Supervised Learning - Regression

Agenda

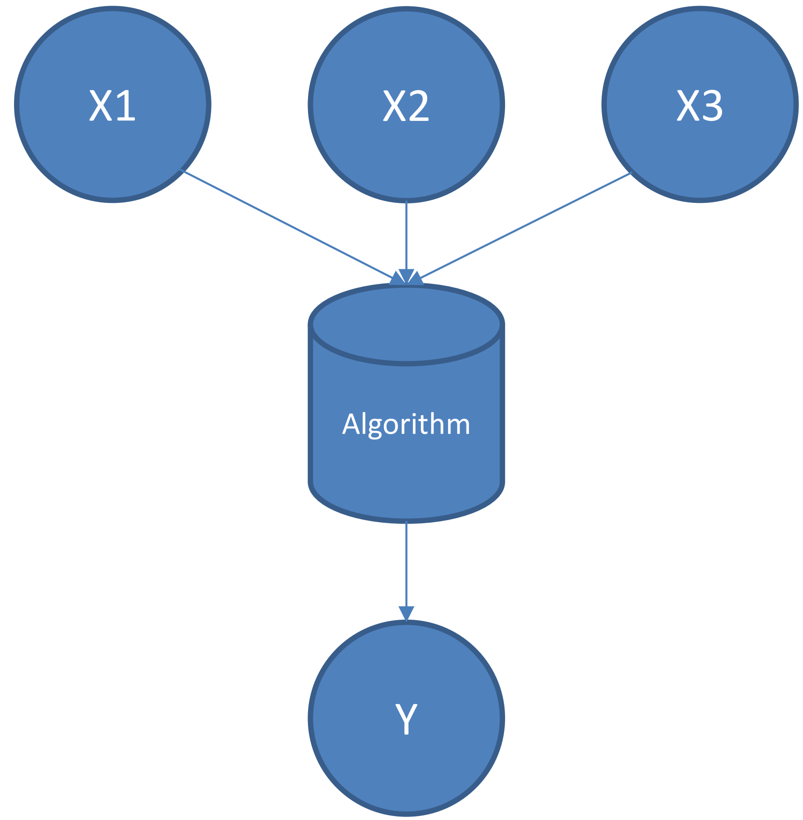
1. Simple Linear Regression
2. Multiple Linear Regression
3. Assumptions of Linear Regression
4. Polynomial Regression
5. R2 and RMSE

Regression: Definition

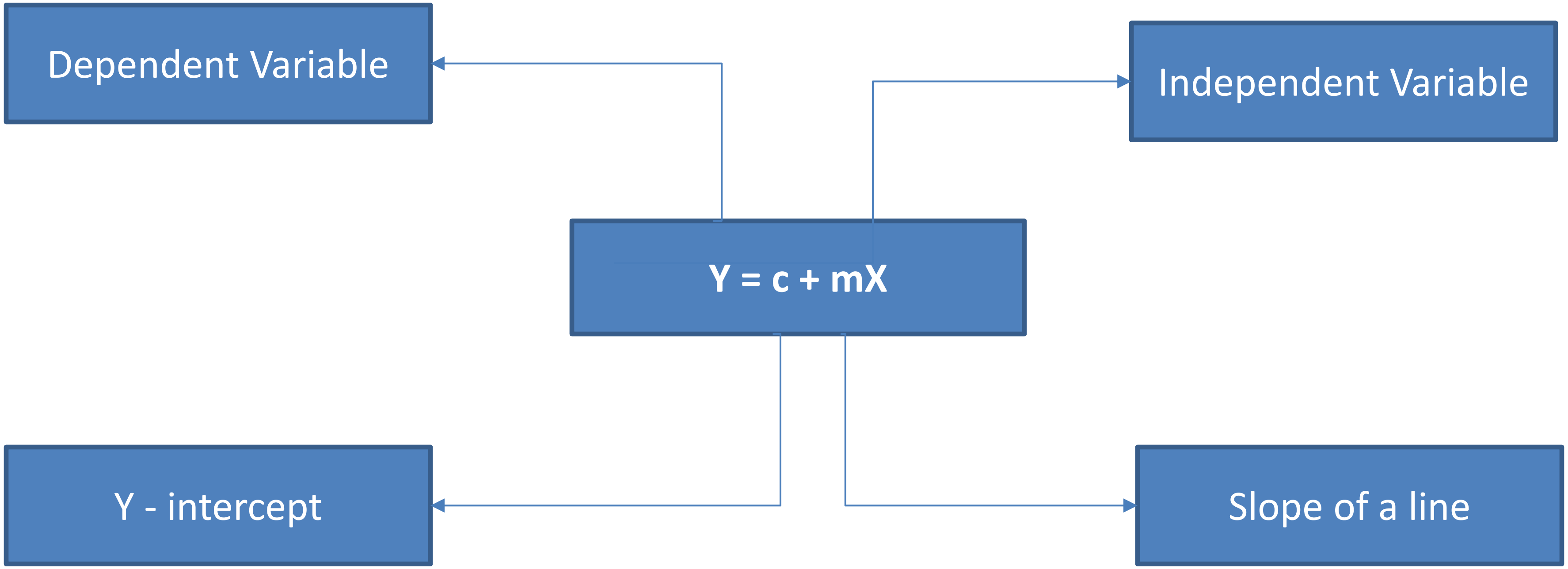
A technique for determining the statistical relationship between two or more variables where a change in a dependent variable is associated with a change in one or more independent variables.

Dependent Variable: The variable to be predicted or explained in a regression model.

Independent Variable: The variable related to the dependent variable in a regression equation.



Regression: Equation



Y – intercept (c) is that value of the dependent variable (y) when the value of the independent variable (x) is zero. It is the point at which the line cuts the y-axis.

Slope (b) is the change in the dependent variable for a unit increase in the dependent variable. It is the tangent of the angle made by the line with the x-axis.

Linear Regression

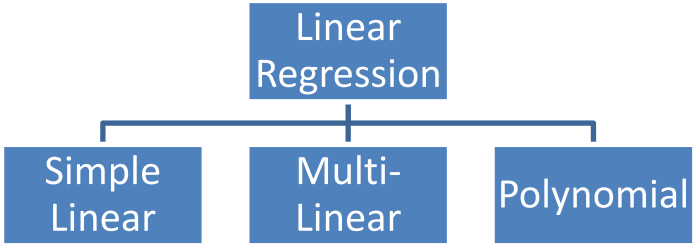
Simple Linear Regression is used to determine the strength of the relationship between one dependent variable and a series of other changing variables (independent).

Linear Regression: one independent variable to predict the outcome.

Multiple Regression: Two or more independent variables to predict the outcome.

Non-Linear Regression: When there is no linearity in the model, we use Non-Linear regression.

Polynomial Regression: “n” number of independent variables to predict the outcome. (basically, running multiple linear regression to fit non-linear data)

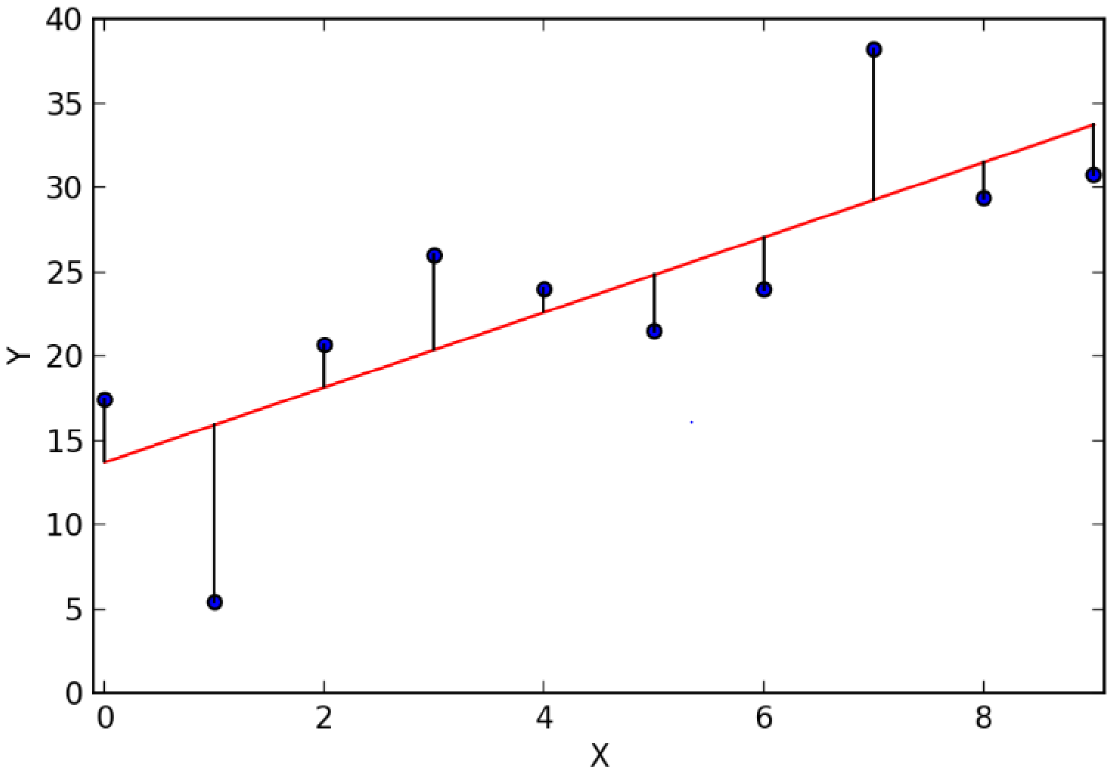


The Regression Line

The regression line is simply a single line that best fits the data (in terms of having the smallest overall distance from the line to the points)

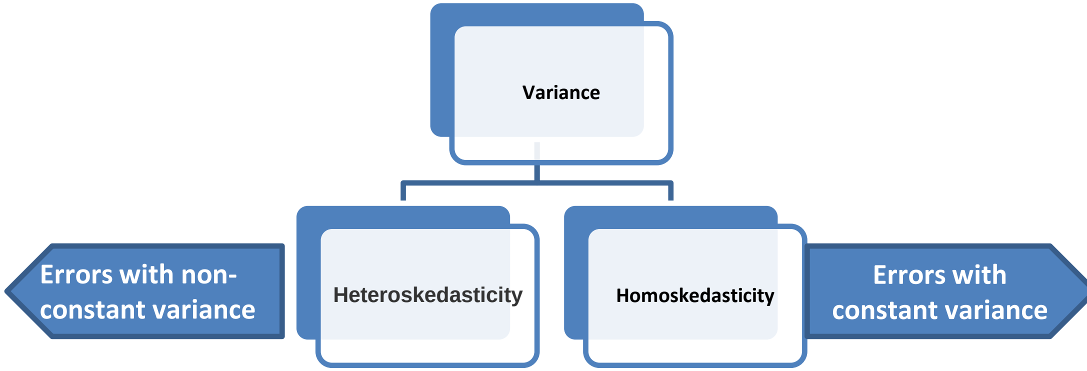
This technique is used for finding the "Best fitting line" using the "Least squares method".

- Dots are the fitted points
- Red line is the regression line
- Black line shows the deviation from the regression line



Assumptions of Linear Regression

1. There should be a linear and additive relationship between dependent variable and independent variable(s).
2. There should be no correlation between the residual (error) terms. Absence of this phenomenon is known as **Autocorrelation.**
3. The independent variables should not be correlated. Absence of this phenomenon is known as **multicollinearity.**
4. The error terms must have constant variance. This phenomenon is known as **homoskedasticity**. The presence of non-constant variance is referred to **hetroskedascity.**
5. The error terms must be normally distributed



Workaround for errors:

- In case of Multicollinearity, Ridge regression can be used.
- In case of correlation, Partial Least Square (PLS) Regression is being used.
- In case of overdispersion, negative binomial model can be used.

Regression: Few Definitions

Autocorrelation: Correlation between the residual value.

Presence of correlation makes the regression model weak. Let's understand this with the help of an example:

If the least square coefficient of X^1 is 15.02 and its standard error is 2.08 (without autocorrelation). But in presence of autocorrelation, the standard error reduces to 1.20. As a result, the prediction interval narrows down to (13.82, 16.22) from (12.94, 17.10).

Multicollinearity: If independent variables are correlated, the phenomenon is known as multicollinearity.

Presence of multicollinearity makes the:

1. Task of finding the true relationship between dependent and independent variables difficult.
2. Standard errors tend to increase.
3. The confidence interval becomes wider.
4. Less precise estimates of slope parameters.

User Case: Regression

A retail store manager wants to know that if he extends the shopping hours, will it greatly increase the sales or not. For this,

- Independent variable = sales (Y)
- Dependent variable = No. of shopping hours (X)
- Regression Equation: Y= b0 +b1*X1 + b2*X2+---bn*Xn

He can run the regression model to find the value of coefficients b0 & b1 and then calculate:

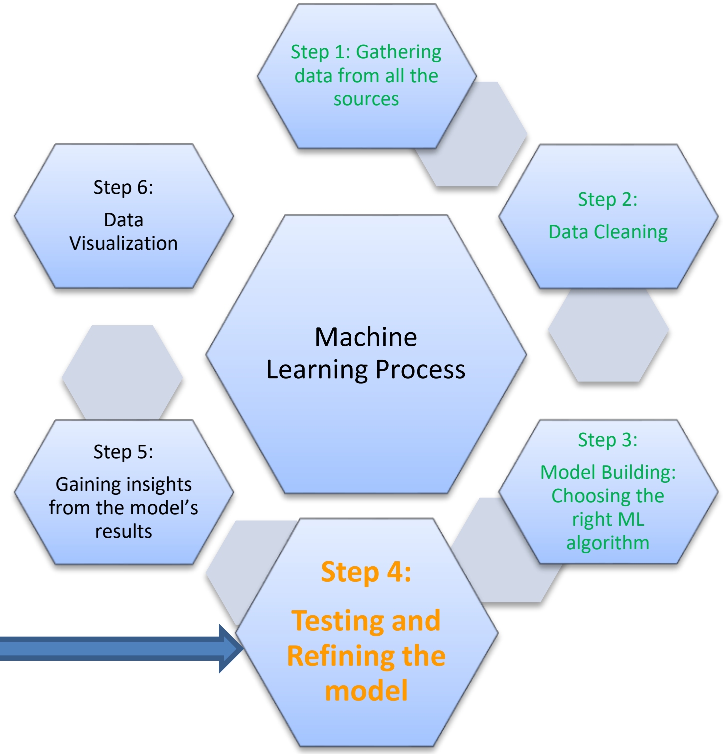
- what number of shopping hours are sufficient to reach the "Maximum" level of sales. (**simple linear)**
- what number of shopping hours are sufficient to reach the "Optimum" level of sales. (**Multi linear**) by adding additional parameter of operational expense. Because extending shopping hour would also increase the operational expenses such as electricity, employee salaries, beverages etc.
- Serving snacks would increase sales, impact can be calculated by **polynomial regression.**
- **Yi= (B0+B1xi+ B11xi^2) + ei**

.

Machine Learning Process: Testing Techniques

Ways to test Model-Fit:

- R-Squared (R^2)
- Overall F-Test
- Root Mean Squared Error (RMSE)



$$R^2 = \frac{N\sum xy - \sum x \sum y}{\sqrt{\left[N\sum x^2 - \left(\sum x\right)^2\right]\left[N\sum y^2 - \left(\sum y\right)^2\right]}}$$

Where,
N = No of scores given
$\sum$ XY = Sum of paired product
$\sum$ X = X score sum
$\sum$ Y = Y score sum
$\sum X^2$ = square of X score sum
$\sum Y^2$ = square of Y score sum

Example: Find the coefficient of determination

Dataset:

X	Y
2	2
5	5
6	4
7	3

N = 4

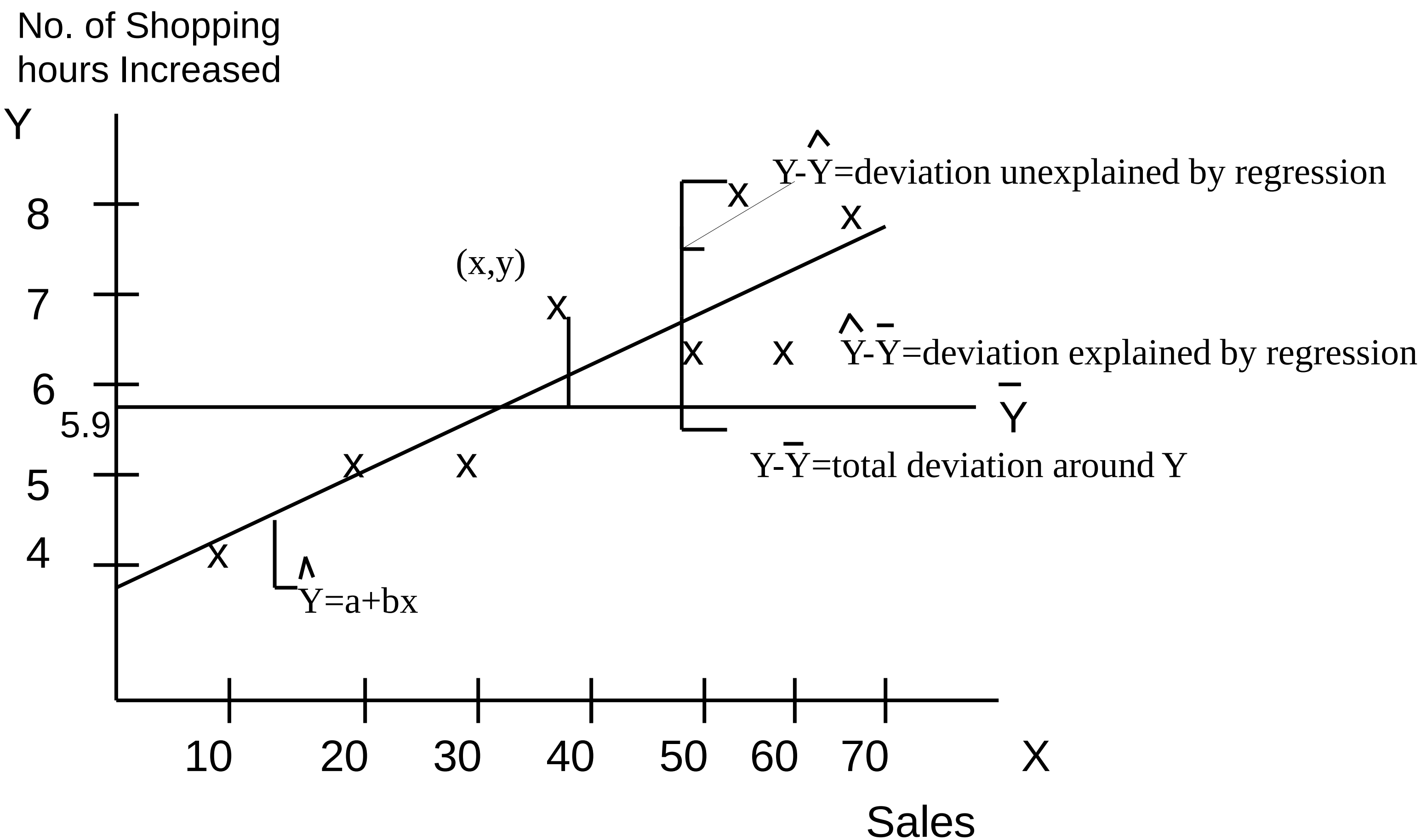
The coefficient of correlation is given by:

Create the table out of given scores

X	Y	XY	X^2	Y^2
2	2	4	4	4
5	5	10	25	25
6	4	24	36	16
7	3	21	49	9

R2=N∑xy−∑x∑y[N∑x2−(∑x)2][N∑y2−(∑y)2]√R2=N∑xy−∑x∑y[N∑x2−(∑x)2][N∑y2−(∑y)2]
= -12540 x 10976
= -0.003

R- Squared Explained



Total Deviation = Explained Deviation + Unexplained Deviation

The total distance from any point to Y. is the sum of the distance from Y to the regression line plus the distance from the regression line to Y.

F-Test and RMSE

The **F-test** tells you whether a group of variables, or even an entire model, is jointly significant in explaining variation in your dependent variable Y.

RMSE is measure of the size of the errors in regression and do not give indication about the explained component of the regression fit.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(X_{obs,i} - X_{model,i}\right)^2}{n}}$$

where Xobs is observed values and Xmodel is modelled values at time/place i.

- RMSE value can be used to compare the individual model performance to that of other predictive models.
- Lower values of RMSE indicate better fit.
- RMSE is a good measure of how accurately the model predicts the response.
- R-squared is a relative measure of fit, RMSE is an absolute measure of fit.
- The value of R2 is always between zero and one

Simple Linear Regression

Problem Statement: There is an electronics manufacturing company which sells an electronic cleaning device. Their sales team has advised to add another feature in the device to the product manager of the company. Now, before taking this further to the senior management, he decided to run simple regression model to find the impact of that feature on the net sales using market data.

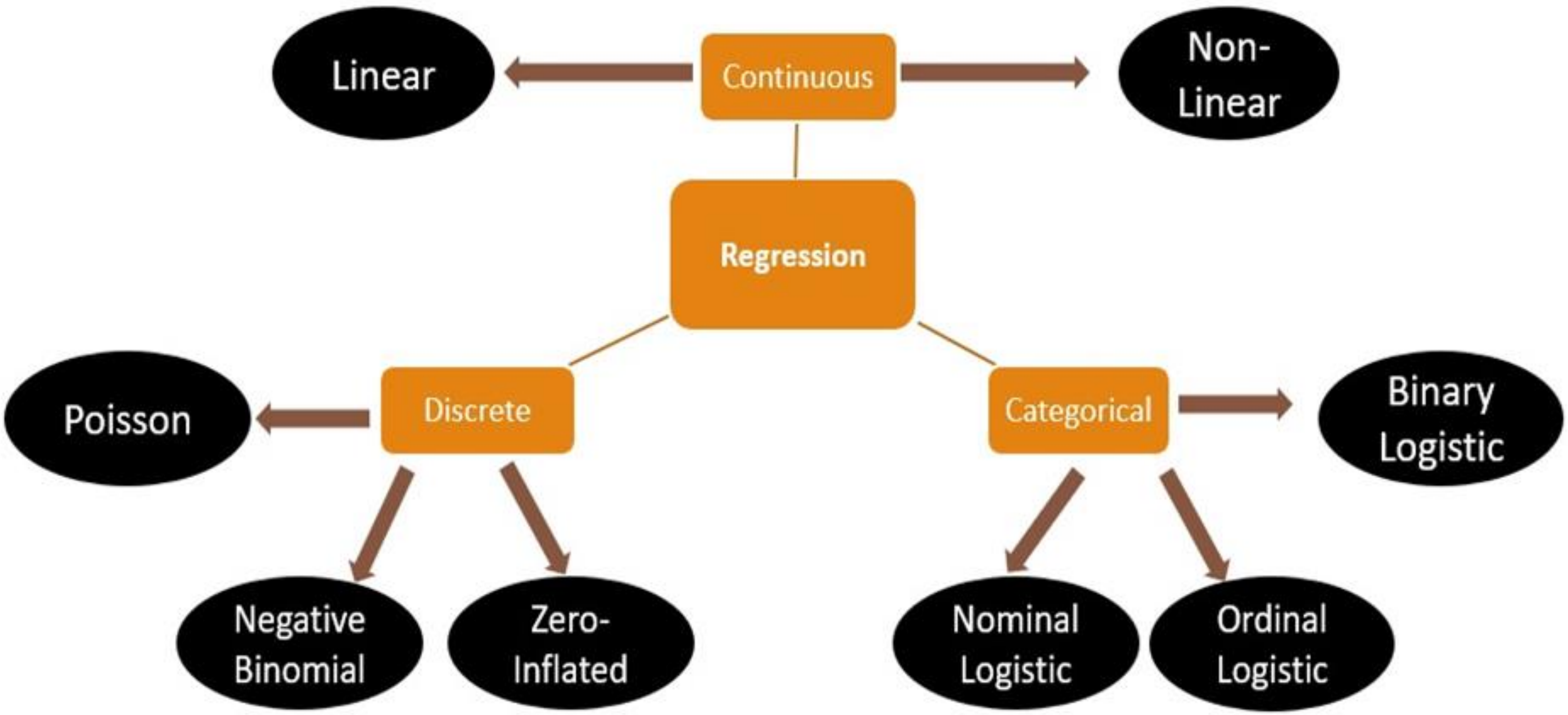
Equation of Simple Linear Regression:

$$Y = b0 + b1 * X$$

Y= Dependent Variable
X= Independent Variable
b0 & **b1** = Coefficients

- Once the coefficients are known, we can use this equation to estimate output values for y given new input examples of x.
- To calculate these coefficients we use regression model in python.

Regression: Summary



Types of Regression

Regression: Use case

A real estate company "ABC" has a new project coming up in which they have to build homes at different locations in California.

They have rough idea about prices but actual price is not decided yet. They want prices so that it will be affordable to common people.

Regression: Step 1

Import the data

Using Pandas library import the data in Python

```
import pandas as pd
import numpy as np
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv('train.csv')
```

dff - DataFrame

Index	OverallQual	GrLivArea	GarageCars	GarageArea	TotalBsmtSF	1stFlrSF	BsmtHalfBath	SaleCondition	GarageType	SalePrice
0	7	1710	2	548	856	856	0	Normal	Attchd	208500
1	6	1262	2	460	1262	1262	1	Normal	Attchd	181500
2	7	1786	2	608	920	920	0	Normal	Attchd	223500
3	7	1717	3	642	756	961	0	Abnorml	Detchd	140000
4	8	2198	3	836	1145	1145	0	Normal	Attchd	250000
5	5	1362	2	480	796	796	0	Normal	Attchd	143000
6	8	1694	2	636	1686	1694	0	Normal	Attchd	307000
7	7	2090	2	484	1107	1107	0	Normal	Attchd	200000
8	7	1774	2	468	952	1022	0	Abnorml	Detchd	129900
9	5	1077	1	205	991	1077	0	Normal	Attchd	118000
10	5	1040	1	384	1040	1040	0	Normal	Detchd	129500
11	9	2324	3	736	1175	1182	0	Partial	BuiltIn	345000
12	5	912	1	352	912	912	0	Normal	Detchd	144000
13	7	1494	3	840	1494	1494	0	Partial	Attchd	279500
14	6	1253	1	352	1253	1253	0	Normal	Attchd	157000
15	7	854	2	576	832	854	0	Normal	Detchd	132000
16	6	1004	2	480	1004	1004	0	Normal	Attchd	149000
17	4	1296	2	516	0	1296	0	Normal	CarPort	90000
18	5	1114	2	576	1114	1114	0	Normal	Detchd	159000
19	5	1339	1	294	1029	1339	0	Abnorml	Attchd	139000
20	8	2376	3	853	1158	1158	0	Partial	BuiltIn	325300
21	7	1108	1	280	637	1108	0	Normal	Attchd	139400
22	8	1795	2	534	1777	1795	0	Normal	Attchd	230000
23	5	1060	2	572	1040	1060	0	Normal	Attchd	129900
24	5	1060	1	270	1060	1060	0	Normal	Attchd	154000
25	8	1600	3	890	1566	1600	0	Normal	Attchd	256300
26	5	900	2	576	900	900	1	Normal	Detchd	134800
27	8	1704	3	772	1704	1704	0	Normal	Attchd	306000
28	5	1600	1	319	1484	1600	0	Normal	Attchd	207500
29	4	520	1	240	520	520	0	Normal	Detchd	68500
30	4	1317	1	250	649	649	0	Normal	Detchd	40000

Format Resize ☑ Background color ☑ Column min/max

Regression: Step 2.1

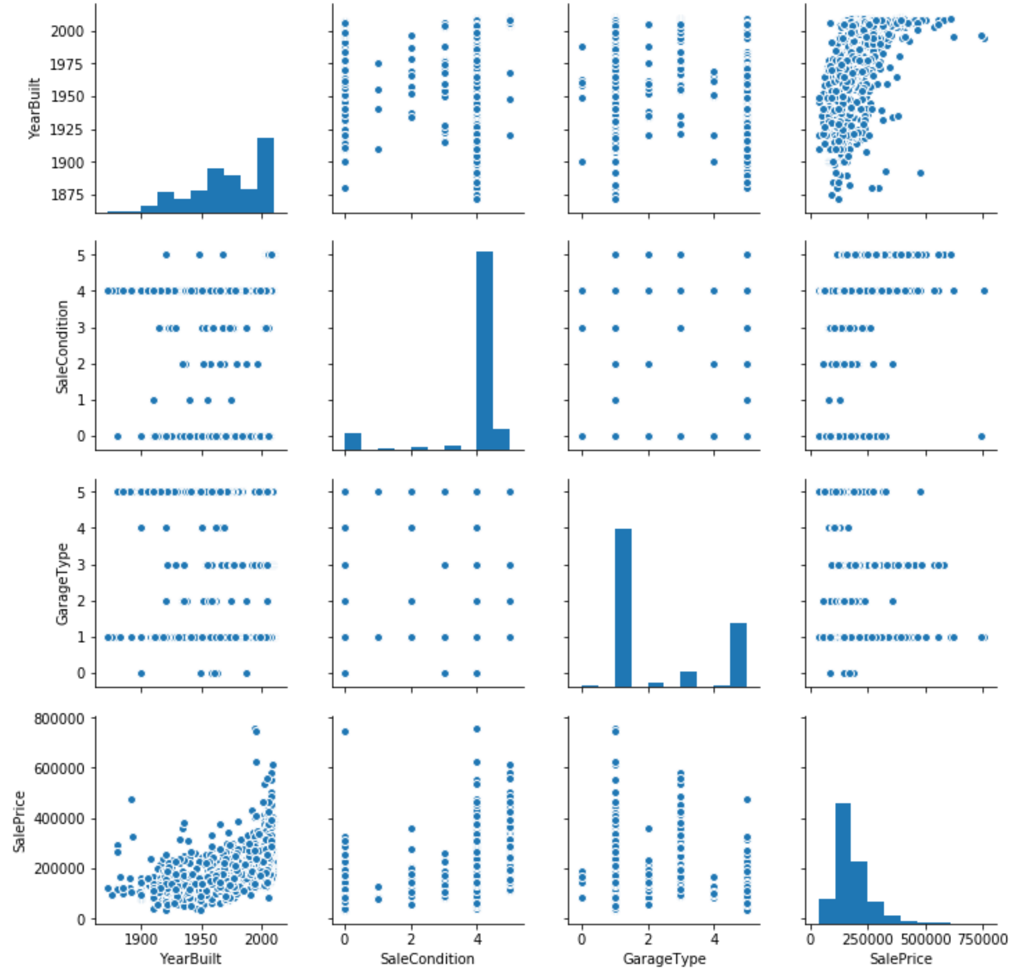
After importing the data we need to clean the data and start exploratory data analysis

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

dff=pd.read_csv('train.csv')

sns.pairplot(dff)
```

Pair plot is helpful to find the relation between the variables.



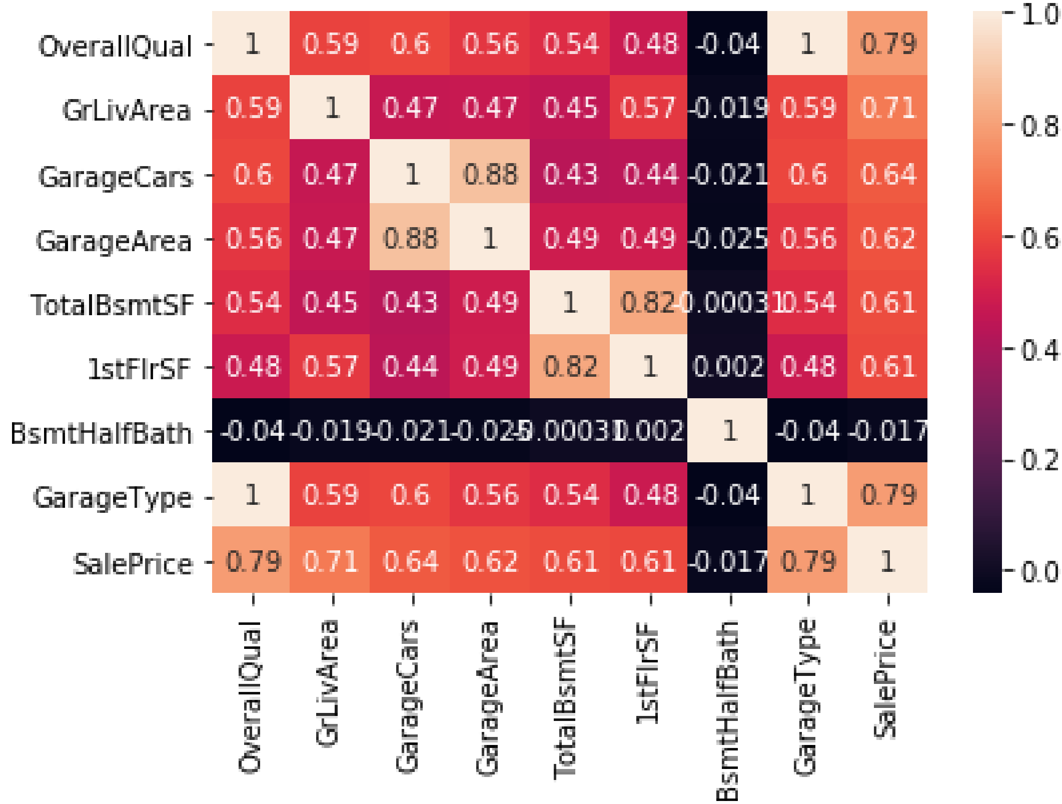
Regression: Step 2.2

After importing the data we need to clean the data and start exploratory data analysis

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns


dff=pd.read_csv('train.csv')

sns.heatmap(dff.corr(),annot=True)
```



Heatmap helps to find the correlation between the variables.

Regression: Step 2.3

During the Exploratory data analysis outliers will be taken case as per the business decisions.

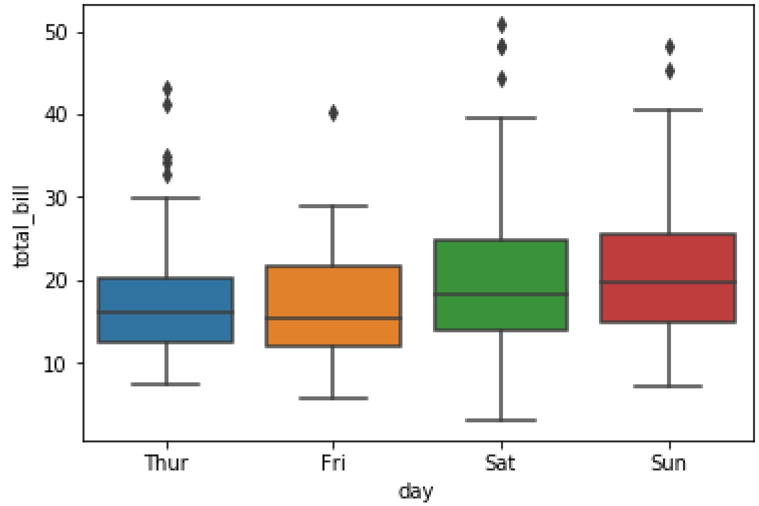
```
def remove_outlier(df_in, col_name):
    q1 = df_in[col_name].quantile(0.25)
    q3 = df_in[col_name].quantile(0.75)
    iqr = q3-q1
    fence_low = q1 - 1.5*iqr
    fence_high = q3 + 1.5*iqr
    df_out = df_in.loc[(df_in[col_name]> fence_low) & (df_in[col_name] < fence_high)]
    return df_out



df_in=FileNameDesc[(FileNameDesc.column_type=='int64') | (FileNameDesc.column_type =='float64')]



for col_na in df_in['column_name']:
    if col_na != IV:
        remove_outlier(dataset,col_na)
```



Regression: Step 3

Once exploratory data analysis done encoding will be the next step where the categorical data will encode to numbers as Python works on the numerical data.

```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categorical_features = [3])
X = onehotencoder.fit_transform(X).toarray()
```

Regression: Step 4

Next step after encoding is feature scaling where we need to bring all the dataset in the same scale so that no other variable will dominate any other variable.

There are two ways for doing the same i.e. Standardization and Normalization.

```
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train)
```

$$x_{new} = \frac{x - \mu}{\sigma}$$

```
from sklearn.preprocessing import Normalizer
Nm_X = Normalizer()
X_train = Nm_X.fit_transform(X_train)
X_test = Nm_X.transform(X_test)
```

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Regression: Step 5

Once feature scaling done the next step is to split the data into Test and Train as it is supervised learning.

```
# Splitting the dataset into the Training set and Test set
from sklearn.cross_validation import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
                                                    random_state = 0)
```

Regression: Step 6

Fit the model on train data and then use test data to check the accuracy of the model

```
#Multi Linear Regression
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Fit the train data on the model

```
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```

Use test data to check the accuracy of the model

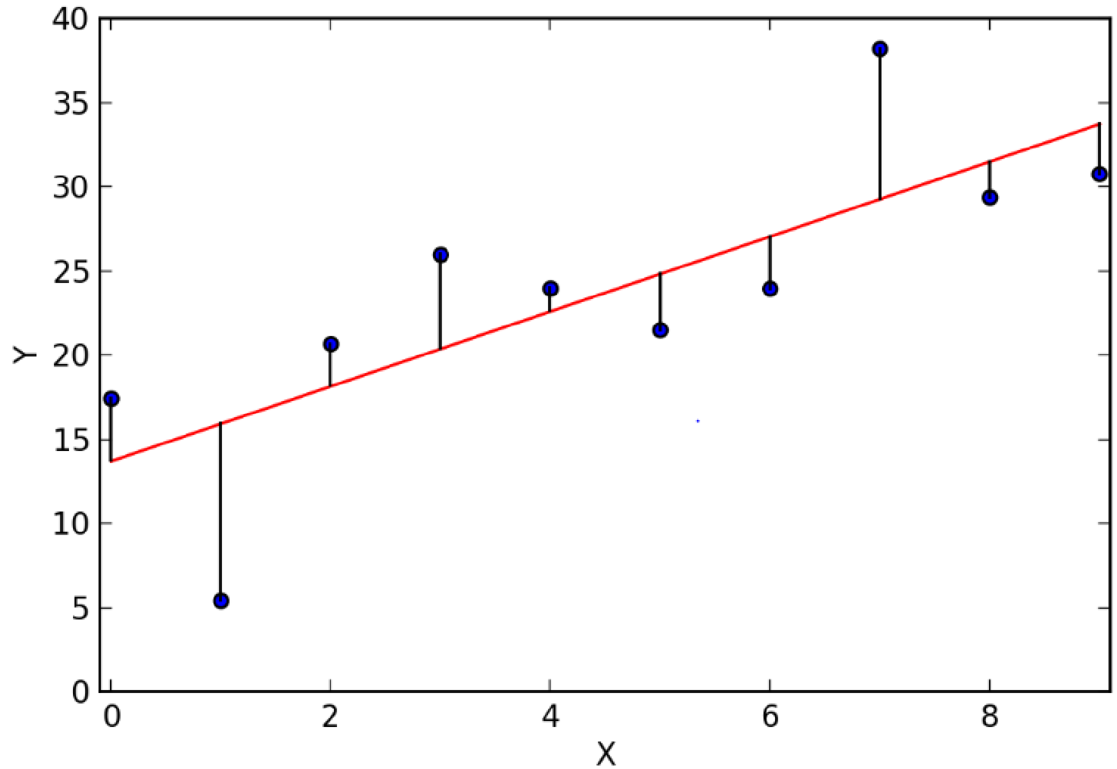
```
# Explained variance score: 1 is perfect prediction
r_square = r2_score(y_test, y_pred)
print('Variance score: {0:.2f}'.format(r_square))
```

R square value will be used to measure the accuracy of the model

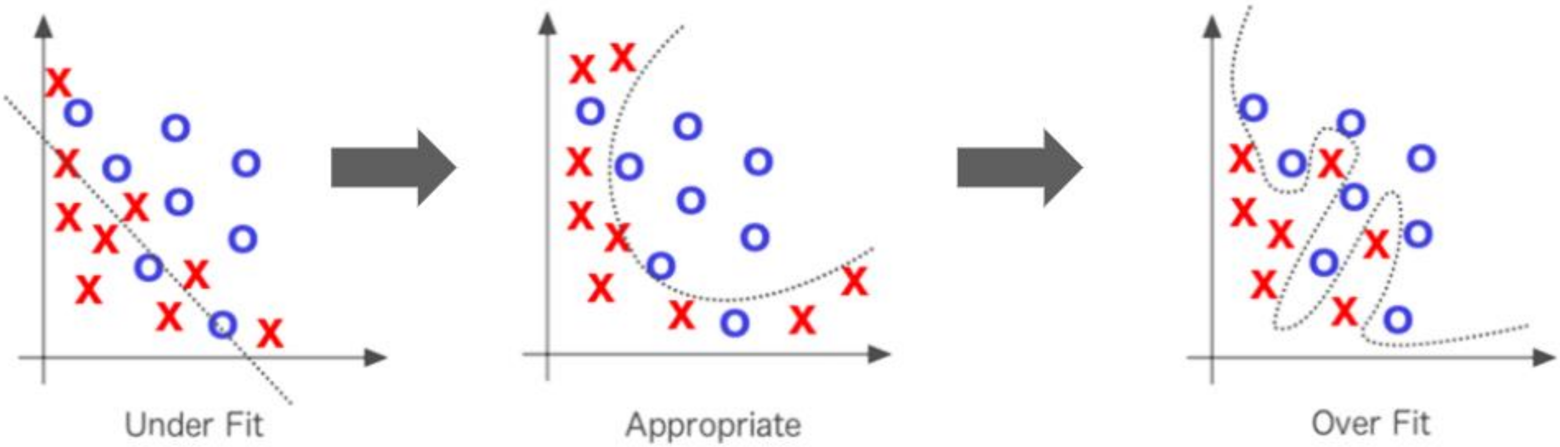
Regression: Model Fitting

Fitting a model means that you are making your algorithm learn the relationship between dependent and independent variables so that you can predict the future values of the outcome.

So the best fitted model has a specific set of parameters which best defines the problem at hand.



Regression: Model Fitting Types



Under fit where we can make our model more accurate

Accurate model as the line passing through most points without over fitting.

Over fitted model

Cognixia

A Collabera LEARNING SOLUTIONS COMPANY

THANK YOU

www.cognixia.com