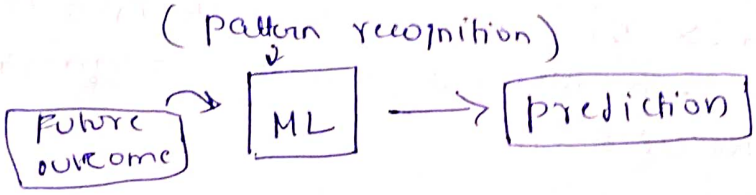
Machine Learning

Notes

By -> Rishabh Dwivedi

Machine - Learning

↳ ML is the ability of machines to learn from past experience & predict the future outcome.



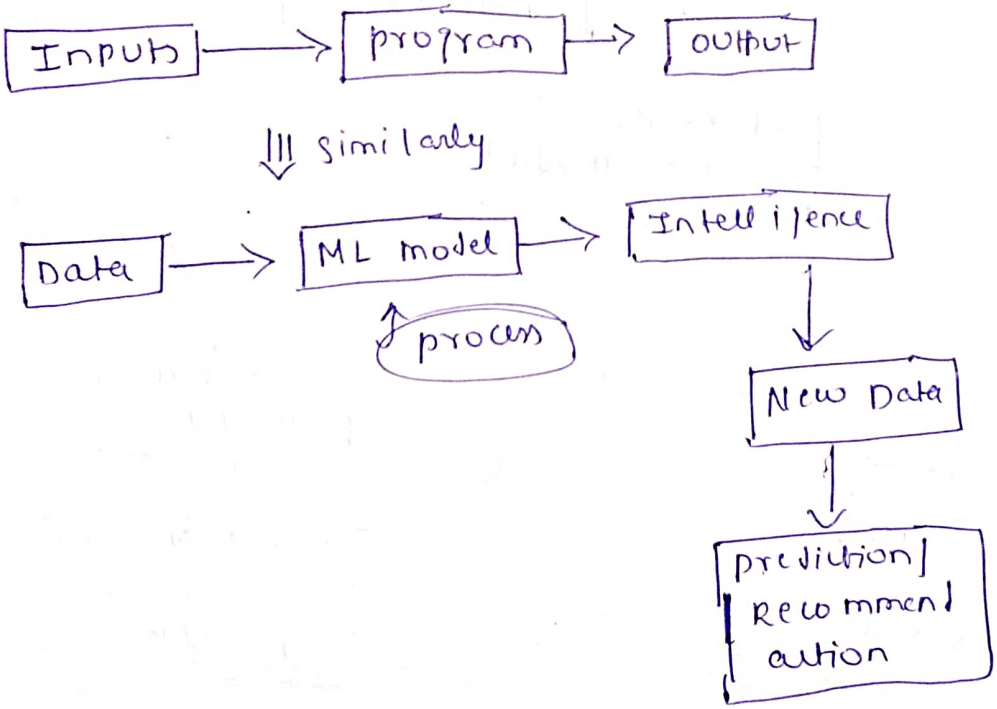
How it works ?

↳ past experience is not but DATA

↓ ?

Any information about anything.

Size of data can be huge
∴ For normal human being it is impossible to derive pattern from it.



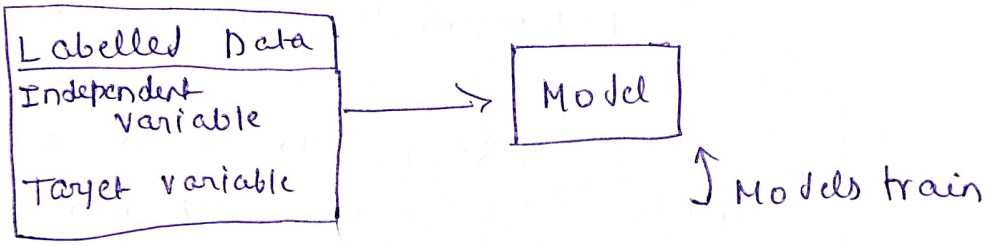
TYPE OF ML

- Unsupervised
- Supervised

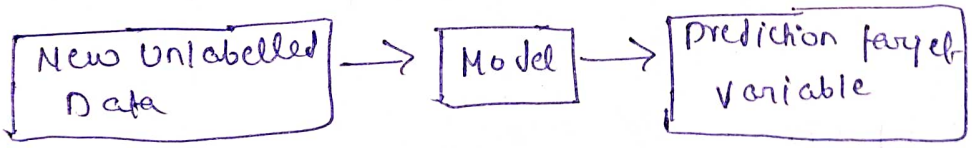
Labelled Data → The data which contains a target variable or an output variable that answers a question of interest is called labelled Data.

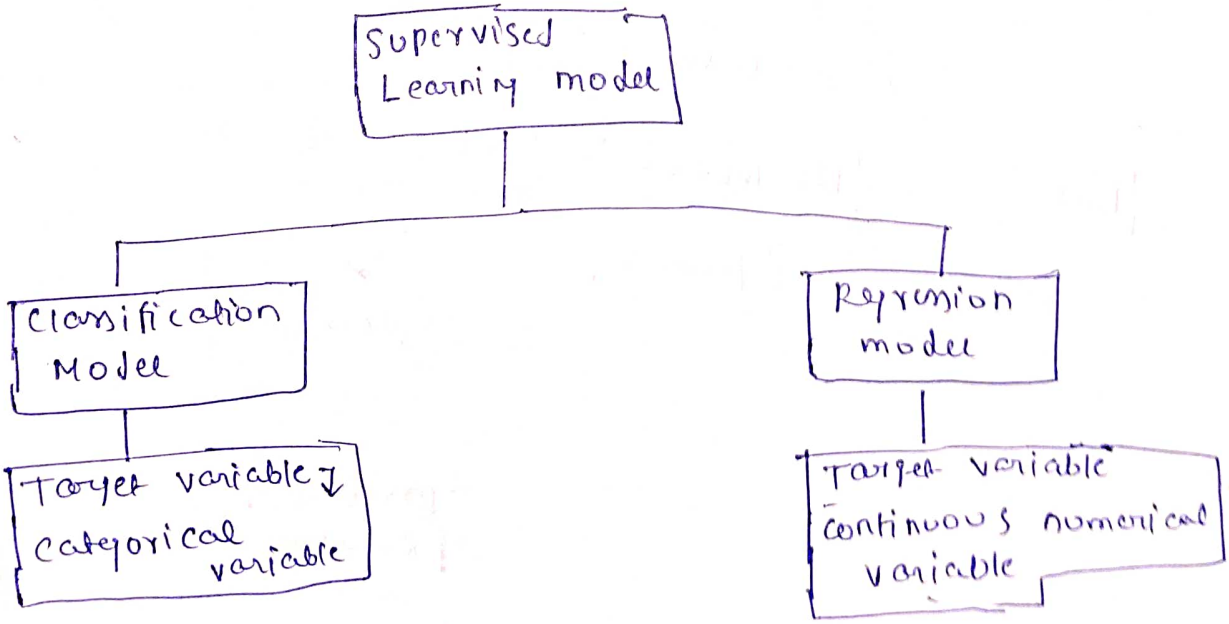
Unlabelled Data → UL Data is one which contains information about something but does not have a predefined target variable.

① Supervised ↙



∴ Now its a time for testing



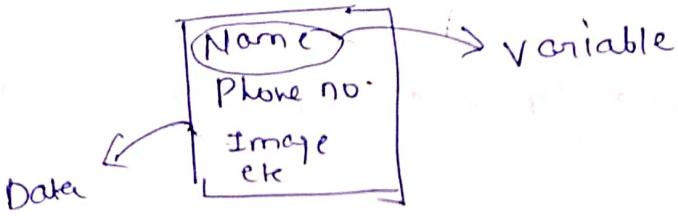


Unsupervised learning

↓

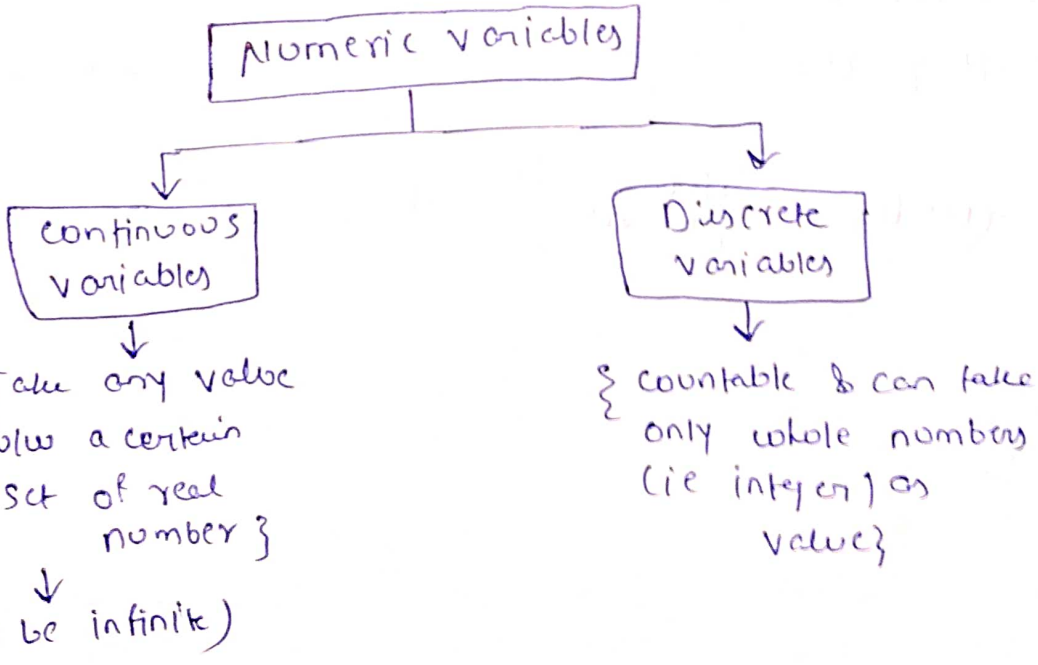
- -> complement of supervised learning
- -> There is no target variable involved.
- -> works on only unlabelled data
- -> Identifies if some pattern exist in the data.

-> **Variables** [Can take any value -> No. or String]



* A variable represents one specific characteristics of the data or tell one specific information about the data under consideration.

- -> Numeric or Quantitative variable
  - -> {Measured or counted}
  - [Numbers -> or Eg 1, 2, --
    (Also known as quantifiers)}
- -> categorical or Qualitative variable
  - -> {can't be measured}
  - [Expresses a feeling or a characteristics]



Categorical variables

- Ordinal variables → { Arranged in logical order } → Eg: Poor < Average < Good
- Nominal variables → { can't be ranked based on their values } → Eg: Colors

Independent Variables

↳ are ones whose value don't depend on any other variables.

Dependent variables

↳ whose value depend on other independent variable & cannot be changed easily.

Structured data → Arranged in structured format.

Roll no	Name
300	Rishabh
500	Rohit

Unstructured data → Not arranged in tabular form.

↑ Eg: Msg received in chats.

=> Graphical representation of data.

↓

Data Analysis

↳ Process of studying the available data and drawing valuable insights or information from it with the help of a software.

Sub	Marks
A	20
B	30
C	60

↝ Analyse it

↳ various softwares like

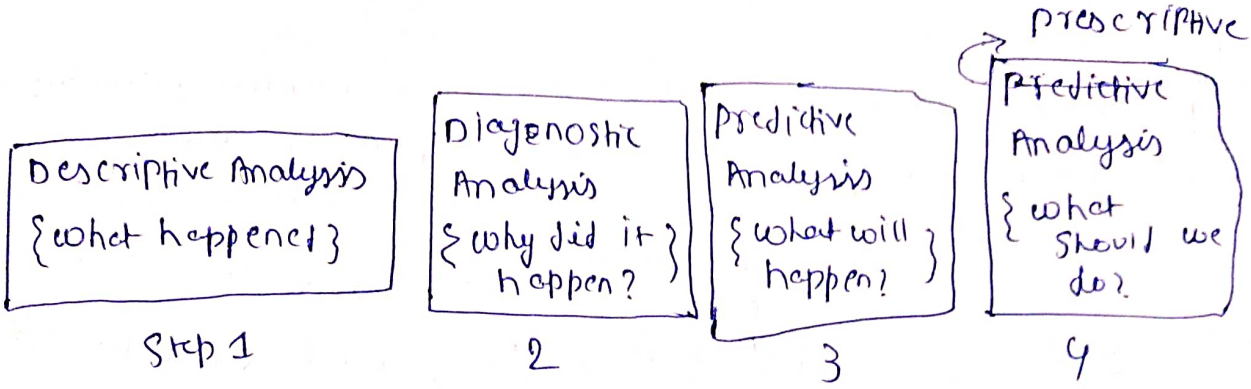
↳ python, R, Excel, Matlab etc.

↝ It's a part of ML which analyses your data & bring out insights before you apply your algo.

↳ Exploratory Data Analysis (EDA)

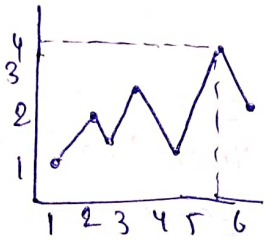
↳ To find out the pattern & relationship among the data variables.

* EDA gives summary about dataset.

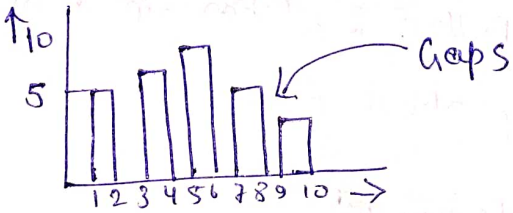


Graph

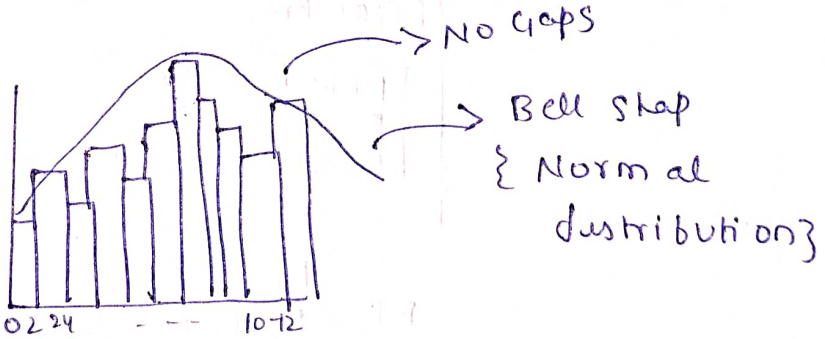
① Line Graph



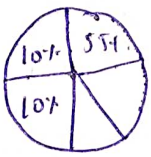
② Bar Graph



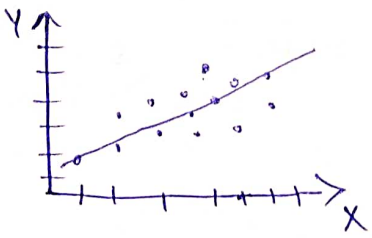
③ Histogram



④ Pie chart

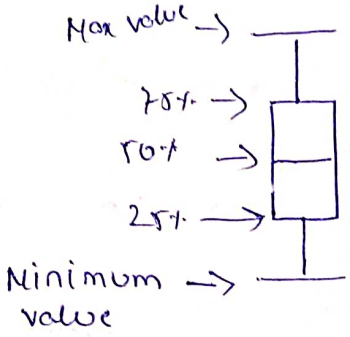


⑤ Scatter Plot

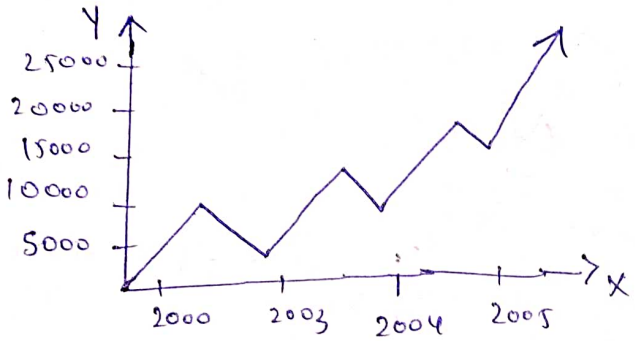


⑥ Box & whisker plot.

↳ Graphical representation of the statical summary of numeric variables.



⑦ Time series



* Some python coding *

↳ Numpy, pandas, Matplotlib & seaborne.

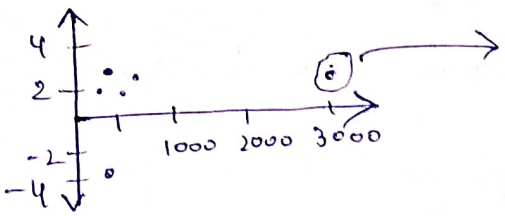
=> Descriptive Statistics.

-> count of values for a variable
-> Mean
-> Standard deviation
-> Minimum value
-> Maximum value
-> Three percentiles — 1st, 2nd & 3rd

-> DS is calculated only for Numerical variable only.

↳ Uses of DS

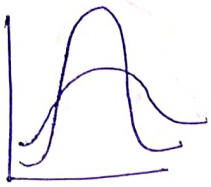
(i) Build a better understanding about data
(ii) Identify & treat missing values.
(iii) Identify any anomalies



* Standard deviation tells the spread from the mean or average value or expected value.

↳ If SD is low → most of the no. are close to the average value.

↳ If SD is high → most of the no. are spread out.



model.describe()

↳ count, mean, std, min, 25%, 50%, 75%, max

* Calculating SD using pandas & Numpy will show different result because ↓

SD using pandas

↳ Formula used

$$SD = \sqrt{\frac{\sum (x-\bar{x})^2}{n-1}}$$

Each value
- x = mean in the data set
- $\bar{x}$ = Mean of all value
- n = no. of values in dataset

SD using numpy

↳ Formula used

$$\sigma = \sqrt{\frac{\sum (x-\bar{x})^2}{n}}$$

Fix it with the help of degree of freedom (ddof) = 1

↓

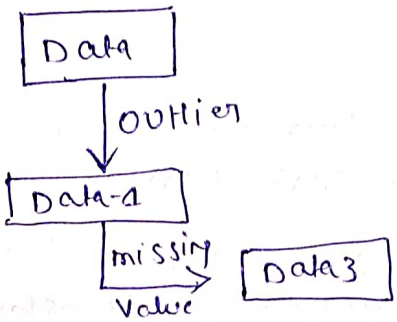
np.std(model['sale price'], ddof=1)

↳ $\sigma = \sqrt{\frac{\sum (x-\bar{x})^2}{n - \text{degree of freedom}}}$

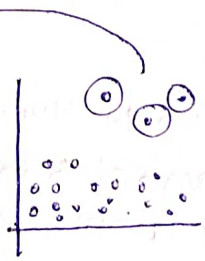
=> Data Exploration

↳ why?

↳ Their may be anamolies / outlier or missing value present in the data, which can create wrong prediction. So to avoid it we perform data exploration.



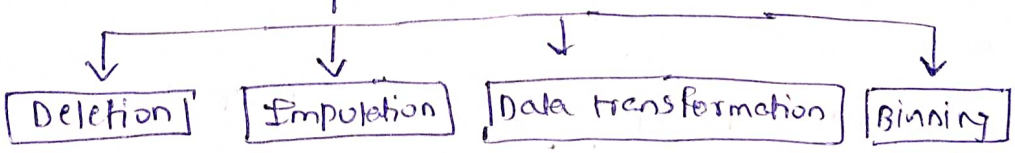
Outliers ?



An outlier is a data point that is distinct from other data point.

∴ Any data point which is higher than upper limit or lower than lower limit is as an outlier.

Treating outlier



↑ For target variable, independent variable

Missing value

↳ NaN form

Treating outlier Missing value

Variable transformation (Vt)

Vt refers to the process of making changes to a variable in a way that it become more useful and meaningful for analysis & modelling process.

why Vt ?

(i) Outlier treatment

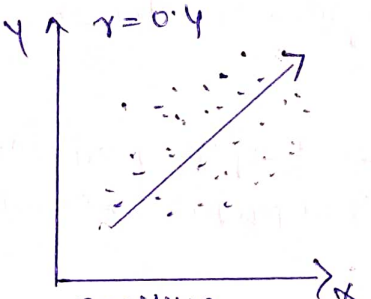
(ii) when a variable does not represent the information it contains in the best possible ways.

(iii) To derive a new variable from an existence variable or combining two variable to transform a variable so that it become more useful.

⇒ correlation ~> {Data exploration for Numerical variable}

↳ It is measure of dependence or association between two variables ie - how does one variable change with change in other.

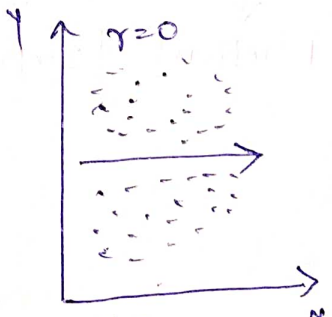
↳ correlation refers to how close two variables are to have a linear relationship with each other.



Positive correlation

↓

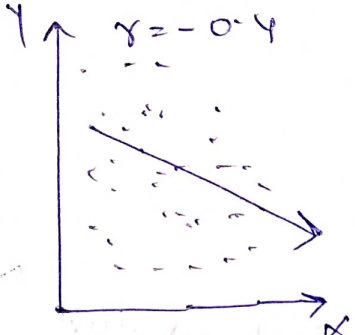
If 1 variable inc. other inc. to.



No correlation

↓

Random changes & no pattern.



Negative correlation

↓

with increase in 1st variable other variable decreases.

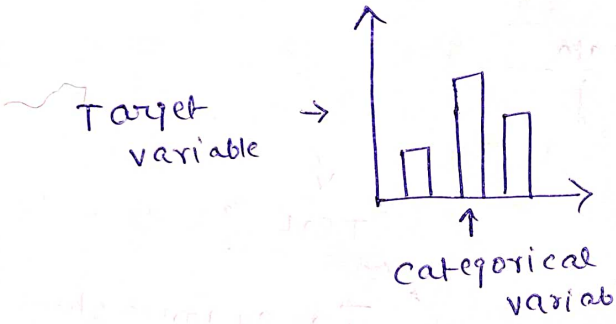
* Any random value b/w -1 to +1.

Eg: model['Sale Price'].corr(model['Flat area'])

Using numpy

↳ np.corrcoef(data[--], data[--])

* only those variables should be selected for building the model which have high correlation (above a certain cut-off value), positive or -ve, with the target variable.



* The method of finding out whether a categorical variable is having a impact on the numerical variable — ANOVA

↓

Analysis of variance

ANOVA

↳ It checks if the mean ie Average value of the target variable across different levels or unique values of a categorical variables are equal or not.

2 types of ANOVA

1-way ANOVA

↳ It deals with one independent variable only.

2-way ANOVA

↳ It deals with two or more independent variables.

Splitting of data into train - test

↳ Before splitting, we need to seprate the dataset into on the basis of dependent & independent variable

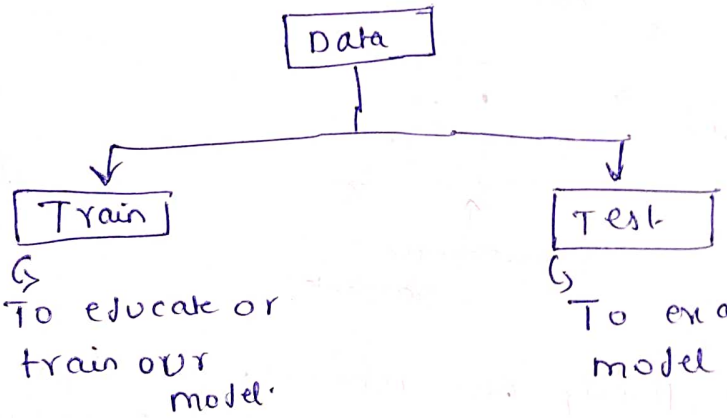
↓

For doing this we per use "iloc" function

Eg: `X = data.iloc[:,0]` ↳ dependent

`Y = data.iloc[:,1:3]` ↳ independent

Now train & test

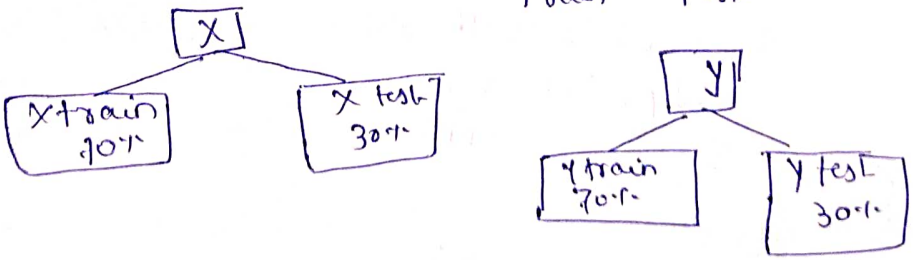


* The data on which a model is built is called training data.

* once the model is trained, it is examined as to how well it has learnt using another subset of the original data which is called Test Data.

ideal split data ratio = 70 : 30

Train Test



=> Feature Scaling

↳ Scaling the feature variables (ie all the independent variable) into the same range.

* Basically sometimes the variation in magnitude or range may be present in dataset.

↓ problem it can create

- -> Variable with higher magnitude will have more impact compare to one which is smaller.
  - -> Model might not predict properly because of unequal weight.

- -> Gradient descent algorithm that is used to find the coefficient of linear ~~algo~~ regression may take a long time to converge.

∴ With Feature scaling this problems will Overcome.

↳ The variables are scaled to have similar magnitude or range so that model is not biased towards a particular variable.

↳ Required for algo like -> Linear regression, Logistic Regression, K Nearest Neighbour etc

↳ Not required for tree based Algo like -> Random Forest, Decision tree etc.

How to perform ?

- -> Standardisation { Standard Normal Distribution => mean -> 0, SD -> 1 }
- -> Min Max scaling { lies b/w [0 1] }
- -> Normalisation { Range -1 to 1 }

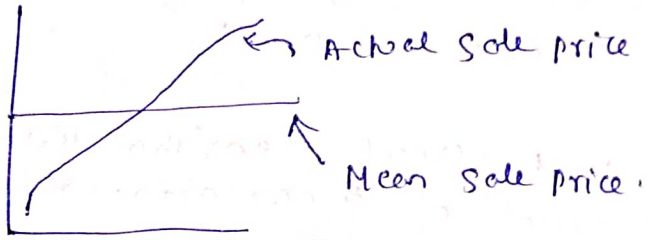
Standd. -> $\frac{x - \mu \text{ (Mean)}}{\sigma \text{ (SD)}}$ | Min max -> $\frac{x - \min(x)}{\max(x) - \min(x)}$ | Not -> $\frac{x - \text{mean}}{\max(x) - \min(x)}$

Linear-Repression (Algo-1)

↳ Before moving, first understand some common terms & error.

(i) Build your first predictive model with mean prediction.

↳ Mean will come out to be same for all rows.
Hence mean sale price is same for even higher price house or lower price



↓

To improve this model, take grade wise mean.

Now plot residual plot for both.
(Mean-sale, Mean-grade_sale)

↓ ↓

More scatter Less scatter

→ Model evalution Metrics
(calculating errors)

① Mean error

↳ Mean of whole model -- but but but it will come out to be zero -- why? because some value are +ve & some are -ve, hence overall zero.

↓

To over come we have absolute error.

② Mean Absolute Error.

Take the absolute value of each individual error term, sum it up ~~all the~~ for all the data points & then take a mean.

$\hat{y}$ = predicted value

y = Actual value

n = Total number of data points.

$$MAE = \frac{1}{n} \times \sum_{i=0}^{n} |\hat{y} - y| \quad \{ \text{or } |y - \hat{y}| \}$$

→ import mean_absolute_error.

③ Mean Square error (MSE)

↳ Same concept, but greater penalties for higher value.

$$MSE = \frac{1}{n} \times \sum_{i=0}^{n} (y - \hat{y})^2 .$$

④ Root mean square error (RMSE)

↳ Gives the expected average distance b/w the actual points & the prediction.

$$RMSE = \sqrt{MSE}$$

Now

MAE = 0 ⟶ 100% accurate

MAE = 250,00 ⟶ How much % accurate

∴ To describe it we have concept of R^2.

⇒ R^2

It give us the Relative error of a regression model wrt the simple mean regression model.

$\bar{y}$ = IS the mean of the "Sale Price"

$\hat{y}$ = IS the predicted values from a regression model.

y = Represent actual "Sale-Price"

$$R^2 = 1 - \frac{\sum_{i=0}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=0}^{n} (\bar{y}_i - y_i)^2}$$

→ Represent n x (MSE of regression model we count to evaluate)

↓

Represent n x MSE of simple mean model.

$$R^2 = 1 - \frac{MSE(model)}{MSE(mean)}$$

$\frac{MSE(model)}{MSE(mean)}$ → The lower this ratio is, the better is the model is.

when $\hat{y} = y$, then $R^2 = 1 \rightarrow 100\%$ or perfect model

if $\hat{y} = \bar{y}$, then $R^2 = 0 \rightarrow$ For mean prediction value $R^2 > 0$

* For any model which is better than mean prediction model but is not a perfect model.

$$0 < R^2 < 1$$

$$0 < R^2 < 100\%.$$

In reality

$-\infty < R^2 < 1$

↓

model we are trying to evaluate is even worse than the mean regression model.

* R^2: The degree to which your input variable explains the variation of your target or output variable.

$R^2 = 0.8 \rightarrow$ 80% of the variable in the output variable is explained by the input variable

* The higher the R^2 is, the more is the variation explained by the input variable & hence better is the model.

* As more input variable are added to the regression model . The value of simple R^2 can no longer be relied upon

* The best linear regression model is the one that has the least error.

* Parameter of Linear regression

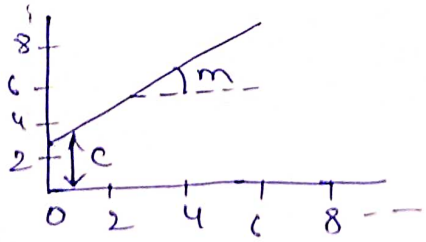
$Y = mX + c$ where $m, c \rightarrow$ parameters of lines.

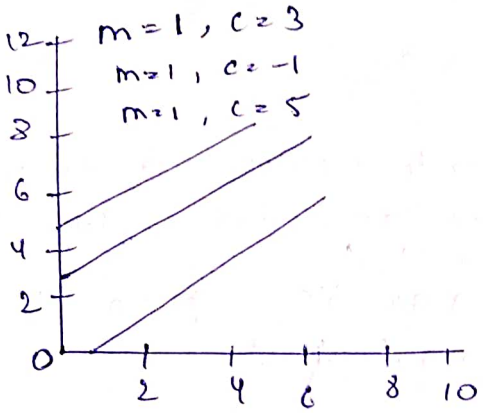
Y = Dependent variable

X = Independent variable

m = slope

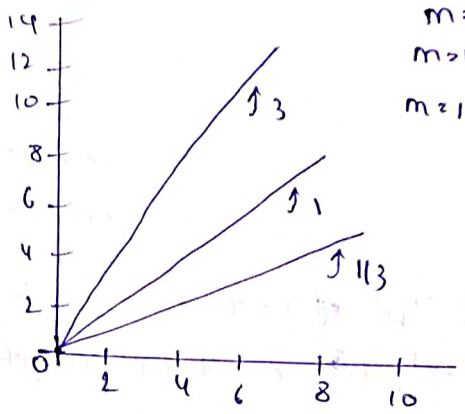
C = Intercept (what point the line cuts the Y axis)





↓

Value of 'c' increases, line shifts upwards but angle of line does not change.



As the value of m increases, the rate at which the value of Y increases with respect to X also increases.

=> Gradient Descent

It is an optimization algorithm that works iteratively & aims to find the minimum value of a conver function with respect to a set of parameters.

↳ Just like hit & trial.

-> Following steps occurs.

Step-1: Random initialization

↳ Take random value of 'm' slope & 'c' intercept.

To avoid everything from scratch, we take value of slope $m = 0.1$

c = mean of "target value"

Step-2: Generate prediction (using initialize value)

$$\hat{y} = mX + c$$

Step-3 : Calculating the cost.

$$L = \sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}$$ { wrt actual point } → MSE

$$L = \sum_{i=1}^{n} \frac{(mx_i + c - y_i)^2}{n}$$

Step 4 : Updation of parameters

$\{ m_{new} = m_{old} - Zm \}$

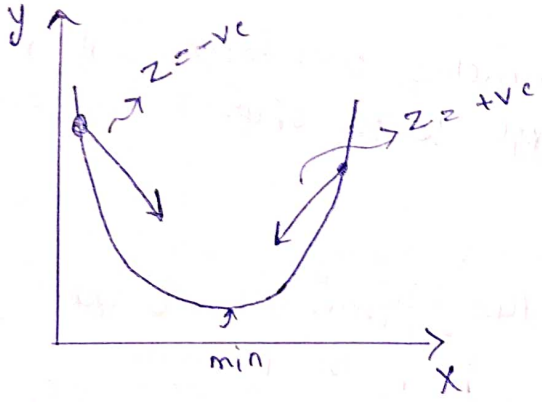
$\{ C_{new} = C_{old} - Zc \}$

[Z → Still unknown but can be positive or -ve]

↓

If Z is +ve, parameter will decrease

If Z is -ve, parameter will increase.



But, are we here

how to know on which side of the minimum cost we ~~here~~

on & what value should ~~be~~ we take for Z?

↓ soln

Partial differentiation

$$\frac{\partial(T)}{\partial m}$$

∴ Partial derivatives of the cost function wrt m & c

$$G_m = \frac{\partial(T)}{\partial m} = 2 \sum_{i=1}^{n} \frac{(x_i m + c - y_i)(x_i)}{n}$$

$$G_c = \frac{\partial(T)}{\partial c} = 2 \sum_{i=1}^{n} \frac{(x_i m + c - y_i)}{n}$$

∴ $z_m = \alpha G_m$ & $z_c = \alpha G_c$

∴ $m_{new} = m_{old} - \alpha G_{mold}$ [G → Gradient]

$c_{new} = c_{old} - \alpha G_{cold}$

α = learning rate = a constant

α → High ⇒ cost function may explode & bounce out far from the minima.

α → Low ⇒ parameters will take lot of iteration to converge to the optimal values.

If $\alpha = 0.01$

Previous value of the cost function. − New value of the cost function < The threshold value. (usually 10^{-x})

↓

The algorithm is converge

Logistic Regression (Algo-2)

↓

Classification Algo

↓

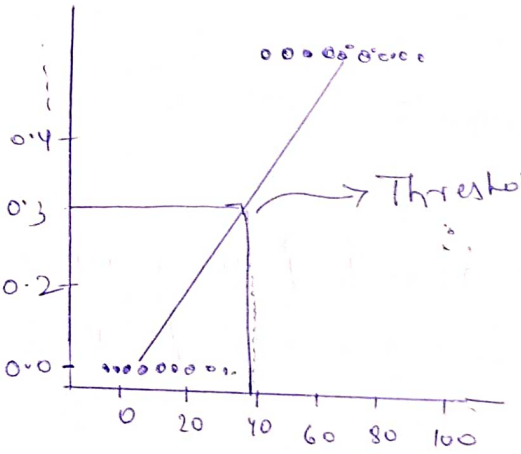
A categorical target variable is used.

↵

A categorical variable is one which represent a characteristic which cannot be measured or counted.

(Q) If logistic regression is a classification algo., then why is there the term "regression" in its name.

↳ { Think, Think, Think -- will answer afterwards }



↵ classification problem

? why not linear regression here

- susceptible to the outlier. → If outlier present then Threshold point can be shifted.
- Interpretation of the model.

} Issues

Here we will be working with 'class' categories hence its probability lies in a range of 0 & 1.

Eqn for linear regression is

$$z = mx + c$$

↓

In order to restrict this value blw 0 & 1 we use Logit (or Sigmoid) function.

The Logit function

↳ A func. which have a range blw 0 & 1, irrespective of any input, it will always give output blw 0 & 1

Given as:

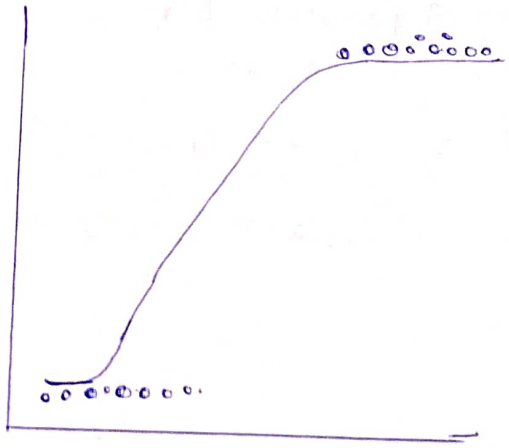
$$g(x) = \frac{1}{1 + e^{-x}}$$

$$\therefore \hat{y} = g(z)$$

$$\hat{y} = \frac{1}{1 + e^{-(mx+c)}}$$

-> If $(mx+c)$ is high then overall $\hat{y}$ comes out to be 1

-> If $(mx+c)$ is less (or -ve) then overall $\hat{y}$ comes out to be 0.



'S' shaped graph

Hence the main reason for using term "regression" is:

-> prediction continuous probability values b/w 0 & 1
   - Regression.

Eg:

probability of 0.82 -> means he belongs to class-1.

-> Loss function

-> In this tech. we use log loss function.

$$J = \sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}$$ -> For linear regression

$$J = -\frac{1}{n} \sum_{i=1}^{n} y_i \cdot \log(\hat{y}_i) + (1-y_i) \log(1-\hat{y}_i)$$

where

$\hat{y}$ = predicted probability of class 1

y = Actual class

n = Number of observation

* -> probability b/w 0 & 1

-> Middle point - 0.5 (Threshold)

-> If probability > 0.50, the observation is regarded as class 1

-> If probability < 0.50, the observation is regarded as class 0.

→ Evaluation Metrics

① Confusion Matrix

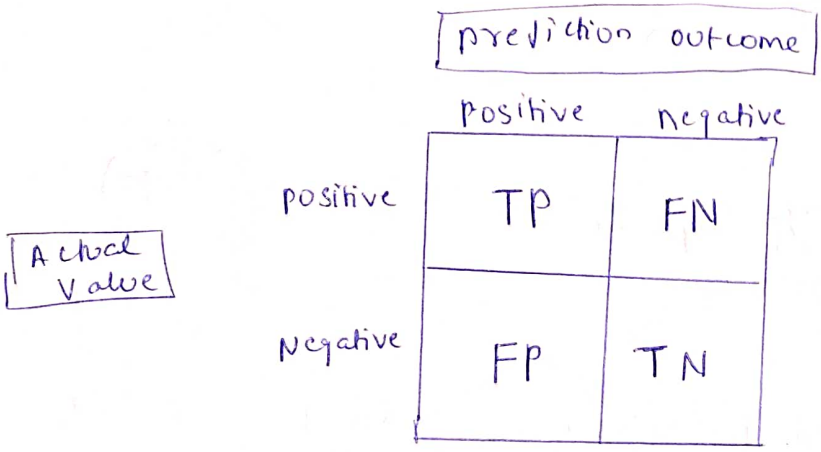
It is used to interpret the model predictions systematically

→ It is nxn matrix, n → no. of distinct classes in the target variable.

→ For 2x2 matrix

↳ Class 1 : Positive class

Class 0 : Negative class



* Confusion matrix serves as the basic platform of representation for most of the classification metrics.

* The metrics which derives from the confusion matrix are:-

① Accuracy

② precision

③ Recall

* Precision & recall handle the imbalanced dataset efficiently.

$$\text{precision} = \frac{\text{Predictions Actually positive}}{\text{Total predicted positive}}$$

$$\Rightarrow \frac{TP}{TP + FP}$$

Precision :

It is used when avoiding false positives is more essential than encountering false negative.

Eg:

4	2
8	486

$$\rightarrow \text{precision} = \frac{4}{4+8} \simeq 33\%.$$

∴ Model having 98% is only 33% precise

$$\text{Recall} = \frac{\text{Predictions Actually positive}}{\text{Total Actual positive}}$$

$$\Rightarrow \frac{TP}{TP + FN}$$

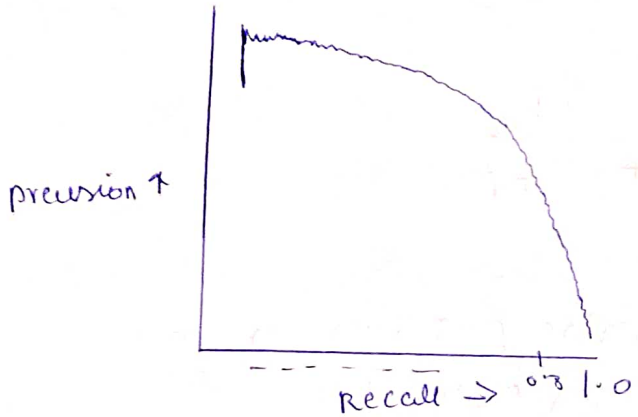
From Above Eq

$$\Rightarrow \frac{4}{4+2} \simeq 66\%.$$

only 66% of actual +ve value were predicted as +ve.

∴ Recall

It is used when avoiding false -ve is prioritized over encountering false +ve.



If precision ↑ then Recall ↓.

But if we are not sure from graph (or metrics ~~method~~) then we use F1 metrics which is nothing but harmonic mean of precision & Recall.

$$F1 = \frac{2}{\frac{1}{\text{precision}} + \frac{1}{\text{recall}}}$$

↓

Max. when precision = Recall.

? But if precision, Recall or F1 score yield the same result, then how do we distinguish which one is better.

↓

we use **log loss**

↓

cost fun. of Logistic regression.

* The smaller the value of log loss, the better the model is.

⇒ The farther the points from actual probability (0 & 1) the more is the loss function.

⇒ AUC - ROC curve

It is a performance measurement for classification problems at various threshold settings.

AUC - Area Under Curve

ROC - Receiver Operating Characteristics.

* Higher the AUC, the better the model is distinguishing between 0 and 1.

(AUC ROC often used interchangibility)

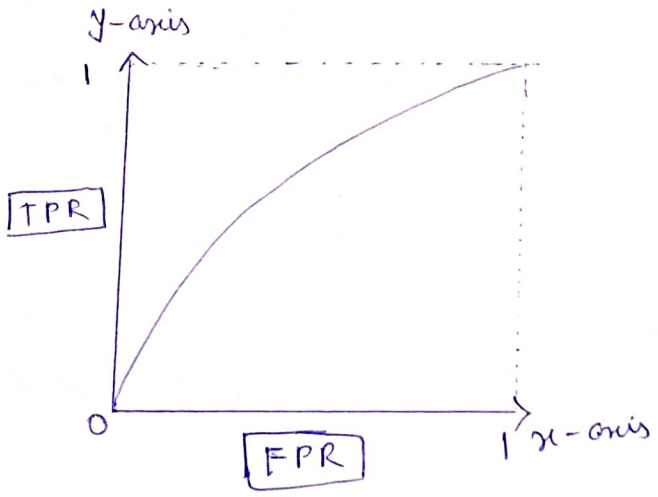
* AUC - ROC works well for balanced data set.

-ve ≃ +ve

* AUC - ROC is not suitable for imbalanced data.

-ve > +ve

(90%) (10%)



Area = 1 unit

FPR (False +ve rate) = $\frac{FP}{TN+FP}$

TPR (True +ve rate) = $\frac{TP}{TP+FN}$

] - b/w 0 & 1.

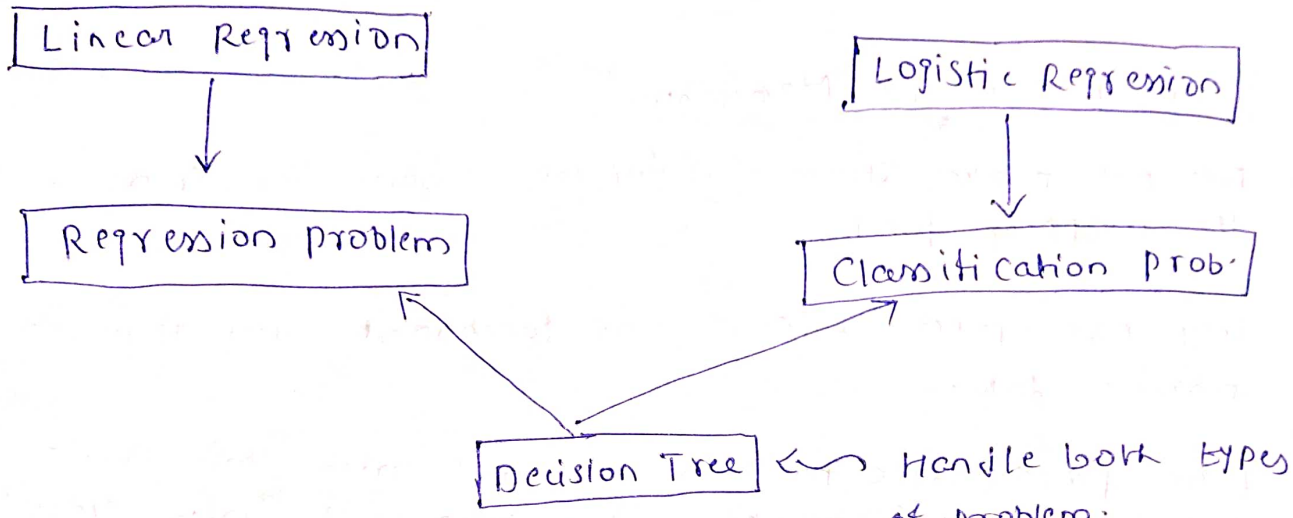
* on this method, create a plot at different threshold & calculate the area.

The higher the AUC-ROC, better is the performance of model.

* Remember ↓

If AUC-ROC > 0.95, there could be something wrong with the model or the dataset.

Decision - Tree (ML/0-3)



Linear & Logistic Regression : Make restrictions or assumptions.

Decision tree : does not make or assumptions.

↳ Predictive model

↓ 2 types.

(i) Parametric

(ii) Non parametric

(i) Parametric

-> Make strong assumptions about the form of the mapping function.

-> very simple & interpretable.

-> set of parameters does not depend upon the amount of data.

* A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called parametric model.

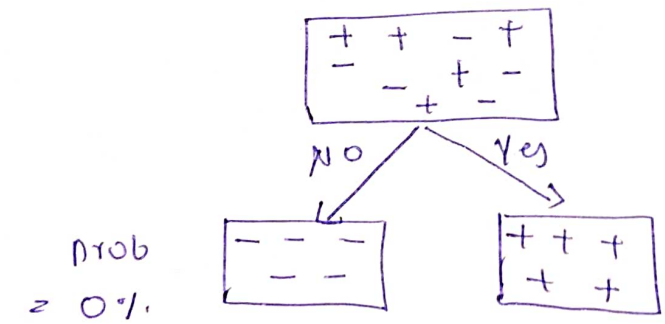
Linear reg. ┐
Logistic reg. ┘ → parametric Alg.

② Non parametric Algorithm

- Do not make strong assumptions about the form of the mapping function.
- They are free to learn any functional form from the training data.
- Non parametric methods are good when you have a lot of data & no prior knowledge and when you don't want to worry too much about choosing just the right features.

⇒ Decision Tree

- Decision to seprate the classes
- All the +ve to be one side & all the -ve on the other side.



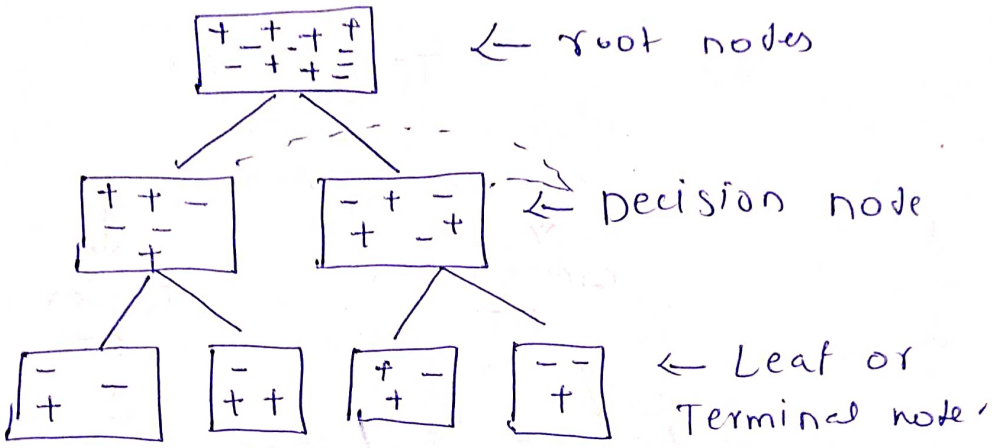
* An ideal care - Pure node after split

↓

A pure node is in which all the data point exhibits the desired behaviour or have the same class.

∴ The objective of decision tree : Is to have pure nodes. (Or close to pure nodes)

→ Various terminology



(i) Splitting : A process of dividing a node into 2 or more nodes.

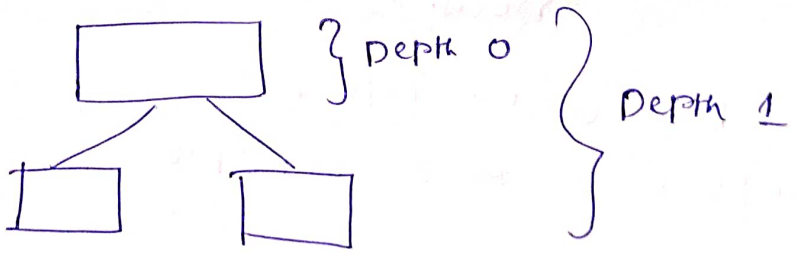
(ii) Decision Node : A sub node that is furthur divided into sub nodes. (Initial Sub Node)

(iii) Leaf Node : A node which does not split furthur.

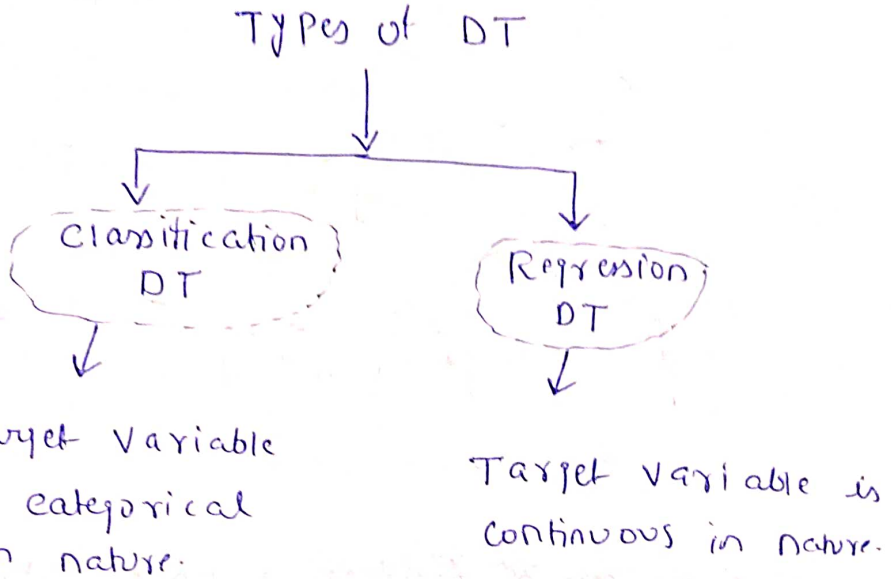
(iv) Parent Node : A node which is divided into sub nodes.

(v) Child Node : A sub node of a perent node.

(vi) Depth of tree : The length of the longest path from the root node till the leaf node.



→ Logic behind DT



↳ Criteria to be used to split a decision node
(i) Gini impurity
(ii) Information gain using entropy.

(i) Gini impurity

↳ It measures the impurity of node.

$$GI = 1 - gini$$

gini → It is the probability, whose ranges are from 0 to 1

↳ Higher the value, the more will be the purity of the node. Lesser the value means lesser purity nodes.

↳ Property of Gini impurity

ⓐ Helps to decide the best split.

ⓑ GI = 1 - Gini

GI → impurity
Gini → purity.

(c) Works only for categorical variable & Not for continuous variable.

(d) Always performs a binary split.

↳ **Gini impurity = 1 − gini**

$$Gini = (P_1^2 + P_2^2 + P_3^2 -- P_n^2)$$

↓

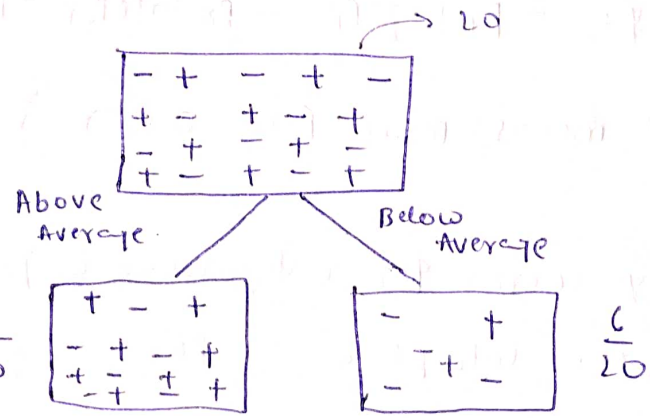
The probability of any 2 random data points belonging to class 1.

→ {For complete tree}

-> Use the weighted impurity of both sub nodes.

$$\text{weight of a node} = \frac{\text{number of samples in that node}}{\text{Total number of samples in its parent node.}}$$

Ex:



GI (Sub node) (Above Ave.) = $1 - [(0.57)^2 + (0.43)^2] = 0.49$

GI (") (Below Avg.) = $1 - [(0.33)^2 + (0.67)^2] = 0.44$

weighted gini impurity : performance in class

$$\left(\frac{14}{20}\right) \times 0.49 + \left(\frac{6}{20}\right) \times (0.44) = 0.475$$

Now check for other split like this.
And choose the tree with less GI.

(ii) Information gain using entropy

Information gain : The difference b/w information needed to describe the parent node & information needed to describe the children node.

* The more is the homogenity of the nodes, the more is the information gain.

$\therefore \uparrow IG \rightarrow$ homogenity $\uparrow$

$$IG = 1 - \text{Entropy (subnodes)}$$

$$\text{Entropy} = -P_1 \log(P_1) - P_2 \log(P_2) - P_3 \log(P_3) \text{ --- } n$$

* Lower entropy means more pure nodes & thus more information gain.

* It ~~also~~ only works for categorical target variable.

↳ Steps to calc. entropy of a split.

-> Cal. the entropy of the parent node.

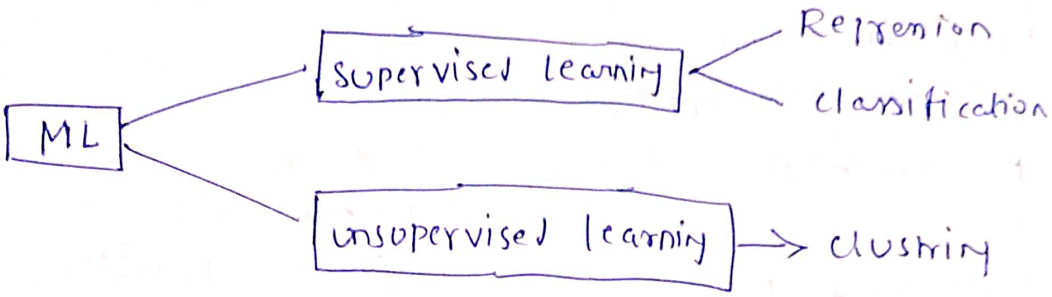
-> Cal. the entropy of each child node.

-> Cal. the weight average entropy of the split.

* -> If the weighted entropy of the child node is greater than the parent node, ignore the split.

$\therefore$ less entropy (or higher IG) ~~let~~ lead to more homogenity.

=> Introduction to clustring.



* Supervised → Develop predictive model based on both input & output data.

* unsupervised → Group & interpret data based only on input data.

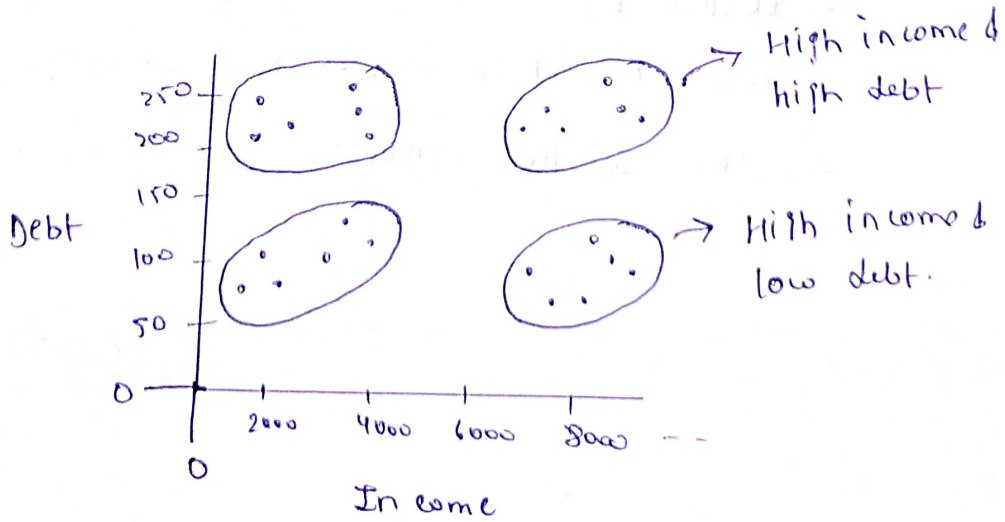
Clustring : we try to find patterns in the observation based on their features are found.

: Target variable are not defined.

→ Properties of clustering.

-> All the data points in a cluster should be similar to each others.

-> The data points from different clusters should be as different as possible.



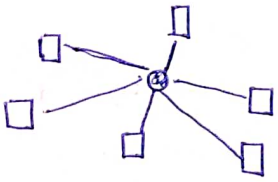
-> Application of clustering

- -> Customer segmentation
- -> Document clustering.
- -> Image segmentation
- -> Recommendation engine.

-> Evaluation metrics → {because visualising & prediction is not possible}

① Inertia

It tells us how far the points within a cluster are.



The distance of a point from the centroid within the cluster is called Intra cluster distance.

∴ Intertia gives as sum of intra cluster distance.

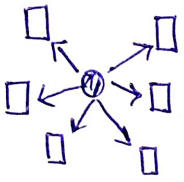
* lesser the inertia value, better the model is.

∴ $$\text{Inertia} = \sum_{i=1}^{m} \sum_{k=1}^{k} w_{ik} (||x_i - \mu_k||)^2$$

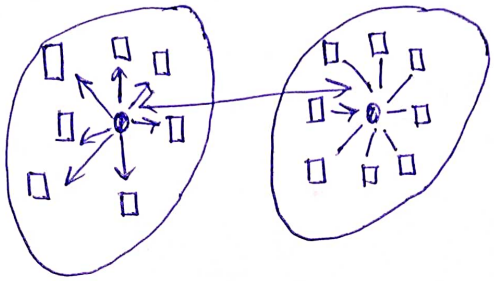
- -> $w_{ik} = 1$, if x_i belongs to cluster k
- -> $w_{ik} = 0$, if x_i does not belongs to cluster k.
- -> μ_k is the centroid of x_i's cluster.

② Dunn index

↳ use to evaluate the clusters.



intra cluster



inter cluster

$$\text{Dunn index} = \frac{\min(\text{inter cluster distance})}{\max(\text{intra cluster distance})}$$

∴ The higher the value of the dunn index, the better the cluster are.