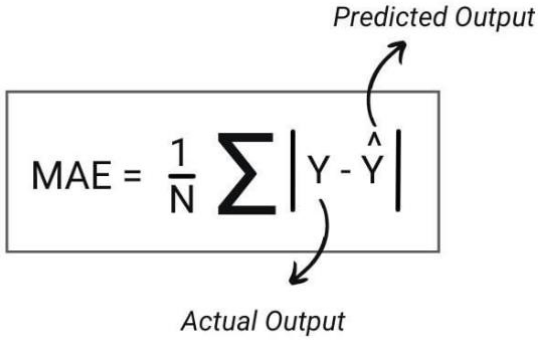
Regression Metrics

Mean Absolute Error

- MAE is a very simple metric which calculates the absolute difference between actual and predicted values.
- Let's take an example you have input data and output data and use Linear Regression, which draws a best-fit line.
- Now you have to find the MAE of your model which is basically a mistake made by the model known as an error.
- Now find the difference between the actual value and predicted value that is an absolute error but we have to find the mean absolute of the complete dataset.
- So, sum all the errors and divide them by a total number of observation and this is MAE. And we aim to get a minimum MAE because this is a loss.



Advantage of MAE

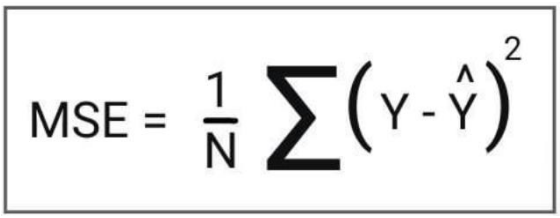
- The MAE you get is in the same unit as the output variable.
- It is most Robust to outliers.

Disadvantage of MAE

- The graph of MAE is not differentiable so we have to apply various optimizers like Gradient descent which can be differentiable.

Mean Square Error

- MSE is a most used and very simple metric with a little bit of change in mean absolute error.
- Mean squared error states that finding the squared difference between actual and predicted value.
- MSE represents the squared distance between actual and predicted values. We perform squared to avoid the cancellation of negative terms and it is the benefit of MSE.



Root Mean Square Error

- As RMSE is clear by the name itself, that it is a simple square root of mean squared error.

$$RMSE = \sqrt{\frac{1}{N}\sum\left(Y - \hat{Y}\right)}$$

Advantages of RMSE

- The output value you get is in the same unit as the required output variable which makes interpretation of loss easy.

Disadvantages of RMSE

- It is not that robust to outliers as compared to MAE for performing RMSE we have to Numpy square root function over MSE.
- Most of the time people use RMSE as an evaluation metric and mostly when you are working with deep learning techniques the most preferred metric is RMSE.

R Squared (R2)

- R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.
- In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.
- With help of R squared we have a baseline model to compare a model which none of the other metric provides.
- The same we have in classification problems which we cam a threshold which is fixwd at 0.5.
- So basically R2 squared calculates how much regression line is better than mean line.
- R2 squared is also known as Coefficant of Determination or sometimes also known as Goodness of fit.

$$\textbf{R2 Squared} = 1 - \frac{SSr}{SSm}$$

SSr = Squared sum error of regression line

SSm = Squared sum error of mean line

How will you interpret the R2 score?

- Suppose if the R2 score is zero then the above regression line by mean line is equal means 1 so 1-1 is zero.
- So, in this case, both lines are overlapping means model performance is worst, It is not capable to take advantage of the output column.
- Now the second case is when the R2 score is 1, which means when the division term is zero and it will happen when the regression line does not make any mistake, it is perfect. In the real world, it is not possible.
- So we can conclude that as our regression line moves towards perfection, the R2 score moves towards one. And the model performance improves.
- The normal case is when the R2 score is between zero and one like 0.8 which means your model is capable to explain 80% of the variance of data.

Adjusted R Squared

- The disadvantage of the R2 score is while adding new features in data the R2 score starts increasing or remains constant but it never decreases because It assumes that while adding more data variance of data increases.
- But the problem is when we add an irrelevant feature in the datasetthen at that time R2 sometimes starts increasing which is incorrect.
- Hence, to control this situation Adjusted R Squared came into existence.

$$R_a^2 = 1 - \left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2)\right]$$

where:
n = number of observations
k = number of independent variables
R_a^2 = adjusted R^2

- Now as K increases by adding some features so the denominator will decreases, n-1 will remain constant.
- R2 score will remain constant or will increases slightly so the complete answer will increase and when we subtract this form one then the resultant score will decrease. So this is the case when we add an irrelevant feature in the dataset.
- If we add a relevant feture then the R2 score will increase and 1-R2 will decreases heavily and the denominator will also decrease so the complete term decreases, and on subtracting from one the score increases.
- Hence, this metric becomes one of the most important metrics to use during the evaluation of the model.

MAE, MSE, RMSE, R2 Score, Adjusted R2 score

In [1]:
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
```

In [2]:
```
data = pd.read_csv("placements.txt",delimiter=",")
```

In [3]:
```
data.head()
```

Out[3]:

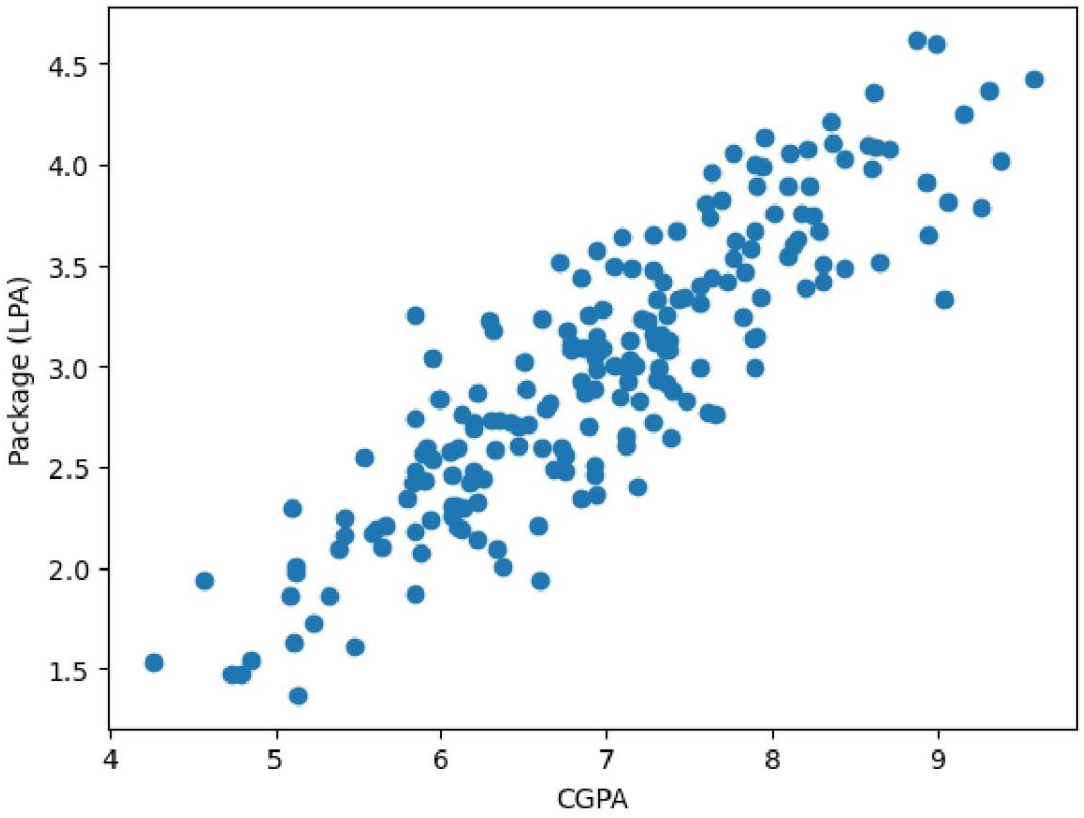
	cgpa	package
0	6.89	3.26
1	5.12	1.98
2	7.82	3.25
3	7.42	3.67
4	6.94	3.57

In [4]:
```
data.shape
```

Out[4]:
```
(200, 2)
```

In [5]:
```
plt.scatter(data['cgpa'],data['package'])
plt.xlabel('CGPA')
plt.ylabel('Package (LPA)')
```

Out[5]:
```
Text(0, 0.5, 'Package (LPA)')
```



In [6]:
```
x = data.iloc[:,0:1]
y = data.iloc[:,-1]
```

In [7]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=4)
```

```
C:\Users\hp\anaconda3\lib\site-packages\scipy\__init__.py:146: UserWarning: A NumP
y version >=1.16.5 and <1.23.0 is required for this version of SciPy (detected ver
sion 1.23.3
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [8]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

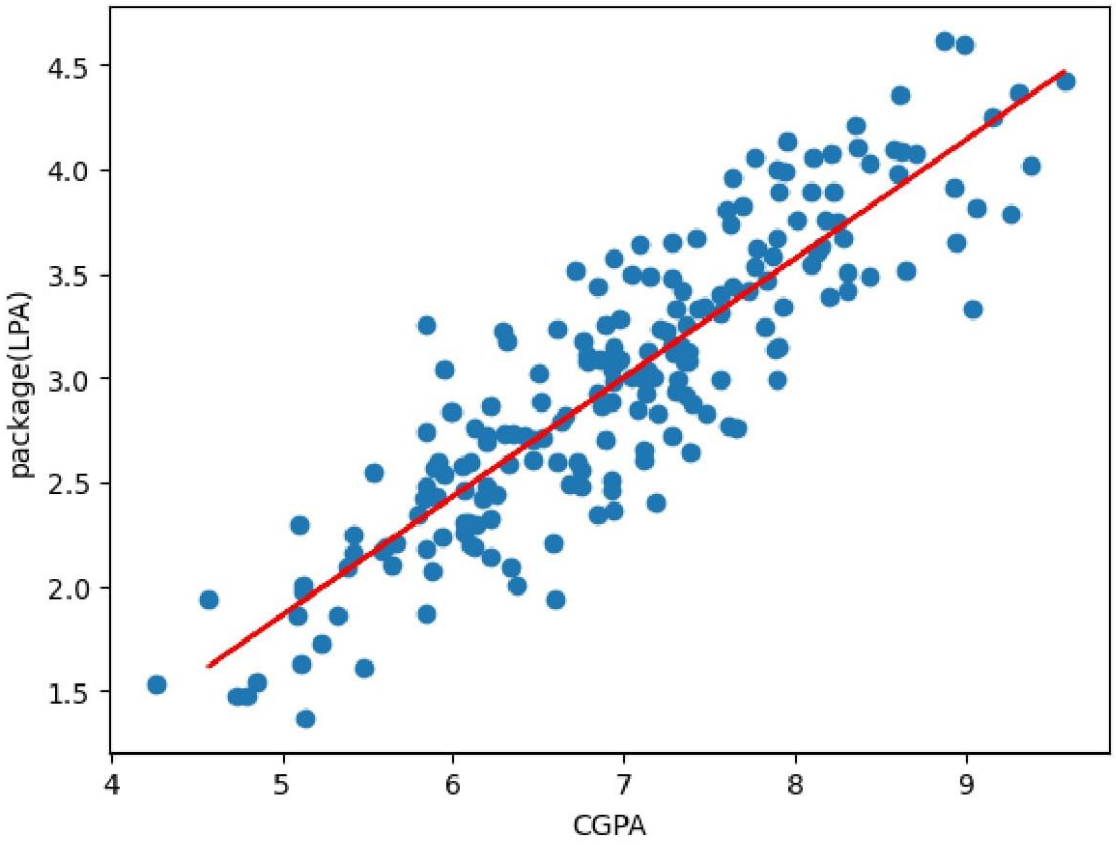
In [9]:
```
lr.fit(x,y)
```

Out[9]:
```
▾ LinearRegression

LinearRegression()
```

In [10]:
```
plt.scatter(data['cgpa'],data['package'])
plt.plot(x_train,lr.predict(x_train),c='r')
plt.xlabel('CGPA')
plt.ylabel('package(LPA)')
```

Out[10]:
```
Text(0, 0.5, 'package(LPA)')
```



In [11]:
```
from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

In [12]:
```
y_pre = lr.predict(x_test)
```

In [14]:
```
df = pd.DataFrame({'Test Value':y_test,'Predicted Value':y_pre})
df
```

	Test Value	Predicted Value
100	4.14	3.542573
83	3.49	3.821673
181	3.89	3.696362
71	3.34	3.269169

In [15]:
```
# Analyze the performance of the model by calculating mean squared error and R2

print('MAE:',mean_absolute_error(y_test,y_pre))

print('MSE:',mean_squared_error(y_test,y_pre))

print('RMSE:',np.sqrt(mean_squared_error(y_test,y_pre)))

print('R2 Score:',r2_score(y_test,y_pre))
```

```
MAE: 0.26913865091404976
MSE: 0.11320931085481958
RMSE: 0.3364659133624379
R2 Score: 0.8001010698130744
```

In [16]:
```
x_test.shape
```

Out[16]:
```
(40, 1)
```

In [17]:
```
# Adjusted R2 score
x_test.shape
1 - ((1-r2_score(y_test,y_pre))*(40-1)/(40-1-1))
```

Out[17]:
```
0.7948405716502606
```

In [18]:
```
# by formula and programing

import numpy as np
error = y_test - y_pre
se = np.sum(error**2)
print('Squared error is ', se)
n = np.size(x)
mse = se/n
print('Mean squared error is ', mse)

rmse = np.sqrt(mse)
print('Root mean square error is ', rmse)
ymean = np.mean(y_test)
SSt = np.sum((y_test - ymean)**2)
R2 = 1- (se/SSt)
print('R2 score is ', R2)
```

```
Squared error is  4.528372434192783
Mean squared error is  0.022641862170963915
Root mean square error is  0.15047213087799322
R2 score is  0.8001010698130744
```