PolynomialRegression_ML_06

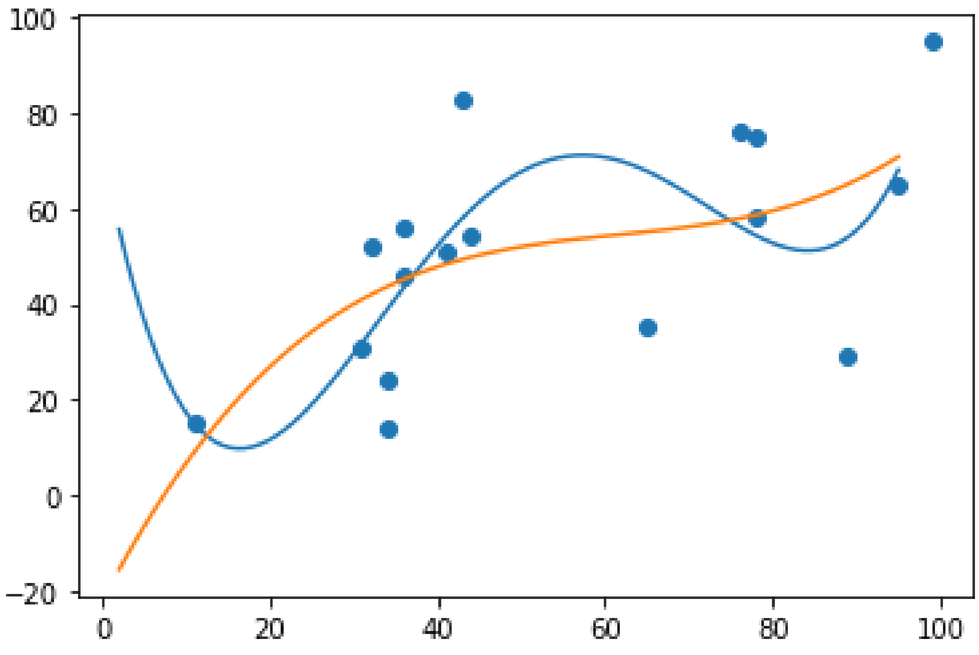
March 15, 2022

0.1 Polynimoal Regression Model for Machine Learning

- Linear pattern with curve
-

[ ]:

```
import numpy as np
import matplotlib.pyplot as plt
x = [89,43,36,36,95,34,34,32,65,76,78,78,99,31,41,44,11]
y = [29,83,46,56,65,14,24,52,35,76,58,75,95,31,51,54,15]
mymodel = np.poly1d(np.polyfit(x,y,4))
mymodel1 = np.poly1d(np.polyfit(x,y,3))
myline = np.linspace(2,95,100)

plt.scatter(x,y)
plt.plot(myline,mymodel(myline))
plt.plot(myline,mymodel1(myline))
plt.show()
```



[ ]:

```python
# R2 Squared  for bad fit
from sklearn.metrics import r2_score
print(r2_score(y,mymodel(x)))
```

```
0.4892680794637757
```

[ ]:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Importing the dataset
dataset = pd.read_csv('position_salaries.csv')
X = dataset.iloc[:, 1:2].values
y = dataset.iloc[:, 2].values
```

[ ]:

```python
dataset.head()
```

[ ]:

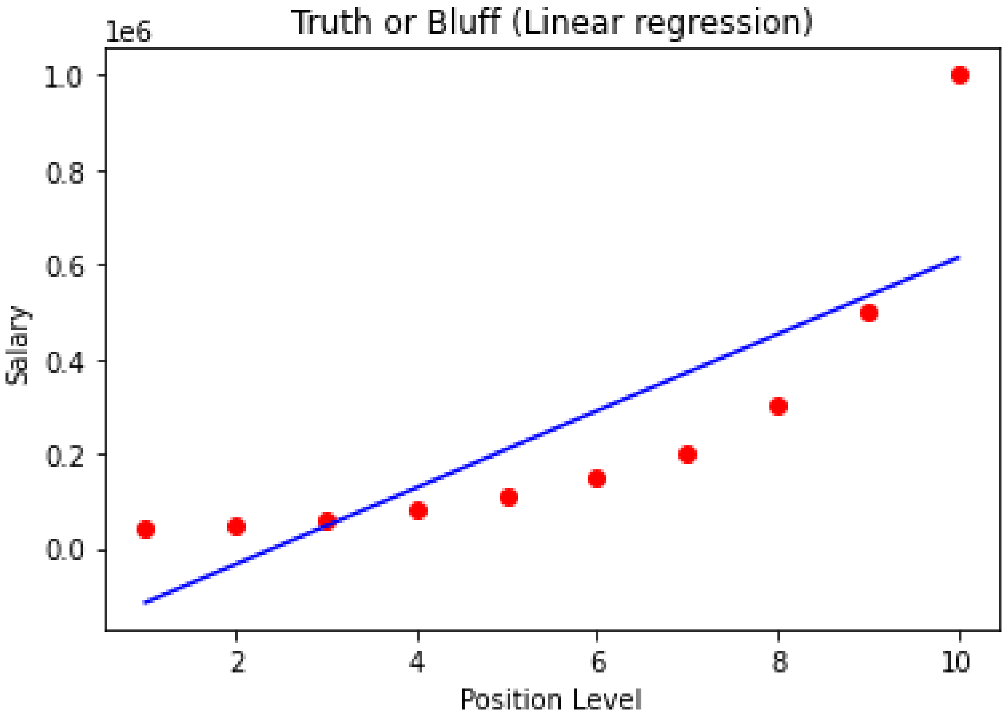
```
            Position  Level  Salary
0   Business Analyst      1   45000
1  Junior Consultant      2   50000
2  Senior Consultant      3   60000
3            Manager      4   80000
4    Country Manager      5  110000
```

[ ]:

```python
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)
```

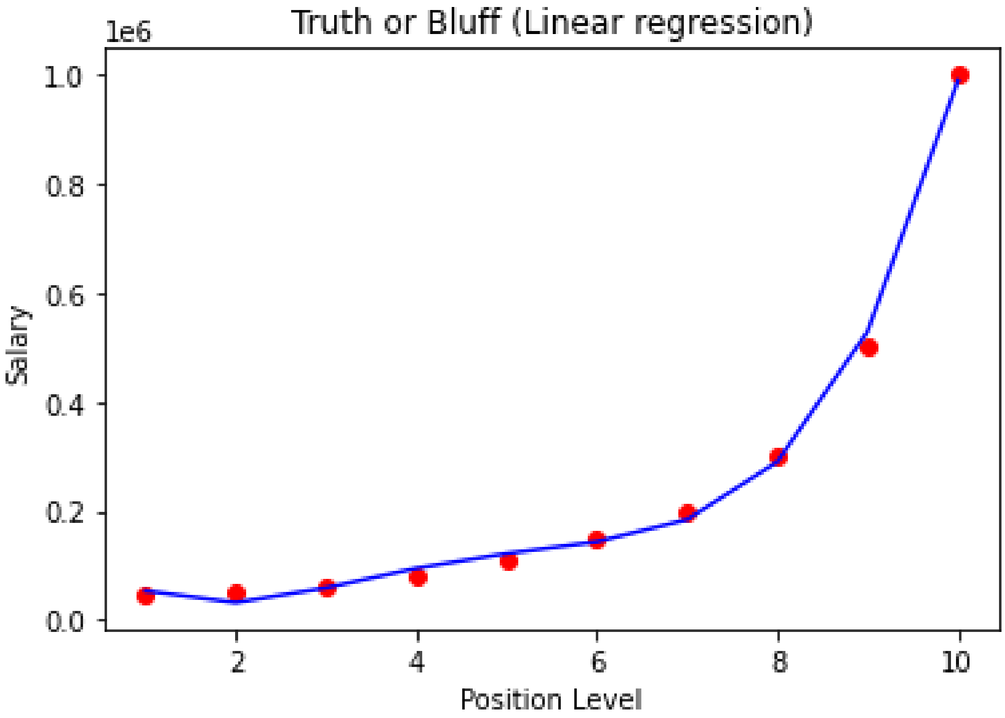
[ ]:

```python
from sklearn.linear_model import LinearRegression
lin_reg = LinearRegression()
lin_reg.fit(X, y)
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()

viz_linear()
```



[ ]:

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(degree=4)
Xp = poly.fit_transform(X)
pol = LinearRegression()
pol.fit(Xp,y)
def viz_linear():
    plt.scatter(X,y, color='red')
    plt.plot(X,pol.predict(Xp), color='blue')
    plt.title("Truth or Bluff (Linear regression)")
    plt.xlabel("Position Level")
    plt.ylabel("Salary")
    plt.show()

viz_linear()
```



```
[ ]: # Predicting a new result with linear regression
     pridLin = lin_reg.predict([[11]])
     pridLin
```

```
[ ]: array([694333.33333333])
```

```
[ ]: # Predicting with polynomial
     pridPol = pol.predict(poly.fit_transform([[11]]))
     pridPol
```

```
[ ]: array([1780833.33333359])
```

```
[ ]: # Difference between LR and PLR
     diff =pridPol - pridLin
     diff
```

```
[ ]: array([1086500.00000025])
```