Machine Learning Lab – Assignment 1

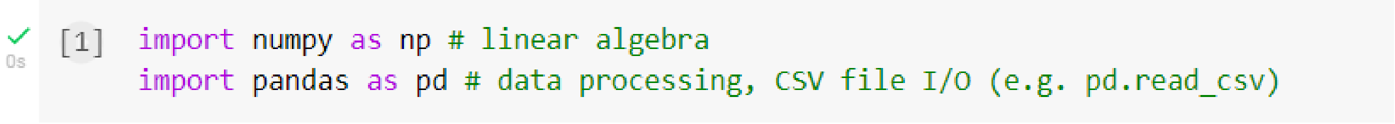
Mayank Jhanwar | Reg No. 189303179 | CCE 7C

Linear Regression Model

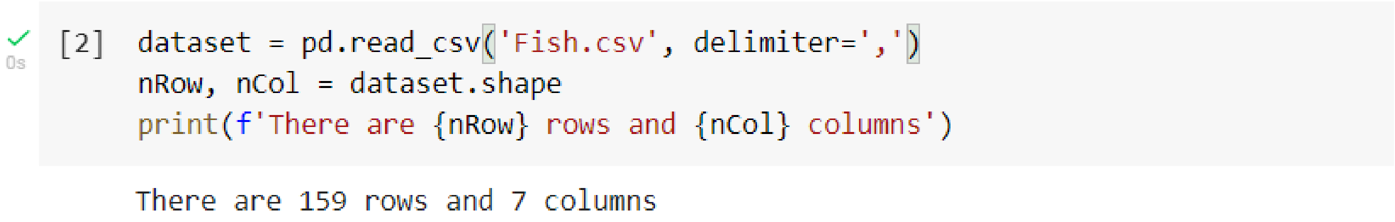
Dataset Taken: Fishweight.csv

1. This dataset contains 7 species of fish data for market sale.
2. There are 159 rows and 7 Columns in the dataset.

Importing Dataset:



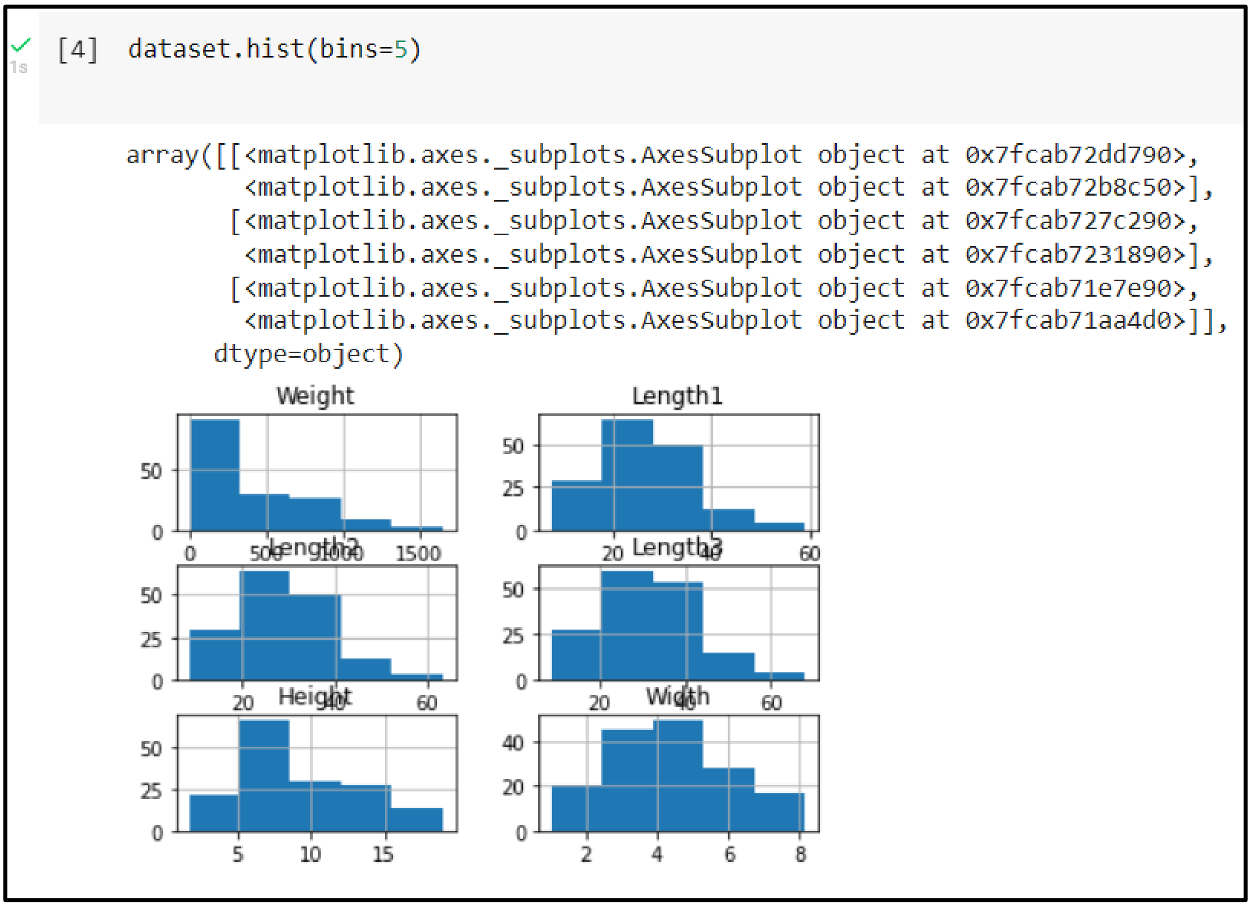
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
```



```
dataset = pd.read_csv('Fish.csv', delimiter=',')
nRow, nCol = dataset.shape
print(f'There are {nRow} rows and {nCol} columns')
```

```
There are 159 rows and 7 columns
```

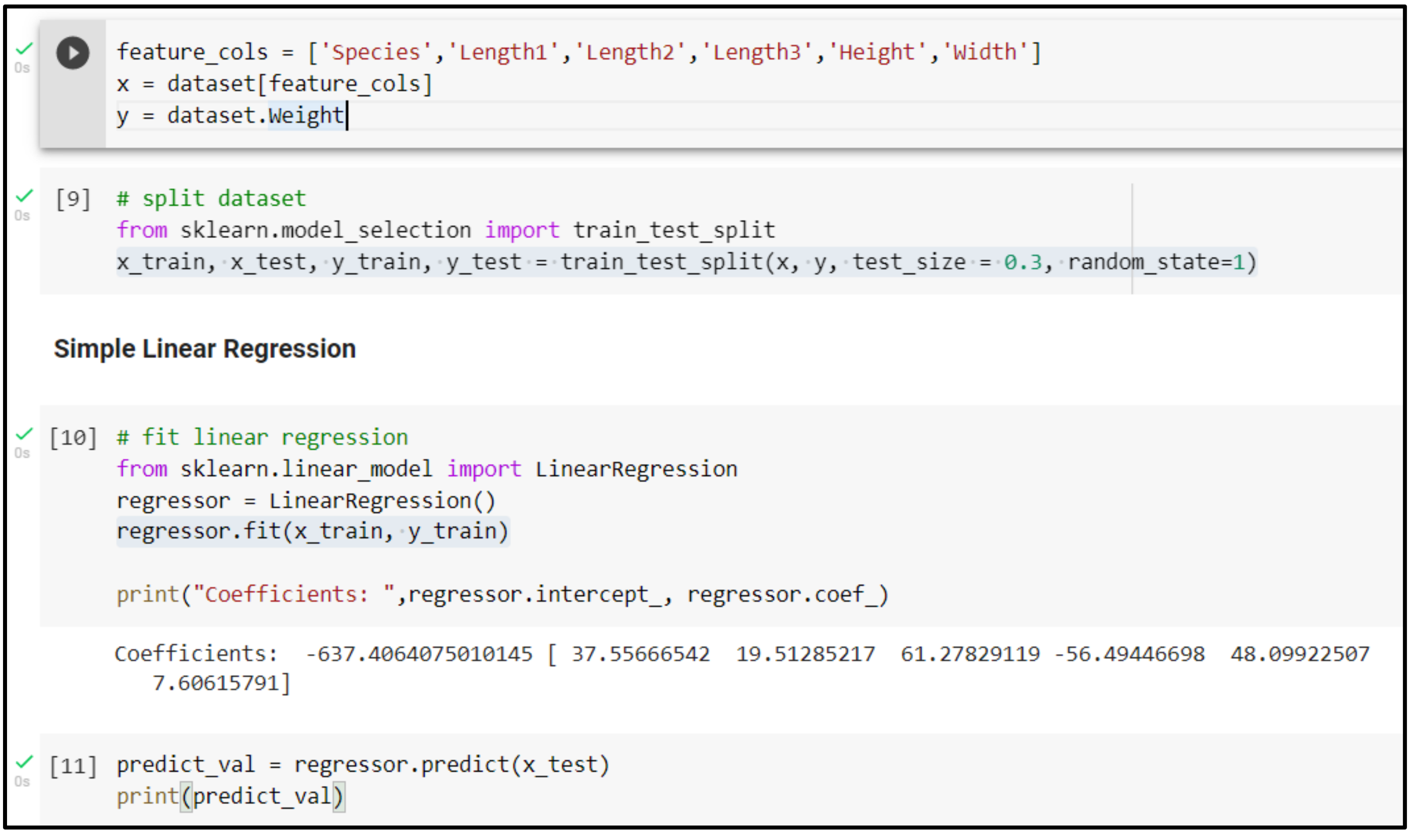
Plotting Histogram for the Dataset:



```
dataset.hist(bins=5)
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7fcab72dd790>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fcab72b8c50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fcab727c290>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fcab7231890>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7fcab71e7e90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7fcab71aa4d0>]],
      dtype=object)
```

Applying Linear Regression



```
feature_cols = ['Species','Length1','Length2','Length3','Height','Width']
x = dataset[feature_cols]
y = dataset.Weight
```

```
[9] # split dataset
    from sklearn.model_selection import train_test_split
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state=1)
```

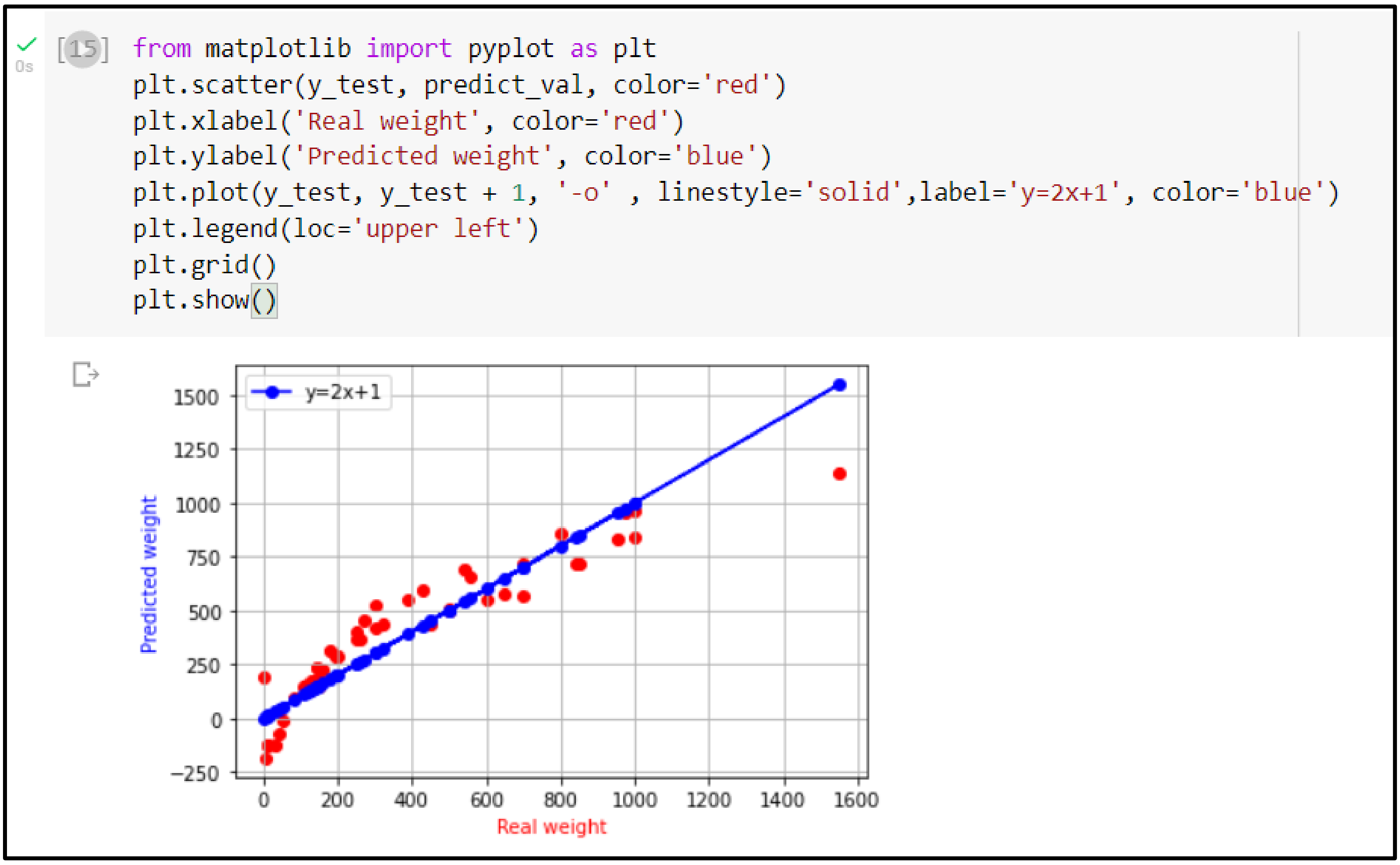
Simple Linear Regression

```
[10] # fit linear regression
     from sklearn.linear_model import LinearRegression
     regressor = LinearRegression()
     regressor.fit(x_train, y_train)

     print("Coefficients: ",regressor.intercept_, regressor.coef_)
```

```
Coefficients:  -637.4064075010145 [ 37.55666542  19.51285217  61.27829119 -56.49446698  48.09922507
   7.60615791]
```

```
[11] predict_val = regressor.predict(x_test)
     print(predict_val)
```

Plotting Results Graph



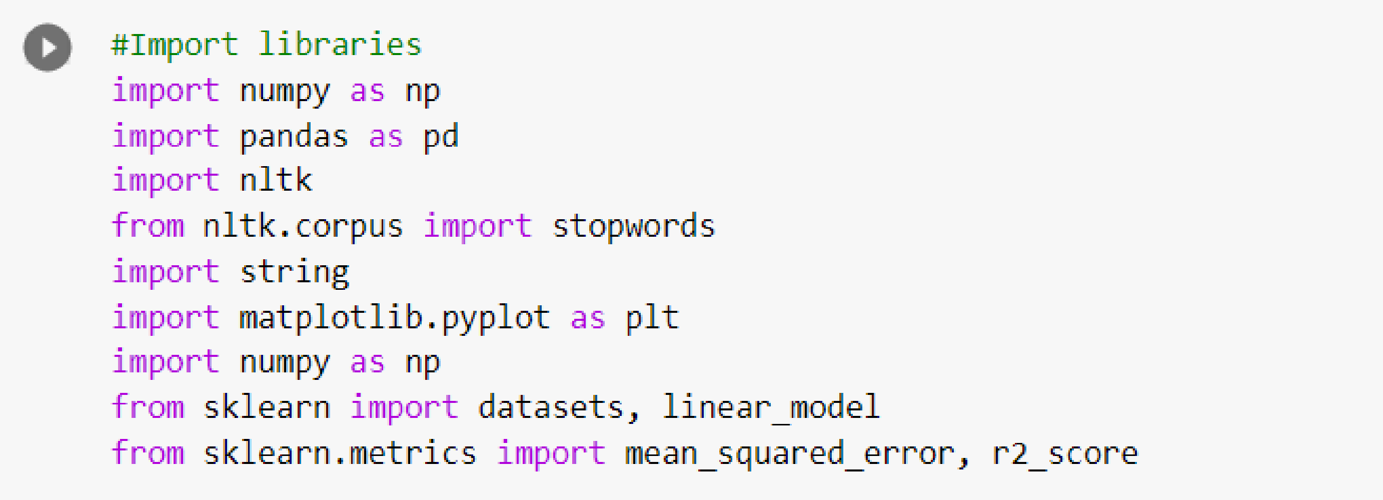
```
[15] from matplotlib import pyplot as plt
     plt.scatter(y_test, predict_val, color='red')
     plt.xlabel('Real weight', color='red')
     plt.ylabel('Predicted weight', color='blue')
     plt.plot(y_test, y_test + 1, '-o' , linestyle='solid',label='y=2x+1', color='blue')
     plt.legend(loc='upper left')
     plt.grid()
     plt.show()
```

Logistic Regression Model

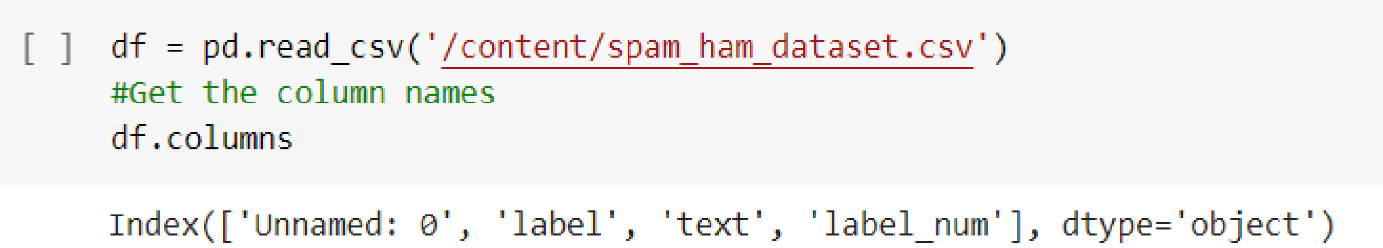
Dataset Taken: spam_ham_dataset.csv

This dataset contains a lot of spam and ham emails.

Importing Dataset:



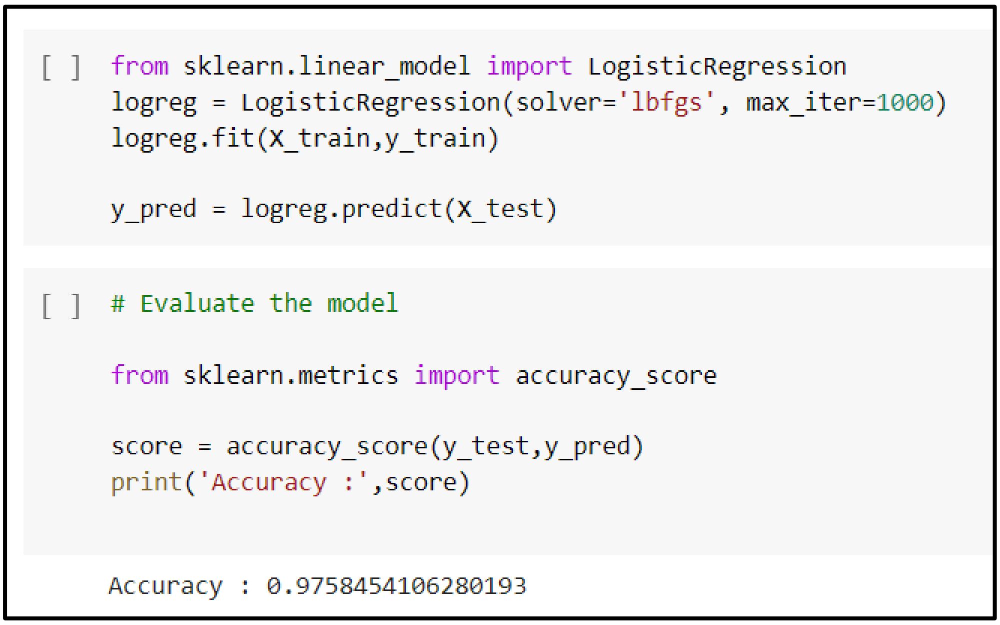
```
#Import libraries
import numpy as np
import pandas as pd
import nltk
from nltk.corpus import stopwords
import string
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
```



```
[ ]  df = pd.read_csv('/content/spam_ham_dataset.csv')
     #Get the column names
     df.columns
```

```
Index(['Unnamed: 0', 'label', 'text', 'label_num'], dtype='object')
```

Applying Logistic Regression



```
[ ]  from sklearn.linear_model import LogisticRegression
     logreg = LogisticRegression(solver='lbfgs', max_iter=1000)
     logreg.fit(X_train,y_train)

     y_pred = logreg.predict(X_test)
```

```
[ ]  # Evaluate the model

     from sklearn.metrics import accuracy_score

     score = accuracy_score(y_test,y_pred)
     print('Accuracy :',score)
```

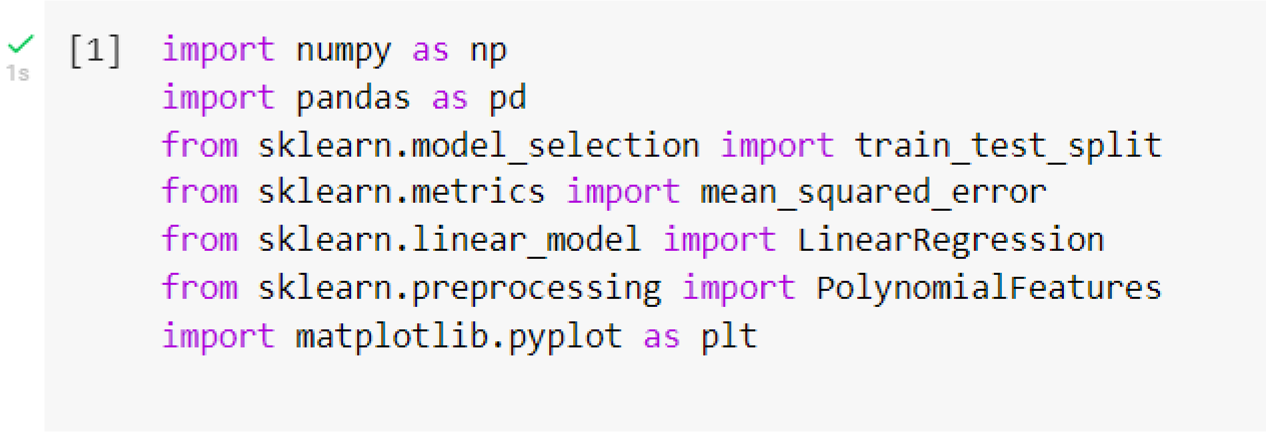
```
Accuracy : 0.9758454106280193
```

Polynomial Regression Model

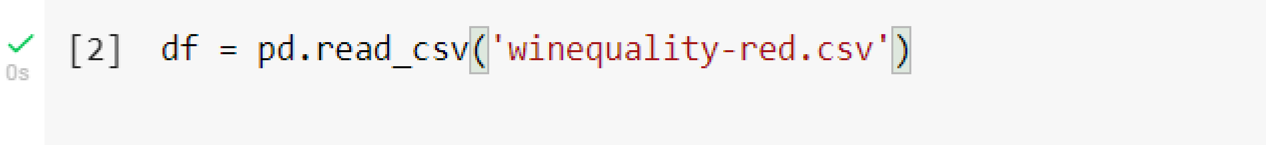
Dataset Taken: winequality_red.csv

The red variations of the Portuguese "Vinho Verde" wine are the subject of this dataset. We will use machine learning to determine which physiochemical properties make a wine 'good'!

Importing Dataset:

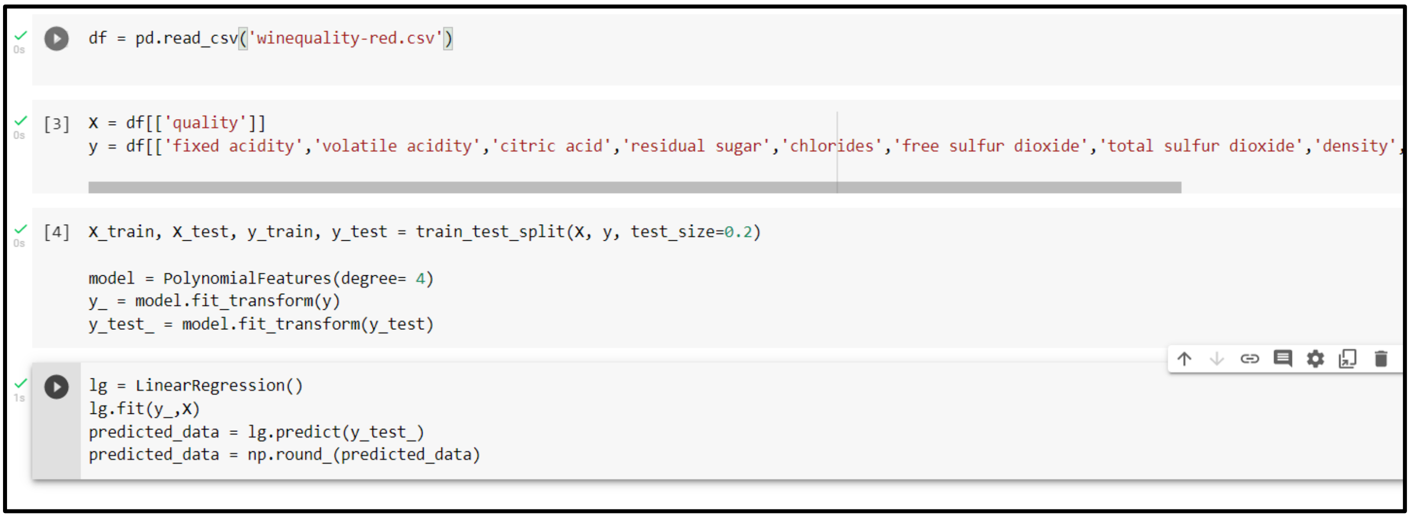


```
[1]  import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import mean_squared_error
     from sklearn.linear_model import LinearRegression
     from sklearn.preprocessing import PolynomialFeatures
     import matplotlib.pyplot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



```
[2]  df = pd.read_csv('winequality-red.csv')
```

Applying Polynomial Regression



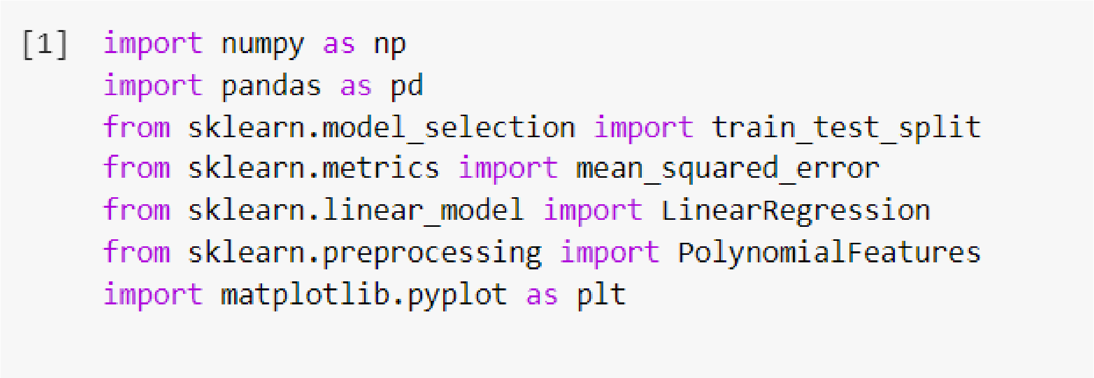
```
df = pd.read_csv('winequality-red.csv')

[3]  X = df[['quality']]
     y = df[['fixed acidity','volatile acidity','citric acid','residual sugar','chlorides','free sulfur dioxide','total sulfur dioxide','density',

[4]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

     model = PolynomialFeatures(degree= 4)
     y_ = model.fit_transform(y)
     y_test_ = model.fit_transform(y_test)

lg = LinearRegression()
lg.fit(y_,X)
predicted_data = lg.predict(y_test_)
predicted_data = np.round_(predicted_data)
```

Multiple Regression Model

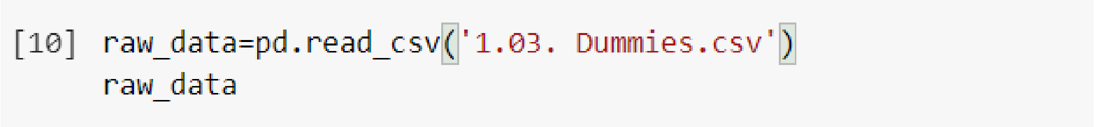
Dataset Taken: dummies.csv

Because GPA cannot be predicted solely by student as a score, but also by their High School GPA, Income, Gender etc. . If we want a good model, we need Multiple Regression, in order to address the higher complexity of problems

Importing Dataset:

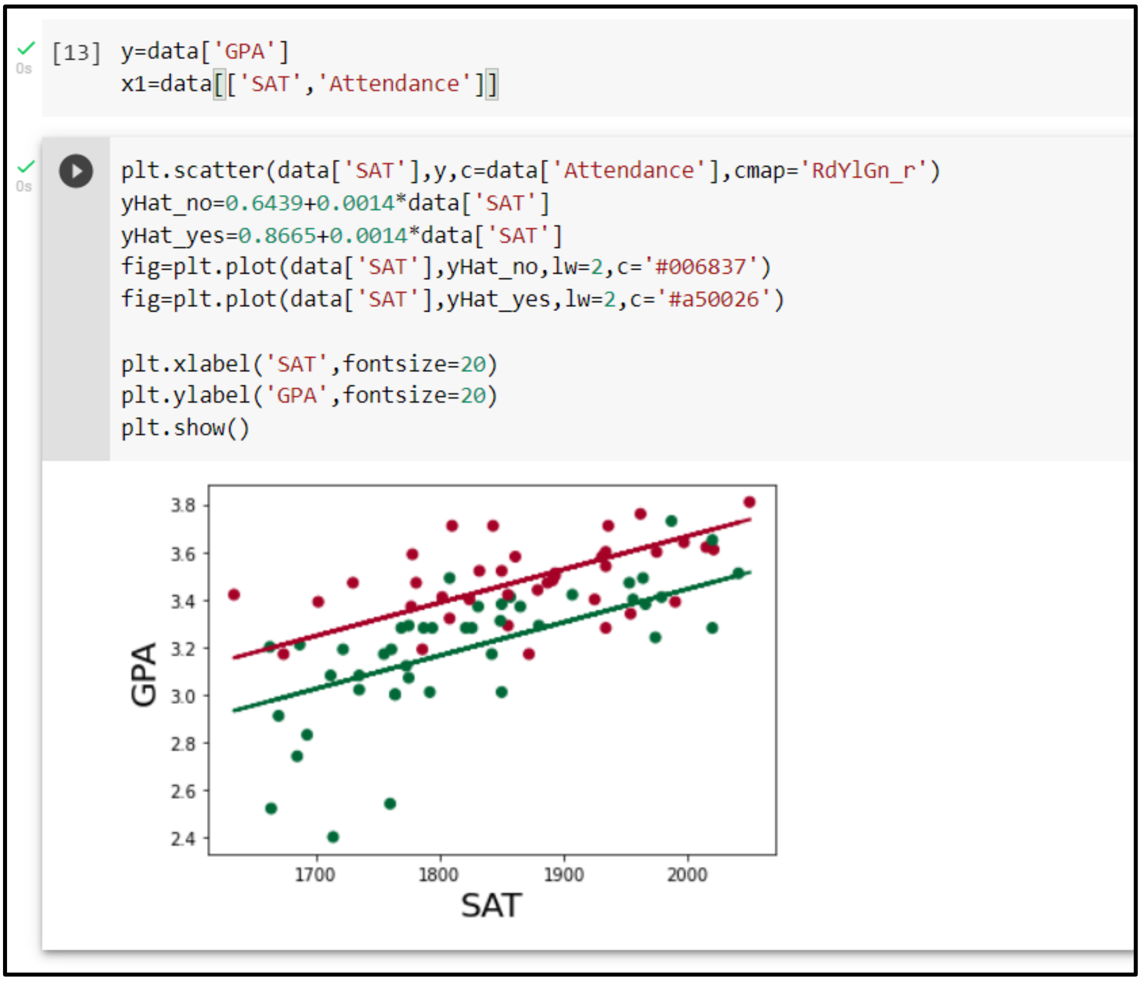


```
[1]  import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import mean_squared_error
     from sklearn.linear_model import LinearRegression
     from sklearn.preprocessing import PolynomialFeatures
     import matplotlib.pyplot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



```
[10] raw_data=pd.read_csv('1.03. Dummies.csv')
     raw_data
```

Applying Multiple Regression Model



```
[13] y=data['GPA']
     x1=data[['SAT','Attendance']]
```

```
plt.scatter(data['SAT'],y,c=data['Attendance'],cmap='RdYlGn_r')
yHat_no=0.6439+0.0014*data['SAT']
yHat_yes=0.8665+0.0014*data['SAT']
fig=plt.plot(data['SAT'],yHat_no,lw=2,c='#006837')
fig=plt.plot(data['SAT'],yHat_yes,lw=2,c='#a50026')

plt.xlabel('SAT',fontsize=20)
plt.ylabel('GPA',fontsize=20)
plt.show()
```