In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
df=pd.read_csv("C:/Users/visha/Downloads/PolyData.csv")
df
```

Out[3]:

	Unnamed: 0	x	y
0	0	-0.216619	2.113105
1	1	2.945493	10.795517
2	2	-2.818077	4.346195
3	3	-1.641737	3.622927
4	4	0.200467	3.759674
...	...	...	...
195	195	0.057998	2.350656
196	196	-2.936630	6.285578
197	197	2.644792	11.962454
198	198	2.009540	6.082032
199	199	-1.916395	2.883002

200 rows × 3 columns

In [9]:
```
X = df.iloc[:, 1:2].values
Y = df.iloc[:, 2].values
```

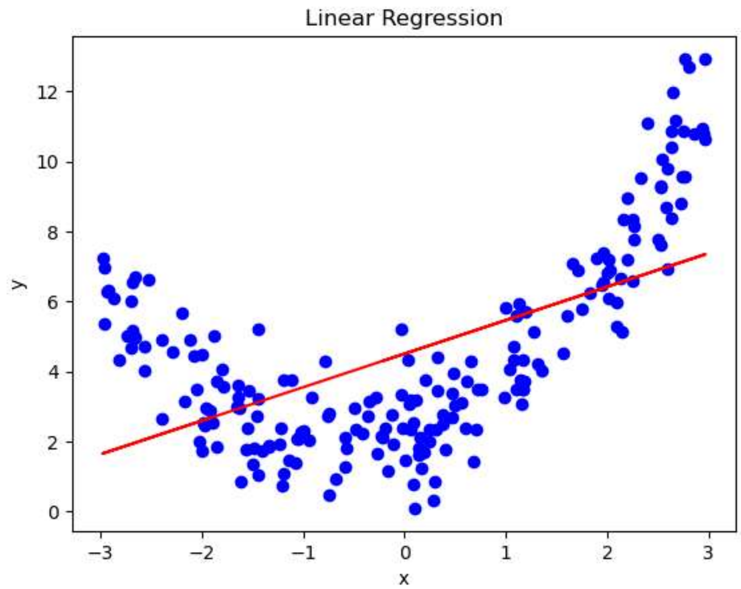
In [10]:
```
from sklearn.linear_model import LinearRegression
lin = LinearRegression()

lin.fit(X, Y)
```

Out[10]: `LinearRegression()`

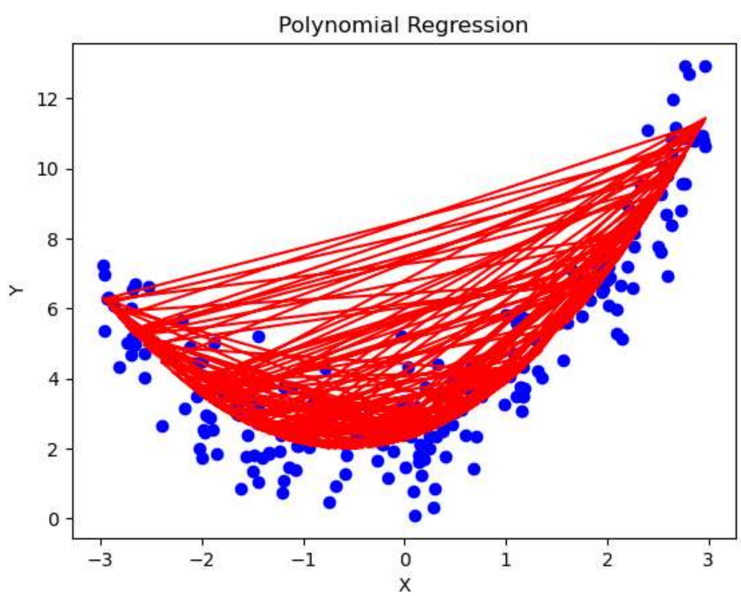
In [11]:
```
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree = 2)
X_poly = poly.fit_transform(X)

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

Out[11]: `LinearRegression()`

In [13]:
```
plt.scatter(X, Y, color = 'blue')

plt.plot(X, lin.predict(X), color = 'red')
plt.title('Linear Regression')
plt.xlabel('x')
plt.ylabel('y')

plt.show()
```



In [14]:
```
# 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('X')
plt.ylabel('Y')

plt.show()
```



In [ ]:
```
#Advantages of using Polynomial Regression:
#A broad range of functions can be fit under it.
#Polynomial basically fits a wide range of curvatures.
#Polynomial provides the best approximation of the relationship between dependent and independent variables.
#Disadvantages of using Polynomial Regression
#These are too sensitive to the outliers.
#The presence of one or two outliers in the data can seriously affect the results of nonlinear analysis.
#In addition, there are unfortunately fewer model validation tools for the detection of outliers in nonlinear regression than the
```