|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | """ |
|  |  |
|  | import numpy as np |
|  | import matplotlib.pyplot as plt |
|  | import pandas as pd |
|  |  |
|  | # Importing the dataset |
|  | dataset = pd.read\_csv('./Sample Data/PART 2. REGRESSION - Polynomial Regression - Polynomial\_Regression/Polynomial\_Regression/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) |
|  |  |
|  | """ |
|  | # Scaling |
|  | from sklearn.preprocessing import StandardScaler |
|  | sc\_X = StandardScaler() |
|  | X\_train = sc\_X.fit\_transform(X\_train) |
|  | X\_test = sc\_X.transform(X\_test) |
|  | """ |
|  |  |
|  | # 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 |
|  | poly\_reg = PolynomialFeatures(degree=4) |
|  | X\_poly = poly\_reg.fit\_transform(X) |
|  | pol\_reg = LinearRegression() |
|  | pol\_reg.fit(X\_poly, y) |
|  |  |
|  | # Visualizing the Polymonial Regression results |
|  | def viz\_polymonial(): |
|  | plt.scatter(X, y, color='red') |
|  | plt.plot(X, pol\_reg.predict(poly\_reg.fit\_transform(X)), color='blue') |
|  | plt.title('Truth or Bluff (Linear Regression)') |
|  | plt.xlabel('Position level') |
|  | plt.ylabel('Salary') |
|  | plt.show() |
|  | return |
|  | viz\_polymonial() |
|  |  |
|  | # Additional feature |
|  | # Making the plot line (Blue one) more smooth |
|  | def viz\_polymonial\_smooth(): |
|  | X\_grid = np.arange(min(X), max(X), 0.1) |
|  | X\_grid = X\_grid.reshape(len(X\_grid), 1) #Why do we need to reshape? (https://www.tutorialspoint.com/numpy/numpy\_reshape.htm) |
|  | # Visualizing the Polymonial Regression results |
|  | plt.scatter(X, y, color='red') |
|  | plt.plot(X\_grid, pol\_reg.predict(poly\_reg.fit\_transform(X\_grid)), color='blue') |
|  | plt.title('Truth or Bluff (Linear Regression)') |
|  | plt.xlabel('Position level') |
|  | plt.ylabel('Salary') |
|  | plt.show() |
|  | return |
|  | viz\_polymonial\_smooth() |
|  |  |
|  | # Predicting a new result with Linear Regression |
|  | lin\_reg.predict([[5.5]]) |
|  | #output should be 249500 |
|  |  |
|  | # Predicting a new result with Polymonial Regression |
|  | pol\_reg.predict(poly\_reg.fit\_transform([[5.5]])) |
|  | #output should be 132148.43750003 |