**Implementation Of Linear And Polynomial Regression In Python**

CODE:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset = pd.read\_csv('C:/Users/prith/Desktop/MACHINE LEARNING/Position\_Salaries.csv')

X = dataset.iloc[:, 1:-1].values

y = dataset.iloc[:, -1].values

"""

Training the Linear Regression model on the Whole dataset

A Linear regression algorithm is used to create a model.

A LinearRegression function is imported from sklearn.linear\_model library.

"""

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(X, y)

#Linear Regression classifier model

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None)

"""

Training the Polynomial Regression model on the Whole dataset

A polynomial regression algorithm is used to create a model.

"""

from sklearn.preprocessing import PolynomialFeatures

poly\_reg = PolynomialFeatures(degree = 4)

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

lin\_reg\_2 = LinearRegression()

lin\_reg\_2.fit(X\_poly, y)

#Polynomial Regression classifier model

LinearRegression(copy\_X=True, fit\_intercept=True, n\_jobs=None)

"""

Visualising the Linear Regression results

Here scatter plot is used to visualize the results. The title of the plot is set to Truth or Bluff

(Linear Regression), xlabel is set to Position Level , and ylabel is set to Salary.

"""

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

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

plt.title('Truth or Bluff (Linear Regression)')

plt.xlabel('Position Level')

plt.ylabel('Salary')

plt.show()

#Visualising the Polynomial Regression results

"""

The title of the plot is set to Truth or Bluff (Polynomial Regression), xlabel is set to Position level,

and ylabel is set to Salary.

"""

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

plt.plot(X, lin\_reg\_2.predict(poly\_reg.fit\_transform(X)), color = 'blue')

plt.title('Truth or Bluff (Polynomial Regression)')

plt.xlabel('Position level')

plt.ylabel('Salary')

plt.show()

OUTPUT:

