

Model Building – Supervised Learning

BY:- SHOBHIT TYAGI



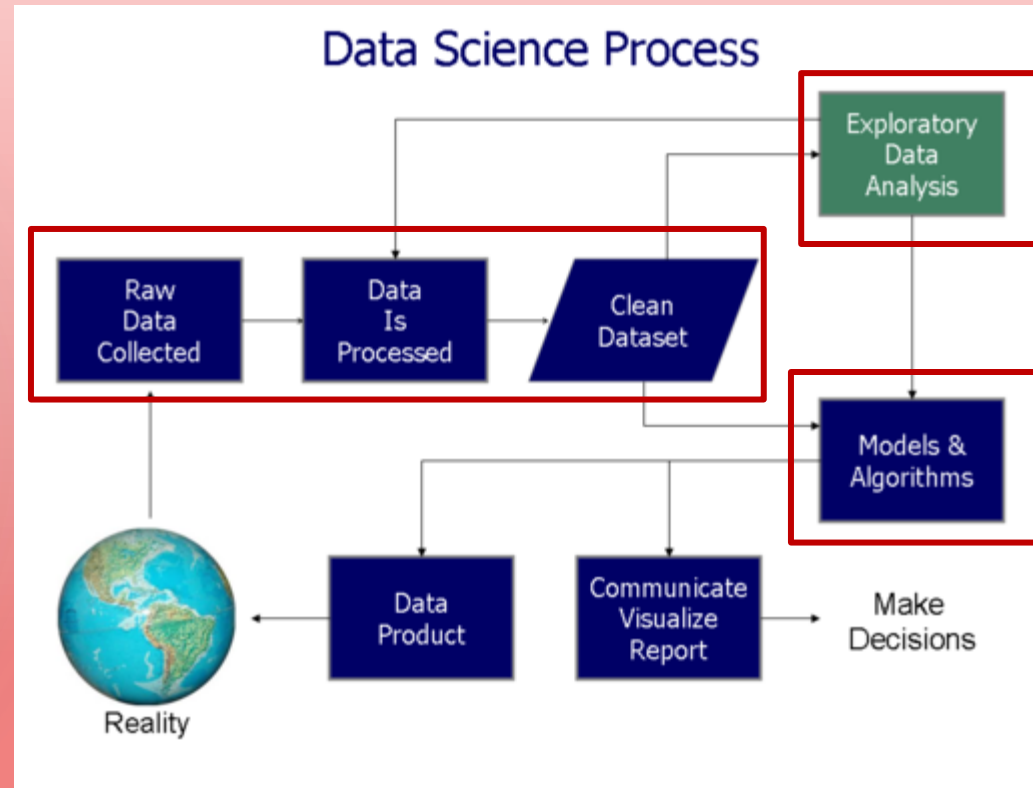
Contents

- Recap
- Regression
- Classification



Recap

Data Science Process Flow



Split the data

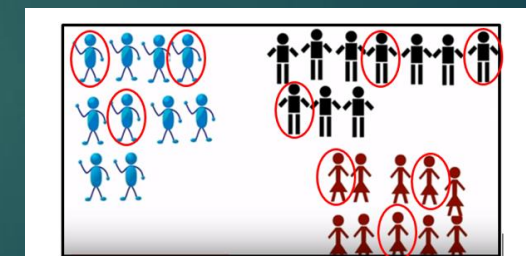
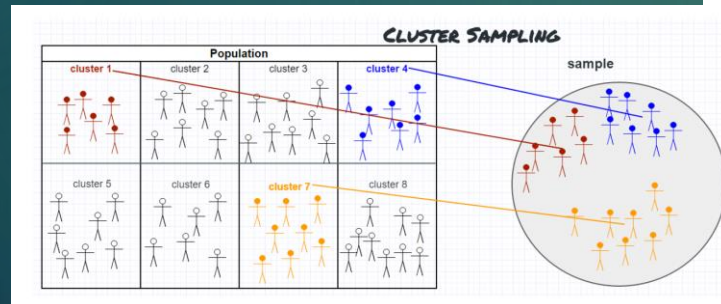
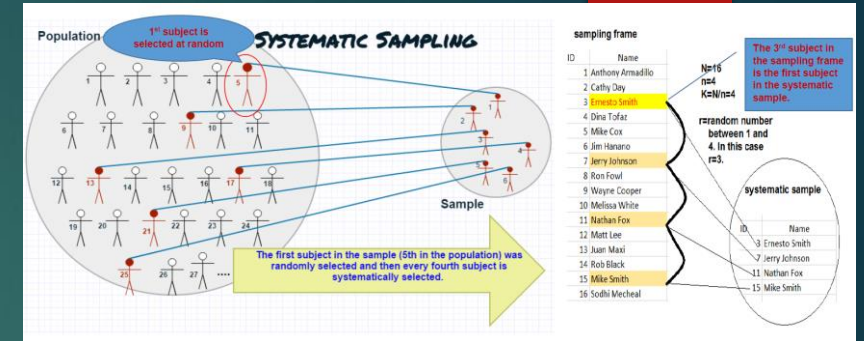
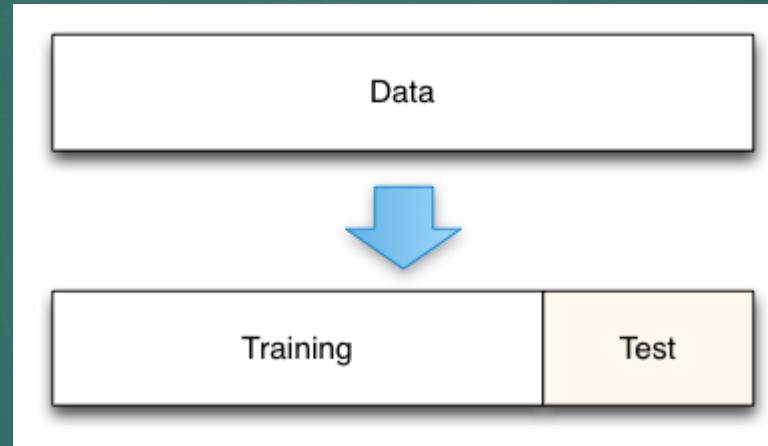
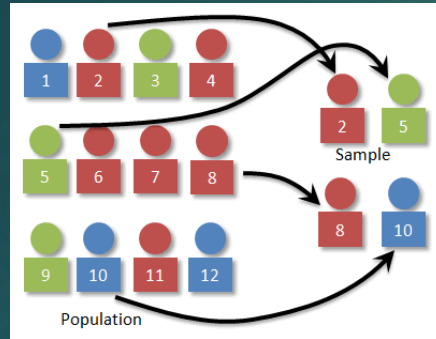
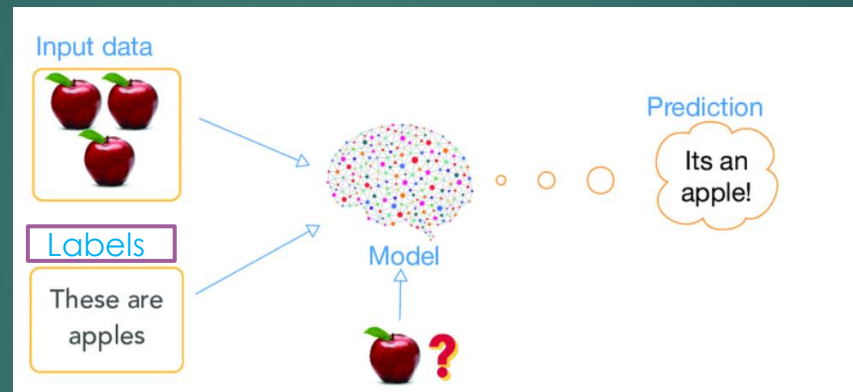


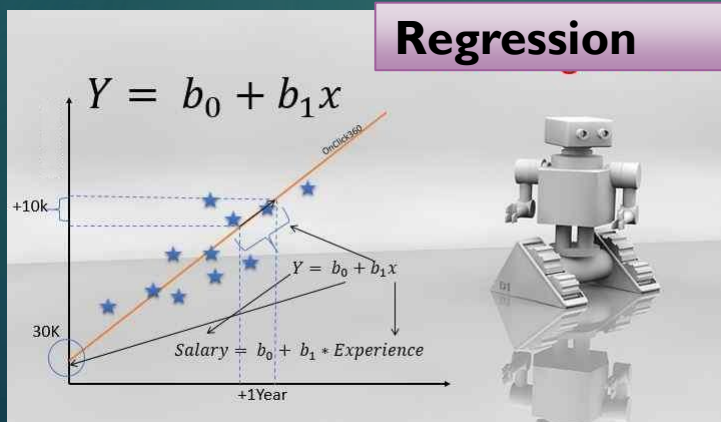
Fig. 3 Selecting datapoint from each subgroup

Modelling – Supervised Learning

The **correct classes** of the training data are **known**



Regression



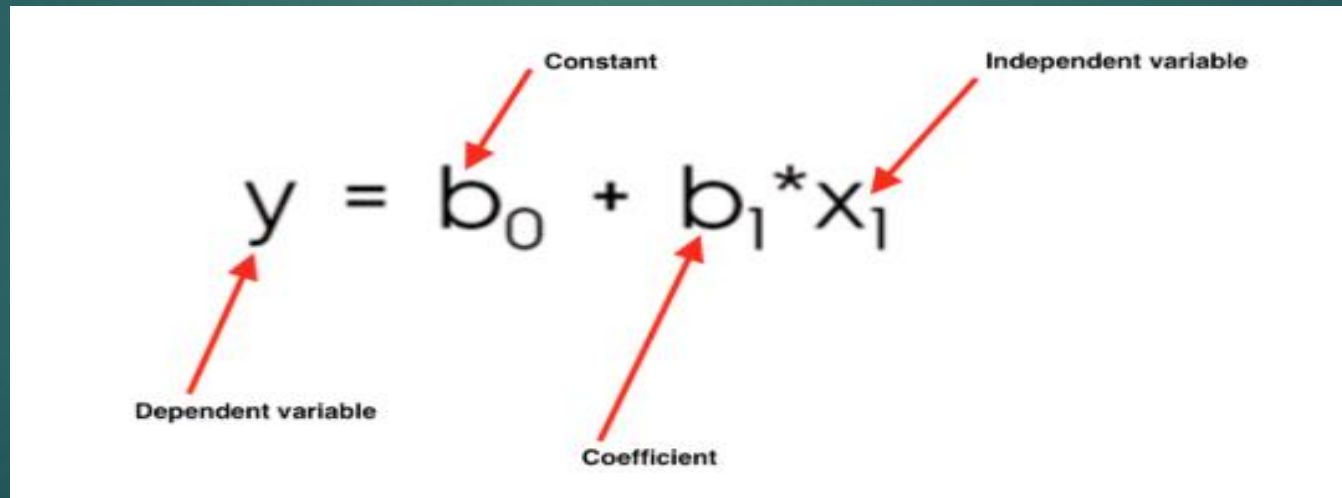
Classification



Supervised Learning - Regression

Linear Regression

- Linear regression is used to predict a quantitative response Y from the predictor variable X .
- Linear Regression is made with an assumption that there's a linear relationship between X and Y .



The diagram shows the linear regression equation $y = b_0 + b_1 * x_1$ with four red arrows pointing to its components and their labels:

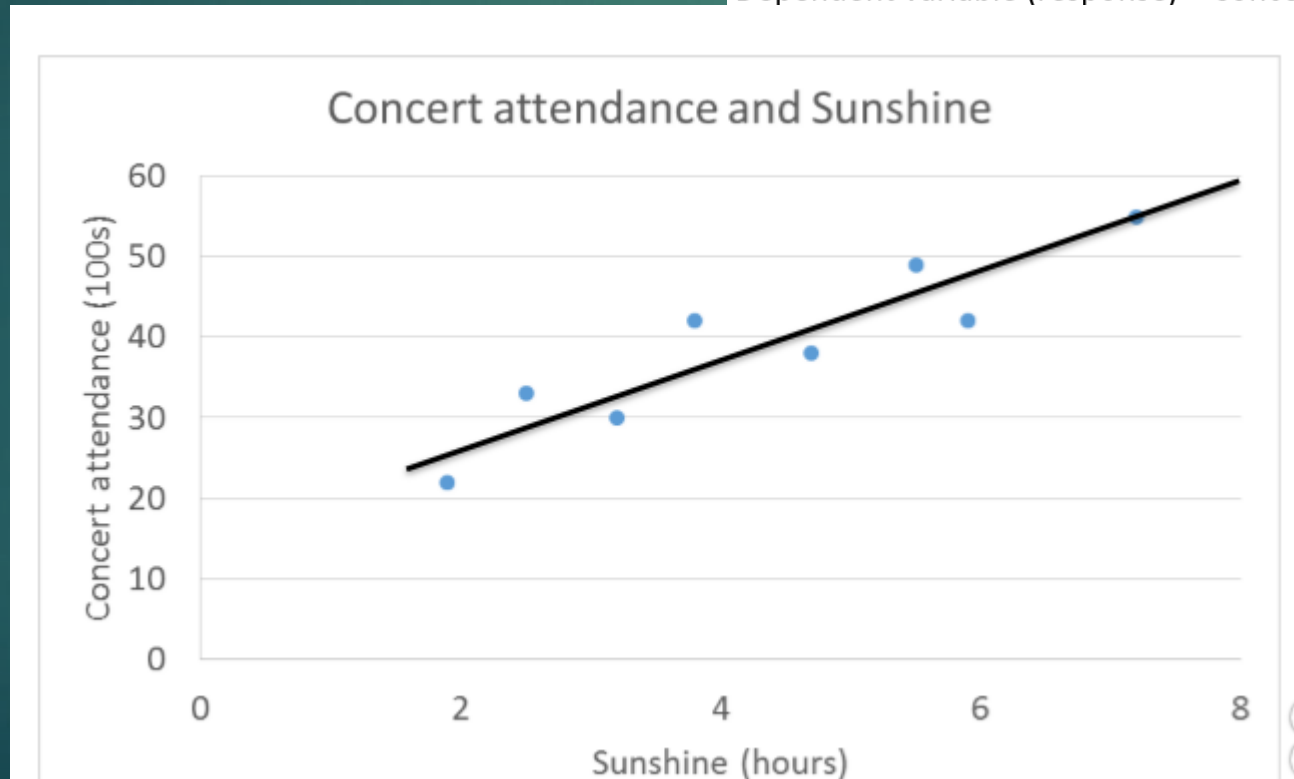
- An arrow points from the label "Dependent variable" to the variable y .
- An arrow points from the label "Constant" to the term b_0 .
- An arrow points from the label "Coefficient" to the term b_1 .
- An arrow points from the label "Independent variable" to the variable x_1 .

Example - Use Case

Sunshine (hours)	1.9	2.5	3.2	3.8	4.7	5.5	5.9	7.2
Concert attendance (100s)	22	33	30	42	38	49	42	55

Independent variable (explanatory) – Sunshine – Plotted on X-axis

Dependent variable (response) – Concert attendance – Plotted on Y-axis

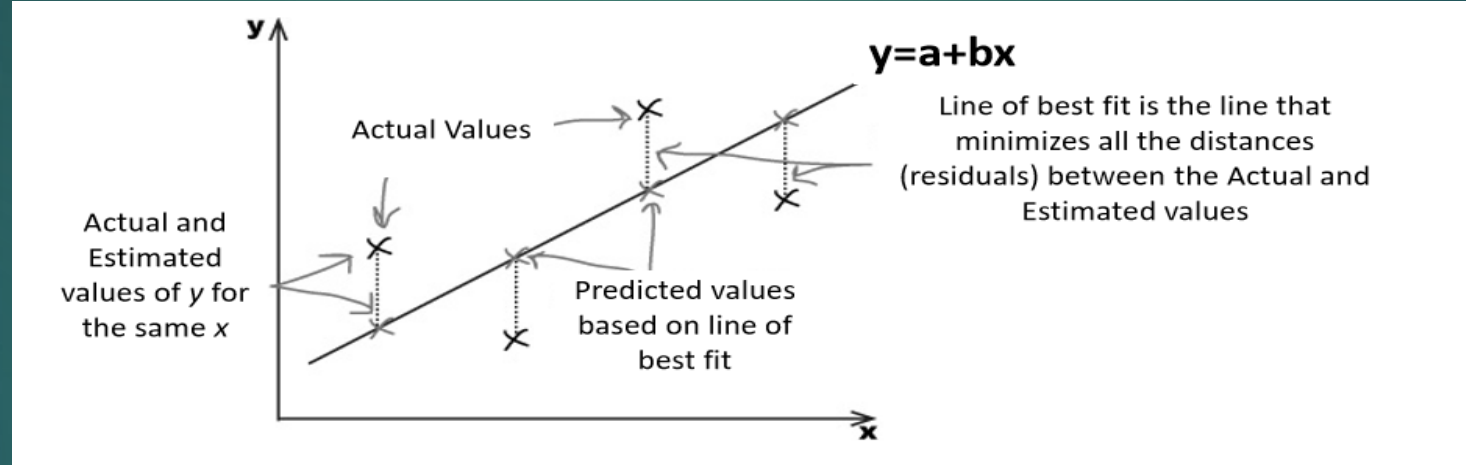


We need to find a
line of best fit

Best Line as errors
are less !!

Line of best fit

Line of best fit is the line which gives least error.



The lines whose residual error on all points is the least is the best line

To ensure residual errors don't cancel, we take square of residual errors.

Cost Function

- ▶ A cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.

$$J = \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2$$

← Objective

$$\text{minimize } \frac{1}{n} \sum_{i=1}^n (\text{pred}_i - y_i)^2$$

Gradient Descent

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

Cost Function

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m [h_{\theta}(x_i) - y_i]^2$$

Predicted Value True Value

Gradient Descent

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

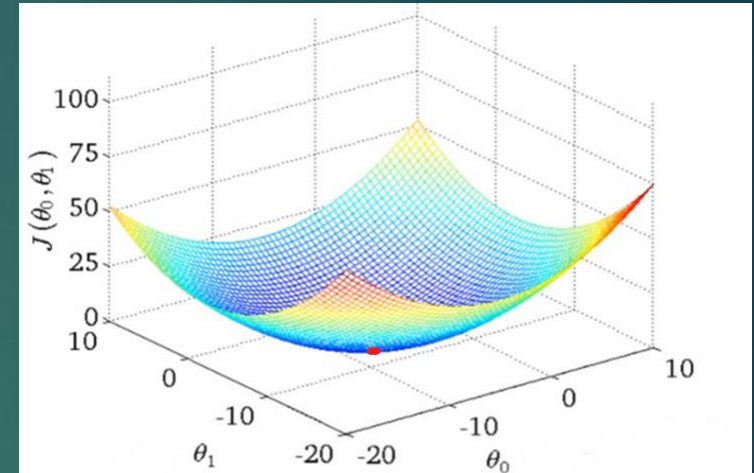
Learning Rate

Repeat until convergence {

$$\theta_1 \leftarrow \theta_1 - \alpha \frac{\partial}{\partial \theta_1} \left(\frac{1}{2m} \sum_{i=1}^m (h(x_i) - y_i)^2 \right)$$

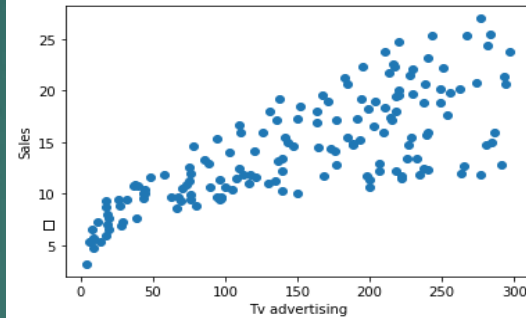
$$\theta_2 \leftarrow \theta_2 - \alpha \frac{\partial}{\partial \theta_2} \left(\frac{1}{2m} \sum_{i=1}^m (h(x_i) - y_i)^2 \right)$$

}



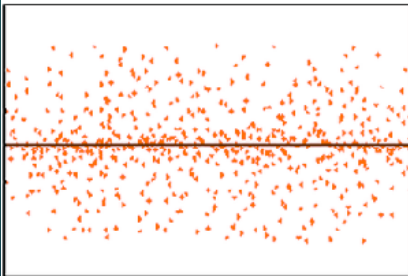
Assumptions of Linear Regression

The model is linear

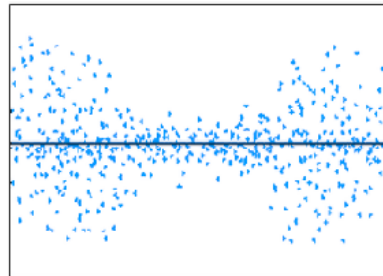


The error terms should have constant variance
(Homoscedasticity)

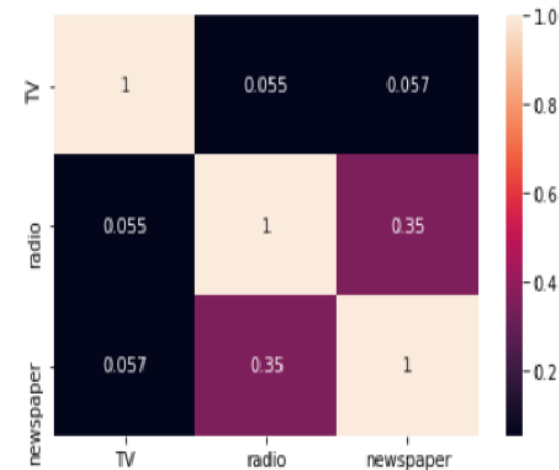
Homoscedasticity



Heteroscedasticity



Little or no Multicollinearity between the features



Performance Metrics

Mean Squared Error

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} = \sqrt{\text{MSE}}$$

Acroynm	Full Name	Residual Operation?	Robust To Outliers?
MAE	Mean Absolute Error	Absolute Value	Yes
MSE	Mean Squared Error	Square	No
RMSE	Root Mean Squared Error	Square	No
MAPE	Mean Absolute Percentage Error	Absolute Value	Yes

Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

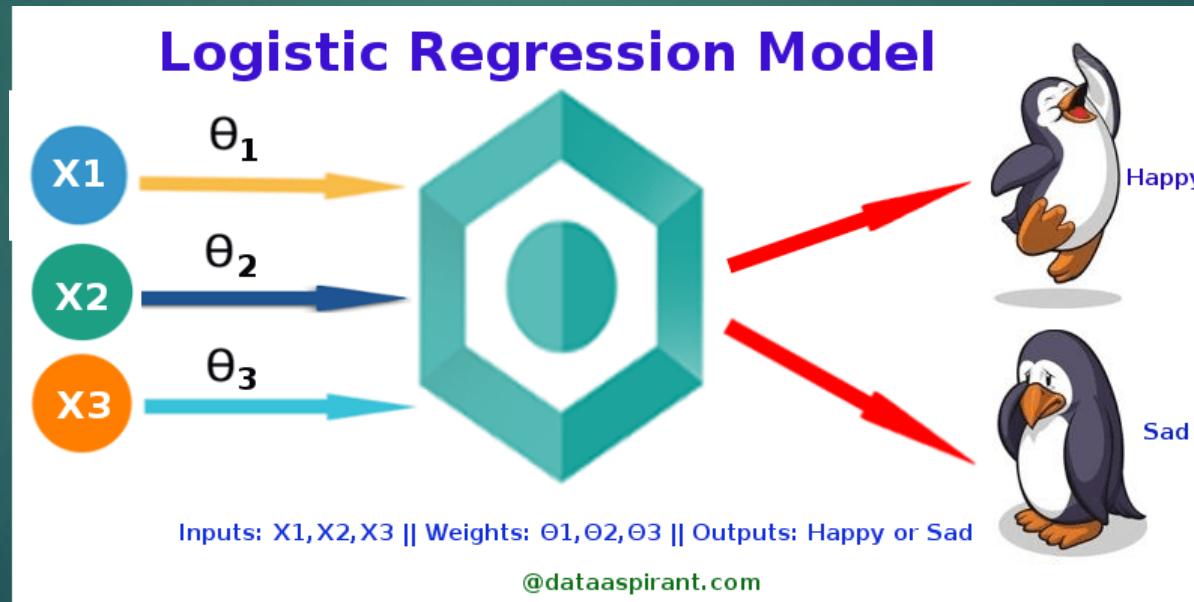
Mean Absolute Percentage Error

$$\frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

Supervised Learning - Classification

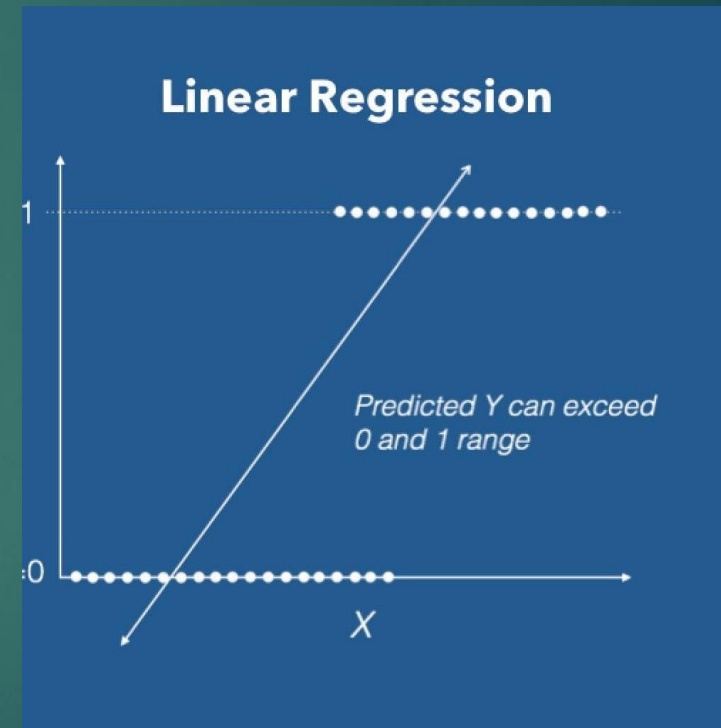
Logistic Regression

- Logistic regression is a classification algorithm used to assign observations to a discrete set of classes.



Compare Linear vs Logistic Regression

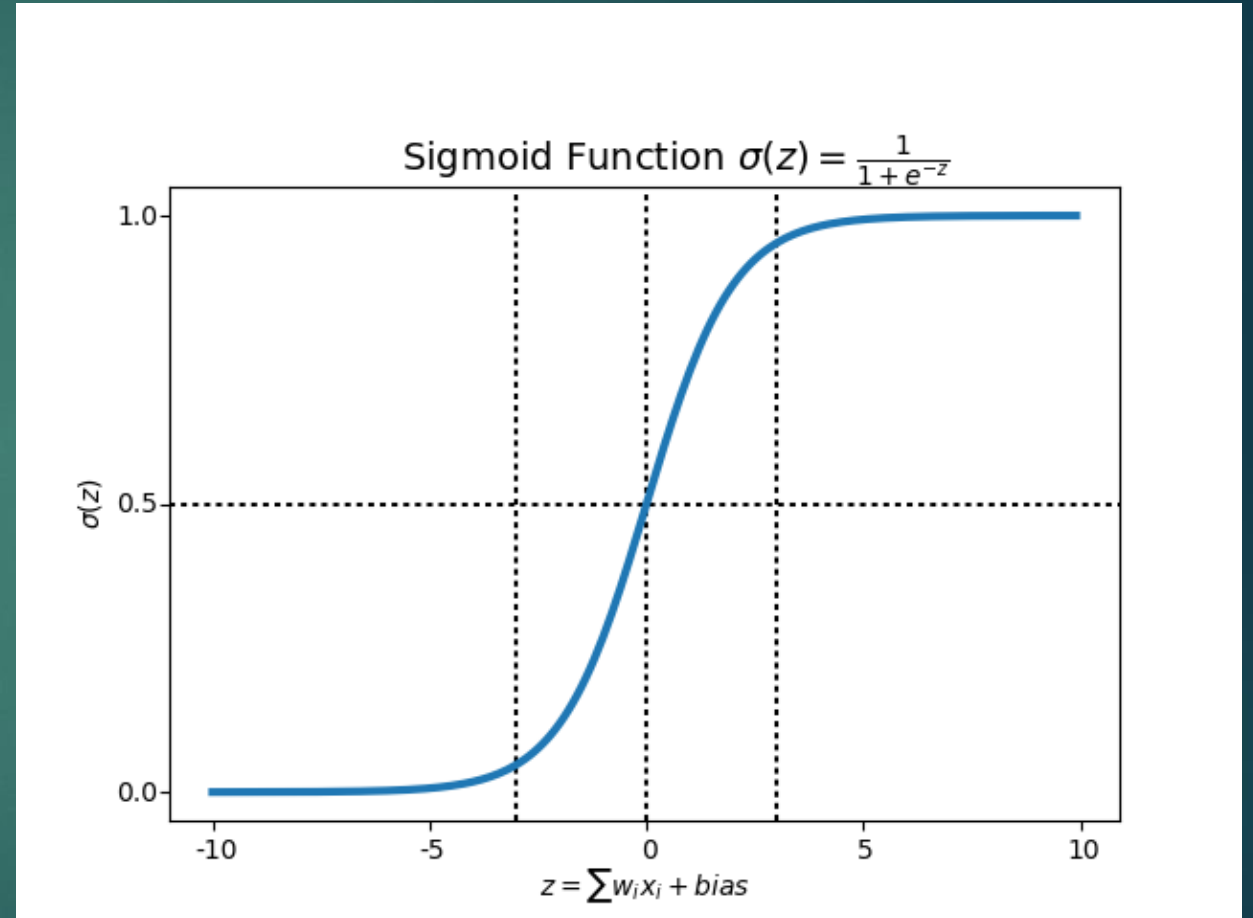
- For solving a fraud transaction problem, your conclusion can be either a yes(1) or no(0), but not a continuous value (1.5, 2.9, 5.6,)
- In linear regression we are trying to find a best fit line, but how can we solve classification problems.



Sigmoid Function

- To map the predicted values to classes we use sigmoid function in logistic regression.
- The function maps any real value into another value between 0 and 1.

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



Logistic Regression - Representation

Equation of the plane

$$h\Theta(x) = \beta_0 + \beta_1 X$$

Sigmoid Function

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

Final Equation of Logistic Regression

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

Log Loss is the negative average of the log of corrected predicted probabilities for each instance.

$$\log \text{ loss} = -1/N \sum_{i=1}^N (\log (P_i))$$

ID	Actual	Predicted Probabilities
ID6	1	0.94
ID1	1	0.9
ID7	1	0.78
ID8	0	0.56
ID2	0	0.51
ID3	1	0.47
ID4	1	0.32
ID5	0	0.1

Corrected Probabilities

$$\log loss = -1/N \sum_{i=1}^N (\log (P_i))$$

ID	Actual	Predicted Probabilities	Corrected Probabilities
ID6	1	0.94	0.94
ID1	1	0.9	0.9
ID7	1	0.78	0.78
ID8	0	0.56	0.44
ID2	0	0.51	0.49
ID3	1	0.47	0.47
ID4	1	0.32	0.32
ID5	0	0.1	0.9

Log for each Probabilities

$$\log loss = -1/N \sum_{i=1}^N (\log (P_i))$$

ID	Actual	Predicted Probabilities	Corrected Probabilities	Log
ID6	1	0.94	0.94	-0.02687
ID1	1	0.9	0.9	-0.04576
ID7	1	0.78	0.78	-0.10791
ID8	0	0.56	0.44	-0.35655
ID2	0	0.51	0.49	-0.3098
ID3	1	0.47	0.47	-0.3279
ID4	1	0.32	0.32	-0.49485
ID5	0	0.1	0.9	-0.04576

Log Loss is the negative average of the log of corrected predicted probabilities for each instance.

$$\log \text{ loss} = -1/N \sum_{i=1}^N (\log (P_i))$$

To find corrected probabilities.

1. Take a log of corrected probabilities.
 - $p(y_i)$ is the probability of 1.
 - $1-p(y_i)$ is the probability of 0.
2. Take the negative average of the values we get in the 2nd step.

$$-\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

Binary Cross-Entropy / Log Loss

Cost Function

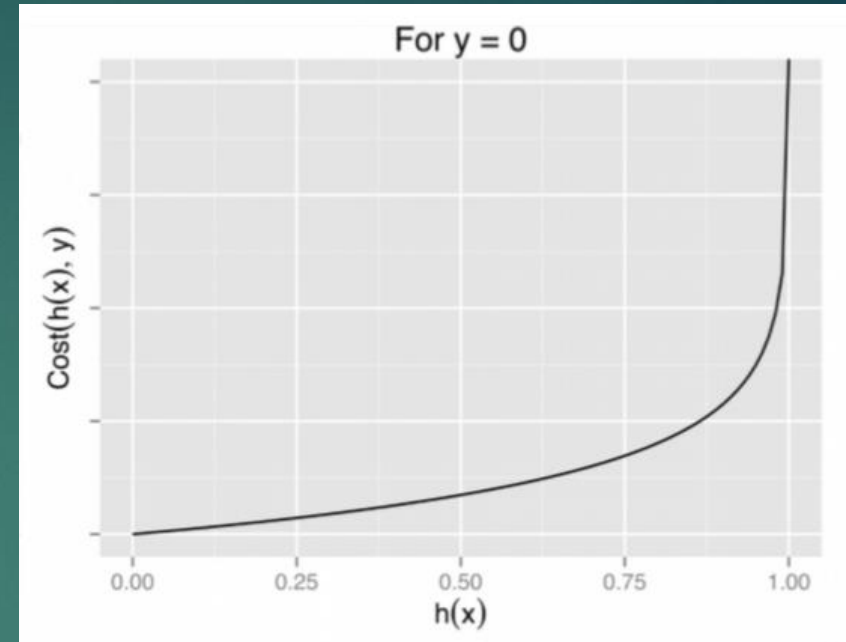
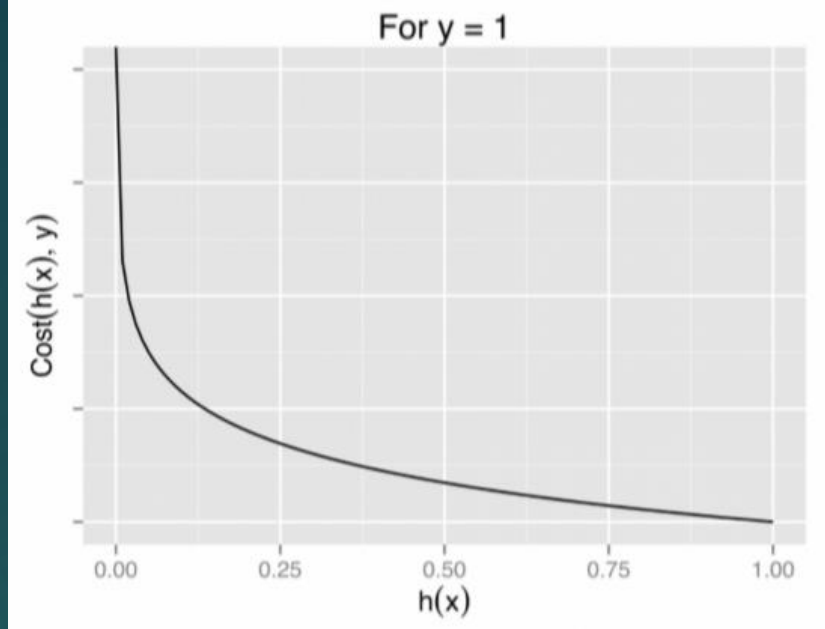
- Generalized Cost Function:

$$J(\Theta) = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} [h_{\Theta}(x^{(i)}) - y^{(i)}]^2$$

- This function would end up being a non convex function with many local minimums, in which it would be very difficult to minimize the cost value and find the global minimum.
- So, for avoiding the problem of local minima's we apply logarithmic function in cost function.

$$Cost(h_{\theta}(x), y) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$

Cost function of Logistic Regression



The above two functions can be compressed into a single cost function

$$J(\theta) = -\frac{1}{m} \sum \left[y^{(i)} \log(h\theta(x(i))) + (1 - y^{(i)}) \log(1 - h\theta(x(i))) \right]$$

Cost Function Optimization

- Optimization of the cost function in logistic regression is done by using Gradient Descent technique.
- The main goal of Gradient descent is to minimize the cost value. i.e. $\min J(\theta)$. So we run the gradient descent function on each parameter.

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Want $\min_{\theta} J(\theta)$:

Repeat {

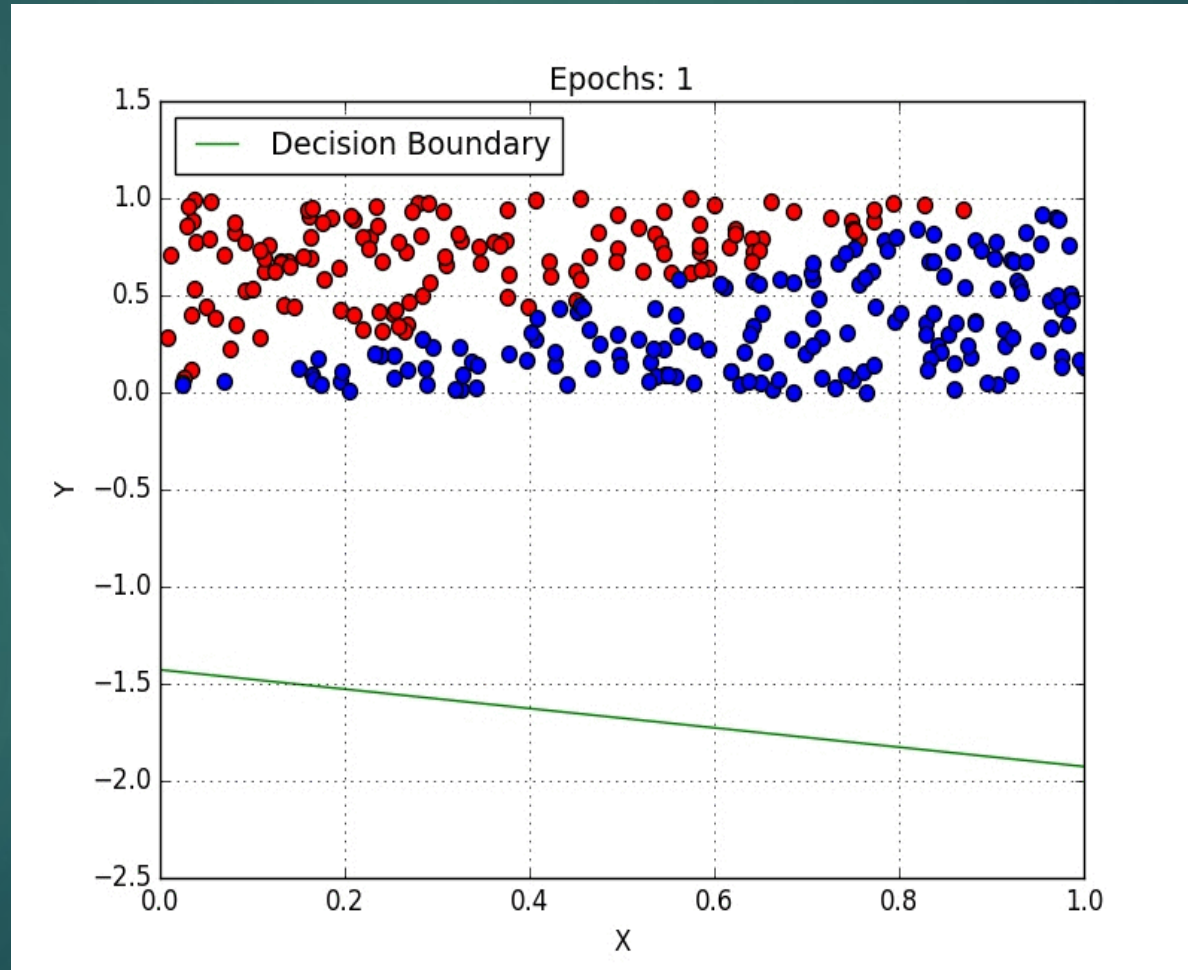
$$\theta_j := \theta_j - \alpha \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

(simultaneously update all θ_j)

}

After minimizing the cost function we save the weights associated with corresponding minima and evaluate the performance of the model.

Cost Function Optimization - Video



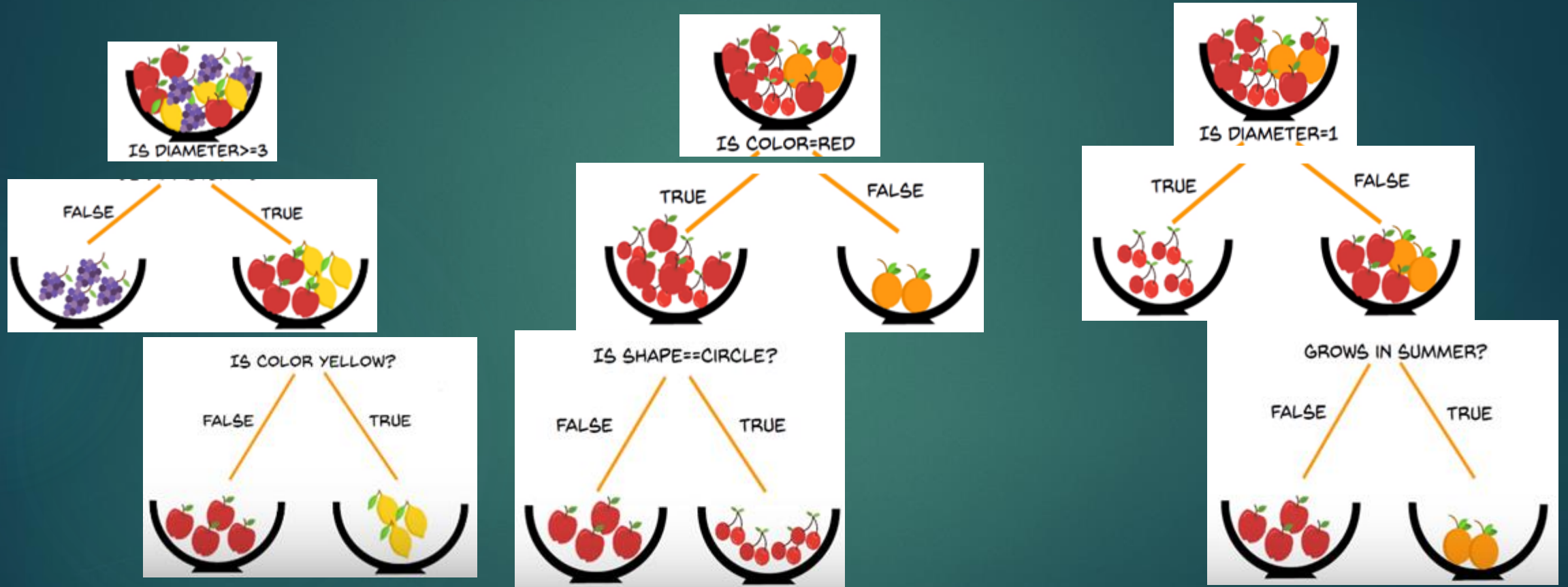
Decision Tree



TRAINING DATASET

COLOR	DIAMETER	LABEL
RED	3	APPLE
YELLOW	3	LEMON
PURPLE	1	GRAPES
RED	3	APPLE
YELLOW	3	LEMON
PURPLE	1	GRAPES

Decision Tree

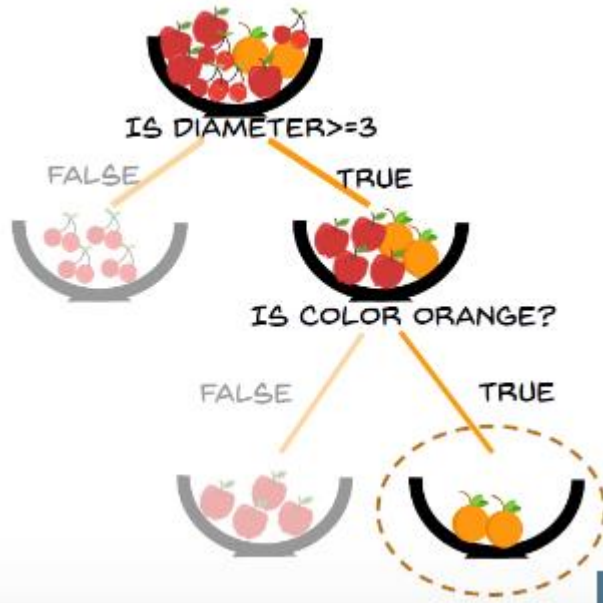


Random Forest Works

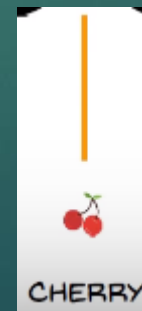
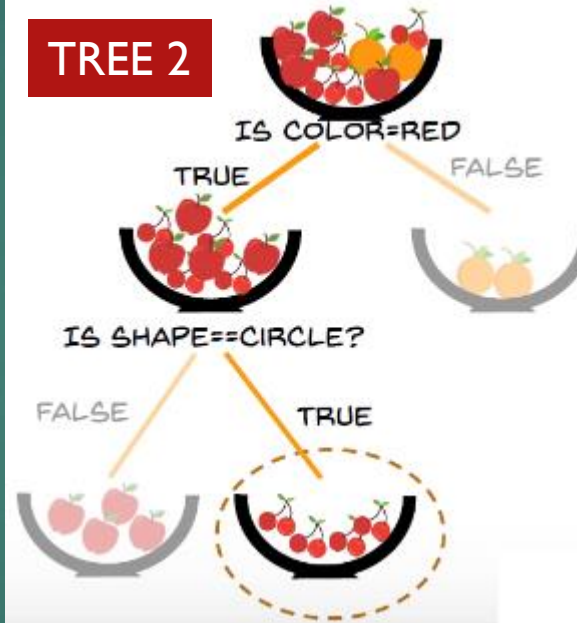


DIAMETER = 3
COLOUR = ORANGE
GROWS IN SUMMER = YES
SHAPE = CIRCLE

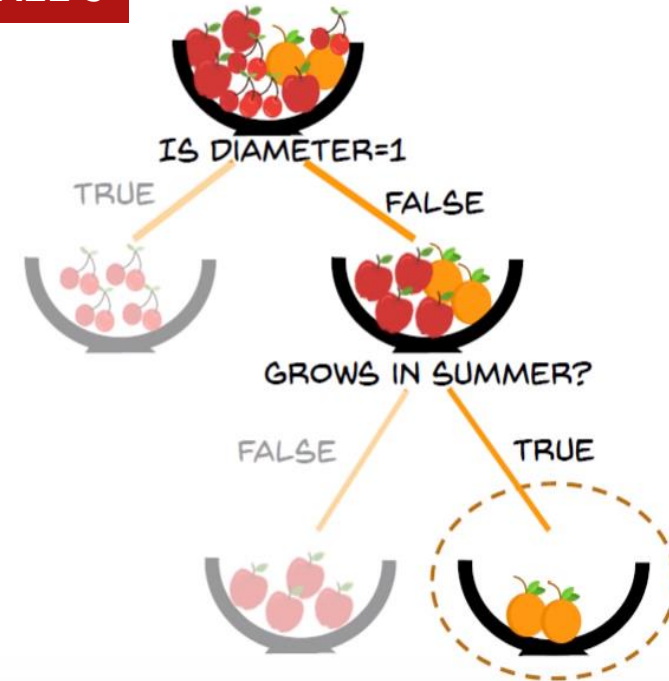
TREE 1



TREE 2



TREE 3







Based on majority voting

Performance Metrics

- Confusion Matrix.
- Accuracy.
- Precision.
- Recall.
- F1 Score.

Confusion Matrix

- Confusion matrix gives us the detailed report of the model output .
- It is extremely useful for measuring Recall, Precision, Accuracy.

		Actual Values		Predicted Values											
		1	0												
1	 TRUE POSITIVE	 FALSE POSITIVE TYPE 1 ERROR	<table><tr><th colspan="2">Actual Values</th><th>Negative (0)</th></tr><tr><th colspan="2"></th><th>0</th></tr><tr><td>1</td><td>FP</td><td></td></tr><tr><td>0</td><td>TN</td><td></td></tr></table>	Actual Values		Negative (0)			0	1	FP		0	TN	
Actual Values		Negative (0)													
		0													
1	FP														
0	TN														
0	 FALSE NEGATIVE TYPE 2 ERROR	 TRUE NEGATIVE													

Accuracy, Precision, Recall, F1

Accuracy is number of correct predictions made by the model over all kinds predictions made

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision is ability of a classification model to identify only the relevant data points or how precise our predictions are.

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

$$\text{Precision} = \frac{TP}{TP + FP}$$

Accuracy, Precision, Recall, F1

Recall is the ability of a model to find all the relevant cases within a dataset

		Actual	
		Positives(1)	Negatives(0)
Predicted	Positives(1)	TP	FP
	Negatives(0)	FN	TN

Recall = $\frac{TP}{TP + FN}$

F1 score is the harmonic mean of precision and recall

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Example

- ▶ Let's consider a use case where we need to identify the **terrorists trying to board flights**
- ▶ -- According to the data out of the 1000 passengers ,there are 19 confirmed terrorist passengers
- ▶
- ▶ **Solution I:** Designed a dummy model where tag every passenger as not a terrorist.
- ▶ 1) Accuracy: **98.10%** {1000 – 19 are correctly predicted} **But what about 19 terrorists?**
- ▶ 2) Recall: 0 {model is unable to find the relevant cases} **Recall is the ability to find the relevant data points terrorist in our case**
- ▶ Accuracy is a not good measure when the target variable classes in the data are nearly balanced or imbalanced.
- ▶ I have designed a model which finds all the terrorist in the data (Recall = 100 %) then
- ▶ **Why do we need Precision?**

Example

Solution2: Designed a dummy model where tag every passenger as terrorist.

1) Recall: 100% {model is able to find all the relevant cases} **Target is achieved but my predictions are wrong or not precise**

Precision can be calculated here

$$\frac{\text{terrorists correctly identified}}{\text{terrorists correctly identified} + \text{individuals incorrectly labeled as terrorists}}$$

Precision tells us how confident the model is about prediction

To build a perfect model we need a **tradeoff** between the both **Precision** and **Recall** which can be achieved by **F1 score** as it's a function of both.

Thank you