1. **Što je strojno učenje? Koja je razlika između eksplicitnog programiranja računala i strojnog učenja? Navedite kakvi problemi se rješavaju pojedinim pristupom (primjeri).**

Strojno učenje je grana umjetne inteligencije koja se bavi razvojem algoritama i tehnika koje omogućuju računalima da nauče iz podataka i obavljaju zadatke bez eksplicitnog programiranja. Strojno učenje omogućuje računalu da “uči” iz primjera i iskustava te samo donese odluke na temelju tih podataka.

Razlika između eksplicitnog programiranja računala i strojnog učenja leži u pristupu rješavanju problema. U eksplicitnom programiranju, programer mora detaljno definirati pravila i korake za rješavanje problema. Ovo je prikladno za probleme za koje se mogu unaprijed predvidjeti svi mogući scenariji (ishodi). Strojno učenje koristi algoritme koji prilagođavaju svoje modele i parametre kako bi minimizirali greške na temelju primjera. Ovo je prikladno za probleme kod kojih ne možemo definirati sve moguće ishode.

Primjeri problema koji se rješavaju **eksplicitnim programiranjem računala**:

* Algoritam sortiranja
* Definiranje pravila za upravljanje prometnim signalima na raskrižjima

Primjeri problema koji se rješavaju **strojnim učenjem**:

* Prepoznavanje lica (face recognition)
* Predviđanje cijena nekretnina na tržištu na osnovu povijesnih podataka
* For you page na društvenim mrežama

1. **Navedite tipove strojnog učenja te za svaki tip minimalno dva primjera.**

Nadzirano učenje (*supervised learning*):

* Procjena cijena nekretnina (regresija)
* E-mail spam filtar (binarna klasifikacija - izlaz je 1 ili 0)
* Prepoznavanje rukom pisanih brojeva (višeklasna klasifikacija)

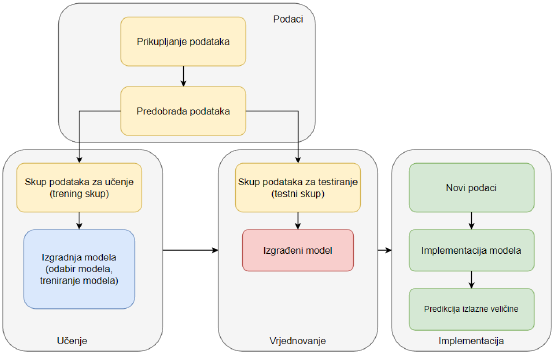
Nenadzirano učenje (*unsupervised learning*):

* Segmentacija kupaca
* Analiza društvenih mreža
* Kompresija podataka

Podržano učenje (*reinforcement leaning*):

* Igra snake
* Autonomna vozila
* Marketing i reklamiranje

1. **Nacrtajte tipičan redoslijed radnji kod primjene algoritma strojnog učenja u prediktivnom modeliranju. Objasnite pojedini korak.**



Prikupljanje podataka: ako podatkovni skup nije na raspolaganju

Predobrada podataka: dobro se upoznati s podacima

Učenje modela: primjenom linearne ili logističke regresije, neuronske mreže, ...

Izgradnja modela → odabir modela: odabir arhitekture

Izgradnja modela → treniranje modela: procjena parametara odabranog modela

**BITNO: Model se gradi isključivo na temelju skupa za učenje!**

Vrjednovanje modela: evaluiranje na skupu za testiranje. Modelu se predaju vrijednosti ulaznih veličina, a zatim se predikcije modela uspoređuju sa stvarnim vrijednostima.

Implementiranje modela: može biti zahtjevan zadatak ako model radi s velikim brojem podataka.

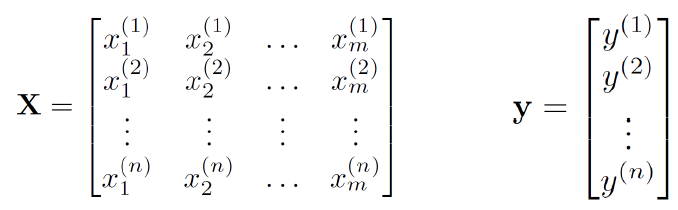
1. **Objasnite tabličnu organizaciju podatkovnog skupa i način zapisivanja podatkovnog skupa u matričnoj notaciji.**

Podaci se najčešće prikazuju u tabličnoj formi (eng. *dataset*).Stupci su veličine, a svaki redak predstavlja jedno mjerenje veličine. Česti formati su CSV, JSON, XML, ...

Npr. ulazne veličine bi bile: marka auta, tip goriva, gradska potrošnja, ...

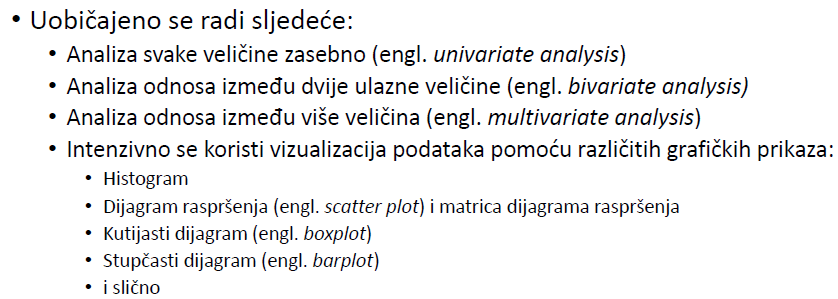
Skup od *n* podataka može se zapisati u obliku matrice X koja sadrži vrijednosti *m* ulaznih veličina i vektor y koji sadrži vrijednosti izlazne veličine za svaki podatak.

Način zapisivanja u matričnoj notaciji:



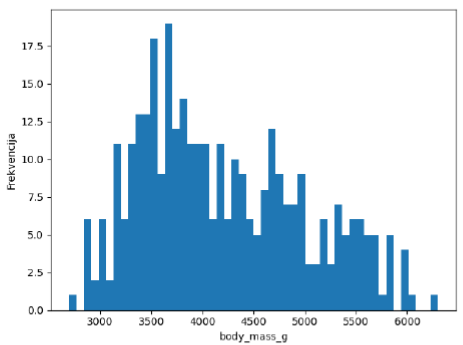
1. **Što je eksplorativna analiza podataka (EDA)? Koji grafički prikazi se koriste prilikom EDA? Objasnite ih.**

Prije predobrade podataka dobro je izvršiti EDA. To nam omogućuje uvid u podatke kao *missing values*, stršeće vrijednosti (eng. *outliers*) te odnos između ulaznih veličina.

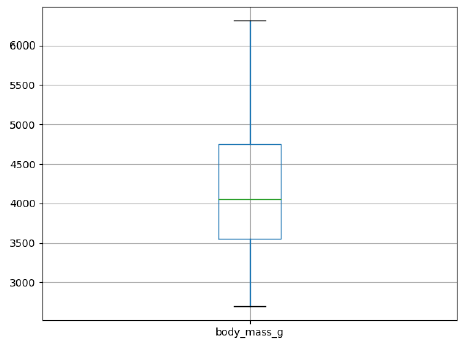


**Grafički prikazi**

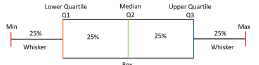
* **Histogram**:
  + Graf koji predstavlja distribuciju skupa kontinuiranih ili diskretnih podataka
  + Na x osi su vrijednosti, a na y osi je učestalost pojavljivanja vrijednosti (frekvencija)
  + Graf:

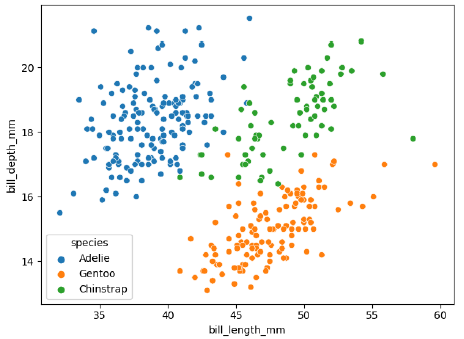


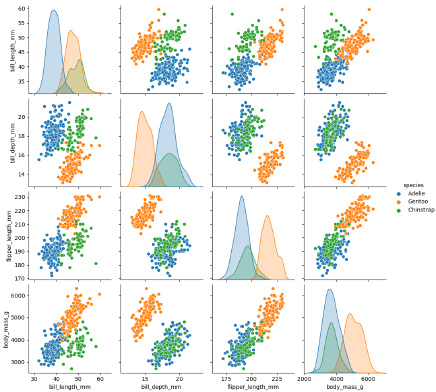
* **Kutijasti**
  + Graf koji predstavlja distribuciju skupa kontinuiranih ili diskretnih podataka
  + Koriste se za uspoređivanje dvije distribucije
  + Nacrtan je od prvog (donjeg) kvartila i trećeg (gornjeg) kvartila, a medijan je označen linijom unutar “kutije”. Ako ima stršećih vrijednosti, prikazuju se pojedinačnim točkama izvan kutije.
  + Graf:



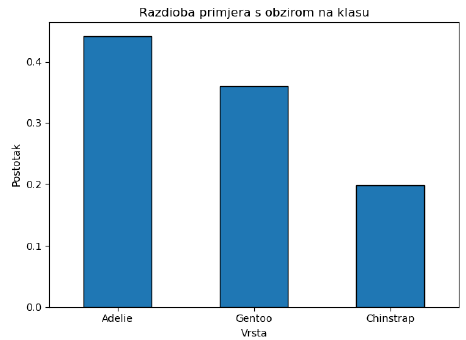
* + Tumač grafa:



* **Dijagram raspršenja**
  + Graf koji predstavlja distribuciju dvije kontinuirane ili diskretne varijable prikazane u dvodimenzionalnoj ravnini
  + Olakšava uvid u odnos između dvije veličine
  + Graf:
* **Matrica dijagrama raspršenja**
  + Prikazuje dijagrame raspršenja svih veličina (kao 1 figure)
  + Prikaz:



* **Stupčasti dijagram**
  + Daje uvid u distribuciju kategoričkih veličina, uključujući učestalost ili broj svake kategorije
  + Stupac = kategorija (x os)
  + Veličina stupca = učestalost ili broj primjera unutar kategorije (y os)
  + Graf:



1. **Objasnite osnovne postupke u predobradi podataka (podatkovnog skupa).**

* Odabrati relevantne ulazne veličine od svih dostupnih
* Ukloniti stršeće i izostale vrijednosti
* Transformirati kategoričke varijable u oblik koji zahtijevaju algoritmi strojnog učenja
* Skalirati numeričke varijable

1. **Objasnite kakve su to kategoričke veličine i načine njihovog kodiranja**

Kategoričke veličine mogu biti:

* **Nominalne** – ne postoji odnos između vrijednosti (npr. boja očiju, spol, ...)
  + Kodiranje: 1-od-K kodiranje (ako veličina ima K mogućih vrijednosti, tada se veličina kodira s K lažnih binarnih veličina (jedna lažna varijabla je 1, ostale su 0)).
* **Ordinalne** – postoji odnos između vrijednosti te je ih je **moguće sortirati** (ocjena, dobna skupina)
  + Kodiranje: Ručno se izvrši mapiranje znakovnih nizova u cjelobrojne vrijednosti
  + Primjer: Low → 0, Medium → 1, High → 2

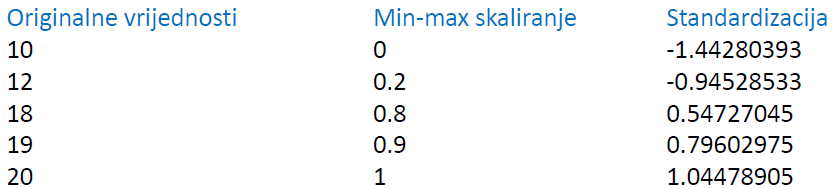
1. **Objasnite načine skaliranja numeričkih varijabli**

Prije procesa učenja skaliramo numeričke veličine tako da sve budu na istoj skali.

Načini skaliranja numeričkih varijabli:

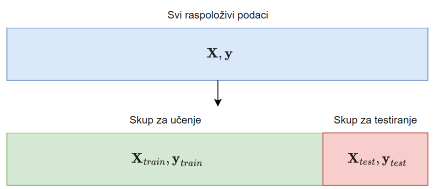
* **Min-max**: transformira ulazne vrijednosti na određeni interval (većinom [0, 1])
* **Standardizacija**: skaliranje gdje podaci za svaku ulaznu veličinu imaju srednju vrijednost 0 i varijancu 1

Primjer:



1. **Objasnite podjelu podatkovnog skupa na dva skupa. Koja je uloga svakog skupa podataka?**

Podatke dijelimo na dva skupa:

* Skup podataka za učenje – odvija se učenje modela i prilagođavanje parametara
* Skup podataka za testiranje – za evaluaciju nakon učenja. Cilj je procijeniti sposobnost modela da primjenjuje naučeno znanje na nove primjere.

**Bitno**:  
Skupovi podataka za učenje i testiranje moraju biti  
međusobno neovisni kako bi se dobila objektivna procjena  
performansi modela.

1. **Objasnite nadzirano učenje i princip označavanja dostupnog skupa podataka. Koja dva glavna tipa problema postoje u okviru nadziranog učenja?**

**Nadzirano učenje (eng. *supervised learning*)** je grana strojnog učenja koja se bavi razvojem algoritama i tehnika za obuku modela na temelju dostupnih podataka. Osnovna ideja je da model uči kako povezati ulazne podatke s odgovarajućim izlaznim podacima.

**Princip označavanja dostupnog skupa podataka** je postupak dodjeljivanja očekivanih izlaznih vrijednosti ulaznim vrijednostima. Npr. ako želimo trenirati model za prepoznavanje slika mačaka i pasa, označavanje podataka uključuje dodjeljivanje oznaka “mačka” ili “pas” svakoj slici u skupu podataka.

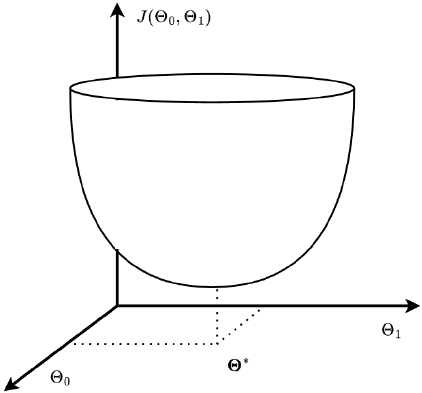
**Dva glavna tipa problema**:

* Klasifikacija
  + Uključuje dodjeljivanje ulaznih podataka u određene diskretne kategorije.
  + Primjer: spam pošta (binarna klasifikacija), prepoznavanje vrsta cvijeća, ...
* Regresija
  + Uključuje predviđanje kontinuiranih vrijednosti na temelju ulaznih podataka.
  + Primjer: predviđanje cijena nekretnina temeljeno na njezinim karakteristikama,  
    vremenske prognoze temeljeno na povijesnim podacima, ...

1. **Objasnite jednostavnu linearnu regresiju. Objasnite kriterijsku funkciju.**

**Jednostavna linearna regresija** modelira odnos između jedne ulazne veličine i kontinuirane izlazne veličine. Cilj je pronaći najbolju linearnu funkciju koja najbolje približava stvarne vrijednosti zavisne varijable na temelju nezavisnih (ulaznih) varijabli.

**Kriterijska funkcija** brojčano iskazuje koliko je dobar model s određenim parametrima na danom skupu podataka za učenje. Koristi se za procjenu koliko dobro linearna regresija približava stvarne vrijednosti zavisne varijable.



Izgled kriterijske funkcije

Minimum kriterijske funkcije – optimalni parametri modela

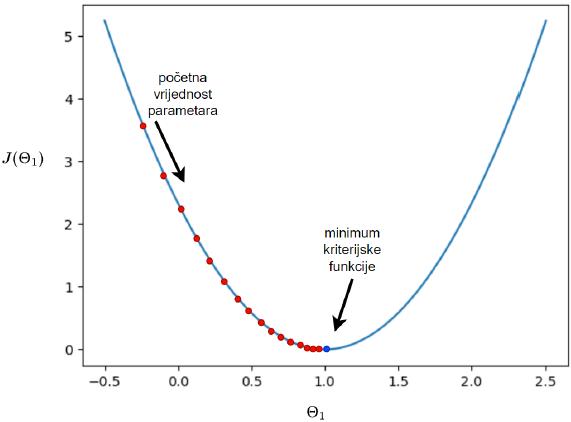
1. **Objasnite načine procjene parametara linearnog regresijskog modela. Navedite prednosti i nedostatke pojedinog pristupa.**

Rješenje u zatvorenoj formi (eng. *closed form solution*)

* Rješenje se dobija izravno, uvrštavanjem brojeva u matematički izraz
* Prednosti:
  + Brz i efikasan pristup
* Nedostaci:
  + Nepraktično za velike skupove podataka

Numerički iterativni postupak minimizacije

* Gradijentni spust
  + Ideja je krenuti od nekih početnih vrijednosti te ih podešavati dok ne dobijemo optimalne paremtre
  + Prednosti:
    - Fleksibilan – može se primijeniti na širok raspon problema optimizacije
    - Može se primijeniti na velike skupove podataka
  + Nedostaci:
    - Može zahtijevati velik broj iteracija
  + Graf:

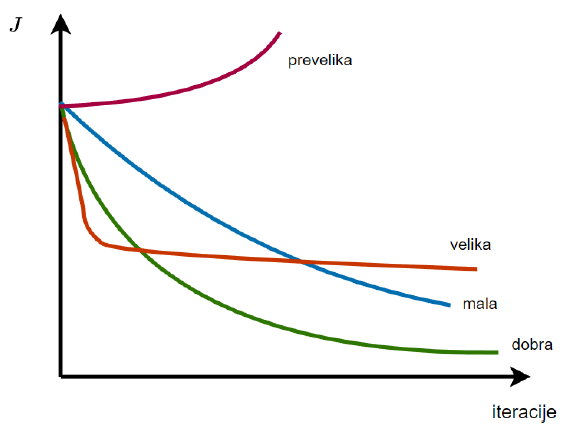


1. **Što je duljina koraka? Kako utječe na minimizaciju kriterijske funkcije? Objasnite na primjeru modela s jednim parametrom.**

**Duljina koraka** (eng. *learning rate*) utječe na proces optimizacije u algoritmima poput gradijentnog spusta. Određuje koliko “velik” korak algoritam treba napraviti u svakom koraku optimizacije pri ažuriranju parametara modela.

**Utječe na minimizaciju kriterijske funkcije** na način:

* Ako je duljina koraka prevelika, algoritam može preskočiti optimalno rješenje
* Ako je duljina koraka premala, algoritam može imati sporo konvergiranje i trebat će više iteracija



**Primjer modela s jednim parametrom**:

Model koji prevdiđa prodaju sladoleda na temelju temperature.

Pretpostavimo da imamo skup podataka o temperaturi (nezavisna (ulazna) vrijabla x) i broju sladoleda (zavisna varijabla y) za nekoliko dana i cilj nam je izraditi linearni regresijski model koji predviđa broj prodanih sladoleda na temelju temperature.

U ovom slučaju, temperatura je naš jedini parametar u modelu. Linearni regresijski model bi izgledao ovako:

y = ϴ0 + ϴ1 \* x

gdje je ϴ0 očekivani broj pdoranih sladoleda kada je temperatura 0, a ϴ1 mjeri koliko se broj prodanih sladoleda mijenja za jedinicu promjene temperature.

Ako model procjeni da je b1 = 3, to znači da očekujemo porast prodanih sladoleda za 3 komada po jednom stupnju temperature.

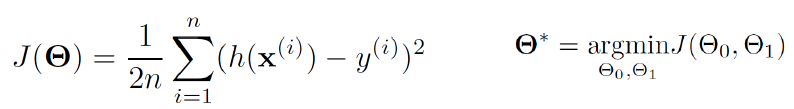
1. **Objasnite višedimenzionalnu linearnu regresiju.**

Ima više ulaznih varijabli. Primjer je određivanje cijene nekretnine gdje bi ulazne varijable bile godina izgradnje, površina, lokacija, broj soba itd...

Postupak procjene može se provesti primjenom metode gradijentnog spusta.

y = ϴ0 + ϴ1x1 + ϴ2x2 + ... + ϴnxn

1. **Objasnite načine procjene parametara višedimenzionalnog regresijskog modela.**

Provodi se na isti način kao i kod linearne regresije s jednom ulaznom veličinom – **minimizacijom kriterijske funkcije**.

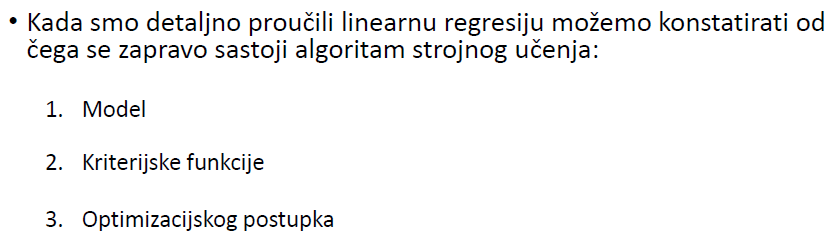
Formula:

1. **Objasnite razliku između batch metode gradijentnog spusta i stohastičke metode gradijentnog spusta. Napišite algoritam podešavanja parametara linearnog regresijskog modela stohastičkom gradijentnom metodom. Što je mini-batch stohastička metoda gradijentnog spusta?**

Razlika između batch metode gradijentnog spusta i stohastičke metode gradijentnog spusta leži u načinu kako se ažuriraju parametri modela i koriste podaci za izračunavanje gradijenta.

U batch metodi koristimo cijeli skup podataka (batch) kako bismo izračunali gradijent. Nakon izračuna gradijenta, parametri se ažuriraju koristeći formulu ažuriranja s dobivenim gradijentom.

U stohastičkoj metodi koristimo samo jedan primjer iz skupa podataka kako bismo izračunali gradijent. To se ponavlja za svaki primjer u skupu podataka ili za određeni broj iteracija.



**Mini-batch stohastička metoda gradijentnog spusta** – hibrid između batch i stohastičke metode. Umjesto da koristimo jedan ili sve podatke, koristimo podskup (mini-batch) određene veličine za izračunavanje gradijenta i ažuriranje parametara.

**Epoha** – jedan prolazak kroz sve podatke za učenje