

## 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:





