

### ---What is linear regression?---

--Linear regression is a linear approach to modeling the relationship between a scalar response and one or more explanatory variables. The case of one explanatory variable is called simple linear regression.

--It is the easiest algorithm among all machine learning algorithm.

--Its quite simple to implement.

### ----About Dataset----

Here is a simple dataset from kaggle which is Weights and Heights dataset tried to apply linear regression algorithm.

The data contains the variables

---Gender

---Height (Inches)

---Weight (Pound)

---Goal---

- Predict the height or weight of a person.

### Step 1--Data Preprocessing

```
In [1]: #Importing libraries.  
import numpy as np  
import pandas as pd
```

```
In [2]: #Read the datasets  
dataset=pd.read_csv("weight-height.csv")
```

```
In [3]: #Check the dataset  
dataset.head()
```

```
Out[3]:
```

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801

```
In [4]: #separating the dependent and independent values....  
#X-Independent variable  
X = dataset.iloc[:, 1:2].values  
print(X)
```

```
[[73.84701702]  
 [68.78190405]  
 [74.11010539]  
 ...
```

203571

1:2  
height  
weight

```
[63.86799221]  
[69.03424313]  
[61.94424588]]
```

```
In [5]: #y-dependent or target variable  
y = dataset.iloc[:, 2].values  
print(y)
```

```
[241.89356318 162.31047252 212.74085556 ... 128.47531878 163.85246135  
113.64910268]
```

## Step 2-Splitting the dataset into training and testing set

```
In [6]: 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=12)
```

## Step 3-Build a linear model

```
In [7]: #Creating linear regression model  
from sklearn.linear_model import LinearRegression  
regressor = LinearRegression()  
regressor.fit(X_train, y_train)
```

```
Out[7]: 

LinearRegression ⓘ ?

  
LinearRegression()
```

```
In [8]: #predicting the test set  
y_pred = regressor.predict(X_test)
```

## Step 5-Checking the accuracy

```
In [9]: # Mean absolute error  
from sklearn.metrics import mean_absolute_error  
print("Mean absolute error: %.2f" % mean_absolute_error(y_test, y_pred))
```

```
Mean absolute error: 9.69
```

```
In [10]: # Mean squared error  
from sklearn.metrics import mean_squared_error  
print("Mean squared error: %.2f" % mean_squared_error(y_test, y_pred))
```

```
Mean squared error: 143.23
```

```
In [ ]:
```