

Linear Regression

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
X = np.array([[1],[2],[3],[4],[5]])
y = np.array([20,40,50,65,80])
```

```
model = LinearRegression()
model.fit(X,y)
```

```
▼ LinearRegression ⓘ ⓘ
LinearRegression()
```

```
y_pred = model.predict(X)
```

```
mae = mean_absolute_error(y,y_pred)
mse = mean_squared_error(y,y_pred)
rmse = np.sqrt(mse)
```

```
print("Mean Absoulte Error (MAE):",mae)
print("Mean Squared Error (MSE):",mse)
print("Root Mean Squared Error (RMSE):",rmse)
```

```
Mean Absoulte Error (MAE): 1.3999999999999992
Mean Squared Error (MSE): 3.5000000000000013
Root Mean Squared Error (RMSE): 1.870828693386971
```

Multiple Regression

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files | No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
import pandas as pd
df = pd.read_csv('Top 10 Test Cricket Batters Rankings.csv')
df
```

	rank	name	country	rating	points	trend	avg	difference
0	1	Joe Root	England	881	881	Flat	50.34	NaN
1	2	Kane Williamson	New Zealand	859	859	Flat	54.99	NaN
2	3	Daryl Mitchell	New Zealand	768	768	Flat	50.25	NaN
3	4	Harry Brook	England	758	758	Up	58.64	3.0
4	5	Steven Smith	Australia	757	757	Flat	56.97	NaN
5	6	Rohit Sharma	India	751	751	Flat	45.47	NaN
6	7	Yashasvi Jaiswal	India	740	740	Up	68.53	1.0
7	8	Virat Kohli	India	737	737	Up	49.16	2.0
8	9	Babar Azam	Pakistan	734	734	Down	45.06	-6.0
9	10	Mohammad Rizwan	Pakistan	728	728	Up	44.83	7.0

```
model = LinearRegression()
model.fit(df[['rating']], df['rank'])
```

▼ `LinearRegression` ⓘ ⓘ
LinearRegression()

```
y_pred = model.predict(df[['rating']])
```

```
mae = mean_absolute_error(df['rank'],y_pred)
mse = mean_squared_error(df['rank'],y_pred)
rmse = np.sqrt(mse)
```

```
print("Mean Absoulte Error (MAE):",mae)
print("Mean Squared Error (MSE):",mse)
print("Root Mean Squared Error (RMSE):",rmse)
```

```
Mean Absoulte Error (MAE): 1.2867058927793906
Mean Squared Error (MSE): 2.3871902466467043
Root Mean Squared Error (RMSE): 1.5450534769536957
```