Vežbe br. 7

MAŠINSKO UČENJE

Šta je mašinsko učenje?

- Podoblast veštačke inteligencije koja se bavi kreiranjem računarskih sistema koji su sposobni da uče na osnovu iskustva
- „Za kompjuterski program se kaže da uči iz iskustva E, vezanog za zadatak T i meru performansi P, ako se njegove performanse na zadatku T, merene metrikom P, unapređuju sa iskustvom E“

Podela mašinskog učenja

NADGLEDANO UČENJE	NENADGLEDANO UČENJE	POLUNADGLEDANO UČENJE	UČENJE SA PODSTICAJEM
• Kreiranje modela na osnovu skupa podataka koji se sastoji i od ulaza i od željenih izlaza	• Pronalaženje struktura, grupa, klastera i sl. u skupu podataka koji ne sadrži obeležene izlaze	• Predstavlja vid kombinacije prethodne dve metode • Definiše se očekivani izlaz za mali deo podataka, pa se vrši „pseudo-labeliranje“ preostalih podataka	• Obučavanje „softverskih agenata“ da izvršavaju određeni zadatak po principu nagrađivanja i kažnjavanja

Nadgledano učenje

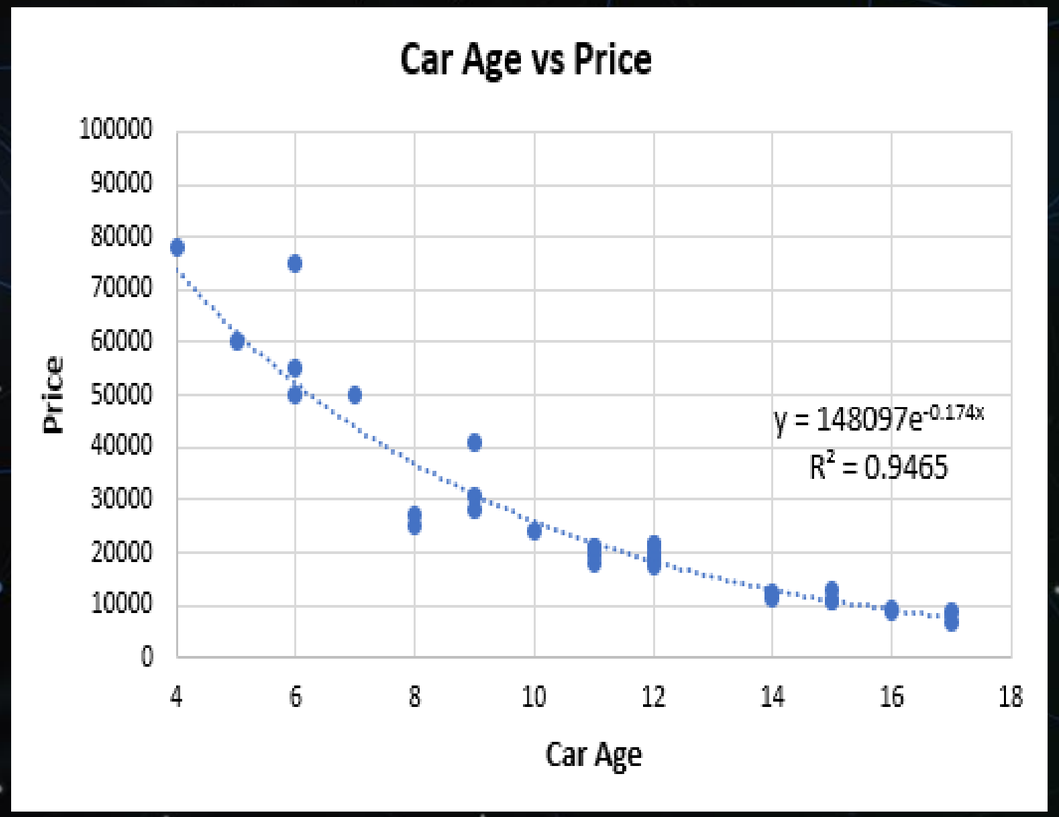
- Kreiranje modela na osnovu skupa podataka koji se sastoji i od ulaza i od željenih izlaza
- Cilj algoritama nadgledanog učenja: „naučiti“ kako formirati izlaz u zavisnosti od ulaza i primeniti naučeno na novim, nepoznatim podacima
- Podela algoritama nadgledanog učenja:
  - Algoritmi za klasifikaciju
  - Algoritmi za regresiju

Regresija

- Predstavlja opisivanje veze između dobijenih ulaza i očekivanog izlaza
- Veza se najčešće predstavlja matematičkom jednačinom
- Kao povratna vrednost regresije dobija se broj koji predstavlja konačni izlaz
- Neki algoritmi za rešavanja regresionih problema:
  - Linearna regresija
  - Ridge regresija
  - Lasso regresija
  - Support Vector Machines (SVM)

Regresija

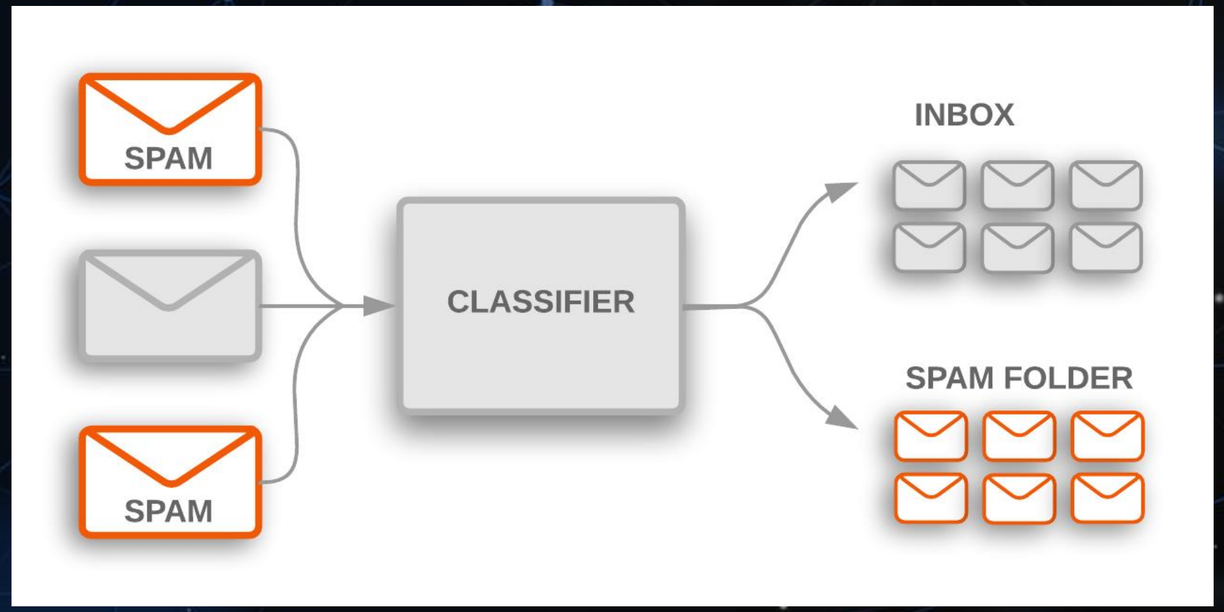

Salary vs Experience
Salary
120000
100000
80000
60000
40000
2
4
6
8
10
Years of Experience


Car Age vs Price
Price
100000
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
4
6
8
10
12
14
16
18
Car Age
y = 148097e-0.174x
R2 = 0.9465

Klasifikacija

- Predstavlja razvrstavanje podataka u različite kategorije na osnovu njihovih obeležja
- Model koji se kreira sa ciljem da vrši klasifikaciju zove se klasifikator
- Neki algoritmi za rešavanje klasifikacionih problema:
  - Logistička regresija (??? ☺)
  - SVM (??? ☺)
  - Stablo odluke (*Decision tree*)
  - Naive Bayes
  - K-Najbližih suseda (*K-Nearest Neighbors*)

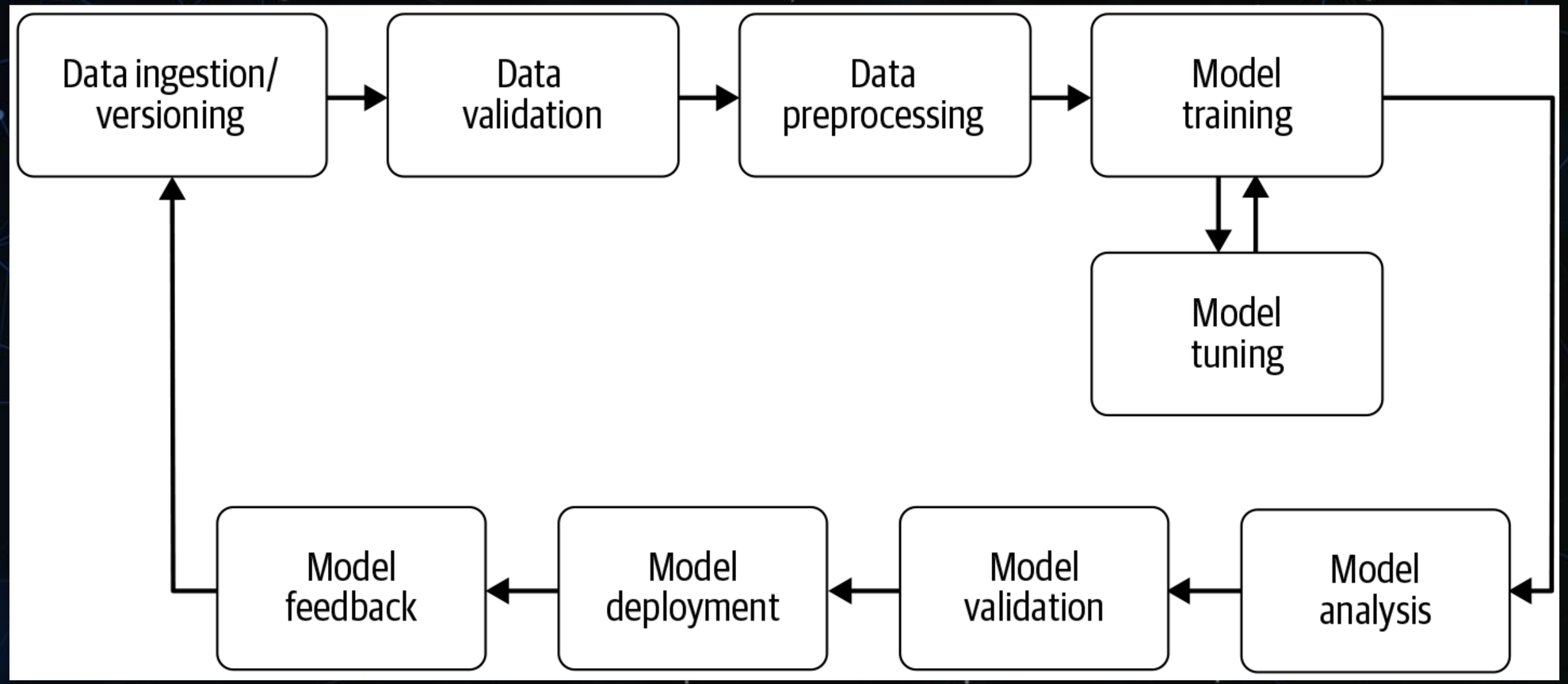
Klasifikacija



Projekat mašinskog učenja

- Faze od kojih se sastoji projekat koji se bavi mašinskim učenjem su:
  - Prikupljanje i priprema podataka - *Data Wrangling*
  - Eksplorativna analiza podataka
  - Odabir algoritma pogodnog za rešavanje datog problema
  - Obučavanje i testiranje modela zasnovanog na odabranom algoritmu
  - Analiza dobijenih rezultata

Projekat mašinskog učenja



Prikupljanje i priprema podataka

- *„Data Wrangling“* obuhvata i do 80% procesa analize podataka
- Cilj je prikupiti i pripremiti podatke, tako da oni budu u obliku pogodnom za prosleđivanje modelu
  - Nedostajuće vrednosti
  - Anomalije
  - Razumevanje postojećih podataka (npr. skraćenica USA)
  - Normalizacija podataka
  - Encoding podataka
  - Balansiranje skupa podataka
- Primeniti u izradi predmetnog projekta ☺

Eksploorativna analiza podataka

- Značajna za bolje razumevanje podataka (samim tim i problema)
- Cilj je ustanoviti da li postoje neki šabloni koji se javljaju u skupu, da li postoje neke zavisnosti između različitih atributa i da li se atributi mogu nekako „povezati“ u cilju uprošćavanja skupa podataka
  - Vizualizacija podataka
  - Prikaz raspodele (varijansa, medijana, srednja vrednost...)
  - Pronalaženje (eventualne) korelacije
  - Redukcija dimenzionalnosti
- Primeniti u izradi predmetnog projekta ☺

Odabir pogodnog algoritma

- Zavisi od:
  - Performansi algoritma
  - Kompleksnosti algoritma
  - Veličine skupa podataka
  - Dimenzionalnosti skupa podataka
  - .....

- Često se problem odabira pogodnog algoritma rešava tako što se kreiraju „bazični“ modeli, proveri njihova uspešnost, pa potom radi sa onim koji od starta daje povoljne rezultate

Obučavanje i testiranje modela

- Kako bi se moglo vršiti obučavanje i testiranje modela, potrebno je podeliti početni skup podataka na **obučavajući** i **testni** skup
- Podela se najčešće vrši u odnosu 70/30 ili 80/20
- Voditi računa o tome kako se vrši podela ☺

Obučavanje i testiranje modela

3.1. Training

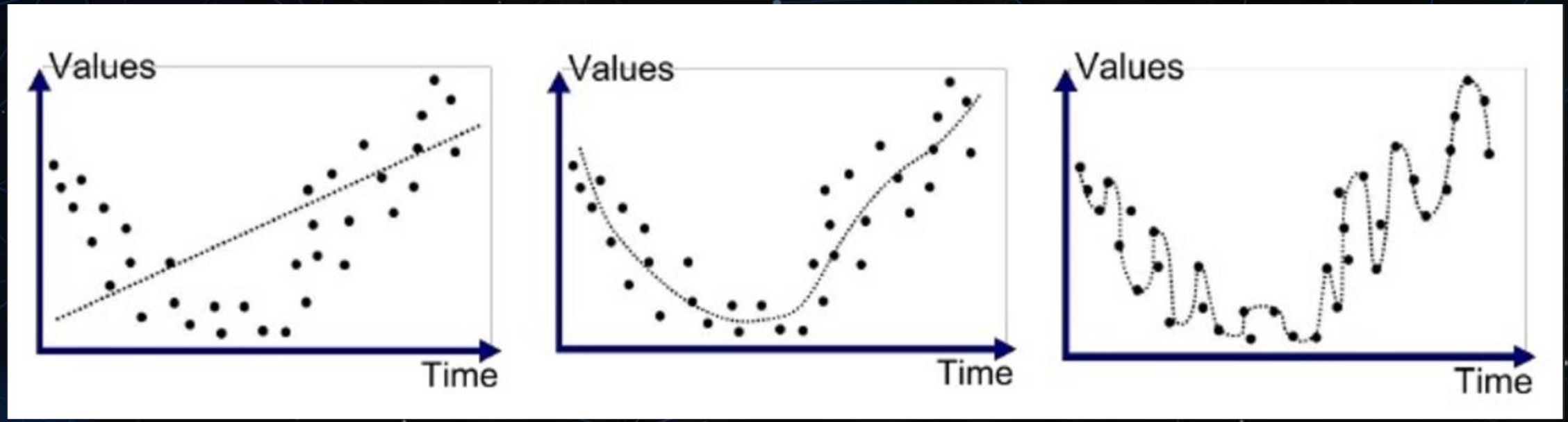
We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples for the pneumonia detection task. We randomly split the entire dataset into 80% training, and 20% validation.

Obučavanje i testiranje modela

3.1. Training

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples. For the pneumonia detection task, we randomly split the dataset into training (28744 patients, 98637 images), validation (1672 patients, 6351 images), and test (389 patients, 420 images). There is no patient overlap between the sets.

Obučavanje i testiranje modela



Obučavanje i testiranje modela

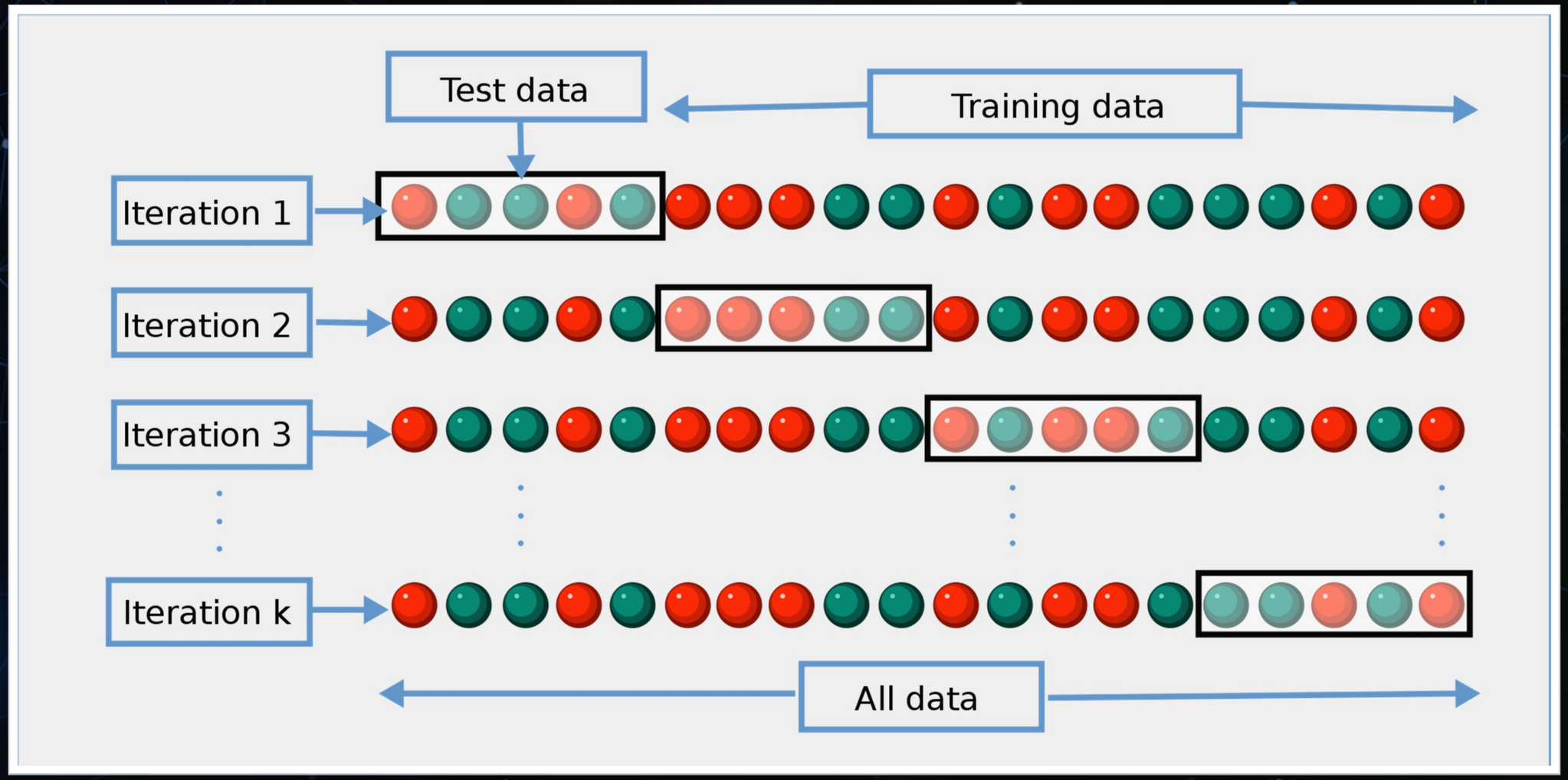
- ***Underfitting*** - model se previše slabo prilagođava obučavajućem skupu podataka (ne uspeva da zaključi kako formirati izlaz)
- Rešenje problema:
  - Promeniti model (odabrati drugi algoritam)
  - Proširiti skup podataka

- ***Overfitting*** – model se previše dobro prilagođava obučavajućem skupu podataka, a na testnom skupu daje jako loše rezultate
- Rešenje problema:
  - Proširiti skup podataka (najprostije rešenje)
  - Regularizacija
  - Učenje u ansamblu

Unakrsna validacija

- Unakrsna validacija (***cross validation***) predstavlja tehniku kojom se detektuje da li postoji overfitting
- Ideja: izdvojiti deo obučavajućeg skupa podataka u novi, validacioni skup podataka, pa posmatrati kako se model ponaša kada mu se prosledi validacioni skup
- Postupak se najčešće primenjuje u više faza
- Primeniti u izradi predmetnog projekta ☺

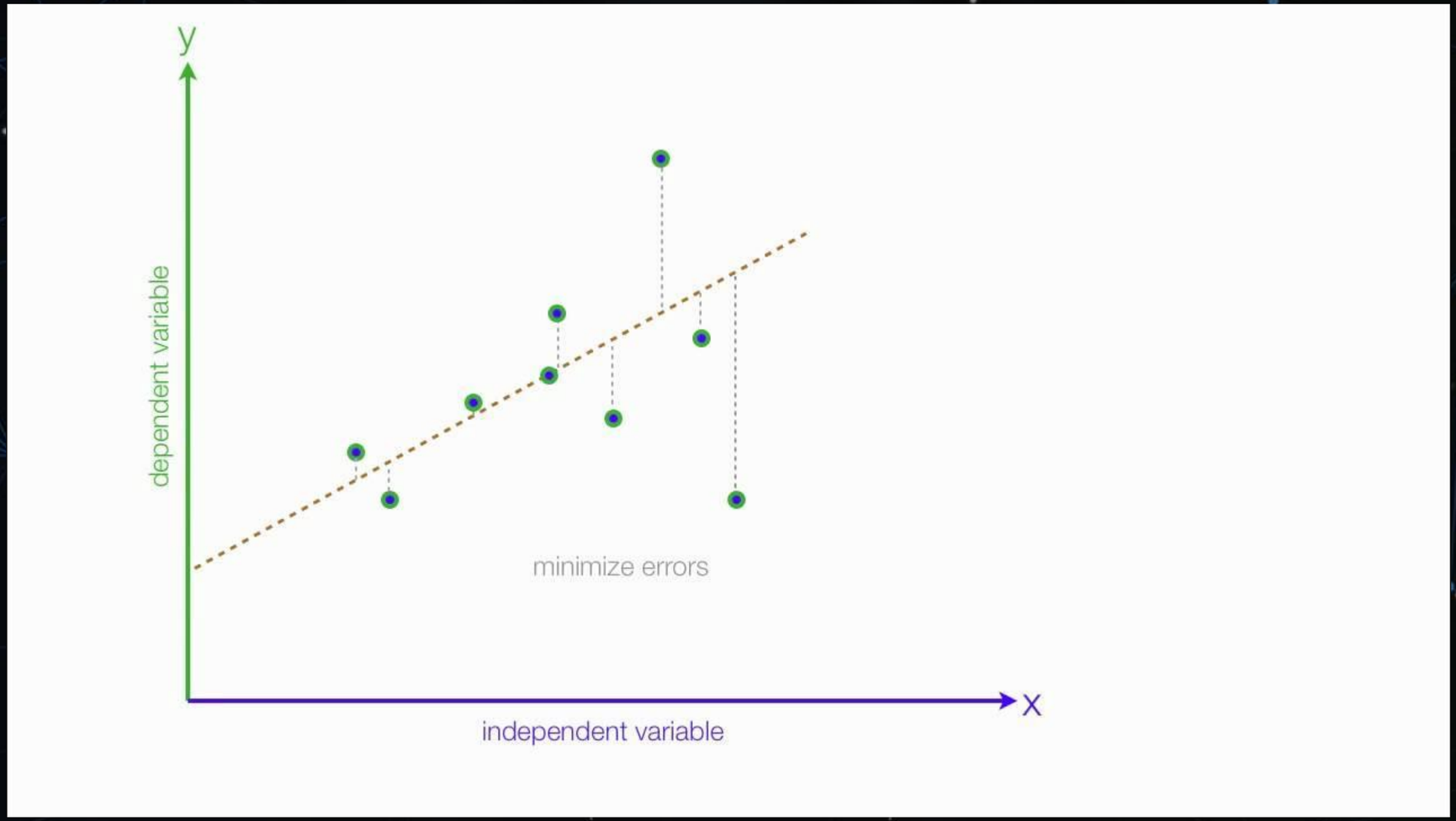
Unakrsna validacija



Linearna regresija

- Modelovanje relacije između izlaza i ulaza, na način da izlaz **linearno** zavisi od nepoznatih parametara
- Relativno lako kada imamo jedan nepoznati parametar, ali šta kada ih je mnogo?
- Cilj linearne regresije: dobiti formulu oblika $y = a*x + b + \varepsilon$
- Kako odrediti *a* i *b*?

Linearna regresija



Metoda najmanjih kvadrata

- Metoda koja se koristi za određivanje koeficijenata u linearnoj regresiji
- Uglavnom ne postoji prava koja će proći kroz sve tačke, zato treba napraviti onu koja će da „najbolje“ prolazi kroz njih
- Metoda se zasniva na načelu da su najbolji parametri *a* i *b* oni za koje važi da će suma kvadrata razlika između očekivanih i izmerenih vrednosti biti najmanja moguća

$$\mathrm{RSS} = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

$$a = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$b = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$$

Linearna regresija

- Prednosti:
  - Najjednostavniji mogući model
  - Model jednostavan za interpretaciju
  - Brzo računanje čak i za velike skupove podataka

- Mane:
  - Uglavnom previše jednostavna
  - Podložna overfitting-u
  - Osetljiva na anomalije

Linearna regresija

```
import math
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression


dataset = pd.read_csv('Salary_Data.csv')
print(dataset.head())

X = dataset["YearsExperience"].values.reshape(-1, 1)
y = dataset["Salary"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)

regressor = LinearRegression()
regressor.fit(X_train, y_train)
y_pred = regressor.predict(X_test)
print("\nProsecna greska: ", "%.2f" % math.sqrt(mean_squared_error(y_test, y_pred)))

plt.scatter(X_test, y_test, color='red')
plt.plot(X_train, regressor.predict(X_train), color='blue')
plt.title("Plata vs Godine iskustva")
plt.xlabel("Godine iskustva")
plt.ylabel("Plata")
plt.show()
```