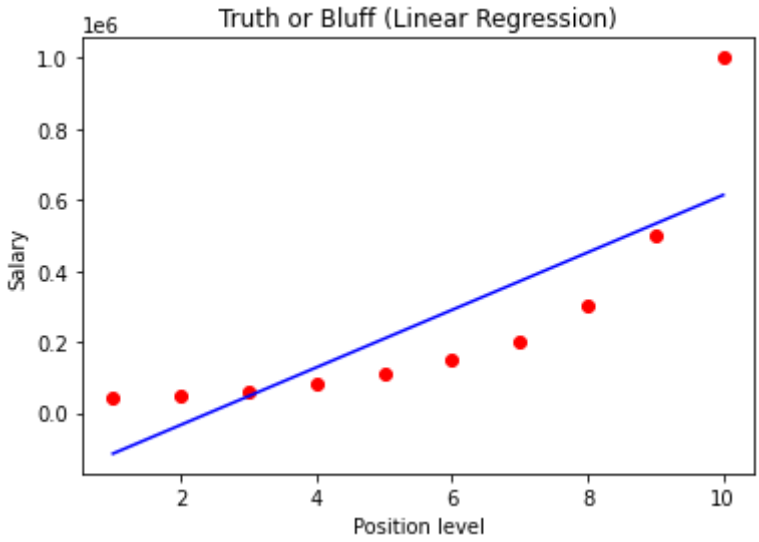
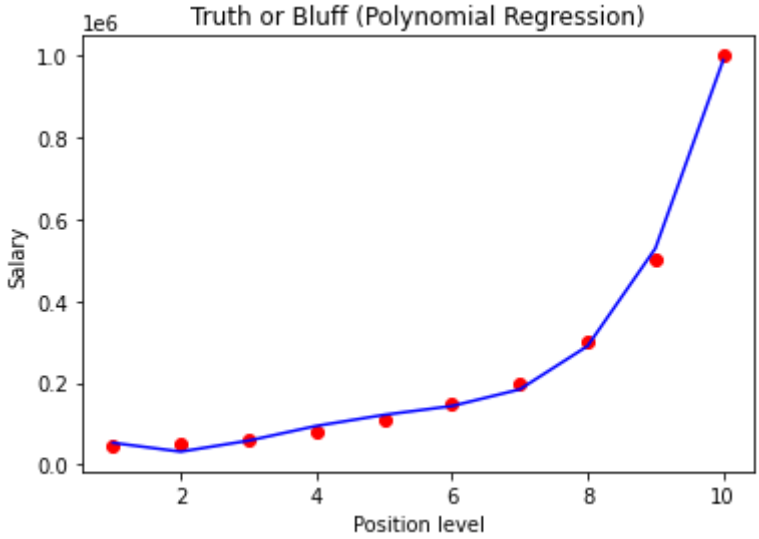
```
# Polynomial Regression
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd


# Importing the dataset
dataset = pd.read_csv('/content/drive/MyDrive/DataSet/Position_Salaries.csv')


#only feature is Level as Position is redundant (just a label for Level)
#so we only import Level as feature
#(1:2 misses indexes 0 and 2 and is preferable to just putting index 1 as 1:2 makes X a ma
X = dataset.iloc[:, 1:2].values
#dependent variable is a vector
y = dataset.iloc[:, 2].values

# Not Splitting the dataset into the Training set and Test set as data set too small
"""from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state =

# No need for Feature Scaling as library does it
"""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 for comparison purposes
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)

# Fitting Polynomial Regression to the dataset
# This is just linear regression with PolynomialFeatures
from sklearn.preprocessing import PolynomialFeatures
#set the maximum degree to use in the model
poly_reg = PolynomialFeatures(degree = 4)
#transform the feature matrix to introduce the derived polynomial features (each is Level
X_poly = poly_reg.fit_transform(X)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```

```
LinearRegression()
```

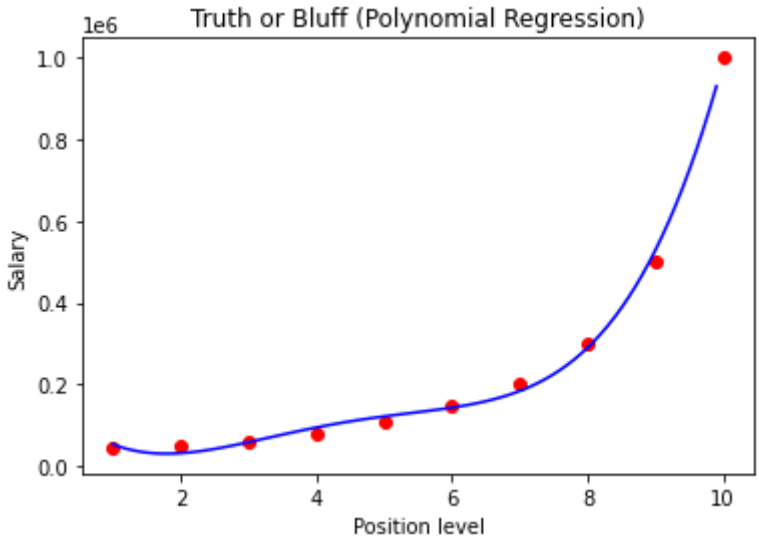
```
# Visualising the Linear Regression results
plt.scatter(X, y, color = 'red')
plt.plot(X, lin_reg.predict(X), color = 'blue')
plt.title('Truth or Bluff (Linear Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```




```
# Visualising the Polynomial Regression results
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()
```




```
# Visualising the Polynomial Regression results (for higher resolution and smoother curve)
X_grid = np.arange(min(X), max(X), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



✓ 0s completed at 8:07 PM