Submitted By

Name - Adarsh Ghimire

Student ID - 100058927

Subject - Machine Learning

In [ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:
```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import seaborn as sns
import matplotlib.pyplot as plt
```

In [ ]:
```
data_path = "drive/MyDrive/COSC 606 Machine learning/COVID-19_Daily_Testing.csv"
df = pd.read_csv(data_path)
```

In [ ]:
```
df.head()
```

Out[ ]:

	Date	Day	Tests	Cases	People Not-Positive - Total	People Tested - Age 0-17	People Tested - Age 18-29	People Tested - Age 30-39	People Tested - Age 40-49	People Tested - Age 50-59	People Tested - Age 60-69
0	03-01-20	Sunday	1	0	1	0	0	1	0	0	0
1	05-02-20	Saturday	2,431	705	1,726	129	470	458	458	412	281
2	05/14/2020	Thursday	4,098	772	3,326	260	805	833	685	604	471
3	03-05-20	Thursday	17	1	16	4	2	0	4	3	2
4	03-06-20	Friday	18	3	15	1	5	1	3	3	2

In [ ]:
```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89 entries, 0 to 88
Data columns (total 59 columns):
 #   Column                                          Non-Null Count  Dtype 
---  ------                                          --------------  ----- 
 0   Date                                            88 non-null     object
 1   Day                                             88 non-null     object
```

```
 2   Tests                                          89 non-null     object
 3   Cases                                          89 non-null     object
 4   People Not-Positive - Total                    89 non-null     object
 5   People Tested - Age 0-17                       89 non-null     int64
 6   People Tested - Age 18-29                      89 non-null     object
 7   People Tested - Age 30-39                      89 non-null     object
 8   People Tested - Age 40-49                      89 non-null     int64
 9   People Tested - Age 50-59                      89 non-null     int64
 10  People Tested - Age 60-69                      89 non-null     int64
 11  People Tested - Age 70-79                      89 non-null     int64
 12  People Tested - Age 80+                        89 non-null     int64
 13  People Tested - Age Unknown                    89 non-null     int64
 14  People Tested - Female                         89 non-null     object
 15  People Tested - Male                           89 non-null     object
 16  People Tested - Gender Unknown                 89 non-null     int64
 17  People Tested - Latinx                         89 non-null     object
 18  People Tested - Asian Non-Latinx               89 non-null     int64
 19  People Tested - Black Non-Latinx               89 non-null     int64
 20  People Tested - White Non-Latinx               89 non-null     int64
 21  People Tested - Other Race Non-Latinx          89 non-null     int64
 22  People Tested - Unknown Race/Ethnicity         89 non-null     object
 23  People Positive - Age 0-17                     89 non-null     int64
 24  People Positive - Age 18-29                    89 non-null     int64
 25  People Positive - Age 30-30                    89 non-null     int64
 26  People Positive - Age 40-49                    89 non-null     int64
 27  People Positive - Age 50-59                    89 non-null     int64
 28  People Positive - Age 60-69                    89 non-null     int64
 29  People Positive - Age 70-79                    89 non-null     int64
 30  People Positive - Age 80+                      89 non-null     int64
 31  People Positive - Age Unknown                  89 non-null     int64
 32  People Positive - Female                       89 non-null     int64
 33  People Positive - Male                         89 non-null     int64
 34  People Positive - Gender Unknown               89 non-null     int64
 35  People Positive - Latinx                       89 non-null     int64
 36  People Positive - Asian Non-Latinx             89 non-null     int64
 37  People Positive - Black Non-Latinx             89 non-null     int64
 38  People Positive - White Non-Latinx             89 non-null     int64
 39  People Positive - Other Race Non-Latinx        89 non-null     int64
 40  People Positive - Unknown Race/Ethnicity       89 non-null     int64
 41  People Not-Positive - Age 0-17                 89 non-null     int64
 42  People Not-Positive - Age 18-29                89 non-null     object
 43  People Not-Positive - Age 30-39                89 non-null     int64
 44  People Not-Positive - Age 40-49                89 non-null     int64
 45  People Not-Positive - Age 50-59                89 non-null     int64
 46  People Not-Positive - Age 60-69                89 non-null     int64
 47  People Not-Positive - Age 70-79                89 non-null     int64
 48  People Not-Positive - Age 80+                  89 non-null     int64
 49  People Not-Positive - Age Unknown              89 non-null     int64
 50  People Not-Positive - Female                   89 non-null     object
 51  People Not-Positive - Male                     89 non-null     object
 52  People Not-Positive - Gender Unknown           89 non-null     int64
 53  People Not-Positive - Latinx                   89 non-null     int64
 54  People Not-Positive - Asian Non-Latinx         89 non-null     int64
 55  People Not-Positive - Black Non-Latinx         89 non-null     int64
 56  People Not-Positive - White Non-Latinx         89 non-null     int64
 57  People Not-Positive - Other Race Non-Latinx    89 non-null     int64
 58  People Not-Positive - Unknown Race/Ethnicity   89 non-null     object
dtypes: int64(44), object(15)
memory usage: 41.1+ KB
None
```

In [ ]:

```
# To replace commas from the numbers and converting the object type to integer t
# Concerned only about Tests done as feature and Cases found as target
# The regression problem will be focused on the basis of finding relation betwee
```

```
df['Cases'] = df['Cases'].str.replace(',', '')
df['Tests'] = df['Tests'].str.replace(',', '')
df['Cases'] = pd.to_numeric(df['Cases'])
df['Tests'] = pd.to_numeric(df['Tests'])
```

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 89 entries, 0 to 88
Data columns (total 59 columns):
 #   Column                                    Non-Null Count  Dtype 
---  ------                                    --------------  ----- 
 0   Date                                      88 non-null     object
 1   Day                                       88 non-null     object
 2   Tests                                     89 non-null     int64 
 3   Cases                                     89 non-null     int64 
 4   People Not-Positive - Total               89 non-null     object
 5   People Tested - Age 0-17                  89 non-null     int64 
 6   People Tested - Age 18-29                 89 non-null     object
 7   People Tested - Age 30-39                 89 non-null     object
 8   People Tested - Age 40-49                 89 non-null     int64 
 9   People Tested - Age 50-59                 89 non-null     int64 
 10  People Tested - Age 60-69                 89 non-null     int64 
 11  People Tested - Age 70-79                 89 non-null     int64 
 12  People Tested - Age 80+                   89 non-null     int64 
 13  People Tested - Age Unknown               89 non-null     int64 
 14  People Tested - Female                    89 non-null     object
 15  People Tested - Male                      89 non-null     object
 16  People Tested - Gender Unknown            89 non-null     int64 
 17  People Tested - Latinx                    89 non-null     object
 18  People Tested - Asian Non-Latinx          89 non-null     int64 
 19  People Tested - Black Non-Latinx          89 non-null     int64 
 20  People Tested - White Non-Latinx          89 non-null     int64 
 21  People Tested - Other Race Non-Latinx     89 non-null     int64 
 22  People Tested - Unknown Race/Ethnicity    89 non-null     object
 23  People Positive - Age 0-17                89 non-null     int64 
 24  People Positive - Age 18-29               89 non-null     int64 
 25  People Positive - Age 30-30               89 non-null     int64 
 26  People Positive - Age 40-49               89 non-null     int64 
 27  People Positive - Age 50-59               89 non-null     int64 
 28  People Positive - Age 60-69               89 non-null     int64 
 29  People Positive - Age 70-79               89 non-null     int64 
 30  People Positive - Age 80+                 89 non-null     int64 
 31  People Positive - Age Unknown             89 non-null     int64 
 32  People Positive - Female                  89 non-null     int64 
 33  People Positive - Male                    89 non-null     int64 
 34  People Positive - Gender Unknown          89 non-null     int64 
 35  People Positive - Latinx                  89 non-null     int64 
 36  People Positive - Asian Non-Latinx        89 non-null     int64 
 37  People Positive - Black Non-Latinx        89 non-null     int64 
 38  People Positive - White Non-Latinx        89 non-null     int64 
 39  People Positive - Other Race Non-Latinx   89 non-null     int64 
 40  People Positive - Unknown Race/Ethnicity  89 non-null     int64 
 41  People Not-Positive - Age 0-17            89 non-null     int64 
 42  People Not-Positive - Age 18-29           89 non-null     object
 43  People Not-Positive - Age 30-39           89 non-null     int64 
 44  People Not-Positive - Age 40-49           89 non-null     int64 
 45  People Not-Positive - Age 50-59           89 non-null     int64 
 46  People Not-Positive - Age 60-69           89 non-null     int64 
 47  People Not-Positive - Age 70-79           89 non-null     int64 
 48  People Not-Positive - Age 80+             89 non-null     int64 
 49  People Not-Positive - Age Unknown         89 non-null     int64 
 50  People Not-Positive - Female              89 non-null     object
```

```
 51  People Not-Positive - Male                    89 non-null     object
 52  People Not-Positive - Gender Unknown          89 non-null     int64
 53  People Not-Positive - Latinx                  89 non-null     int64
 54  People Not-Positive - Asian Non-Latinx        89 non-null     int64
 55  People Not-Positive - Black Non-Latinx        89 non-null     int64
 56  People Not-Positive - White Non-Latinx        89 non-null     int64
 57  People Not-Positive - Other Race Non-Latinx   89 non-null     int64
 58  People Not-Positive - Unknown Race/Ethnicity  89 non-null     object
dtypes: int64(46), object(13)
memory usage: 41.1+ KB
```

In [ ]:

```
data_numeric = df.select_dtypes(include=['float64', 'int64'])
plt.figure(figsize=(20, 10))
sns.pairplot(data_numeric)
plt.show()
```

In [ ]:

```
X = df['Tests'].values.reshape(-1,1)
y = df['Cases'].values.reshape(-1,1)
```

In [ ]:

```
print(X.shape)
print(y.shape)
```

```
(89, 1)
(89, 1)
```

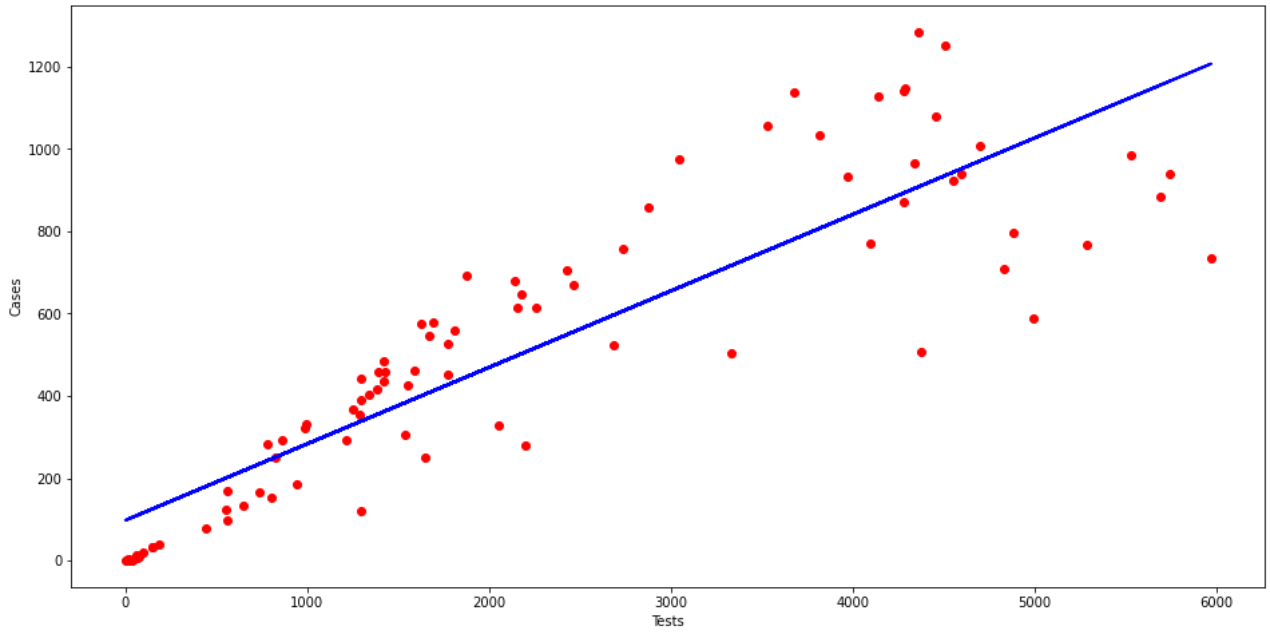
Linear Regression Model with out any feature manipulations

In [ ]:

```
reg = LinearRegression()
reg.fit(X, y)
predictions = reg.predict(X)
print("The linear model is: Y = {:.5} + {:.5}X".format(reg.intercept_[0], reg.co
plt.figure(figsize=(16, 8))

plt.scatter(X, y, c='red')
plt.plot(X, predictions, c='blue', linewidth=2)
plt.xlabel("Tests")
plt.ylabel("Cases")
plt.show()
```

```
The linear model is: Y = 97.777 + 0.18572X
```



In [ ]:
```
print('RMSE for Linear Regression : ',np.sqrt(mean_squared_error(y,predictions))
```

```
RMSE for Linear Regression :  171.79768160540917
```

The RMSE calculated with the Linear regression without any feature manipulation is **171.79**

Linear Regression with just feature scaling

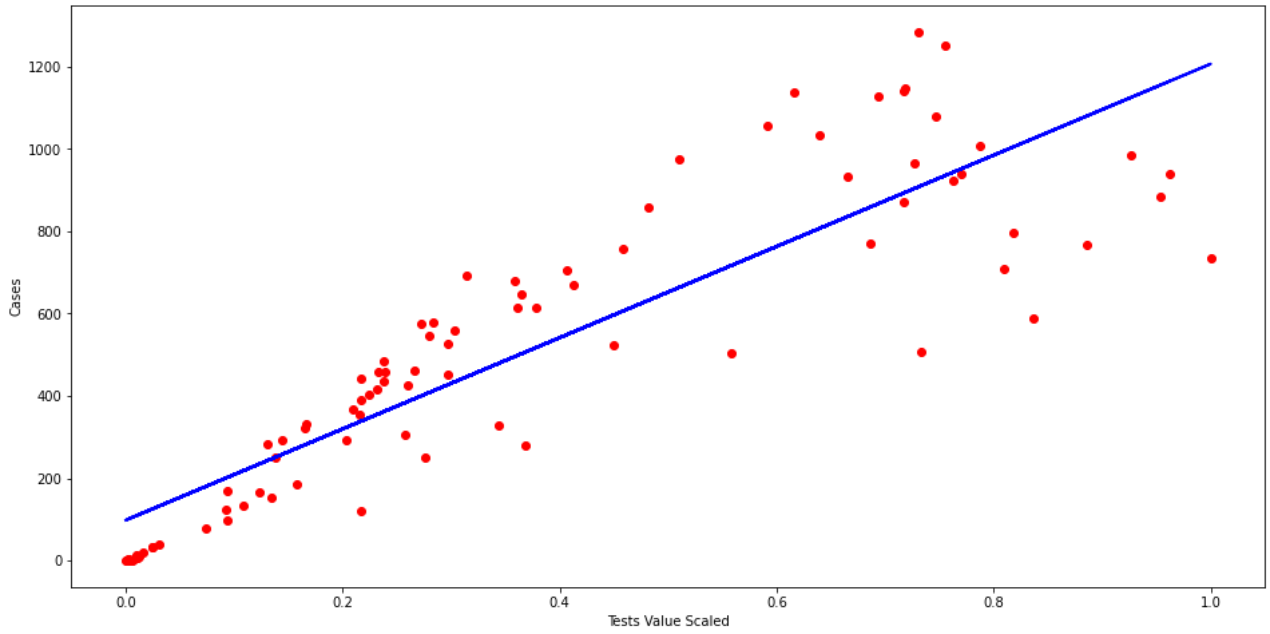
In [ ]:
```
# Feature Scaling, which means changing values from 0 to 1
X_scaled = (X - min(X))/(max(X)-min(X))
print(X_scaled[:3])
```

```
[[0.        ]
 [0.40703518]
 [0.68626466]]
```

In [ ]:
```
reg_scaled = LinearRegression()
reg_scaled.fit(X_scaled, y)
predictions_scaled = reg_scaled.predict(X_scaled)
print("The linear model is: Y = {:.5} + {:.5}X".format(reg_scaled.intercept_[0],
plt.figure(figsize=(16, 8))

plt.scatter(X_scaled, y, c='red')
plt.plot(X_scaled, predictions_scaled, c='blue', linewidth=2)
plt.xlabel("Tests Value Scaled")
plt.ylabel("Cases")
plt.show()
```

```
The linear model is: Y = 97.963 + 1108.8X
```



In [ ]:
```
print('RMSE for Linear Regression : ',np.sqrt(mean_squared_error(y, predictions_
```

```
RMSE for Linear Regression :  171.79768160540914
```

The RMSE calculated with the Linear regression with feature scaled is **171.7976**

Linear Regression with just feature normalization

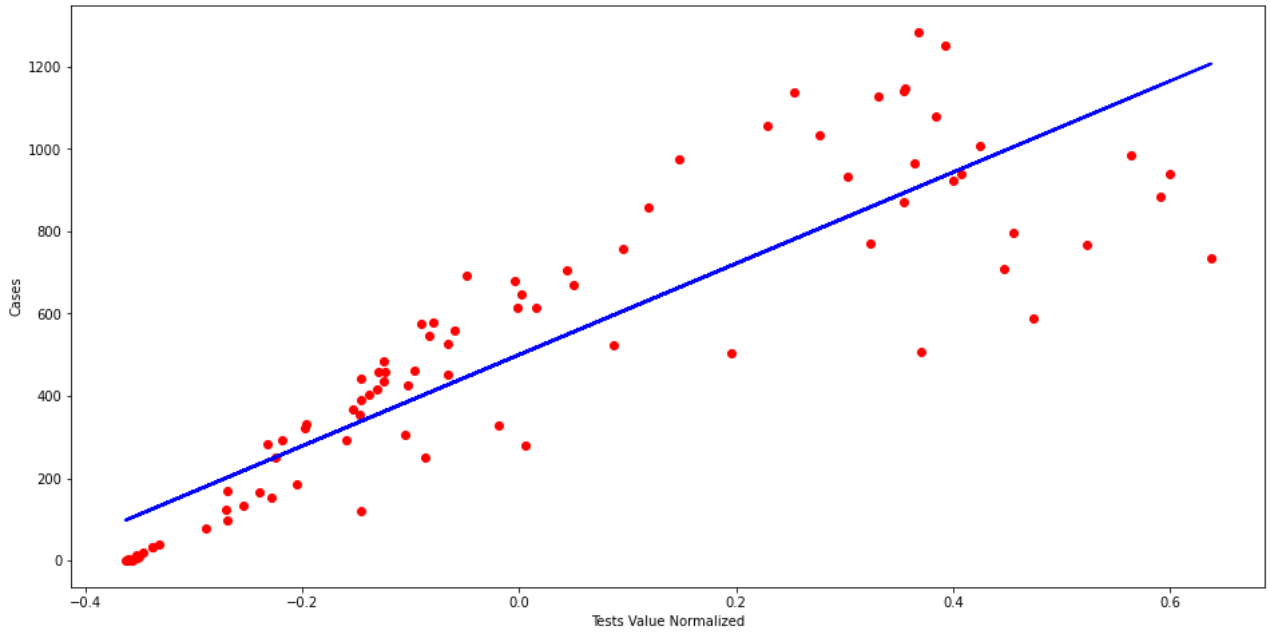
In [ ]:
```
# Feature Normalization, which means changing feature values such that distribut
X_normalized = (X - np.mean(X))/(max(X)-min(X))
print(X_normalized[:3])
```

```
[[-0.36255811]
 [ 0.04447707]
 [ 0.32370655]]
```

In [ ]:
```
reg_normalized = LinearRegression()
reg_normalized.fit(X_normalized, y)
predictions_normalized = reg_normalized.predict(X_normalized)
print("The linear model is: Y = {:.5} + {:.5}X".format(reg_normalized.intercept_
plt.figure(figsize=(16, 8))

plt.scatter(X_normalized, y, c='red')
plt.plot(X_normalized, predictions_normalized, c='blue', linewidth=2)
plt.xlabel("Tests Value Normalized")
plt.ylabel("Cases")
plt.show()
```

```
The linear model is: Y = 499.96 + 1108.8X
```



In [ ]:
```
print('RMSE for Linear Regression : ',np.sqrt(mean_squared_error(y, predictions_
```

```
RMSE for Linear Regression :  171.79768160540917
```

The RMSE calculated with the Linear regression with feature normalized is **171.7976**

Linear Regression with just feature standardization

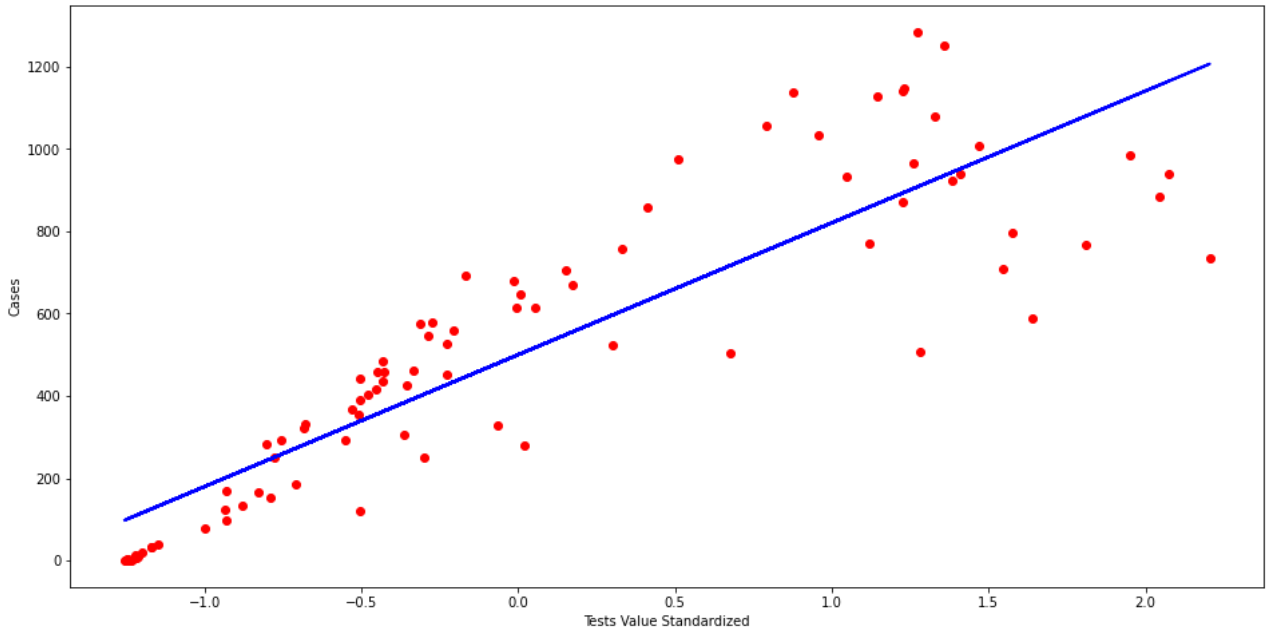
In [ ]:
```
# Feature Standardization, which means changing feature values such that distrib
X_std = (X - np.mean(X))/(np.std(X))
print(X_std[:3])
```

```
[[-1.25398817]
 [ 0.15383387]
 [ 1.11961137]]
```

In [ ]:
```
reg_std = LinearRegression()
reg_std.fit(X_std, y)
predictions_std = reg_std.predict(X_std)
print("The linear model is: Y = {:.5} + {:.5}X".format(reg_std.intercept_[0], re
plt.figure(figsize=(16, 8))

plt.scatter(X_std, y, c='red')
plt.plot(X_std, predictions_std, c='blue', linewidth=2)
plt.xlabel("Tests Value Standardized")
plt.ylabel("Cases")
plt.show()
```

```
The linear model is: Y = 499.96 + 320.57X
```



In [ ]:
```
print('RMSE for Linear Regression : ',np.sqrt(mean_squared_error(y, predictions_
```

```
RMSE for Linear Regression :  171.79768160540917
```

The RMSE calculated with the Linear regression with feature normalized is **171.7976**

The feature standarizations, normalizations, and scaling, have not affected the regression problem at hand. Because the linear regression problem is only univariate, and the scikit-learn uses the least square technique to compute linear regression model. Thus, all the feature manipulation techniques do not affect the overall result of the system. However, it would have affected if only we have used the gradient descent approach to compute the regression model.

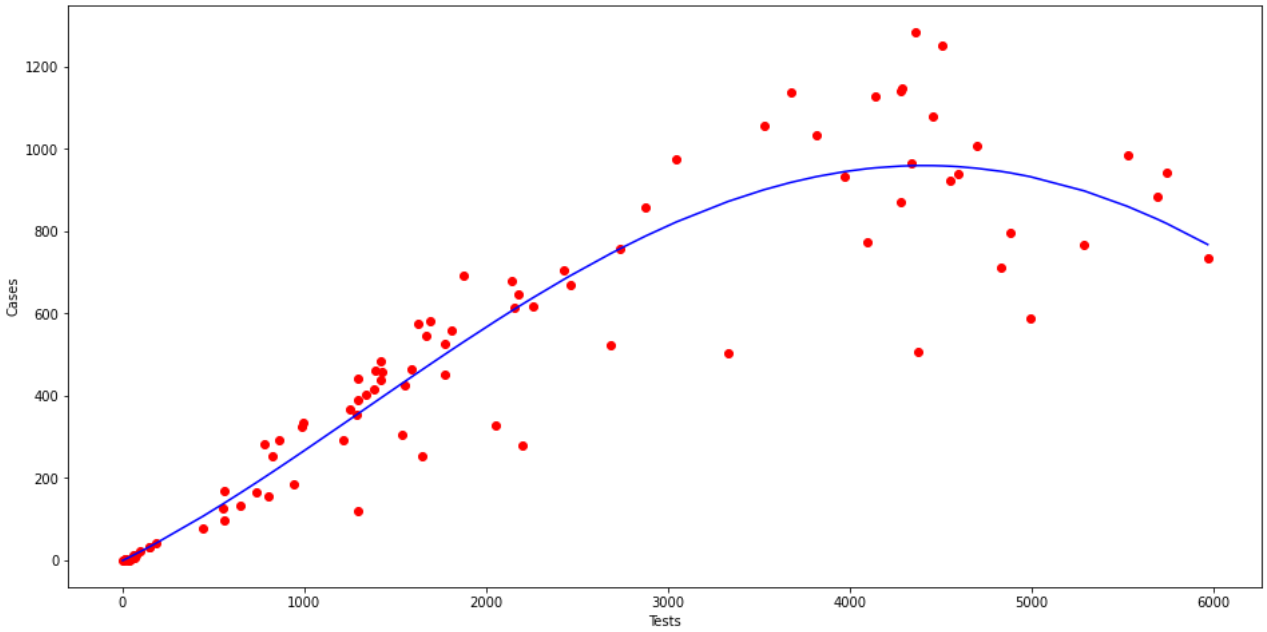
Polynomial Regression

Polynomial regression with degree 4

In [ ]:
```
poly = PolynomialFeatures(degree = 4)
X_poly = poly.fit_transform(X)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)
pred = lin2.predict(X_poly)
new_X, new_y = zip(*sorted(zip(X, pred)))

plt.figure(figsize=(16, 8))
plt.scatter(X, y, c='red')
plt.plot(new_X, new_y, c='blue')
plt.xlabel("Tests")
plt.ylabel("Cases")
plt.show()
print('RMSE for Linear Regression=>',np.sqrt(mean_squared_error(y,lin2.predict(p
```




```
RMSE for Linear Regression=> 131.076775703435
```

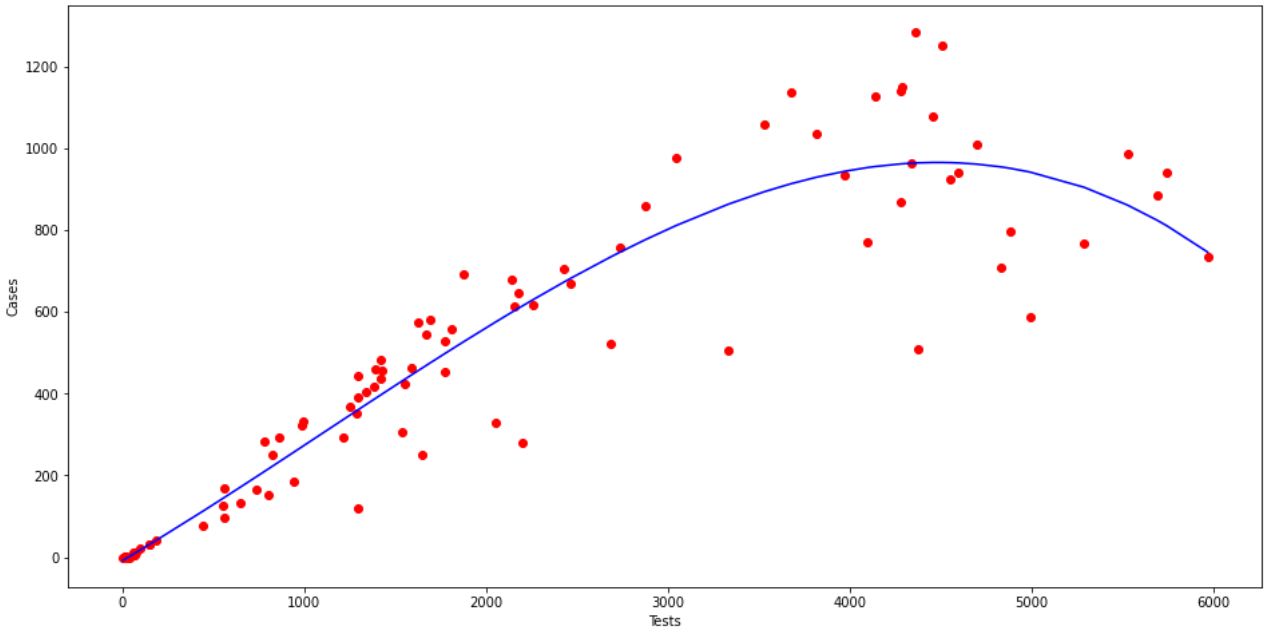
Polynomial regression with degree 3

In [ ]:

```
poly = PolynomialFeatures(degree = 3)
X_poly = poly.fit_transform(X)

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)
pred = lin2.predict(X_poly)
new_X, new_y = zip(*sorted(zip(X, pred)))

plt.figure(figsize=(16, 8))
plt.scatter(X, y, c='red')
plt.plot(new_X, new_y, c='blue')
plt.xlabel("Tests")
plt.ylabel("Cases")
plt.show()
print('RMSE for Linear Regression=>',np.sqrt(mean_squared_error(y,lin2.predict(p
```




```
RMSE for Linear Regression=> 131.2391556826983
```

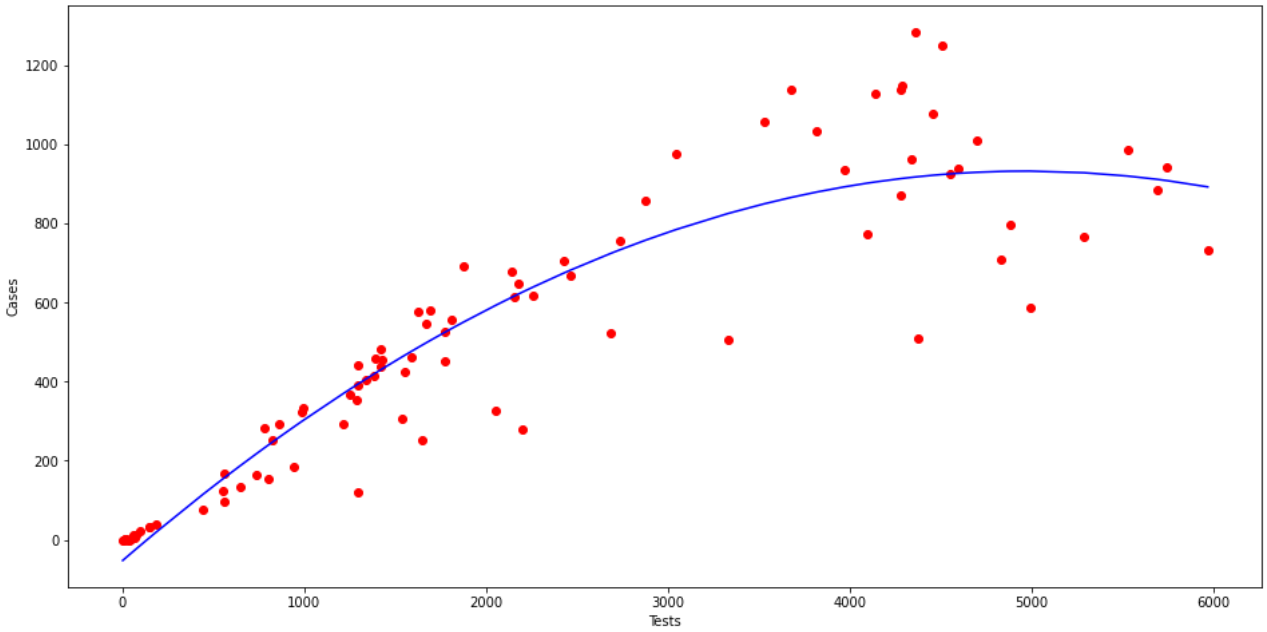
Polynomial regression with degree 2

In [ ]:

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

poly.fit(X_poly, y)
lin2 = LinearRegression()
lin2.fit(X_poly, y)
pred = lin2.predict(X_poly)
new_X, new_y = zip(*sorted(zip(X, pred)))

plt.figure(figsize=(16, 8))
plt.scatter(X, y, c='red')
plt.plot(new_X, new_y, c='blue')
plt.xlabel("Tests")
plt.ylabel("Cases")
plt.show()
print('RMSE for Linear Regression=>',np.sqrt(mean_squared_error(y,lin2.predict(p
```




```
RMSE for Linear Regression=> 136.7701535452868
```

The regression model based on degree of polynomial can be seen to have produced varying RMSE score, because of the fact that the higher order polynomials curves are much complex and can easily fit the data, however the lower order polynomials are simple and might not completely fit the data. Thus, need to take into account the consideration for increasing complexity or increasing generalizability of the model.

In [ ]: