

```
In [2]: 1 import pandas as pd
        2 df = pd.read_csv("Student.csv")
        3 df.head()
```

Out[2]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

```
In [3]: 1 df.tail()
```

Out[3]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
395	396	324	110	3	3.5	3.5	9.04	1	0.82
396	397	325	107	3	3.0	3.5	9.11	1	0.84
397	398	330	116	4	5.0	4.5	9.45	1	0.91
398	399	312	103	3	3.5	4.0	8.78	0	0.67
399	400	333	117	4	5.0	4.0	9.66	1	0.95

```
In [4]: 1 df.drop("Serial No.",axis=1,inplace=True)
        2 y = df['Chance of Admit ']
        3 df.drop("Chance of Admit ",axis=1,inplace=True)
```

In [5]:

1 df

Out[5]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	337	118	4	4.5	4.5	9.65	1
1	324	107	4	4.0	4.5	8.87	1
2	316	104	3	3.0	3.5	8.00	1
3	322	110	3	3.5	2.5	8.67	1
4	314	103	2	2.0	3.0	8.21	0
...
395	324	110	3	3.5	3.5	9.04	1
396	325	107	3	3.0	3.5	9.11	1
397	330	116	4	5.0	4.5	9.45	1
398	312	103	3	3.5	4.0	8.78	0
399	333	117	4	5.0	4.0	9.66	1

400 rows × 7 columns

In [6]:

```
1 from sklearn.model_selection import train_test_split
2 x_train,x_test,y_train,y_test = train_test_split(df,y,test_size=0.2)
```

In [7]:

```
1 from sklearn.linear_model import LinearRegression
2 lr = LinearRegression()
3 lr.fit(x_train,y_train)
4 pred = lr.predict(x_test)
```

In [9]:

```
1 import numpy as np
2 from sklearn import metrics
3 rmse = np.sqrt(metrics.mean_squared_error(y_test,pred))
4 rmse
```

Out[9]: 0.06429067993817045

```
In [10]: 1 from sklearn.preprocessing import StandardScaler
2         sc = StandardScaler()
3         sc.fit(df)
4         df = sc.transform(df)
5         from sklearn.model_selection import train_test_split
6         x_train,x_test,y_train,y_test = train_test_split(df,y,test_size=0.2)
```

```
In [11]: 1 from sklearn.linear_model import LinearRegression
2         lr = LinearRegression()
3         lr.fit(x_train,y_train)
4         pred = lr.predict(x_test)
```

```
In [12]: 1 from sklearn import metrics
2         rmse = np.sqrt(metrics.mean_squared_error(y_test,pred))
3         rmse
```

Out[12]: 0.0582904925571127

```
In [ ]: 1
```