#importing library

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

#importing dataset

dataset=pd.read\_csv("Position\_Salaries.csv")

x=dataset.iloc[:,1:2].values

y=dataset.iloc[:,2].values

'''#splitting into test and training dataset

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

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.2,random\_state=0)

#feature scaling

from sklearn.preprocessing import StandardScaler

sc\_x=StandardScaler()

x\_train=sc\_x.fit\_transform(x\_train)

x\_test=sc\_x.transform(x\_test)'''

#linear regression

from sklearn.linear\_model import LinearRegression

lin\_reg=LinearRegression()

lin\_reg.fit(x,y)

#polynomial regression

from sklearn.preprocessing import PolynomialFeatures

poly\_reg=PolynomialFeatures(degree=5)

x\_poly=poly\_reg.fit\_transform(x)

lin\_reg2=LinearRegression()

lin\_reg2.fit(x\_poly,y)

#visualsing the linear reg

plt.scatter(x,y,color='red')

plt.plot(x,lin\_reg.predict(x),color='blue')

plt.title('truth or bluff(linear)')

plt.xlabel('position level')

plt.ylabel('Salary')

plt.show()

#visualising poly reg

plt.scatter(x,y,color='red')

plt.plot(x,lin\_reg2.predict(poly\_reg.fit\_transform(x)),color='blue')

plt.title('truth or bluff(poly degree=5)')

plt.xlabel('position level')

plt.ylabel('Salary')

#predicting the results by lin reg

y\_predl=lin\_reg.predict(x)

#predicting the results by multiple reg

y\_predpoly=lin\_reg2.predict(poly\_reg.fit\_transform(x))