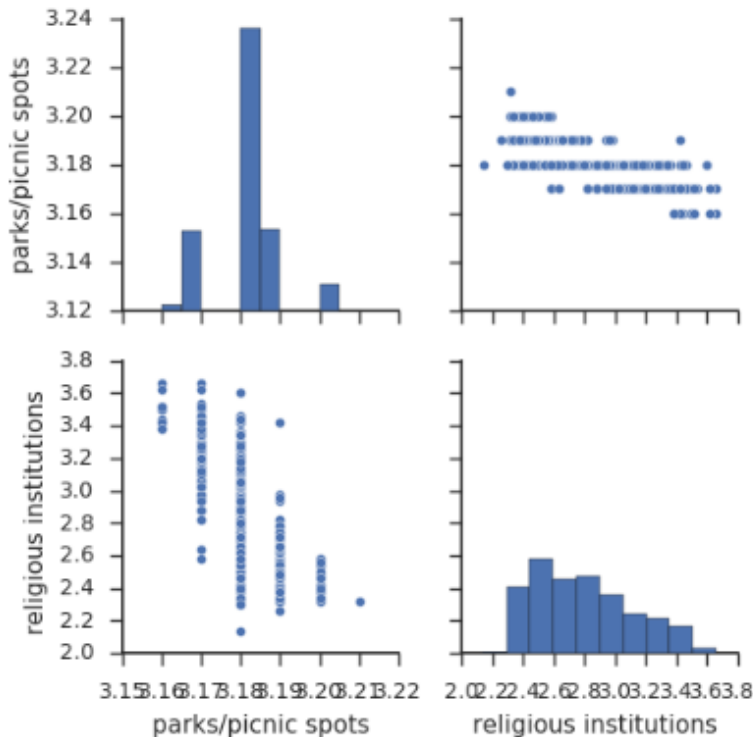
<matplotlib.figure.Figure at 0x7f7fd3b9710>



**Slika 7.15.** Grafici pojedinih kategorija

Kako bi donijeli finalnu odluku koja je to kategorija najpoželjnija upariti ćemo na grafiku dvije kategorije od značaja parkove/piknik mjesta i religijske institucije. Iskoristit ćemo i logističku regresiju kako dobili konkretan broj.

```
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 dolazimo do zaključka da su parkovi ili piknik mjesta najpoželjnija za posjetiti u istočnoj Aziji, a i procjenom logističke regresije do 98% možemo biti sigurni da smo došli do dobrog zaključka.

## 8 Zaključak

Mašinsko učenje izgrađeno je na statističkom okviru. Pored toga mašinsko učenje obuhvata i veliki broj drugih područja matematike i računarskih nauka. Može se istaknuti da se statističko modeliranje više bavi pronalaženjem odnosa između varijabli dok mašinsko učenje pored toga u prvi plan stavlja i predviđanje (predikciju), evaulaciju tih predviđanja i upotrebljivost sistema za predviđanje. [7]

Na kraju su usvojena mnoga znanja o mašinskom učenju, tehnikama mašinskog učenja i na konkretnom primjeru prikazana je i primjena tih tehnika.

Što se tiče analize recenzija putovanja dobijen je zaključak da su parkovi/piknik mjesta nabolja za posjetiti u istočnoj Aziji. Parkova ima mnogo, često je ulaz besplatan, koriste ga i djeca i mladi i starije osobe tako da je za sve generacije, zato je to najposjećenije mjesto. 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. To su mjesta koja su zaista najpoželjnija u čitavoj toj regiji.

## 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]	Tehnnike 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]	Procedure 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>
[16]	Skup podataka 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>
[17]	Jeff Dean intervju- mašinsko učenje, NLP i BERT 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>
[18]	Deep learning Dostupno na: <a href="https://www.investopedia.com/terms/d/deep-learning.asp">https://www.investopedia.com/terms/d/deep-learning.asp</a>