# Practice Project - Polynomial Regression based : Using dataset "Position_Salaries.csv"

## Table of Contents

## Problem Statement:

```
 Problem using Polynomial Regression - Predicting new results with Polynomial Regression. Note that the input variable must be in a numpy 2D array.
```

# polynomial regression

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below Equation: y= b0+b1x1+ b2x12+ b2x13+...... bnx1n

In [ ]:
```
#Polynomial Steps:
```

# 1.Importing Libraries

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

# 2.Importing and reading the Dataset

In [ ]:
```
df=pd.read_csv('../Polynomial regression/data/Position_Salaries.csv')
df
```

Out[ ]:

| | 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 |
| **5** | Region Manager | 6 | 150000 |
| **6** | Partner | 7 | 200000 |
| **7** | Senior Partner | 8 | 300000 |
| **8** | C-level | 9 | 500000 |
| **9** | CEO | 10 | 1000000 |

## 3.Dividing the dataset into 2 components

*Divide dataset into two components that is X and y.X will contain the Column between 1 and 2. y will contain the 2 columns.*

In [ ]:
```
X = df.iloc[:, 1:2].values
y = df.iloc[:, 2].values
```

## 4.Fitting Linear Regression to the dataset

*Fitting the linear Regression model On two components.*

In [ ]:
```
# Fitting Linear Regression to the dataset
from sklearn.linear_model import LinearRegression
lin = LinearRegression()

lin.fit(X, y)
```

Out[ ]:
```
LinearRegression()
```

## 5.Fitting Polynomial Regression to the dataset

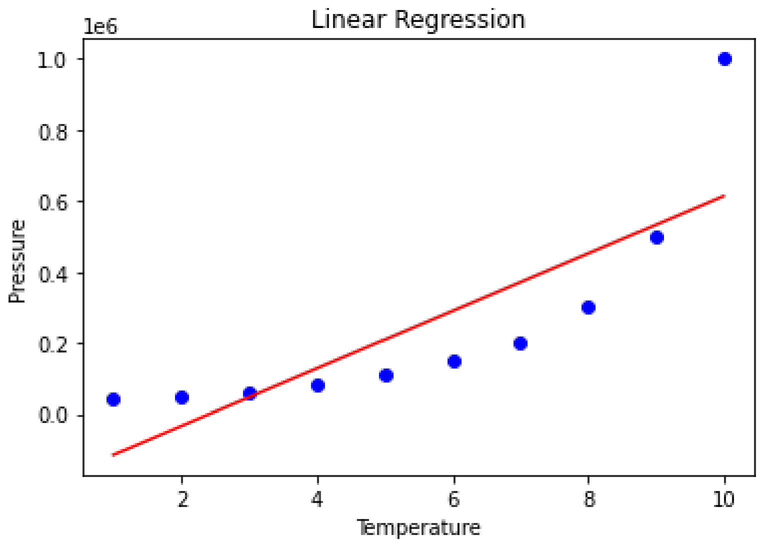
*Fitting the Polynomial Regression model on two components X and y.*

In [ ]:
```
# Fitting Polynomial Regression to the dataset
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree = 4)
X_poly = poly.fit_transform(X)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)
```

Out[ ]:
```
LinearRegression()
```

## 6.In this step, we are Visualising the Linear Regression results using a scatter plot.

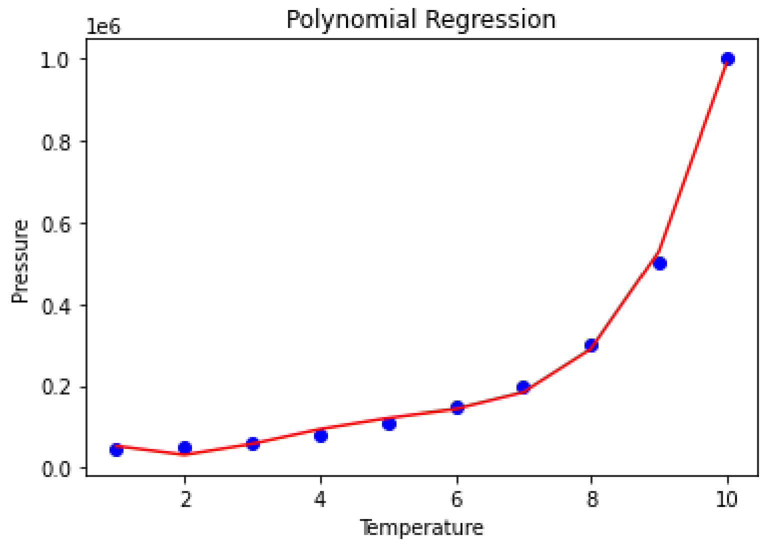
In [ ]:
```
# Visualising the Linear Regression results
plt.scatter(X, y, color = 'blue')

plt.plot(X, lin.predict(X), color = 'red')
plt.title('Linear Regression')
plt.xlabel('Temperature')
plt.ylabel('Pressure')
plt.show()
```



## 7.Visualising the Polynomial Regression results using a scatter plot.

In [ ]:

```
# Visualising the Polynomial Regression results
plt.scatter(X, y, color = 'blue')

plt.plot(X, lin2.predict(poly.fit_transform(X)), color = 'red')
plt.title('Polynomial Regression')
plt.xlabel('Temperature')
plt.ylabel('Pressure')

plt.show()
```



## 8.Predicting new results with both Linear and Polynomial Regression. Note that the input variable must be in a numpy 2D array.

In [ ]:

```
# Predicting a new result with Linear Regression after converting predict variable to 2D array
pred = 110.0
predarray = np.array([[pred]])
lin.predict(predarray)
```

Out[ ]:

```
array([8701333.33333333])
```

In [ ]:

```
# Predicting a new result with Polynomial Regression after converting predict variable to 2D array
pred2 = 110.0
pred2array = np.array([[pred2]])
lin2.predict(poly.fit_transform(pred2array))
```

Out[ ]:

```
array([1.10869084e+11])
```