Linear & Polynomial Regression

Code:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('https://s3.us-west-2.amazonaws.com/public.gamelab.fun/dataset/position\_salaries.csv')

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

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

# Splitting the dataset into the Training set and Test set

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)

# Fitting Linear Regression to the dataset

from sklearn.linear\_model import LinearRegression

lin\_reg = LinearRegression()

lin\_reg.fit(X, y)

# Visualizing the Linear Regression results

def viz\_linear():

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()

return

viz\_linear()

# Fitting Polynomial Regression to the dataset

from sklearn.preprocessing import PolynomialFeatures

# Visualizing the Polymonial Regression results

def viz\_polymonial(deg):

poly\_reg = PolynomialFeatures(degree=deg)

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

pol\_reg = LinearRegression()

pol\_reg.fit(X\_poly, y)

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

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

plt.title('Truth or Bluff (Polynomial Regression deg {deg})'.format(deg=deg))

plt.xlabel('Position level')

plt.ylabel('Salary')

plt.show()

return

viz\_polymonial(2)

viz\_polymonial(3)

viz\_polymonial(4)

viz\_polymonial(5)

Plots:









