

Simple Linear Regression

In this application, a Simple Linear Regression model needs to be applied for the linear relationship between GPA and SAT Score

Import Libraries

```
In [2]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

Load Scores.csv Dataset into scores dataframe

```
In [4]: data = pd.read_csv("scores.csv")
data.head()
```

Out[4]:

	GPA	SAT Score
0	2.714	1480
1	3.418	1620
2	4.325	2070
3	2.552	1200
4	2.634	1300

Define X input and y output features with `iloc[:, :-1].values` for X and `iloc[:, -1]` for y

```
In [7]: X = data.iloc[:, :-1].values
y = data.iloc[:, -1].values
```

Split the dataset into Train and Test sets

```
In [8]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1/3, random_state=0)
```

Create the LinearRegression model object and call fit() to fit the model to training set

```
In [9]: lin_reg = LinearRegression()  
model = lin_reg.fit(X_train,y_train)
```

Predict the y with Test dataset using model object's predict()

```
In [10]: y_pred = model.predict(X_test)
```

Plot the scatter plot with X_test and y_test and also regression line with x as X_train and y as regressor.predict(X_train)

```
In [11]: mse = mean_squared_error(y_test, y_pred)  
rmse = np.sqrt(mse)  
print("RMSE Value: ", rmse)
```

RMSE Value: 198.5639232590927

Calculate and print Root Mean Square Error(RMSE)

```
In [13]: plt.scatter(X_test,y_test, label="Scatter Plot")  
y1 = model.predict(X_train)  
plt.plot(X_train, y1, color = 'black', linewidth=3, label = 'Regression Line')
```

Out[13]: [<matplotlib.lines.Line2D at 0x24aefc8fcf8>]

