Big Basket Ads in App iOS

Date : 18 July 20233

Project Start Date - End Date	● Start Date – 18 -07 -2023 ● End Date – 18 -07 2023
Objectives	● To analyze event revenue to find the next day's expected event revenue ● General exploratory analysis ● Polynomial Regression and Linear Regression
Milestones accomplished the week of Start Date - End Date:	● Exploratory analysis ● Polynomial Regression and Linear Regression ● Classification of data with respect to term

Contact Information

This project is performed for educational purpose of under the guidance of Siddhivinayak Sir .

Project Manager
Name : Siddhivinayak Phulwadkar
Mobile: 9028965955
Email:
siddhivinayakphulwadkar@gmail.com

Student Name
Name : Megalarasan
Mobile:9600802073
Email: smarasuoct2003@gmail.com

Project Abstract

The dataset is about showing advertisements to App iOS users for product purchases. Our target was to predict the next day's expected event revenue from the given event revenue. We have to apply either Simple Linear Regression or Polynomial Regression based on the accuracy of the datasets. For this dataset, we have applied both Polynomial Regression and Linear Regression and performed exploratory analysis

Big Basket Ads in App iOS

Importing the libraries

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

Importing the dataset

```
dataset = pd.read_csv('C:/Users/USER/Documents/BIG BASKET DATA/18 july in app ios.csv')
dataset.shape
```

```
(44, 57)
```

```
dataset
```

	Sr no.	Attributed Touch Type	Event Name	Event Value	Event Revenue	Event Revenue Currency	Event Revenue USD	Cost Model	Cost Value	Cost Currency	...	Is Retargeting	Retargeting Conversion Type	Is Prim Attribut
0	1	click	placeorder	{"af_content_type":"product","order id":"21217...	775.00	INR	10.298229	NaN	NaN	NaN	...	False	NaN	T
1	2	click	placeorder	{"af_content_type":"product","order id":"21216...	802.55	INR	10.664314	NaN	NaN	NaN	...	False	NaN	T
2	3	click	placeorder	{"af_content_type":"product","order id":"21216...	748.00	INR	9.938152	NaN	NaN	NaN	...	False	NaN	T
3	4	click	placeorder	{"af_content_type":"product","order id":"21215...	1129.00	INR	14.994386	NaN	NaN	NaN	...	False	NaN	T
4	5	click	placeorder	{"af_content_type":"product","order id":"21214...	2235.00	INR	29.689273	NaN	NaN	NaN	...	False	NaN	T

```
# NA values are not suitable for regression analysis. So, we replace with zero
```

```
dataset1= dataset.fillna(0)
```

```
X = dataset1.iloc[:,:-56].values
y = dataset1.iloc[:, -53].values
```

```
X
```

```
array([[ 1],
       [ 2],
       [ 3],
       [ 4],
       [ 5],
       [ 6],
       [ 7],
       [ 8],
       [ 9],
       [10],
       [11],
       [12],
       [13],
       [14],
       [15],
       [16],
       [17],
       [18],
       [19],
       [20],
       [21],
       [22],
       [23],
       [24],
       [25],
       [26],
       [27],
       [28],
       [29],
       [30],
       [31],
       [32],
       [33],
       [34],
       [35],
       [36],
       [37],
       [38],
       [39],
       [40],
       [41],
       [42],
       [43],
       [44]], dtype=int64)
```

```
y
```

```
array([ 775.  ,  802.55,  748.  , 1129.  , 2235.  ,  198.  ,  605.2 ,
       1434.7 , 3102.59, 2573.  ,  922.3 ,  935.68,  777.95, 1033.  ,
       1073.54,  287.31,  207.85,  522.5 ,  522.5 ,  114.45,    0.  ,
          0.  , 1332.5 ,    0.  ,    0.  , 1138.11,  220.  ,  708.  ,
       1310.  , 2845.76,  568.  , 2517.69, 1185.1 ,   76.75,  462.39,
       1709.26,  862.96, 1453.  , 1993.  , 1514.08,  834.67,  614.72,
        130.13, 1112.31])
```

```
dataset1
```

	Sr no.	Attributed Touch Type	Event Name	Event Value	Event Revenue	Event Revenue Currency	Event Revenue USD	Cost Model	Cost Value	Cost Currency	...	Is Retargeting	Retargeting Conversion Type	Is Prim Attribut
0	1	click	placeorder	{"af_content_type":"product","order id":"21217...	775.00	INR	10.298229	0.0	0.0	0.0	...	False	0.0	T
1	2	click	placeorder	{"af_content_type":"product","order id":"21216...	802.55	INR	10.664314	0.0	0.0	0.0	...	False	0.0	T
2	3	click	placeorder	{"af_content_type":"product","order id":"21216...	748.00	INR	9.938152	0.0	0.0	0.0	...	False	0.0	T
3	4	click	placeorder	{"af_content_type":"product","order id":"21215...	1129.00	INR	14.994386	0.0	0.0	0.0	...	False	0.0	T
4	5	click	placeorder	{"af_content_type":"product","order id":"21214...	2235.00	INR	29.689273	0.0	0.0	0.0	...	False	0.0	T

Training the Linear Regression model on the whole dataset

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

```
▾ LinearRegression
LinearRegression()
```

Training the Polynomial Regression model on the whole dataset

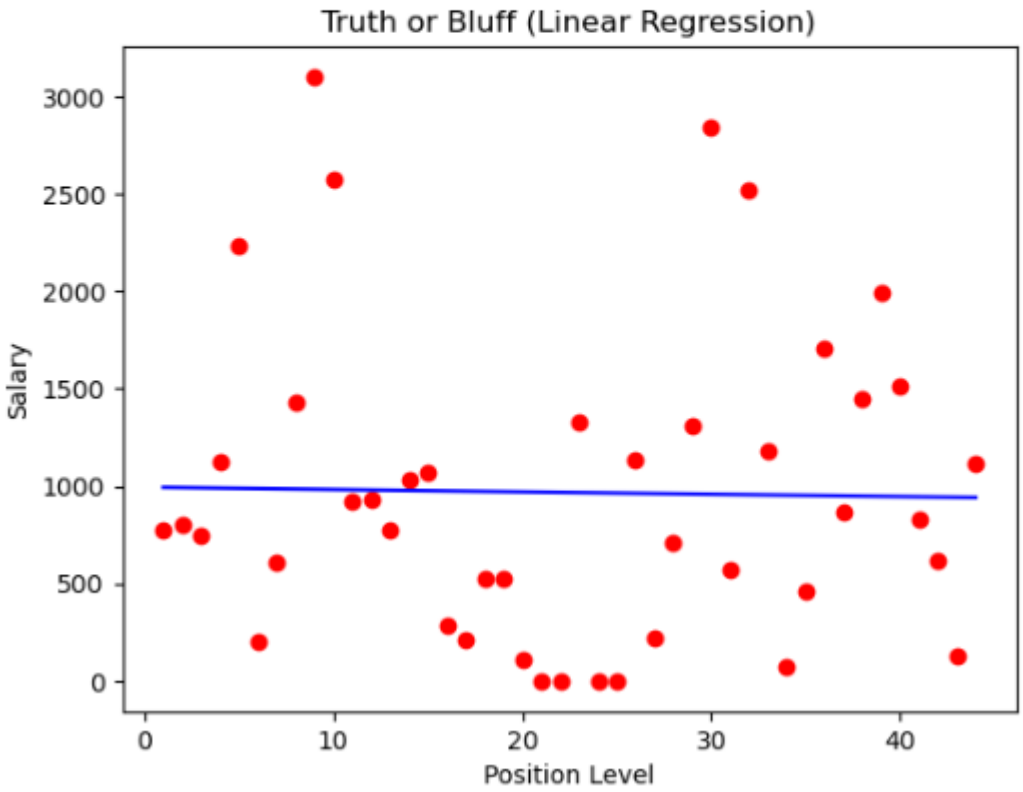
```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 5)
X_poly = poly_reg.fit_transform(X)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```

```
▾ LinearRegression
LinearRegression()
```

Data visualization

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()
```



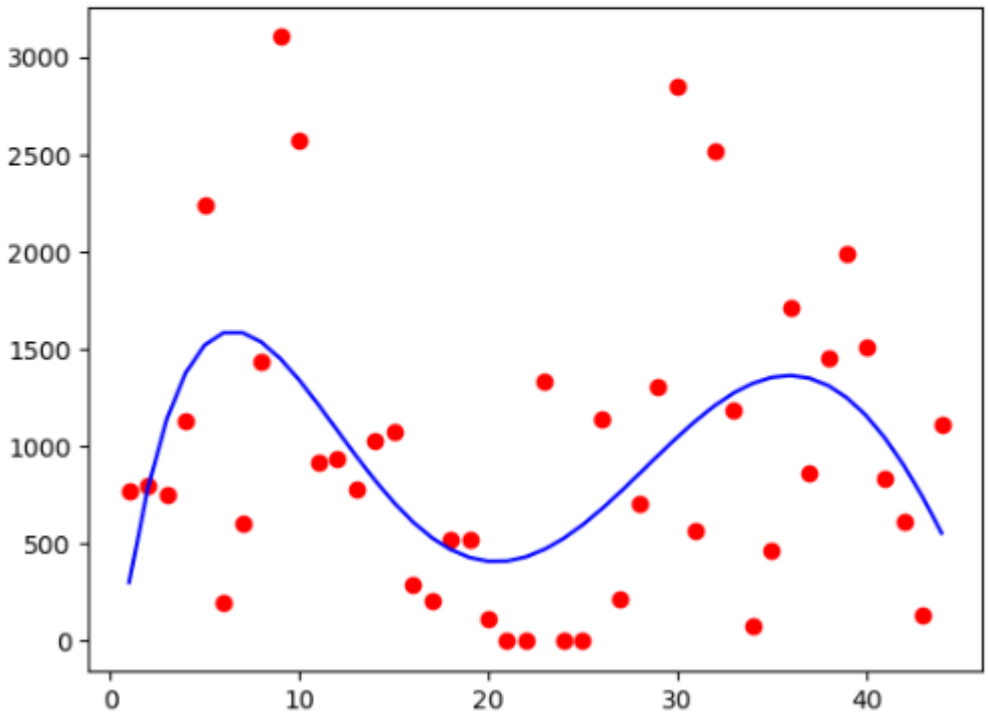
Insights of Linear Regression Graph

- **Some** of the **points** are **closer** to the **Best-fit line**. But **most** of the **points** are **away** from the **Best-fit line.**
- This caused the **error** in the **regression**. As a result of this regression, the **accuracy level** became **very poor** to find the expected value for the next day.
- At this level of accuracy, we **can't find** the **expected revenue** value of the **next day**.
- The **accuracy level** is **low** because of **insufficient data.**

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.show()
```



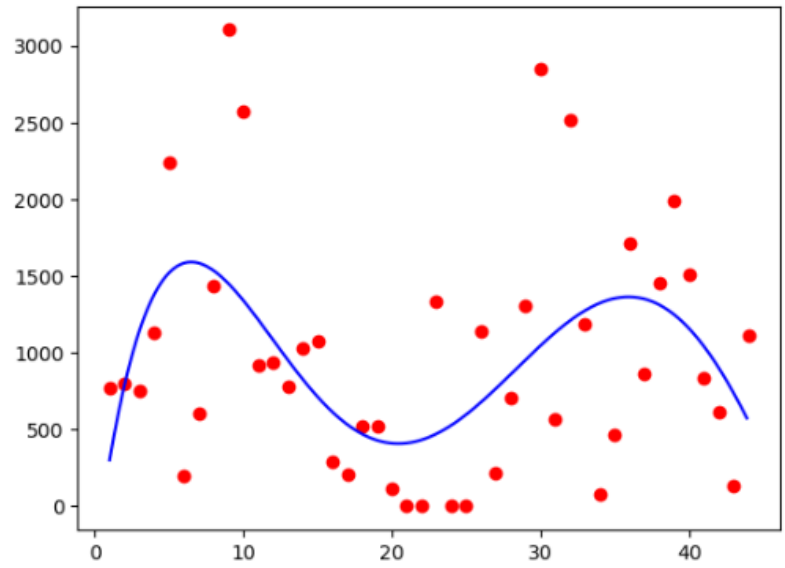
Insights of Polynomial Regression Graph

- **Same** as the **Linear regression**, **most** of the **points** are **away** from the **Best-fit line**.
- As a result of this regression, the **accuracy level** became **very poor** to find the expected value for the next day.
- At this level of accuracy, we **can't find** the **expected revenue** value of the **next day.**
- In this polynomial Regression, all the **possible degrees** are **not feasible** to find the next day's expected revenue values.
- In the **above graph** is **degree 4**. Similarly for all the degree's accuracy level is poor to find the expected values of the next day

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.show()
```



Insights of Polynomial Regression Graph

- After the higher resolution and smoother, the **curve doesn't make** any change in the **accuracy level**.
- We **can't find** the **expected revenue** value of the next day.
- In this polynomial Regression, all the **possible degrees** are **not feasible** to find the next day's expected revenue values.
- The **given data** set is **not suitable** for this r**egression**. It is **better** with the **huge amount** of **data** with **unbiased featured data.**

Predicting a new result with Linear Regression

```
lin_reg.predict([[45]])
```

```
array([940.70139535])
```

```
lin_reg.predict([[46]])
```

```
array([939.49160888])
```

```
lin_reg.predict([[47]])
```

```
array([938.28182241])
```

Predicting a new result with Polynomial Regression

```
lin_reg_2.predict(poly_reg.fit_transform([[45]]))
```

```
array([354.2486861])
```

```
lin_reg_2.predict(poly_reg.fit_transform([[46]]))
```

```
array([142.0285335])
```

```
lin_reg_2.predict(poly_reg.fit_transform([[47]]))
```

```
array([-76.14476272])
```

Conclusion

- Both **the Linear and Polynomial Regression** are **not given** the **best accuracy** for the given dataset.
- In **Linear Regression**, **values** are **highly deviating** from the best-fit line. So, there is **highly error** on the regression part.
- In **Polynomial Regression**, In **revenue**, there is **no negative value**. But in this polynomial regression, the **values** are **gone** for **negative values.**
- These are because of **insufficient data or biased data features**. So, we **can't find** the **near future values** with these data.
- **Regression accuracy** is **better** with **huge amounts of data** unlike the given data and **unbiased featured data.**