# lab 4

# linear regression using lasso and ridge regularization

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.datasets import fetch\_california\_housing

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression,Lasso,Ridge

from sklearn.metrics import r2\_score,mean\_squared\_error

#loading dataset

housing=fetch\_california\_housing()

#convert dataset into datafrfame

df=pd.DataFrame(housing.data,columns=housing.feature\_names)

df['target']=housing.target

#feature and target

X=df.drop('target',axis=1)

y=df['target']

#TRAIN TEST SPLIT

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,test\_size=0.2,random\_state=42)

#USING linear Regression(NO regularization)

lr=LinearRegression()

lr.fit(X\_train,y\_train)

lr\_pred=lr.predict(X\_test)   #y predict gareko ho esma

op:

R2 score: 0.5757877060324508

MSE: 0.5558915986952444

# metrics using linear regression ,also print (w/o regularization)

print('R2 score:',r2\_score(y\_test,lr\_pred))

print('MSE:',mean\_squared\_error(y\_test,lr\_pred))

# using lasso regularization

lasso=Lasso(alpha=1)

lasso.fit(X\_train,y\_train)

lasso\_pred=lasso.predict(X\_test)

#metrics using linear regression

print('R2 score:',r2\_score(y\_test,lasso\_pred))

print('MSE:',mean\_squared\_error(y\_test,lasso\_pred))

op:

R2 score: 0.2841671821008396

MSE: 0.9380337514945427

# using ridge regularization

ridge=Ridge(alpha=1)

ridge.fit(X\_train,y\_train)

ridge\_pred=ridge.predict(X\_test)

#metrics using linear regression

print('R2 score:',r2\_score(y\_test,ridge\_pred))

print('MSE:',mean\_squared\_error(y\_test,ridge\_pred))

op:

R2 score: 0.5758549611440126

MSE: 0.5558034669932211