

SALARY PREDICTION using POLYNOMIAL REGRESSION

Importing Libraries

In [1]:

```
import pandas as pd
```

Load Dataset from Local Directory

In [2]:

```
from google.colab import files
uploaded = files.upload()
```

Choose File

No file selected

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving Dataset.csv to Dataset.csv

Load Dataset

In [5]:

```
dataset = pd.read_csv('Dataset.csv')
```

Summarize Dataset

In [6]:

```
print(dataset.shape)
print(dataset.head(5))
```

```
(10, 2)
   Level  Salary
0       1   45000
1       2   50000
2       3   60000
3       4   80000
4       5  110000
```

Segregate Dataset into Input X & Output Y

In [7]:

```
X = dataset.iloc[:, :-1]
X
```

Out[7]:

| | Level |
|---|-------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 4 | 5 |
| 5 | 6 |

| 6 | Level |
|---|-------|
| 7 | 8 |
| 8 | 9 |
| 9 | 10 |

In [8]:

```
Y = dataset.iloc[:, -1].values
Y
```

Out[8]:

```
array([ 45000,  50000,  60000,  80000, 110000, 150000, 200000,
        300000, 500000, 1000000])
```

Training Dataset using Linear Regression

In [9]:

```
from sklearn.linear_model import LinearRegression
modelLR = LinearRegression()
modelLR.fit(X, Y)
```

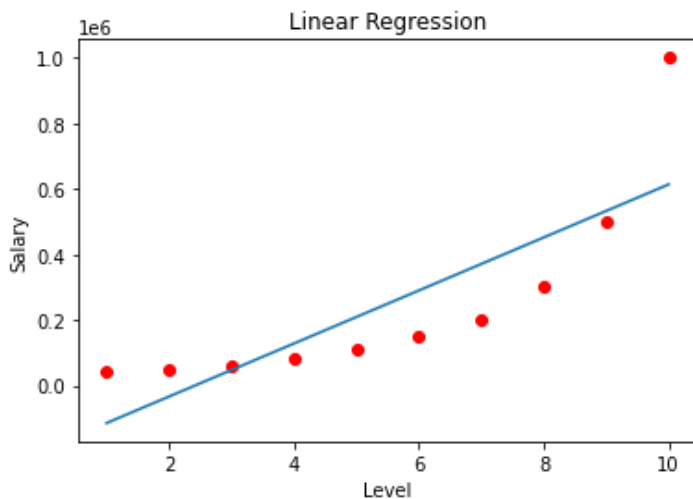
Out[9]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Visualizing Linear Regression results

In [10]:

```
import matplotlib.pyplot as plt
plt.scatter(X,Y, color = 'red')
plt.plot(X, modelLR.predict(X))
plt.title("Linear Regression")
plt.xlabel("Level")
plt.ylabel("Salary")
plt.show()
```



- fit() - Training Model - Calculating the initial Parameters
- transform() - After Training we will transform Data by using above calculated values
- fit_transform() - First fit & Transform

Convert X to Polynomial Format(X^n)

- n -degree
- n = 2 consist x & x^2
- n = 3 consist x & x^2 & x^3

In [12]:

```
from sklearn.preprocessing import PolynomialFeatures
modelPR = PolynomialFeatures(degree = 2)
xPoly = modelPR.fit_transform(X)
```

Train same Linear Regression with X-Polynomial instead of X

In [13]:

```
modelPLR = LinearRegression()
modelPLR.fit(xPoly, Y)
```

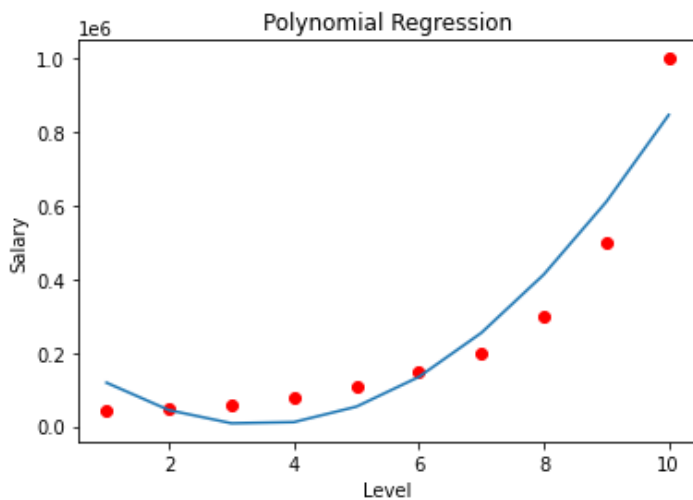
Out[13]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Visualizing Polyomial Regression results

In [16]:

```
plt.scatter(X,Y, color = "red")
plt.plot(X, modelPLR.predict(modelPR.fit_transform(X)))
plt.title("Polynomial Regression")
plt.xlabel("Level")
plt.ylabel("Salary")
plt.show()
```



Prediction using Polynomial Regression

In [17]:

```
x = 8.5
salaryPred = modelPLR.predict(modelPR.fit_transform([[x]]))
print('Salary of a person with Level {0} is {1}'.format(x, salaryPred))
```

Salary of a person with Level 8.5 is [506710.22727273]