Assignment 1 Set B1

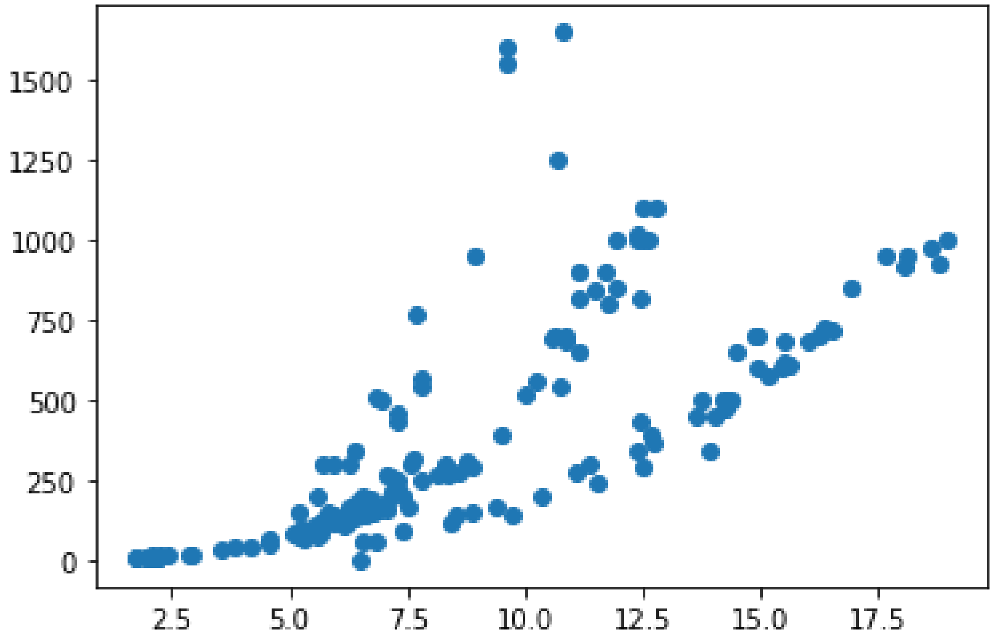
March 2, 2024

1 Build a simple linear regression model for Fish Species Weight Prediction.

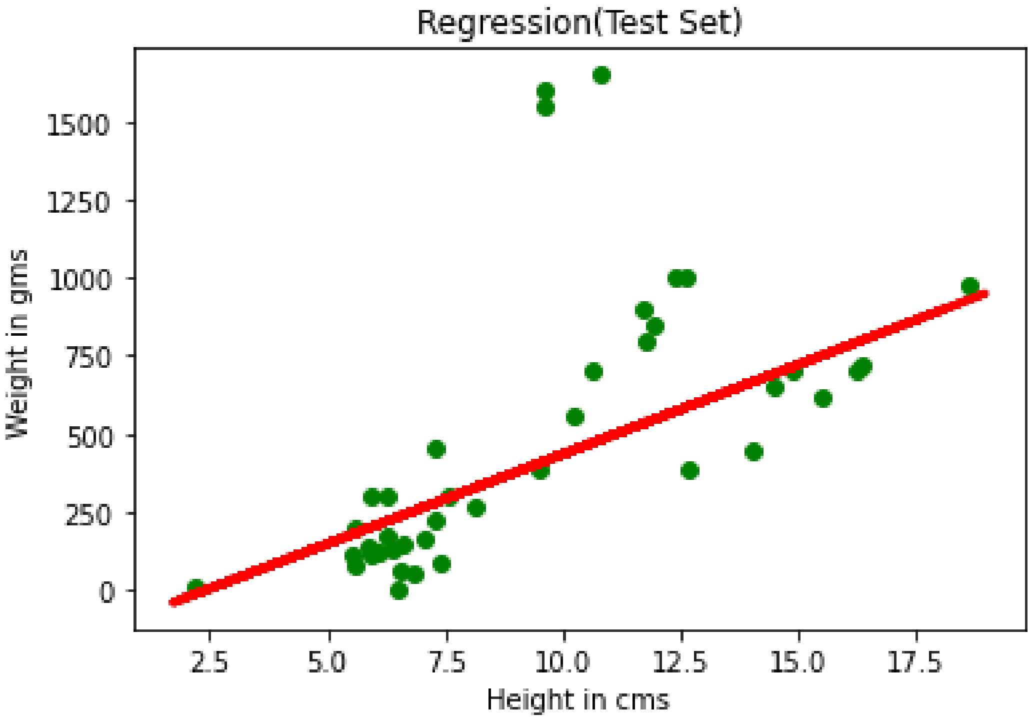
[1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,mean_squared_error
%matplotlib inline
db=pd.read_csv('/home/mmcc/Desktop/DA Data Sets/Fish.csv')
X=db['Height'].values.reshape(-1, 1)
y=db['Weight'].values.reshape(-1, 1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.25,random_state=0) #random_state=0 means you will get same testing and training data as many times as youu run the program
#print("X_TEST:\n ",X_test)
print("X_TEST Size:\n",X_test.shape)
#print("X_TRAIN:\n ",X_train)
print("X_TRAIN Size:\n",X_train.shape)
#print("Y_TEST:\n ",y_test)
print("Y_TEST Size:\n",y_test.shape)
#print("Y_TRAIN:\n ",y_train)
print("Y_TRAIN Size:\n",y_train.shape)
print("Scatter Plot Graph")
plt.scatter(X,y)
plt.show()
```

```
X_TEST Size:
 (40, 1)
X_TRAIN Size:
 (119, 1)
Y_TEST Size:
 (40, 1)
Y_TRAIN Size:
 (119, 1)
Scatter Plot Graph
```

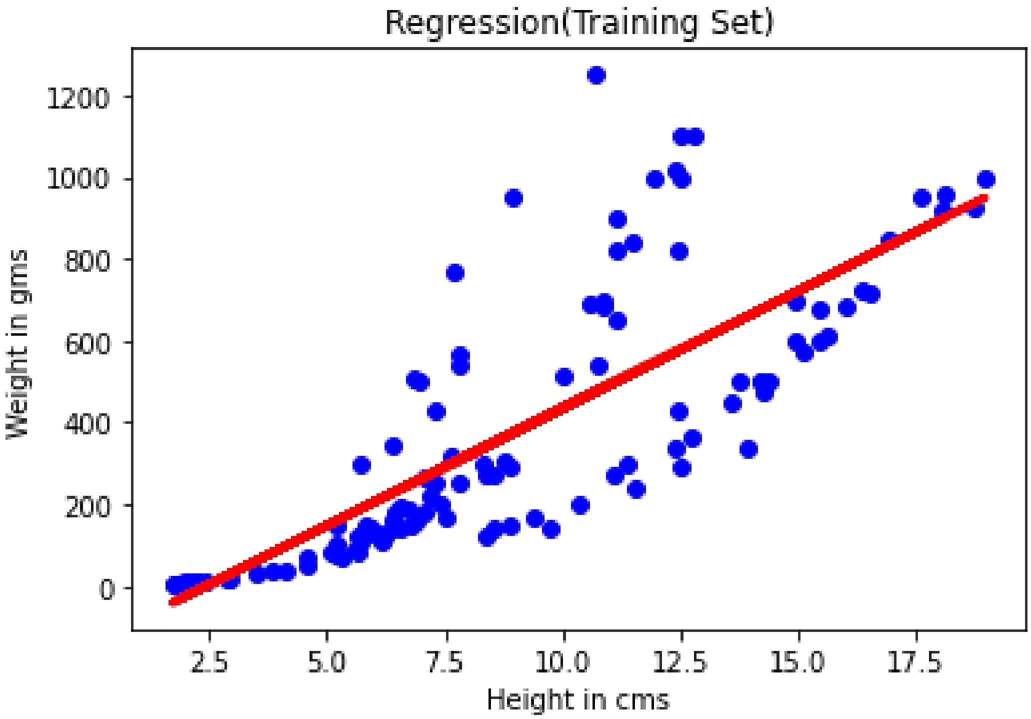


[2]:
```
# Spliting into train & test dataset
regressor = LinearRegression() # Creating a regressior
regressor.fit(X_train,y_train) # Fiting the dataset into the model
plt.scatter(X_test,y_test,color="green") # Plot a graph with X_test vs y_test
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3) #␣
 ↪Regressior line showing
plt.title('Regression(Test Set)')
plt.xlabel('Height in cms')
plt.ylabel('Weight in gms')
plt.show()
```



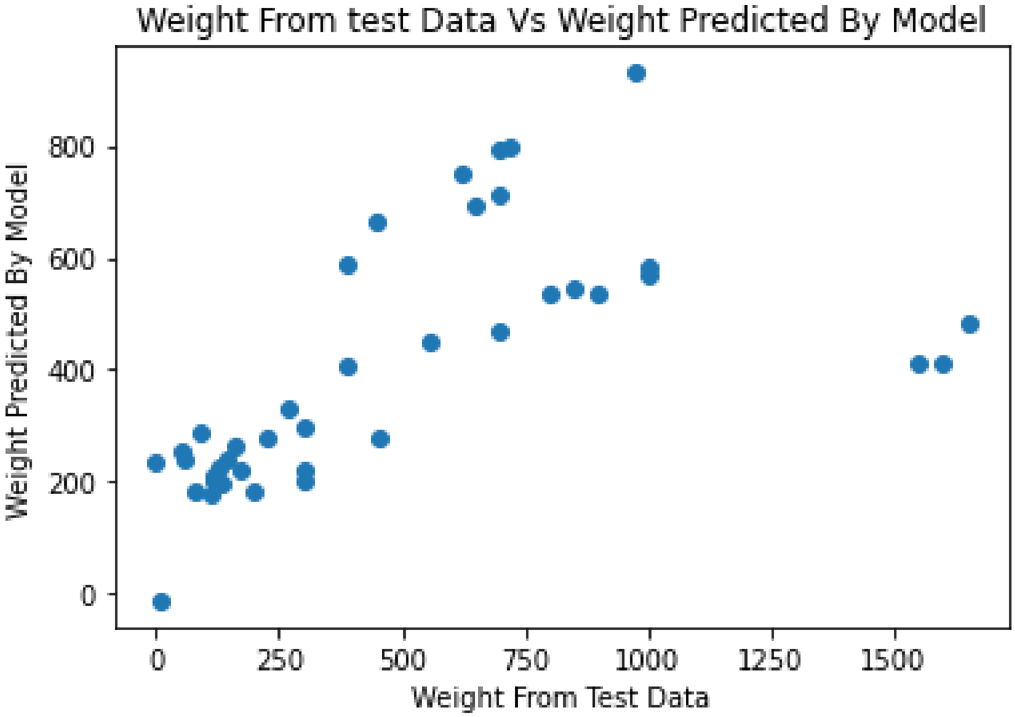
[3]:

```
plt.scatter(X_train,y_train,color="blue") # Plot a graph with X_train vs y_train
plt.plot(X_train,regressor.predict(X_train),color="red",linewidth=3) #␣
 ↪Regressior line showing
plt.title('Regression(Training Set)')
plt.xlabel('Height in cms')
plt.ylabel('Weight in gms')
plt.show()
```

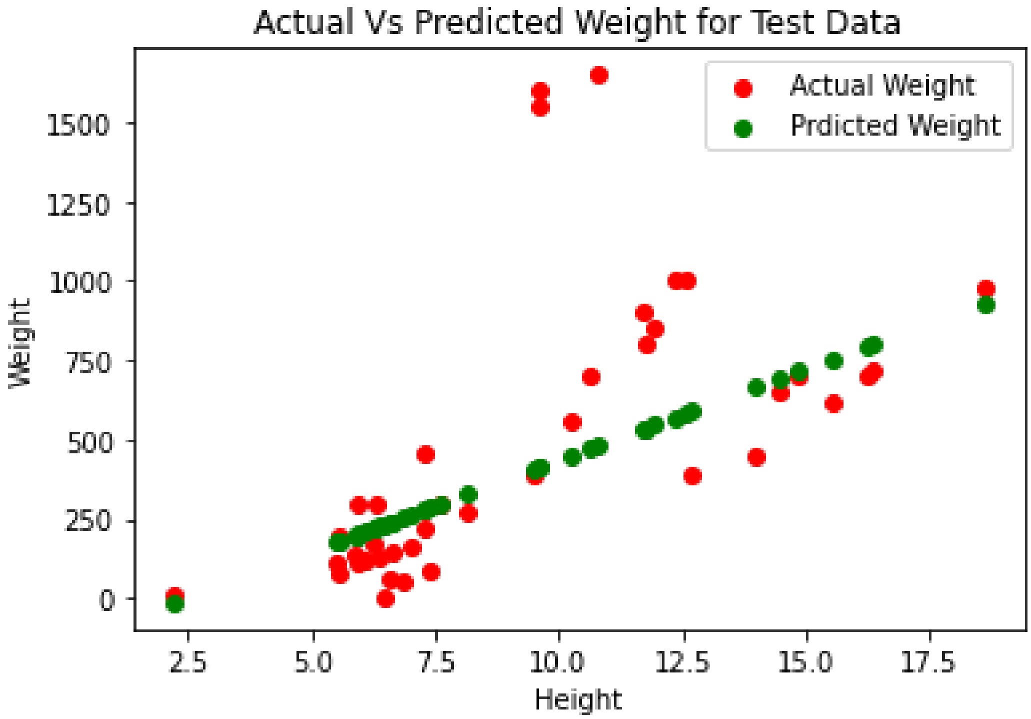


[4]:
```
y_pred = regressor.predict(X_test)
plt.scatter(y_test, y_pred)
plt.xlabel('Weight From Test Data')
plt.ylabel('Weight Predicted By Model')
plt.title("Weight From test Data Vs Weight Predicted By Model")
```

[4]:
```
Text(0.5, 1.0, 'Weight From test Data Vs Weight Predicted By Model')
```



[5]:
```
plt.scatter(X_test, y_test, color='red', label = 'Actual Weight')
plt.scatter(X_test, y_pred, color='green', label = 'Prdicted Weight')
plt.xlabel('Height')
plt.ylabel('Weight')
plt.title('Actual Vs Predicted Weight for Test Data')
plt.legend()
plt.show()
```



```
[6]: y_pred = regressor.predict(X_test)
     print('R2 score: %.2f' % r2_score(y_test,y_pred)) # Priniting R2 Score
     print('Mean Error :',mean_squared_error(y_test,y_pred)) # Priniting the mean␣
      ↪error
     def fish_weight(height): # A function to predict the fish weight according to␣
      ↪fish height
      result = regressor.predict(np.array(height).reshape(1, -1))
      return(result[0,0])
     fish_height = float(input('Enter Fish Height : '))
     print('The Weight of Fish will be : ',int(fish_weight(fish_height)),'gms')
```

```
R2 score: 0.32
Mean Error : 129936.44998018215
Enter Fish Height : 11
The Weight of Fish will be :  492 gms
```