



## Day-12 | Exam mark prediction using Linear Regression-multipleVariable

### Import Libraries

```
import pandas as pd
from sklearn.linear_model import LinearRegression
```

### Load Dataset from Local Directory

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

0s completed at 19:17



+ Code + Text

RAM Disk Editing

Load Dataset

```
[3] dataset = pd.read_csv('data.csv')
```

Load Summarize

```
print(dataset.shape)
print(dataset.head(5))
```

	(201, 4)
	hours age internet marks
0	6.83 15 1 78.50
1	6.56 16 0 76.74
2	NaN 17 1 78.68
3	5.67 18 0 71.82
4	8.67 19 1 84.19

0s completed at 19:17



### Finding & Removing NA values from our Features X

```
[5] dataset.columns[dataset.isna().any()]  
  
Index(['hours'], dtype='object')
```

```
dataset.hours = dataset.hours.fillna(dataset.hours.mean())
```

### Segregate Dataset into Input X & Output Y

```
[ ] X = dataset.iloc[:, :-1].values  
    print(X.shape)  
    X
```

```
[ ] Y = dataset.iloc[:, -1].values
```

0s completed at 19:18



+ Code + Text

```
[7] [ 7.55, 16., 1., ],  
[ 6.35, 17., 1., ],  
[ 7.53, 18., 0., ],  
[ 8.56, 19., 1., ],  
[ 8.94, 20., 1., ],  
[ 6.6, 15., 1., ],  
[ 8.35, 16., 1., ],  
[ 4.15, 15., 0., ]])
```

```
Y = dataset.iloc[:, -1].values  
Y
```

### Training Dataset using Linear Regression

```
[ ] model = LinearRegression()  
model.fit(X,Y)
```

0s completed at 19:18



+ Code + Text

```
model = LinearRegression()  
[9] model.fit(X,Y)  
  
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

▼ Predicted Price for Land sq.Feet of custom values

```
a=[[9.2,20,0]]  
PredictedmodelResult = model.predict(a)  
print(PredictedmodelResult)  
  
[86.26599847]
```

