Regression

- The term regression is used, when trying to find the relationship between variables.
- In ML, and in statistical modeling, that relationship is used to predict the outcome of future events.

Linear Regression

- LR uses the relationship between the data-points to draw a straight line through all them.
- This line can be used to predict future values.

$$y = m * x + b$$

$$m = \frac{n\sum xy - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{\sum y - m\sum x}{n}$$

$$sse = \sum (y_{org} - y_{pred})^2$$

In [53]:
```
#Using Linear Regression - Supervised Machine Learning Algorithm

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [54]:
```
### Load Data

df = {"X" : [i for i in range(1, 8)],
      "Y" : [1.5, 3.8, 6.7, 9.0, 11.2, 13.6, 16.0]}
df = pd.DataFrame(data=df)
```

In [61]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(df['X'], df['Y'], test_size=0.30, random_state=51)
X_train
```

Out[61]:
```
5    6
0    1
1    2
6    7
Name: X, dtype: int64
```

In [65]:
```
X_train= np.array(X_train).reshape((-1,1))
X_test=np.array(X_test).reshape((-1, 1))
y_train=np.array(y_train).reshape((-1,1))
y_test=np.array(y_test).reshape((-1,1))
```

In [66]:
```
print('Shape of X_train = ', X_train.shape)
print('Shape of y_train = ', y_train.shape)
print('Shape of X_test = ', X_test.shape)
print('Shape of y_test = ', y_test.shape)
```

```
Shape of X_train =  (4, 1)
Shape of y_train =  (4, 1)
Shape of X_test =  (3, 1)
Shape of y_test =  (3, 1)
```

In [67]:
```
## Linear Regression - ML Model Training

from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, y_train)
```

Out[67]:
```
LinearRegression()
```

In [72]:
```
## Predict the value of Home and Test
lr.predict([X_test[0, :]])
lr.score(X_test, y_test)
```

Out[72]:
```
0.9763516902275321
```

In [34]:
```
np.array([5,15,25,35,45,55]).reshape((-1,1))
```

Out[34]:
```
array([[ 5],
       [15],
       [25],
       [35],
       [45],
       [55]])
```

In [74]:
```
lr.predict([X_test[0, :]])
y_pred = lr.predict(X_test)
y_pred
```

Out[74]:
```
array([[ 6.29807692],
       [11.15192308],
       [ 8.725     ]])
```

In [ ]:
```
y_pred.s
```

In [79]:
```
from sklearn.metrics import mean_squared_error

mse = mean_squared_error(y_test, y_pred)
mse
```

Out[79]:
```
0.07982618343195289
```

In [75]:
```
from sklearn.metrics import mean_absolute_error

mae = mean_absolute_error(y_test, y_pred)
mae
```

Out[75]:
```
0.24166666666666656
```

In [80]:
```
from sklearn.metrics import r2_score

r2 = np.sqrt(mean_squared_error(y_test, y_pred))
r2
```

Out[80]:
```
0.2825352782077893
```

In [ ]:
```

```