In [1]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

In [2]:
```
# Import data set csv
data_set = pd.read_csv('filtered_data_set.csv')
```

In [3]:
```
# Convert to np array
data_set = np.array(data_set)
```

In [4]:
```
# Determine the number of rows in the raw data.
num_rows = np.size(data_set, 0)
```

In [5]:
```
# Using the number of rows, create a column of 1s to place as the first column ir
ones_col = (np.ones((num_rows, 1)))
```

In [6]:
```
# Set X to all values to our inputs.
X = np.concatenate((ones_col, data_set), axis=1)
```

In [7]:
```
# Set y to only the output column
y = data_set[:,-1]
```

In [8]:
```
# Drop the last column
X = X[:, :-1]
```

In [9]:
```
# Create the X training and testing set, and Y training and testing set where 70%
# are for the training set and the rest to the testing set.
x_testing_set, x_training_set, y_testing_set, y_training_set = train_test_split()
                                                                                s
```

In [10]:
```
# Create linear regression object
lm = LinearRegression()
```

In [11]:
```
# Fit the linear model to the training data
lm.fit(x_training_set,y_training_set)
```

Out[11]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [12]:
```
# Find the predicted y values
predictions = lm.predict(x_testing_set)
```

In [13]:
```
predictions
```

Out[13]:
```
array([0.19431141, 0.30192189, 0.30117236, ..., 0.20793911, 0.26909863,
       0.09884488])
```

In [14]:
```
# Compare the predicted outputs to a threshold value
scaled_predictions = []
for i in range(0,len(predictions)):
    if(predictions[i] >= 0.5):
        scaled_predictions.append(1)
    else:
        scaled_predictions.append(0)
```

In [15]:
```
# Calculate the accuracy
correct = 0
total = 0
for i in range(0, len(predictions)):
    if(y_testing_set[i] == scaled_predictions[i]):
        correct = correct + 1
    total = total + 1
print("Correct = ", correct, " Total = ", total, " ",((correct/total)*100), "%")
```

```
Correct =  2299  Total =  2961   77.64268828098615 %
```

In [ ]:

In [ ]:

In [ ]: