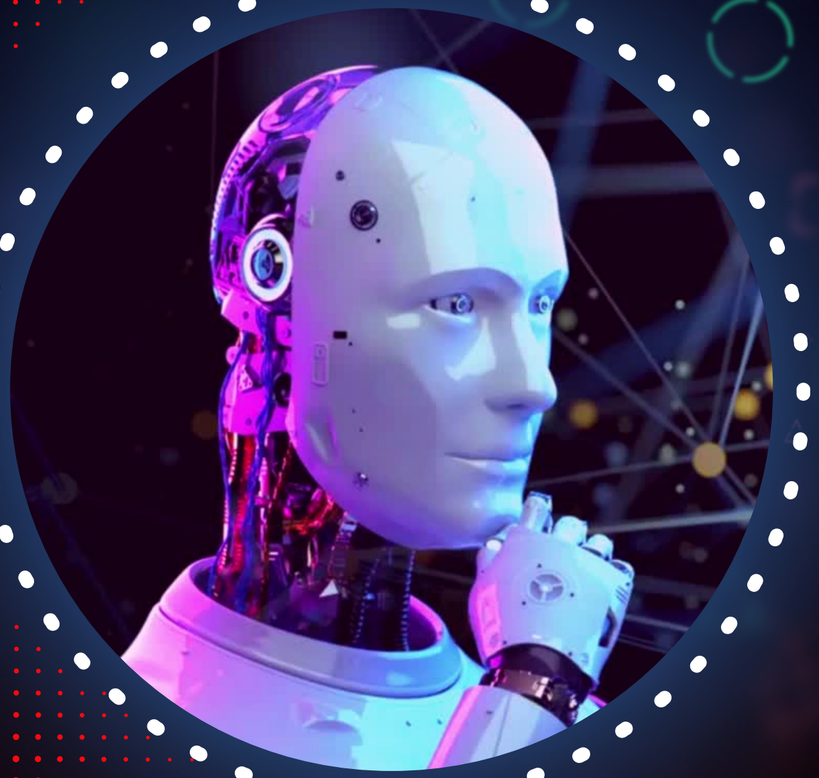
Pantech e Learning
DIGITAL LEARNING SIMPLIFIED



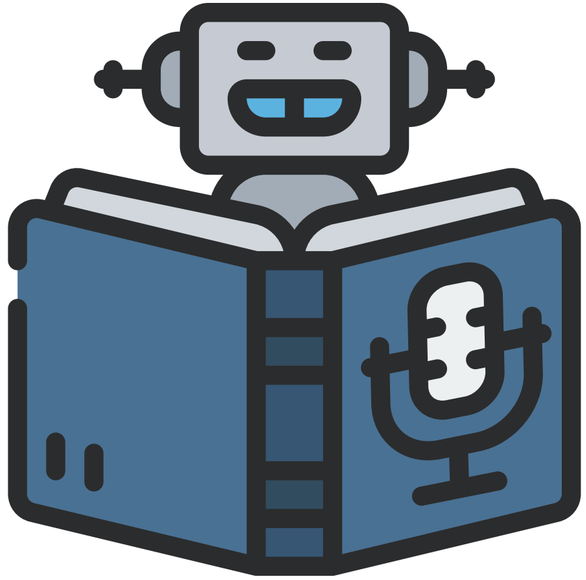
MACHINE LEARNING

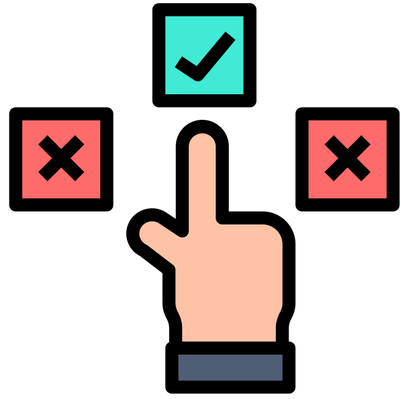
DAY – 14

Steps in Machine Learning

- Data collection
- Preprocessing
- train, test, split
- Feature selection
- Model - selection, training
- Predictions
- Evaluate and improve

HOW TO CHOOSE THE RIGHT MACHINE LEARNING ALGORIHTM ?

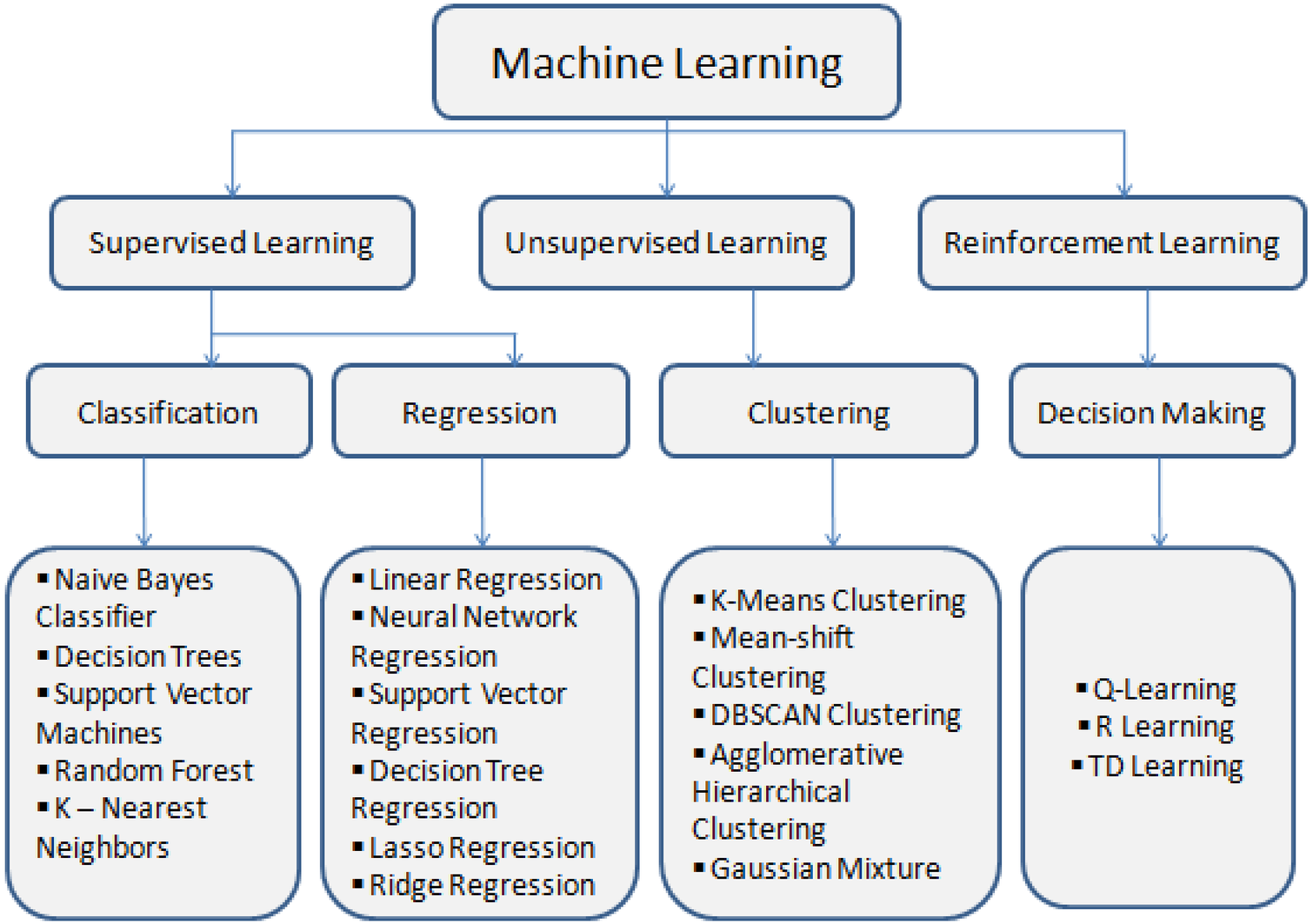




FACTORS TO BE CONSIDERED

- Nature of the problem
- Nature of algorithms
- Performance comparison

1. NATURE OF PROBLEM



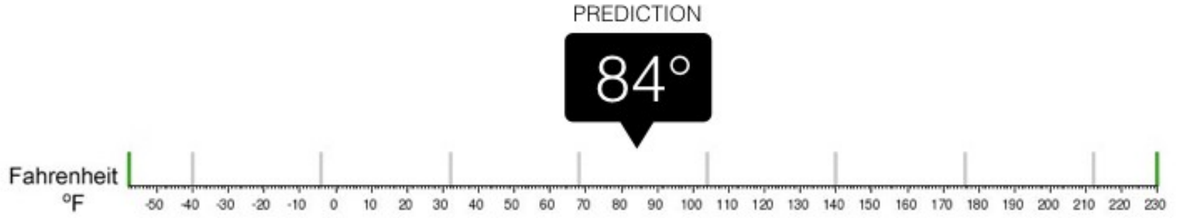
1. SUPERVISED MACHINE LEARNING

- Supervised learning is the types of machine learning in which machines are trained using well "labelled" training data, and on basis of that data, machines predict the output.
- The labelled data means some input data is already tagged with the correct output.
- In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly.
- It applies the same concept as a student learns in the supervision of the teacher.

TYPES OF SUPERVISED MACHINE LEARNING

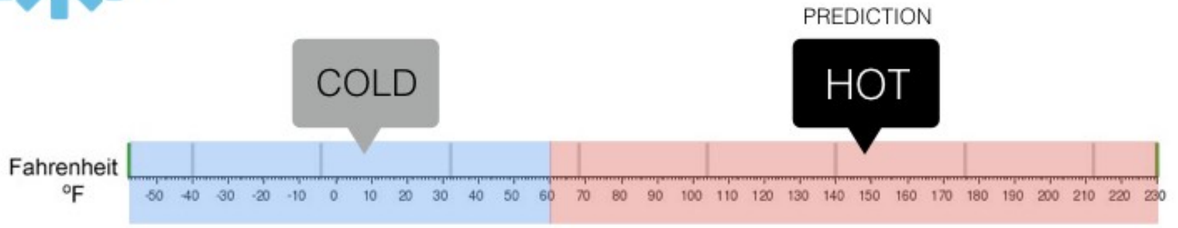
Regression

What is the temperature going to be tomorrow?



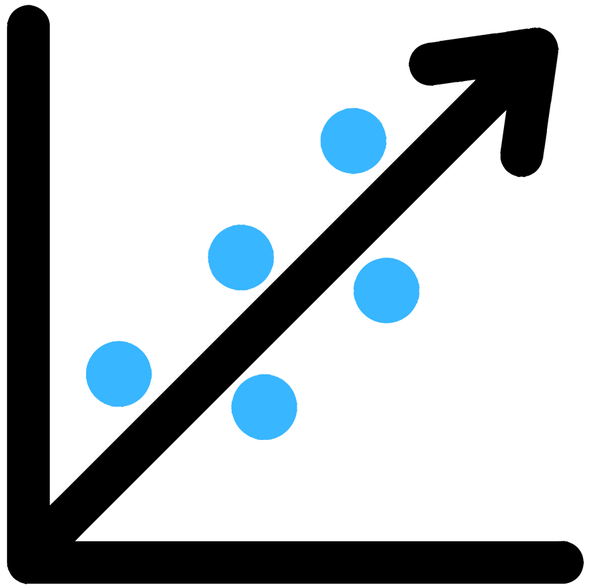
Classification

Will it be Cold or Hot tomorrow?



2. NATURE OF ALGORIHTM

LINEAR REGRESSION



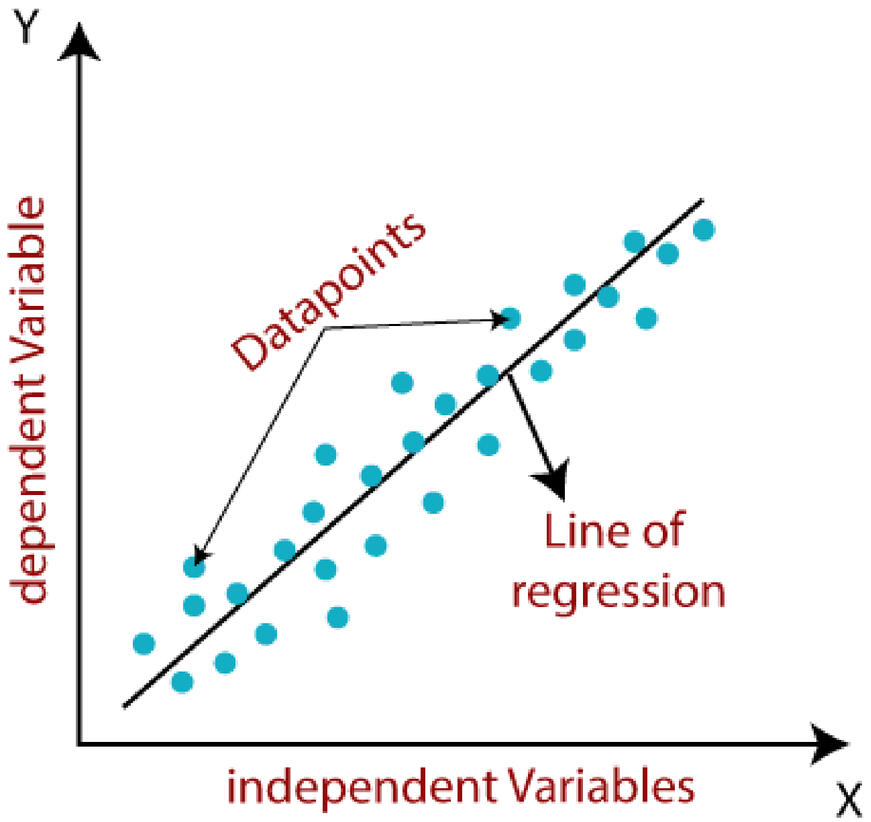
Linear Regression

- Linear regression is a popular and widely-used supervised learning algorithm in machine learning.

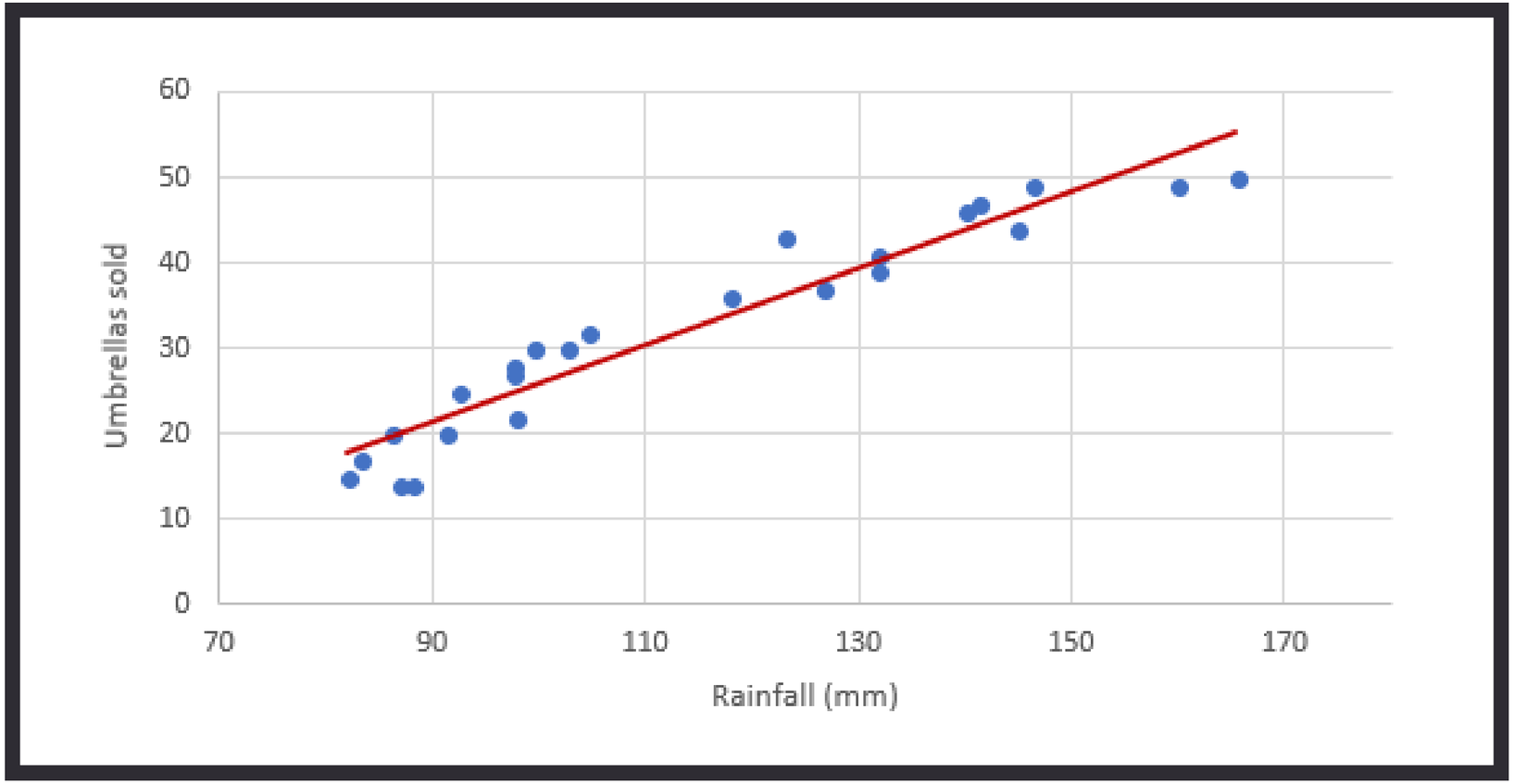
- It is a statistical method which assumes a linear relationship between the variables, meaning the relationship can be represented by a straight line on a scatter plot.

- Linear regression can be used for prediction of dependent variable with one or more independent variables .

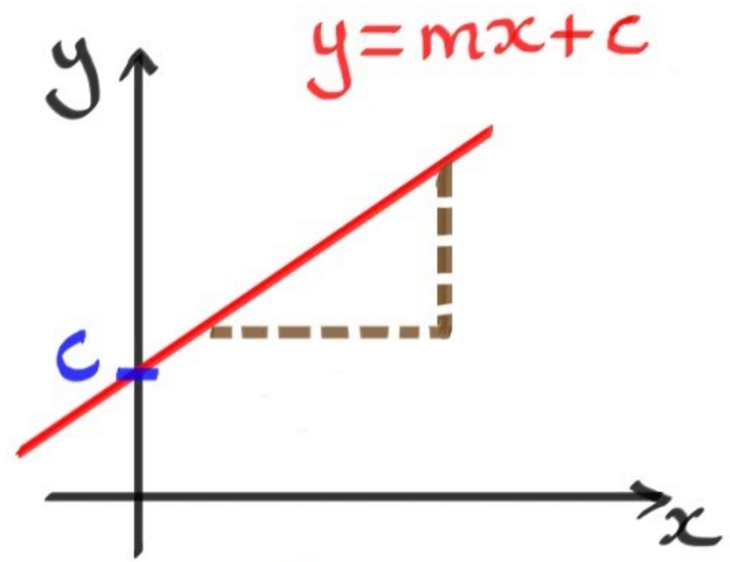
Linear Regression



Linear Regression



Equation of Straight Line



m = gradient

$$= \frac{\text{difference in } y}{\text{difference in } x}$$

$$y = mx + c$$

m : gradient

c : intercept

Used Car Dataset

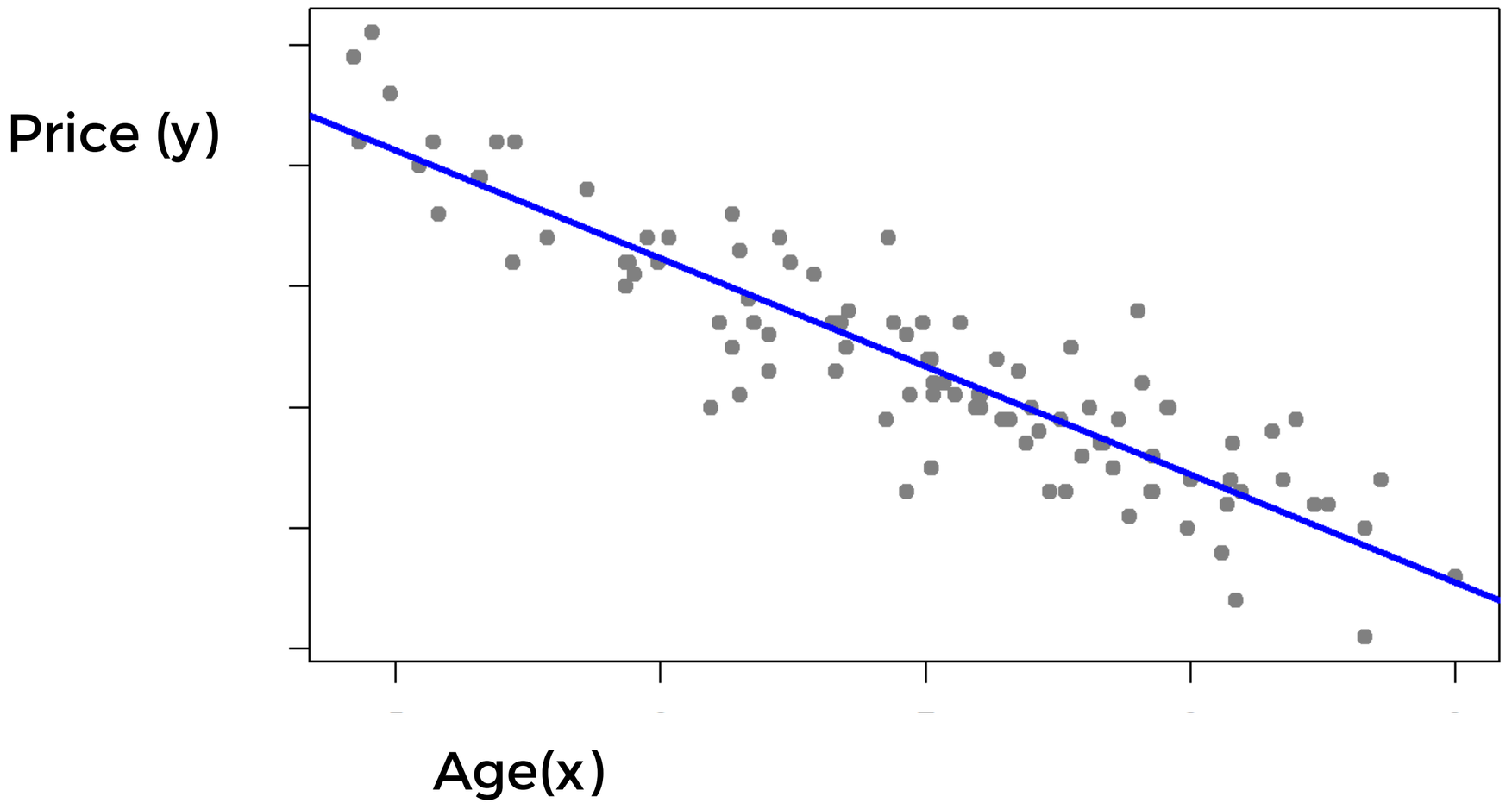
	Car_Name	Year	Present_Price	Fuel_Type	Seller_Type	Transmission	Owner	Kms_Driven	age	Selling_Price
0	ritz	2014	5.59	Petrol	Dealer	Manual	0	27000	9	3.35
1	sx4	2013	9.54	Diesel	Dealer	Manual	0	43000	10	4.75
2	ciaz	2017	9.85	Petrol	Dealer	Manual	0	6900	6	7.25
3	wagon r	2011	4.15	Petrol	Dealer	Manual	0	5200	12	2.85
4	swift	2014	6.87	Diesel	Dealer	Manual	0	42450	9	4.60
5	vitara brezza	2018	9.83	Diesel	Dealer	Manual	0	2071	5	9.25
6	ciaz	2015	8.12	Petrol	Dealer	Manual	0	18796	8	6.75
7	s cross	2015	8.61	Diesel	Dealer	Manual	0	33429	8	6.50
8	ciaz	2016	8.89	Diesel	Dealer	Manual	0	20273	7	8.75
9	ciaz	2015	8.92	Diesel	Dealer	Manual	0	42367	8	7.45

Simple vs Multiple Linear Regression

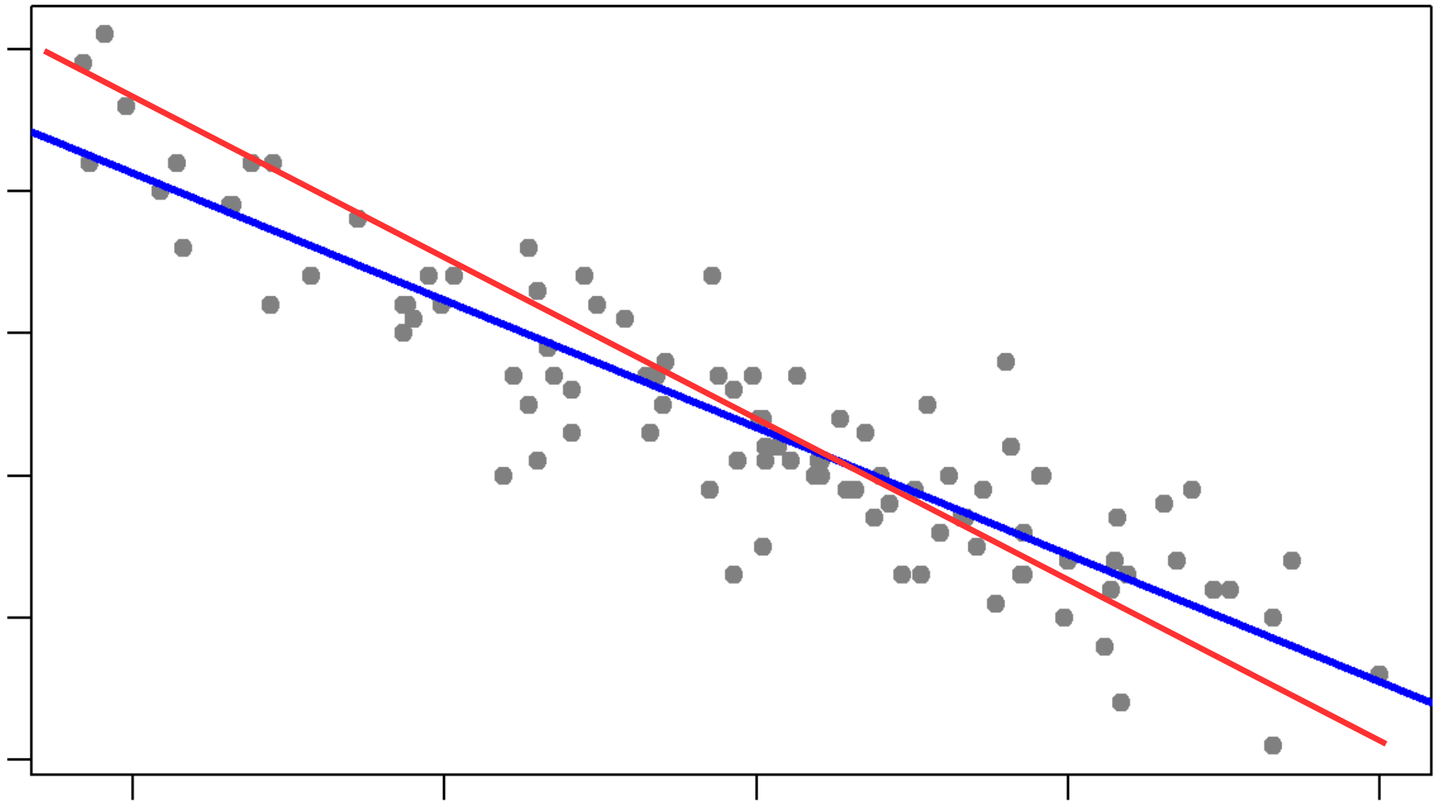
- **Simple linear regression** is a regression model that estimates the relationship between one independent variable and one dependent variable using a straight line.

- **Multiple Linear regression** estimates the relationship between two or more independent variables

Simple Linear Regression



Best Fitting Line

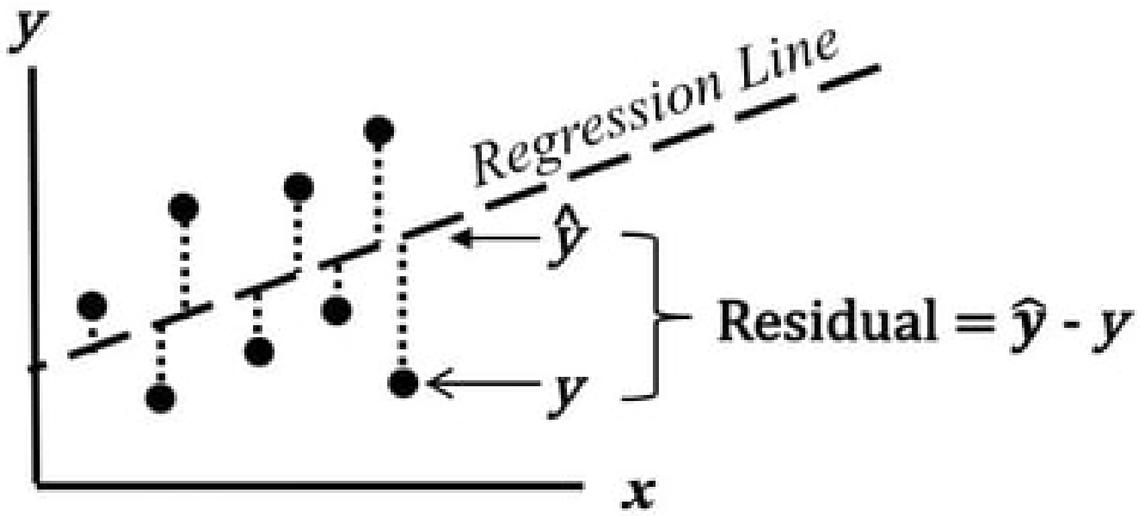


How Linear Regression works?

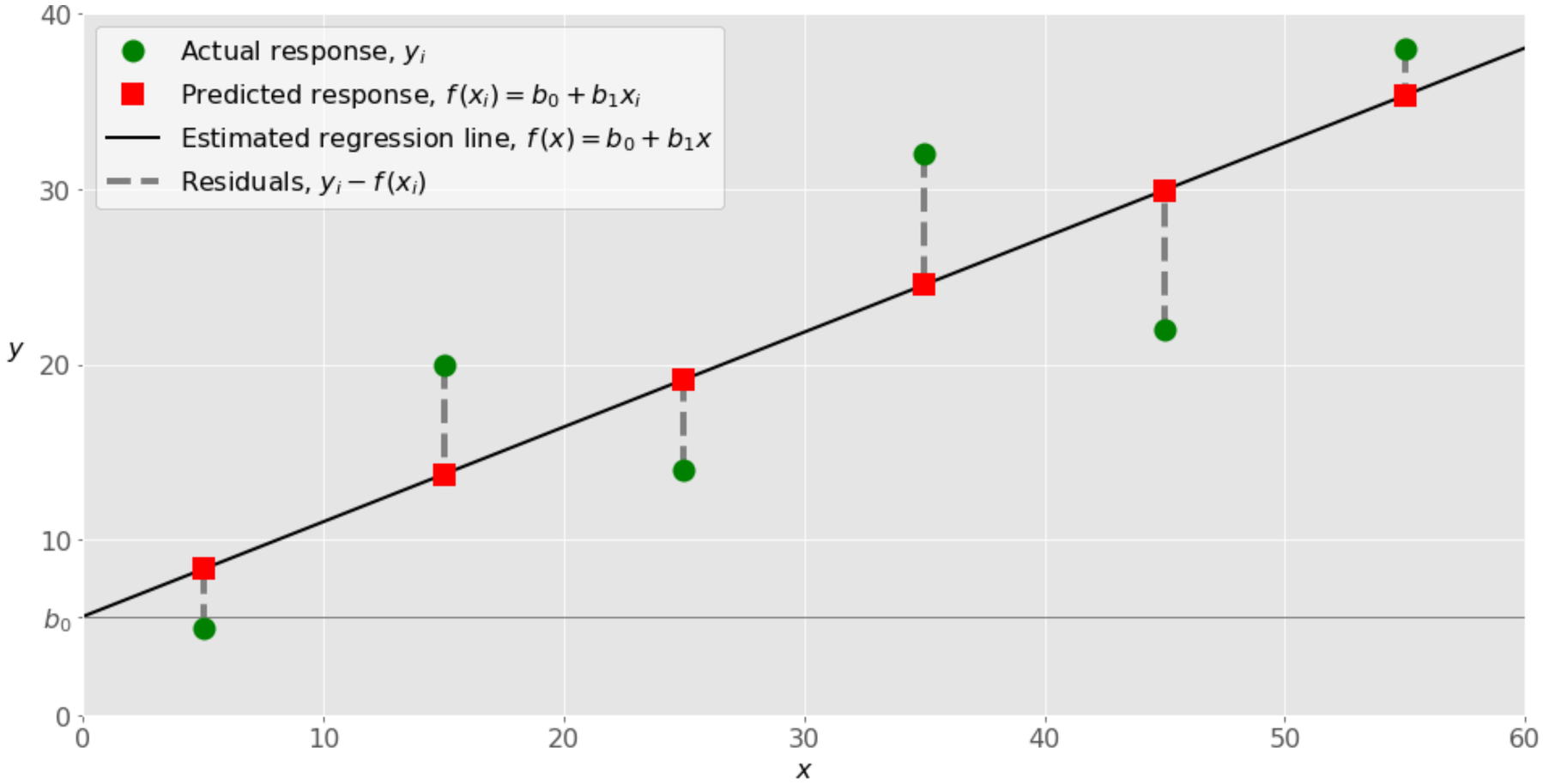
- The whole idea of the linear Regression is to find the best fit line, which has very low error that can accurately predict the value of the dependent variable based on the values of the independent variables.

- The line of best fit is determined by minimizing the difference between actual and predicted values of the dependent variable.

Error Calculation



Error Calculation



Selecting Best fitted line

- If the line passes exactly through all the points, then the value of error will be zero.

- So, by comparing the error value of all the lines, the line which has low error will be the best fitted line

Formulas

Simple Linear Regression

$$y = b_0 + b_1 * x_1$$

Multiple Linear Regression

$$y = b_0 + b_1 * x_1 + b_2 * x_2 + \ldots + b_n * x_n$$

y predicted

$$y\ (predicted) = b0 + b1^{*}x1 + b2^{*}x2 + \ldots\ldots + bn^{*}xn$$

X_1, x_2, x_n – Indepdent Variable(IV)

b_0 – intercept

b_1, b_2 – coefficients

n – No. of observations

Error formula

Error = y(original) - y(predicted)

y (predicted) = b0 + b1*x1 + b2*x2 + ……+ bn*xn

Error = y(original) - b0 + b1*x1 + b2*x2 + ……+ bn*xn

The values b0 and b1,b2…bn must be chosen so that they minimize the error.

Ordinary Least Squares (OLS)

- The OLS method is used to estimate β0 and β1. The OLS method seeks to minimize the sum of the squared residuals.

- Ordinary least square method is non-iterative method to fit a model by seeking to minimize sum of squared errors.

Ordinary Least Squares (OLS)

$$b_0 = \bar{Y} - b_1 \bar{X}$$

$$b_1 = \frac{\sum(x - \bar{x}) * (y - \bar{y})}{\sum(x - \bar{x})^2}$$

Ordinary Least Squares (OLS)

- OLS uses calculus to find the values of the parameters that minimize the sum of squared errors between the predicted values and the actual values.

- This is done by taking the partial derivatives of the cost function with respect to the parameters and setting them equal to zero.

- The solutions to these equations are the values of the parameters that minimize the cost function.

Ordinary Least Sqaures (OLS) Matrix Form

- OLS matrix form is a way of writing the ordinary least squares (OLS) regression model in matrix notation.

- This can be helpful for understanding the OLS model and for performing calculations with the model.

Ordinary Least Sqaures (OLS) Matrix Form

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$Y = X\beta + \varepsilon$$

$$\hat{\beta} = (X^T . X)^{-1} . X^T . Y$$

Linear Regression from Scikit-Learn

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
```

Regularization in Linear Regression

- Regularization is a technique used to prevent overfitting in machine learning models.

- Overfitting occurs when a model learns the noise in the data too well, and as a result, it does not generalize well to new data.

Regularization in Linear Regression

- Regularization helps to prevent overfitting by adding a penalty to the model's complexity.

- This penalty penalizes the model for having large coefficients, which helps to keep the model's predictions from being too extreme.

Regularization in Linear Regression

There are 2 main types of regularization:

- L1 regularization: **Lasso Regression**
- L2 regularization: **Ridge Regression**

Ridge Regression

- Ridge Regression works by adding a penalty term to the cost function of a linear regression model, called the regularization term.

- This regularization term prevents the model from overfitting by penalizing the large coefficients.

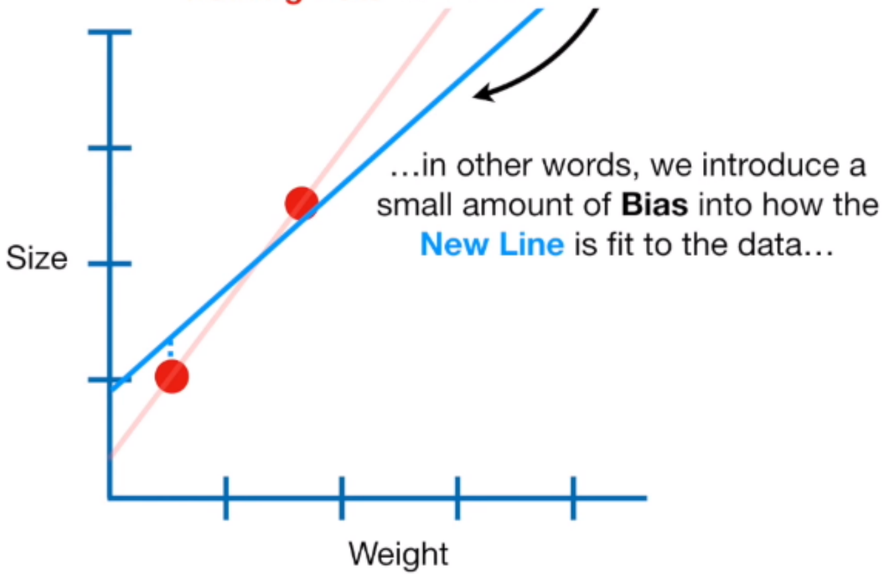
Ridge Regression

$$Cost\ Function\ = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x)^i - y^i\right)^2 + \lambda\,(slope)^2$$

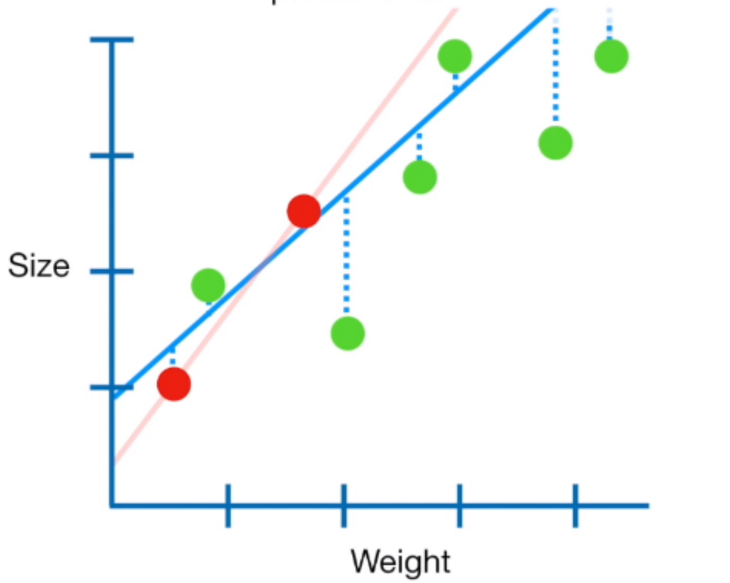
$$\lambda = Hyperparameter$$

Ridge Regression

The main idea behind **Ridge Regression** is to find a **New Line** that doesn't fit the **Training Data** as well…



In other words, by starting with a slightly worse fit, **Ridge Regression** can provide better long term predictions.



Ridge Regression

- Lambda in ridge regression is a hyperparameter that controls the amount of regularization that is applied to the model.

- A higher value of lambda leads to stronger regularization, resulting in more pronounced shrinkage of the coefficient values.

- Conversely, a lower value of lambda reduces the amount of regularization, allowing the model to fit the data more closely.

Ridge Regression

```
from sklearn.linear_model import Ridge

model = Ridge()
```

Lasso Regression

- Lasso regularization penalizes the sum of the absolute values of the coefficients.

- This means that the coefficients are encouraged to be small, and some of them may even be set to zero. This can help to prevent overfitting

Lasso Regression

$$Cost\ Function\ = \frac{1}{n}\sum_{i=1}^{n}\left(h_\theta(x)^i - y^i\right)^2 + \lambda \sum_{i=1}^{n} |slope|$$

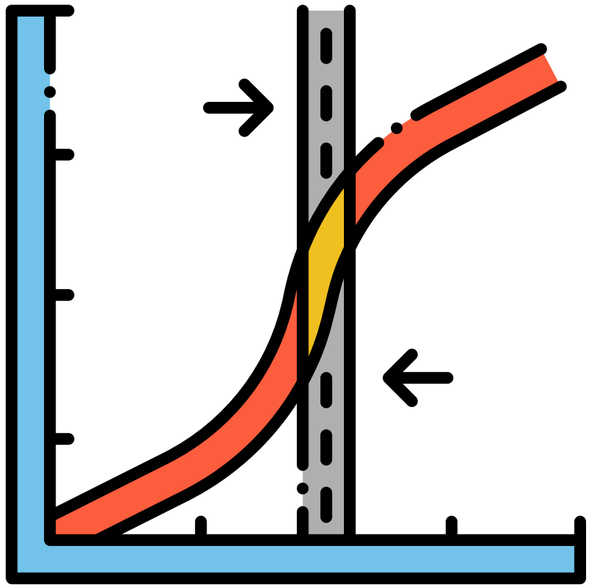
$$\lambda = Hyperparameter$$

Lasso Regression

```
from sklearn.linear_model import Lasso

model = Lasso()
```

LOGISTIC REGRESSION



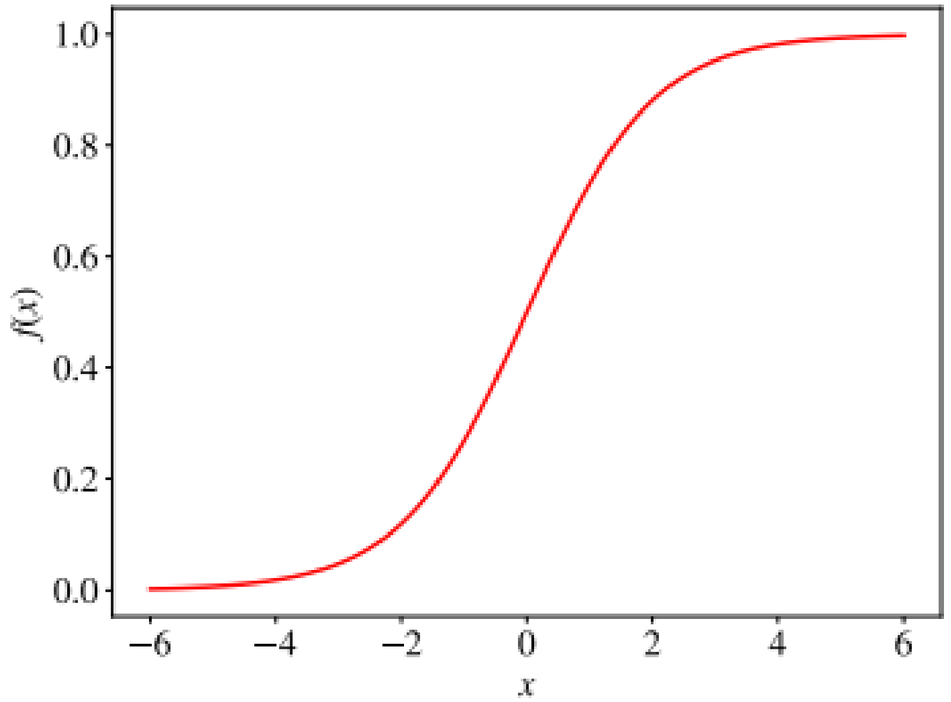
Logistic Regression

- Logistic Regression is much similar to the Linear Regression except that how they are used.

- Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

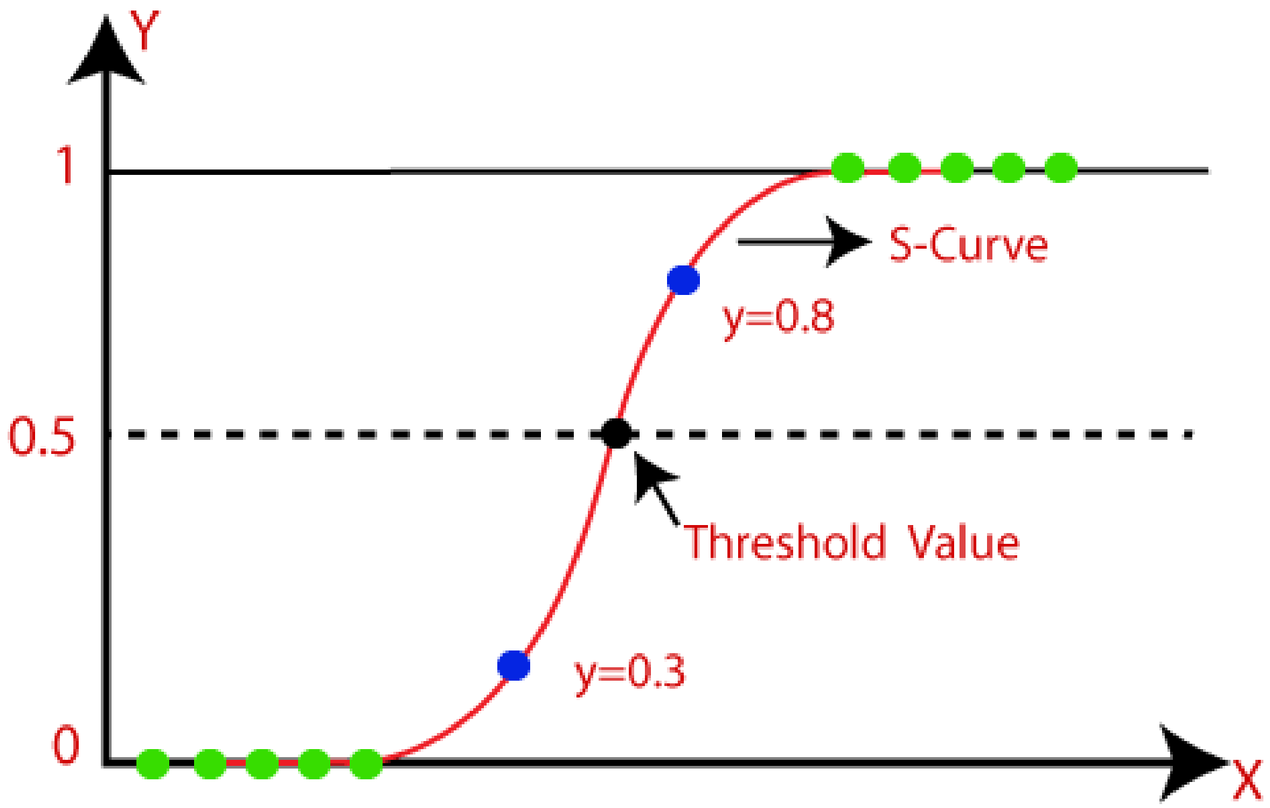
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

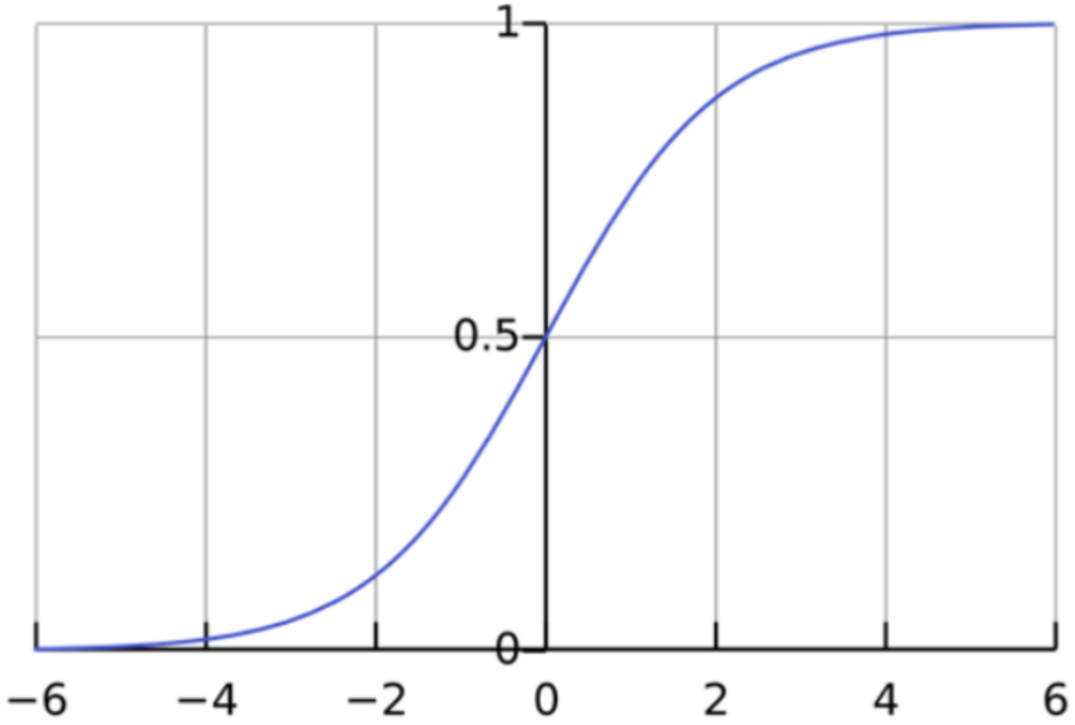
Logistic Regression



$$f(x) = \frac{1}{1 + e^{-(x)}}$$

Logistic Regression





Logistic Regression

- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1.

- Such as values above the threshold value of 0.5 tends to 1, and a value below the threshold values of 0.5 tends to 0.

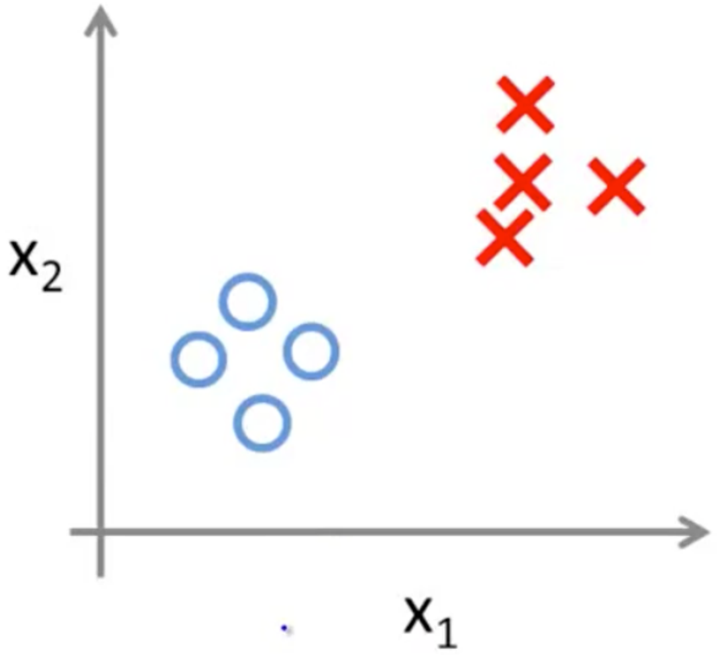
Types of Logistic Regression

- **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

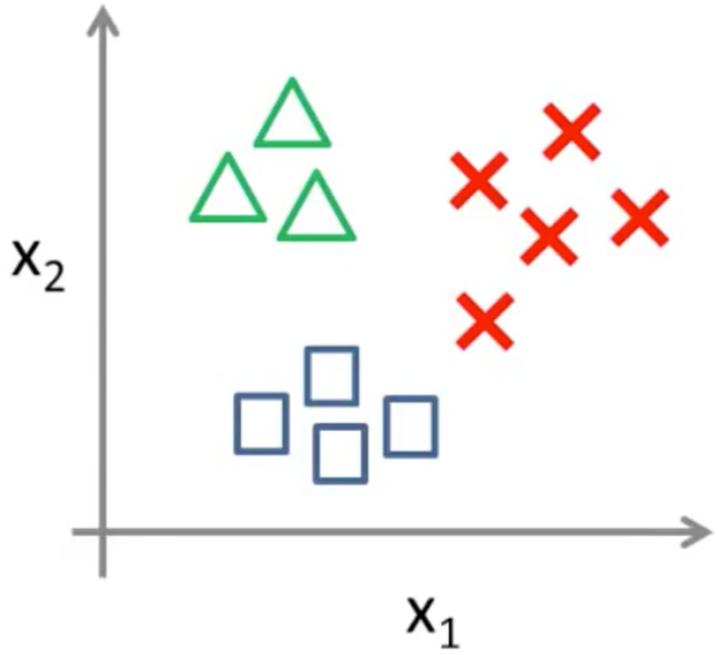
- **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"

Types of Logistic Regression

Binary classification:



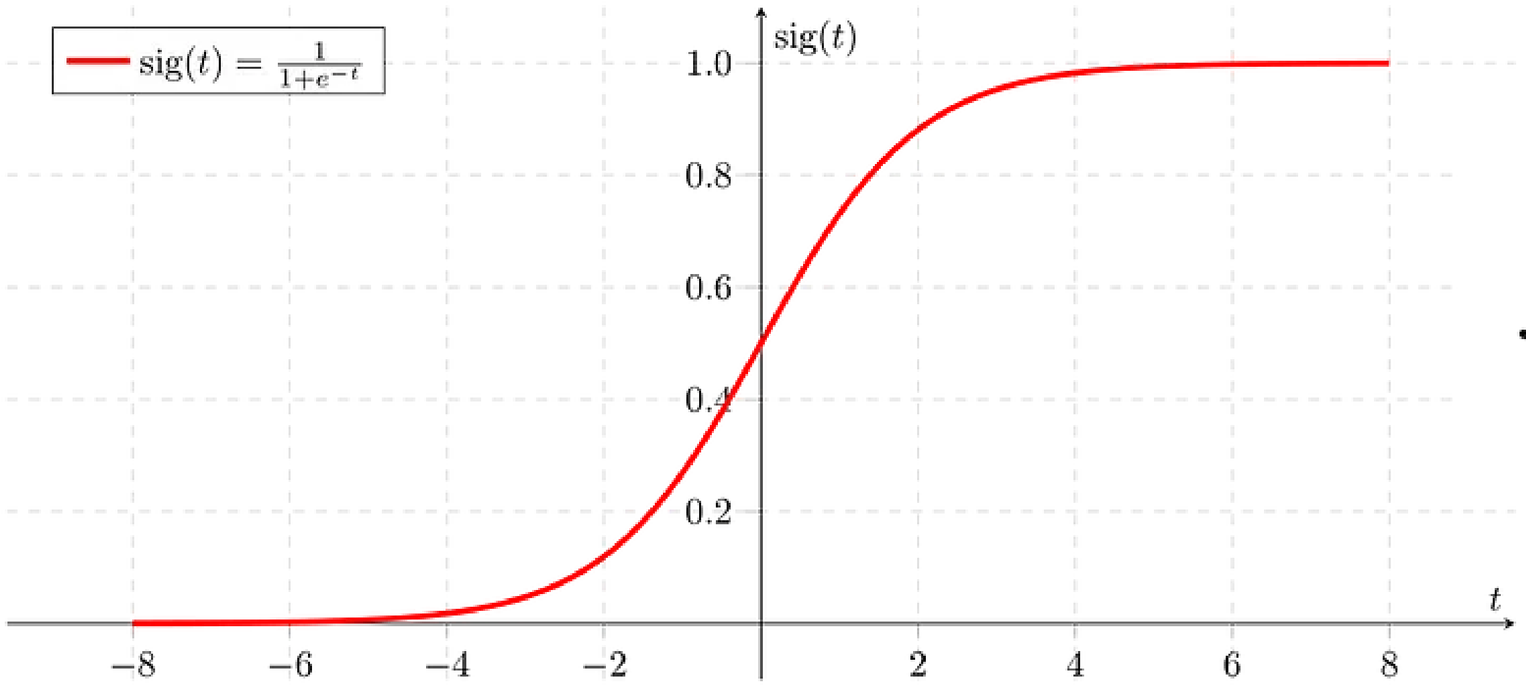
Multi-class classification:



Logistic Regression

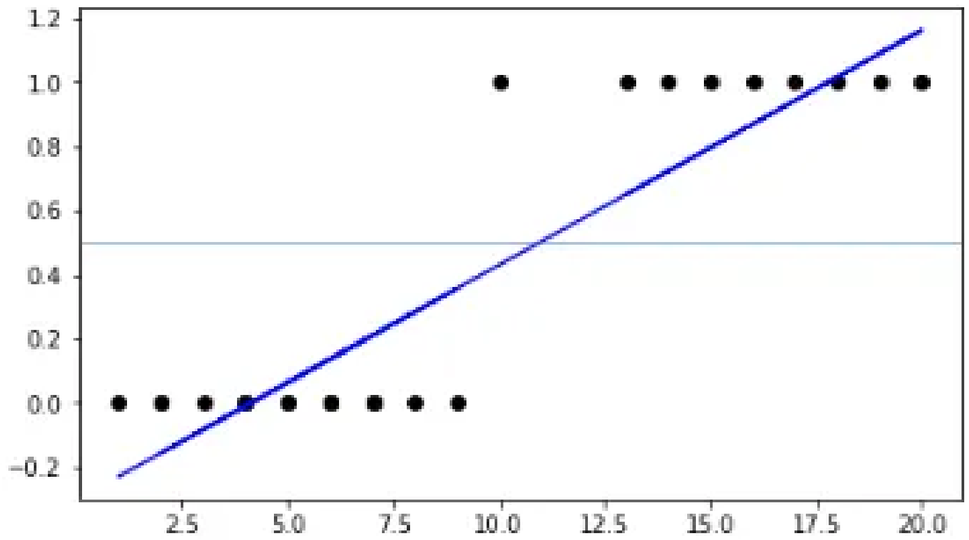
	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
5	0	3	male	NaN	0	0	8.4583	Q	Third	man	True	NaN	Queenstown	no	True
6	0	1	male	54.0	0	0	51.8625	S	First	man	True	E	Southampton	no	True
7	0	3	male	2.0	3	1	21.0750	S	Third	child	False	NaN	Southampton	no	False
8	1	3	female	27.0	0	2	11.1333	S	Third	woman	False	NaN	Southampton	yes	False
9	1	2	female	14.0	1	0	30.0708	C	Second	child	False	NaN	Cherbourg	yes	False

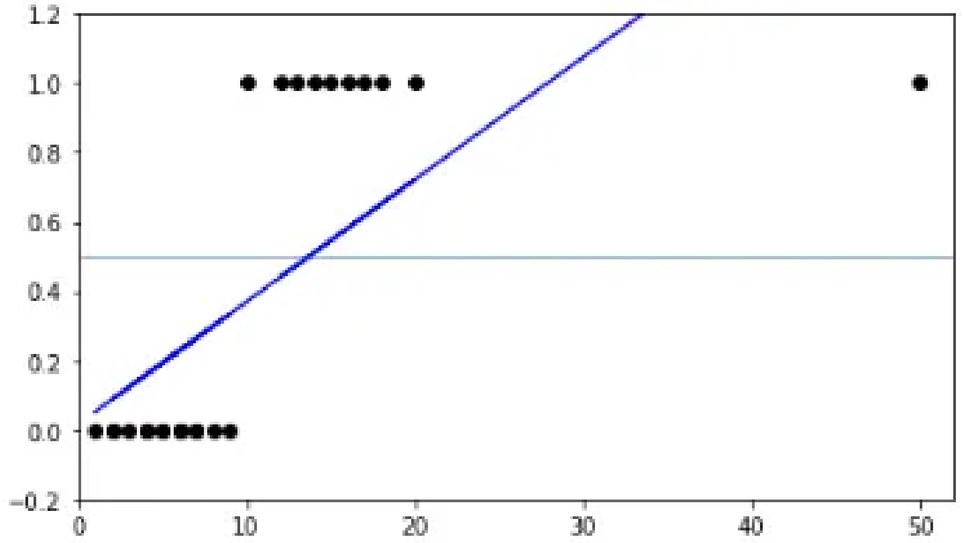
Logistic Regression



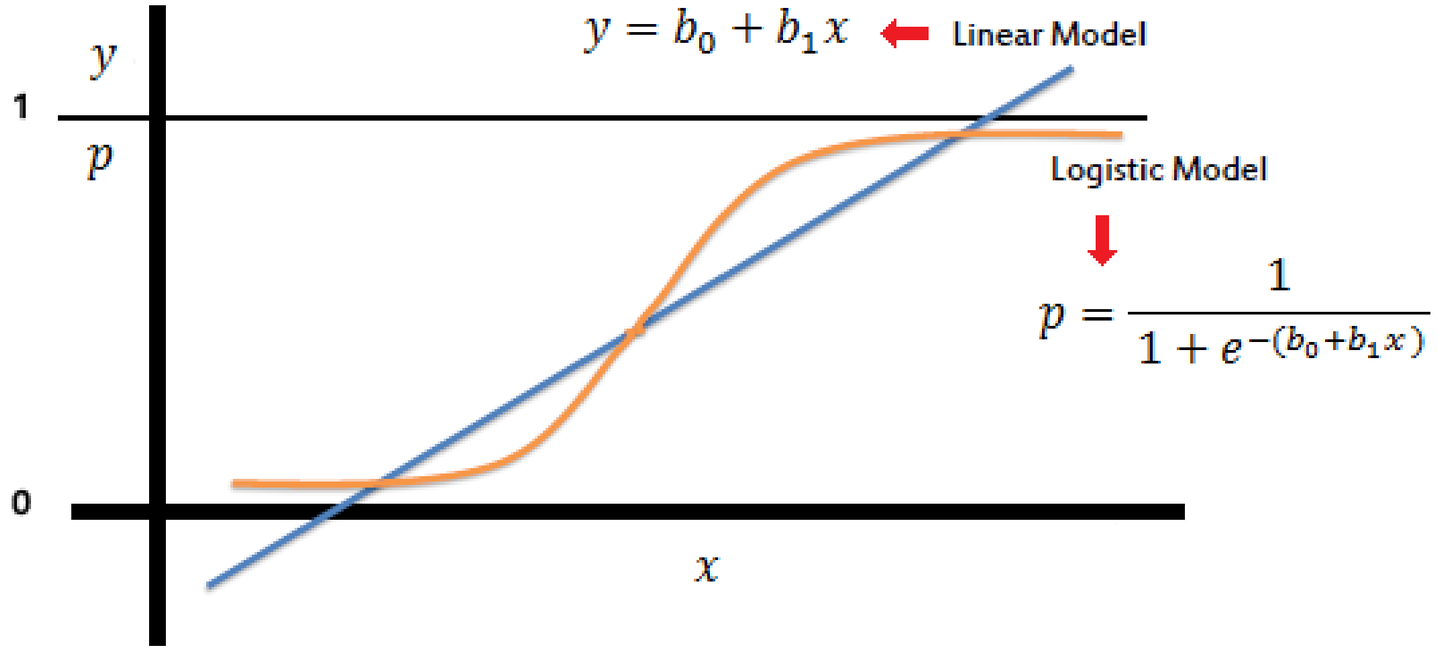
$$f(x) = \frac{1}{1+e^{-(x)}}$$

Logistic Regression

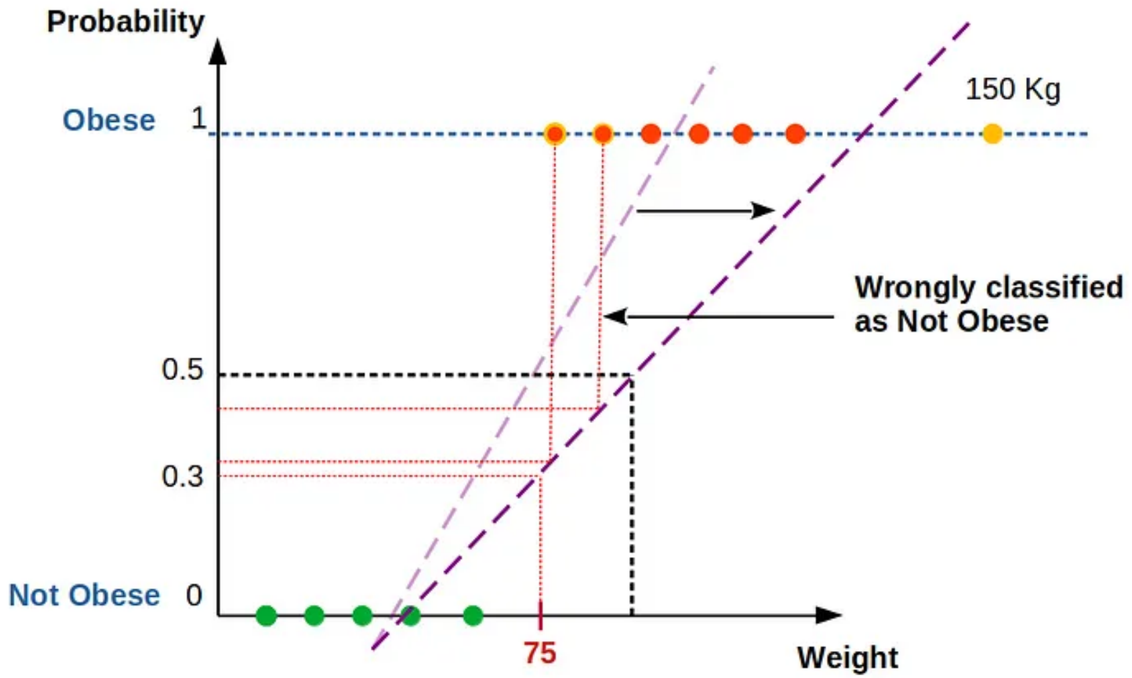




Logistic Regression

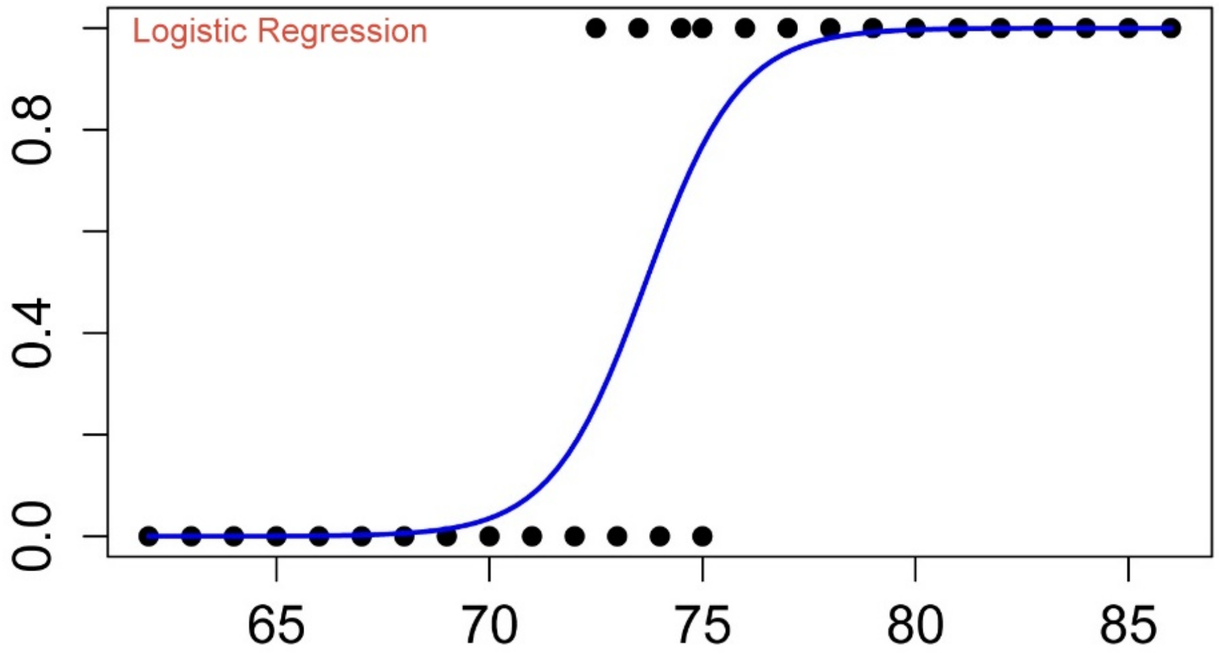


Why not Linear Regression for Classification?



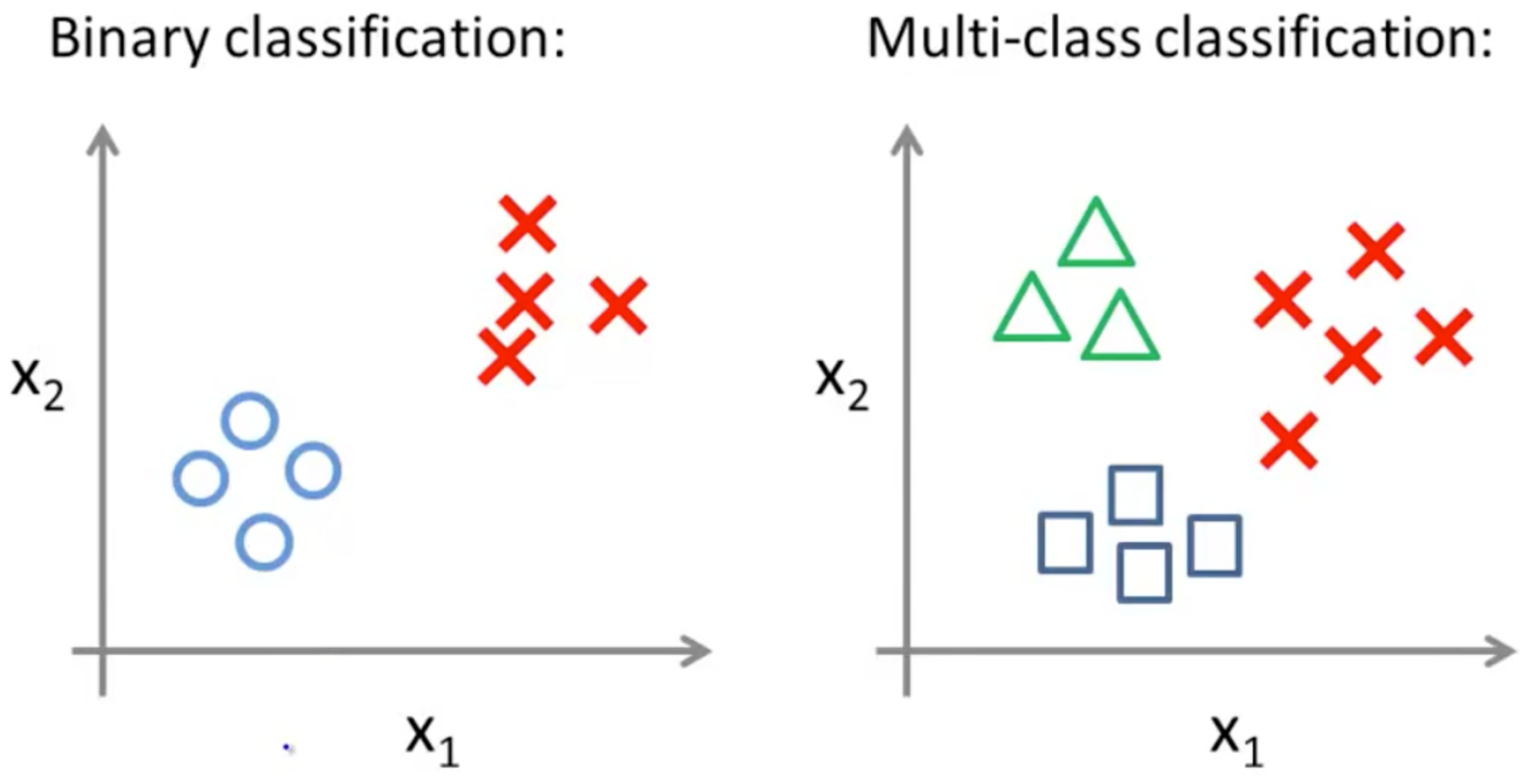
- There are two things that explain why Linear Regression is not suitable for classification.
- The first one is that Linear Regression deals with continuous values whereas classification problems mandate discrete values.
- The second problem is regarding the shift in threshold value when new data points are added.

Equation of Logistic Regression



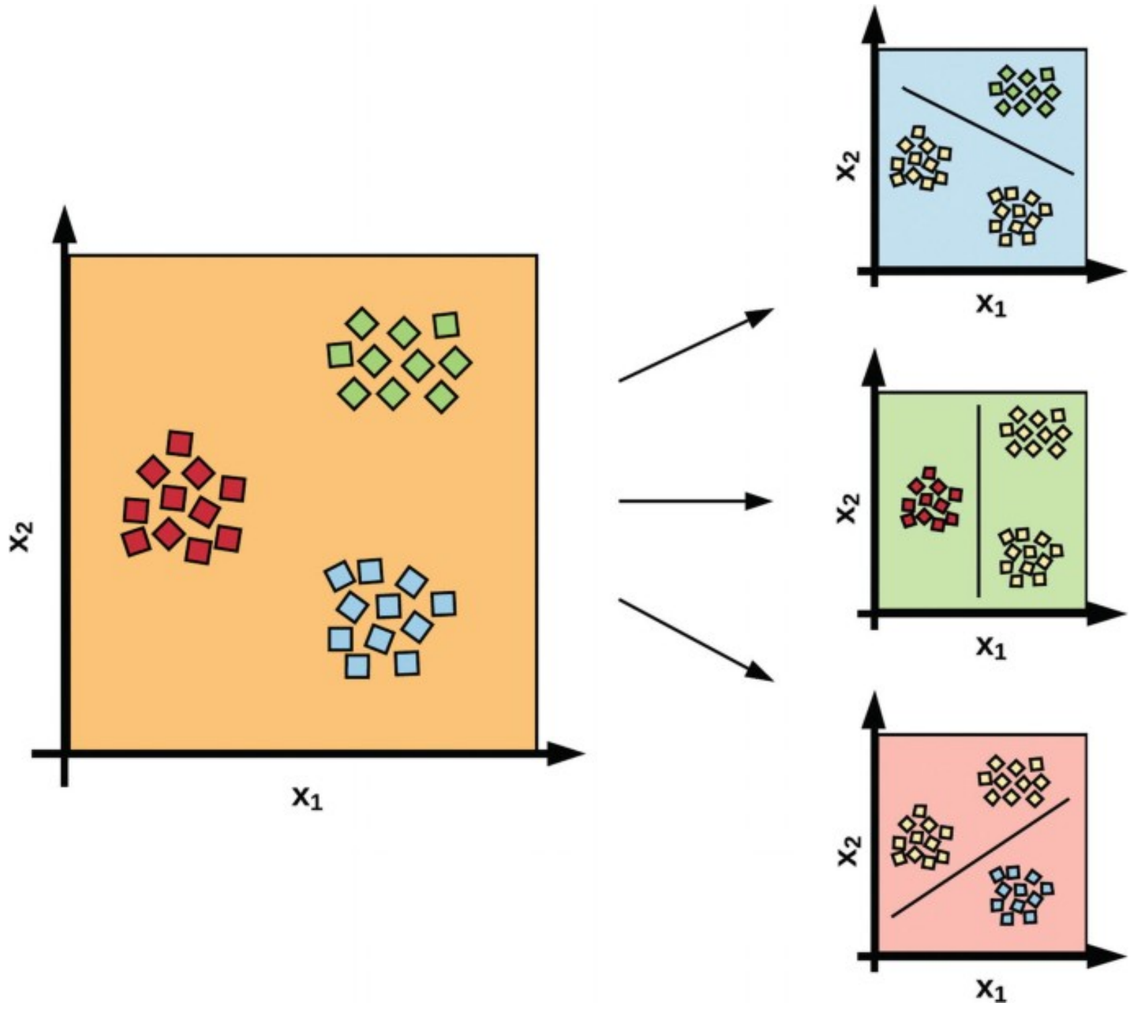
$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Multiclass Classification with Logistic Regression



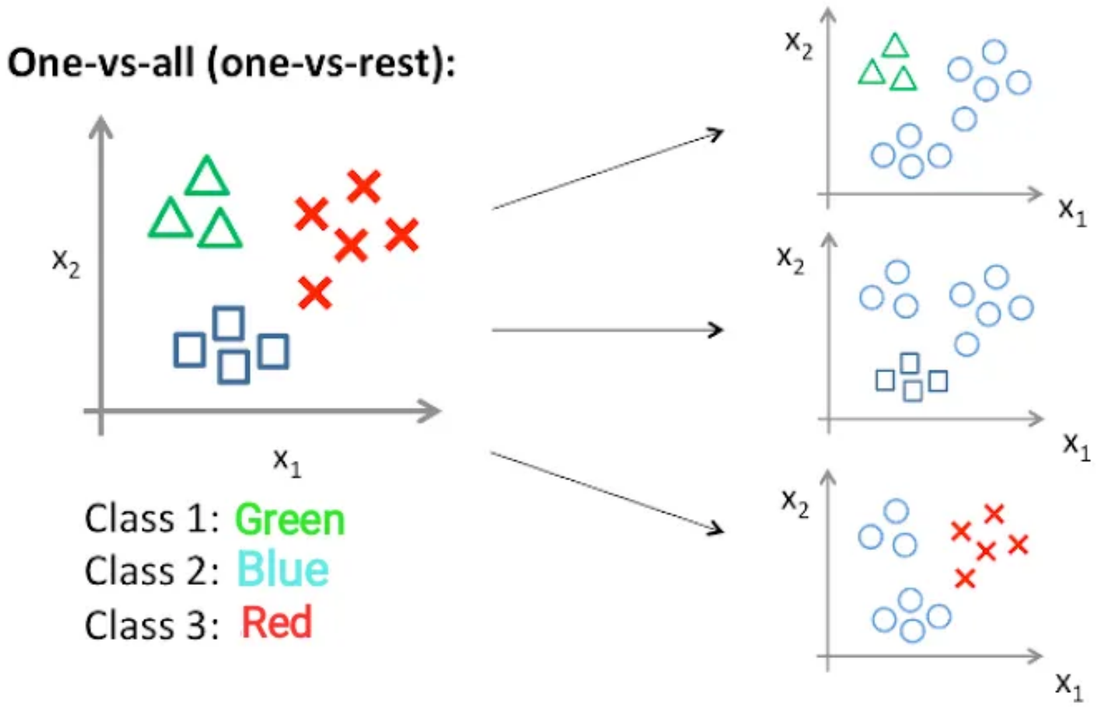
Multiclass Classification with Logistic Regression

	f1	f2	f3	output
0	1	7	13	O1
1	2	8	14	O2
2	3	9	15	O3
3	4	10	16	O2
4	5	11	17	O1
5	6	12	18	O2



Multiclass Classification with Logistic Regression

	f1	f2	f3	output	O1	O2	O3
0	1	7	13	O1	1	-1	-1
1	2	8	14	O2	-1	1	-1
2	3	9	15	O3	-1	-1	1
3	4	10	16	O2	-1	1	-1
4	5	11	17	O1	1	-1	-1
5	6	12	18	O2	-1	1	-1



Test data Prediction based on probability

$$[O1, O2, O3] = [0.20, 0.25, 0.55]$$

THANK YOU