[Machine Learning]

Machine learning (ML) is a subfield of artificial intelligence (AI) that focuses on developing algorithms and model that can automatically learn pattorns and insights from data withought being explicitly programmed. The goal of ML is to enable computers to learn and improve from experience, just like a human.

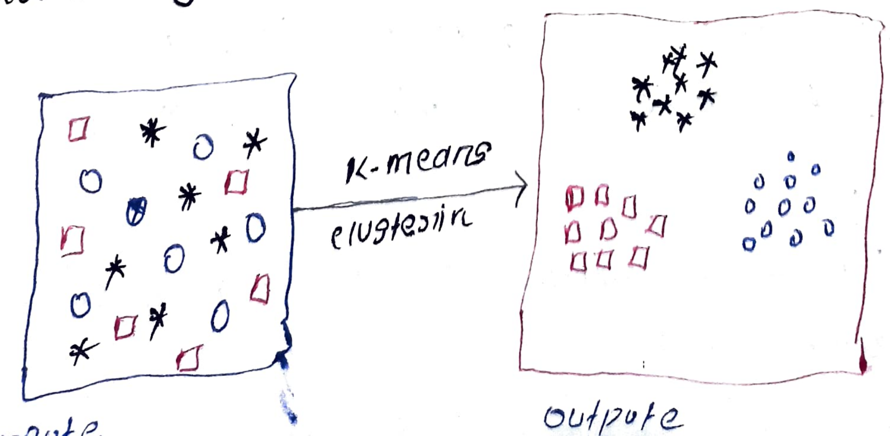
There are three main types of machine learning:

Supervised Learning: Supervised learning is a type of machine learning that uses labeled data to train machine learning model. A labelled dataset is one that has both inpute and output parameter. The model just need to map the inpute to the respective outputs or target valve.

Supervised learning algorithms:
- Logistic Regression
- SVM(support vector machine
- Linear Regression
- KNN
- Decision Tree
- Naive Bayes.

Unsupervised Learning: unsupervised learning is a type of machine learning that uses unlabeled data to train machine. Unlabeled data doesn't have a fixed output variable. The model learn from the data, discovers the patterns and feature in the data, and return outpute.

- clustering
- PCA
- K-means clusteinin
- Hierarchical clustering



Reinforcement Learning: In reinforcement learning, the algorithm learns by interacting with an environment and receive feedback in the form of rewards or punishment. The Algorithms learn to take action that maximize the cumulative reward over time. This type of learning is often used in robotic, gaming and control system

Well posed Problems

Well-posed learning problem in machine learning is a problem that is well-defined, has a clear objective and has feasible solution.

i) Well-defined problem statement: The problem statement should be clear and unambiguous. The inputs, outputs and the objective of the problem should be well-defined.

ii) Accessible and representative data: The data used to train the model should be representative of the real-world problem and should be accessible to the algorithms. Sufficient data required.

iii) Appropriate evaluation metrics: The evaluation matrix should be appropriate for problem at hand.

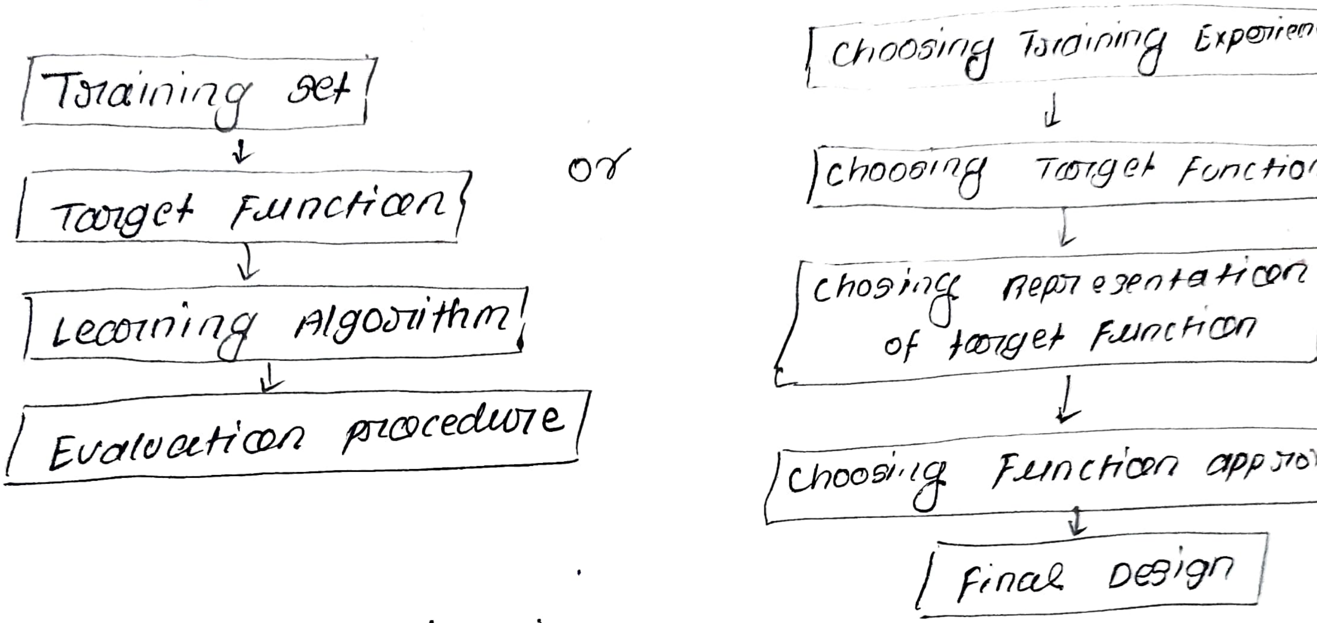
iv) Feasible solution: The solution should be practical and should be achievable within a reasonable timeframe.

Generalizability: The solution obtain should be able to generalize well to unseen data.

posed learning problem include:

- image classification
- sentiment analysis
- recomendation system
- House price prediction
- image caption Generator

Designing learning system



Empirical risk minimization

Empirical risk minimization (ERM) is a common approach i supervised learning, which involves minimizing the average los over a training dataset to find the best model parameters

on ERM, a model is trained on a training dataset, which con of input-output pairs. The model is then evaluated using a function, which measure the difference between the predic output to the truth output. The goal of ERM is to find model parameters that minimize the average loss over training dataset.

Given a set of training data $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)$ —— $(x_n, y_n)\}$

where x_i is the inpute and y_i is the outpute, and model parametorized by θ, we want to find the value of θ that minimizes the average loss over training data.

$$L_S(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i) \Big/ = L_S(\Theta) \sum_{i=1}^{n} L(\Theta(x_i), y_i)$$

$$\boxed{\text{minimize } \theta \frac{1}{n}\sum_{i=1}^{n} \{L(y_i, f(x_i;\theta))\}}$$

L = loss function

f = model Θ = model parameter

PAC (Probably Approximately correct).

It is a theoretical framework in machine learning that provides bounds on the number of training example need to learn a concept to a certain degree of accuracy whith high probability. The goal of PAC learning is to find a hypothesis the true concept underlying a set of traing example.

PAC is a way to measure how well a machine learning algorithm can learn a concept from a set of examples. It like trying to learn new language by looking at example of sentences in that language.

PAC learning tell us that if we have set of example and a hypothesis (a guess about what the answer is). we can measure how ~~our~~ clouse our guess is to the truth answer. The goal is to find a hypothesis that is close to the true answer.

[Data preprocessing]

- Data cleaning
- Data integration
- Dimension reduction
- Feature extraction
- Data transformation
- Data spliting
- One hote encoding
- word embeding
- Tokenization
- Normalization

Normalization

Normalization is a data preprocessing technique that involves scaling numerical feature in dataset to a standard range. The goal of normalization is to bring all the feature to a similar scale, so that no single feature dom to others, and to make it easier for machine learning alg to learn from the data.

[Gradient Descent]

Gradient descent is a genetic optimization algorithm capable of finding optimal solution to a wide range of problem. The general idea of gradient decsent is to tweek parameters iteratively in order to minimize a cost function.

① Batch Gradient Descent

$$\frac{\partial}{\partial \theta_j} MSE(\theta) = \frac{2}{m}\sum_{i=1}^{m}(\theta^T x^i - y^i)x_j^i$$

Gradient vector of the cost function

$$\nabla_\theta MSE(\theta) = \begin{pmatrix} \frac{\partial}{\partial \theta} MSE(\theta) \\ \frac{\partial}{\partial \theta} MSE(\theta) \\ \vdots \\ \frac{\partial}{\partial \theta_n} MSE(\theta) \end{pmatrix} = \frac{2}{m} X^T(X\theta - y)$$

Notice that this formula involves calculation over the full training set X at each gradient descent step! This is why the algorithm is called batch gradient descent: it uses the whole batch of training data at every step (actually. full gradient decent would probably be a better name)

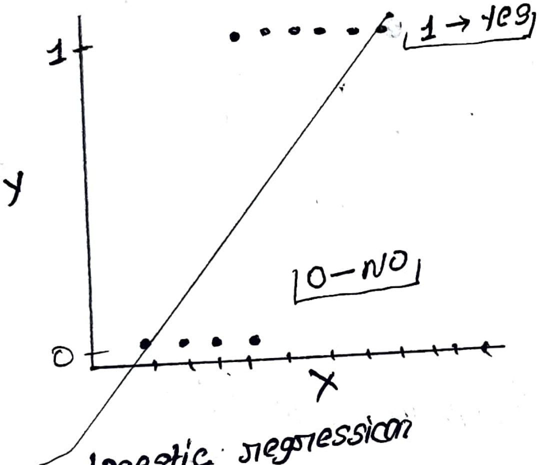
[Logistic Regression]

logistic regression is a statistical method used to predict the likelihood of a binary outcome like "yes" or "no".

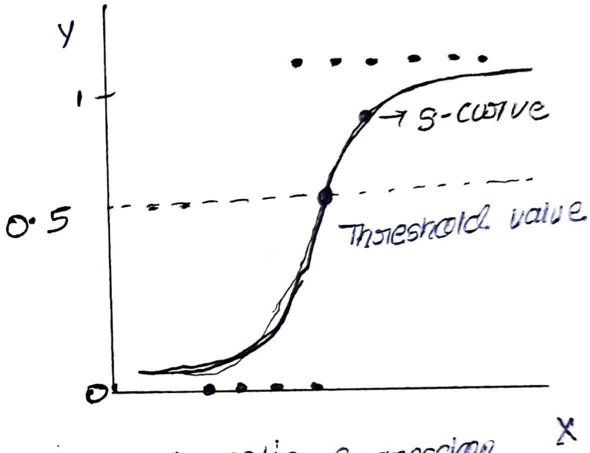
It is a type of regression analysis that predicts the probability of occurrence of a categorical dependent variable based on one or more independent variables.

$$\text{logistic Regression} = \boxed{y = \frac{1}{1+e^{-x}}}$$

sigmoid



Logestic regression



Logestic Regression

Here logestic regression is fail.

The value of the logistic must be between 0 and 1, which canno go beyond this limit, so it forms a curve like "S" form. The S-form curve is called the sigmoid function or the logestic function.

sigmoid function help to convert the linear combination of predicto variable into probability estimate that use to make prediction.

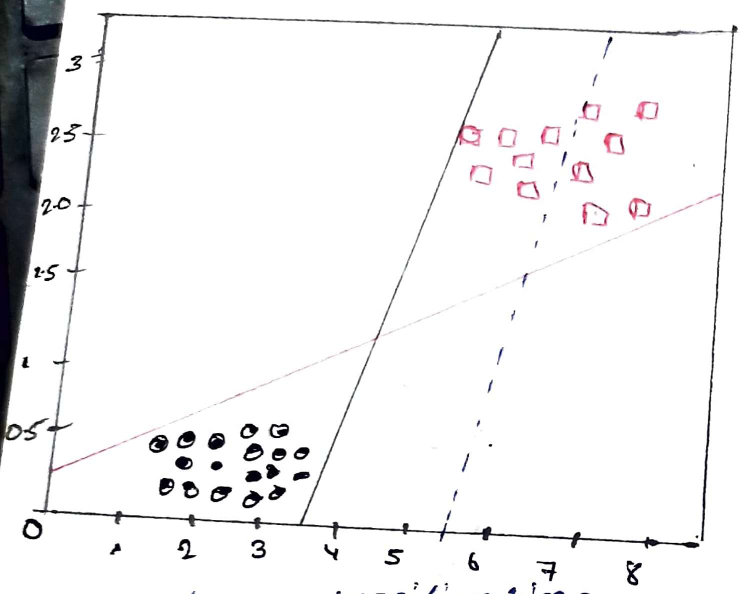
[Support vector Machines]

A support vector machine(SVM) is a powerful and versatile machine learning model capable of performing linear or nonlinear classification regression and oven novellty detection.

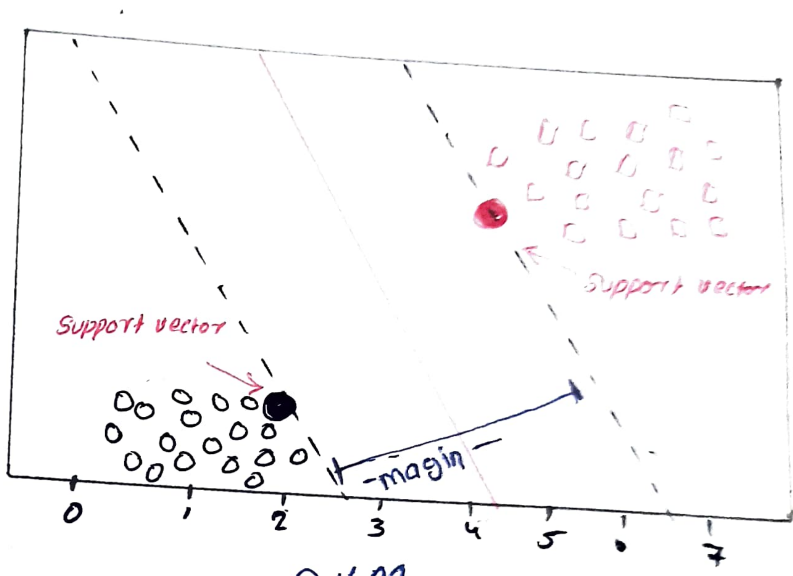
• it perform sensitive with outlier

outlier

$\frac{2}{||w||}$



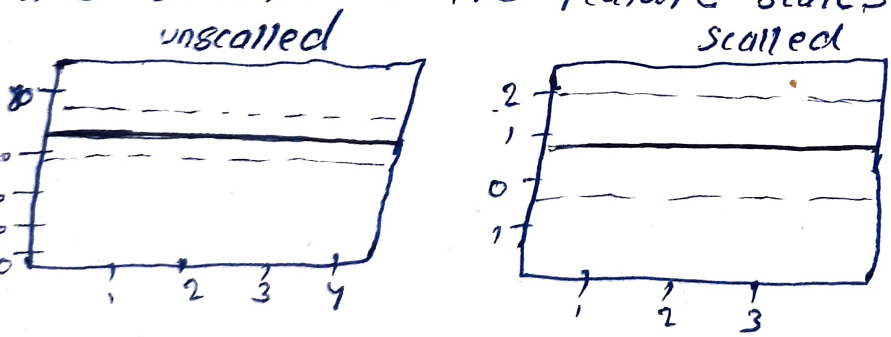
linear classification of 3 model



SVM

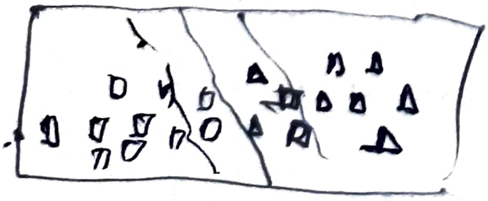
SVM is a maximum margin classifier. #[Go throug book.]

• it is sensitive to the feature scales



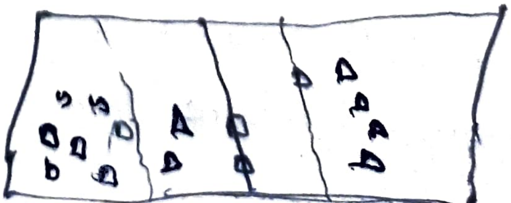
Sensitive to outlier henie we use soft margin instant of hard margin.

C₁



C = 100

[if underfit then reduce



KNN, short for K-nearest neighbours, is a machine learning algo- used for classification and regression problem The KNN algorithm work by finding the K closest data point to the given inpute, and then using those data points to make prediction.

Advantage of KNN

- Simple to understand and easy to implement
- NO training period: unlike other machine learning algorithm that requisre training on a datset.

KNN does not requisred ~~a~~ training period. It uses the the enti dataset for making prediction.

- Good performance on small dataset.
- works well with non-linear data.

Euclidean Distance

$$d(p,q) = \sqrt{\sum_i (p_i - q_i)^2}$$

Disadvantage

- computationally expansive
- sensitive to outliers
- Requisres normalization
- The choice of K: The performance of KNN is highly depend on the choice of K, which can be a challanging task case. choosing a larger K-value can lead to smoother deci boundaries, but it also increase risk of ~~overfitting~~. Underfittin

K-less chance of ~~underfitting~~ overfiting.

[Bayesian Learning]

Bayesian learning is a machine learning approach that involves using Bayesian statistics to make prediction and decisions

Bayesian learning is based on the Bayes theorem, which is a mathematical formula that describes the probability of an event based on prior knowledge and new evidence.

$P(x|Y)$ → probability of Y given X

Bayes theorem

$$\boxed{P(X|Y) = \frac{P(Y|X)\,(P(x).}{P(Y)}}$$

$P(X||Y)$→is called a **posterior**, which we need to calculate

$P(Y|X)$→is called the **likelihood**. It is the probability of evidence when hypothesis is true.

$P(X)$→ called **prior probability**, probability of hypothesis before considering the evidence.

$P(Y)$→called **marginal** probability.

(Q)

$P(\text{cancer}) = (0.008)$

$P(\text{pos} \mid \text{cancer}) = 0.98$

$P(\text{pos} \mid \sim\text{cancer}) = 0.03$

@ if a new patient comes in with a positive test result, what is the probability that he has cancer?

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

$$P(\text{cancer} \mid \text{pos}) = \frac{P(\text{pos} \mid \text{cancer}) * P(\text{cancer})}{P(\text{pos})}$$

$P(\text{pos}) = P(\text{pos} \mid \text{cancer}) * P(\text{cancer}) + P(\text{pos} \mid \sim \text{cancer}) * P(\sim\text{canc}$

$P(\sim\text{canser}) = 1 - P((\text{cancer})$ → probability of having no cancer

Substituting these value we will get -

$$P(\text{cancer} \mid \text{pos}) = \frac{(P(\text{pos} \mid \text{cancer}) * P(\text{cancer}))}{P(\text{pos} \mid \text{cancer}) * P(\text{cancer}) + P(\text{pos} \mid \sim \text{caner} \; 1 - P(\text{cancer}))}$$

$$= \frac{0.98 \times 0.008}{0.98 \times 0.008 + 0.03 * (1 - 0.008)}$$

$P(\text{can} \mid \text{pos}) = 0.208$

$$P(\text{cancer} \mid \text{positive}) = \frac{P(\text{postive} \mid \text{cancer}) \times P(\text{canser})}{P(\text{positive})} \quad (1)$$

we have to find $P(\text{positive})$

$$P(\text{positive}) = P(\text{positive} \mid \text{cancer}) \times P(\text{cancer}) + P(\text{postive} \mid \sim\text{cancer}) * P(\sim\text{canser})$$

$$P(\sim\text{cancer}) = 1 - P(\text{cancer})$$

$$\Rightarrow 1 - 0.008 = 0.992$$

$$\text{positive}) = P(\text{positive} \mid \text{cancer}) P(\text{cancer}) + P(\text{positive} \mid \sim\text{canser}) P(\text{cancer})$$

$$= 0.98 \times 0.008 + 0.03 \times 0.992$$

$$\Rightarrow 0.0376$$

$P(\text{positive}) = 0.0376$

e can put in equation (1)

$$= P(\text{cancer} \mid \text{positive}) = \frac{P(\text{positive} \mid \text{cancer}) \times P(\text{cancer})}{P(\text{positive})}$$

$$= \frac{0.98 \times 0.008}{0.0376}$$

$$\approx 0.208$$

Decision tree are versatile machine learning algorithms that can perform both clasification and regression tasks and even multioutput tasks.

CART (classification and Regression Tree)

- CART algorithm is a greedy algorithm.

cost function for classification

$$\boxed{J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}}$$

where $G_{left/right}$ → measures the impurity of the left / right subset

$M_{left/right}$ – is the number of instances in the left / right su

The algorithm work by first spliting the training set into two subset using a single feature k and a threshold t_k

it search for the pair (k, t_k) That produces the purest subs weight by their size.

once the CART algorithm has successfully split the training set in two, it splits the subset using the same logic, then the sub-subset, and so on, recursively. It stops recursion once it reaches the maximum depth (defined by the max-depth hyperparameter) or if it cannot find the split that will reduce impurity.

ID3 (Iterative Dichotomiser 3)

It as decision tree algorithm used in ML for classification task. The algorithm builds a decision tree by recursively selecting the inpute feature that provides the most information gain about the target variable, based on the entropy or impurity of the dataset.

Disadvantag

- Attributes must be in nominal values
- Dataset must not include missing data
- The algorithm tend to fall into over-fitting.

$$\left[\text{Entropy (Decision)} = \sum P(I) \cdot \log_2 P(I)\right]$$

$$\left[\text{Gain}(S,A) = \text{Entropy}(S) - \sum [P(S|A) \cdot \text{Entropy}(S|A)]\right]$$

(Q) Data set Given

- Firstly, we need to calculate global entropy

There are 14 example; 9 instances says → yes
5 instanies says → no

Entropy (Decision) = $\sum - p(I) . \log_2 p(I)$

= $- p(yes) . \log_2 (yes) - p(NO) . \log_2 p(NO)$

= $- (9/14) . \log_2 (9/14) - (5/14) . \log_2 (5/14)$

= 0.9407

1) # Wind factor on decision

Gain (Decision, wind) = Entropy(Decision) - $\sum [p(Decision | W) . Entropy (Decision | W)]$

⇒ wind attribute has two labels: weak and strong.

calculate (Decision | wind = weak) and (Decision | wind = strong)

~~Day~~	~~wind~~	~~Decision~~
~~1~~	~~week~~	~~no~~
~~3~~	~~week~~	~~yes~~
~~4~~	~~week~~	~~yes~~
~~5~~	~~week~~	~~yes~~
~~8~~	~~week~~	~~no~~
~~9~~	~~week~~	~~yes.~~
~~10~~	~~week~~	~~yes~~
~~13~~	~~week~~	

There are 8 instances for weak wind. Decision of 2 items are no and 6 items are yes.

Entropy (Decision / wind = weak) = $-p(no) \cdot \log_2 p(no) - p(yes) \cdot \log_2 p(yes)$

$$= -\left(\frac{2}{8}\right) \cdot \log_2 (2/8) - (6/8) \cdot \log_2 (6/8) = 0.811$$

[Strong wind factor on decision]

Entropy (Decision / wind = strong) = $-p(no) \cdot \log_2 p(no) - p(yes) \cdot \log_2 p(yes)$

$$= -(3/6) \cdot \log_2 (3/6) - (3/6) \cdot \log_2 (3/6) = 1$$

Gain (Decision, wind) = [P(Decision / wind = weak). Entropy (Decision / wind = weak)] − [P(Decision / wind = strong). Entropy (Decision / wind = strong)]

Entropy of Strong wind

$$= 0.940 - [(8/14) \cdot 0.811] - [(6/14) \cdot 1] = 0.048$$

week wind   weak + strong wind   strong   Entropy of strong wind

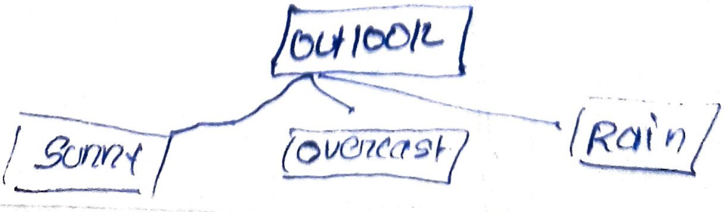
[Information Gain of all attributes]

1) Gain (Decision, outlook) = 0.246

2) Gain (Decision, Temperatore) = 0.029

3) Gain (Decision, Humidity) = 0.151

4) Gain (Decision, wind) = 0.048



#

[CART Algorithm]

it can handle both classification and regression and regression tasks.

This algorithm uses a new metric named gini index to create decision points for classification tasks.

Gini index

$$\text{Gini} = 1 - \Sigma (p_i)^2 \quad \text{for } i=1 \text{ to number of class.}$$

outlook	yes	no	number of instances	Gini
sunny	2	3	5	0.48
overcast	4	0	4	0
Rain	3	2	5	0.48

$$\text{Gini (outlook = sunny)} = 1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 = 0.48$$

$$\text{Gini (outlook = overcast)} = 1 - (4/4)^2 - (0/4)^2 = 0$$

$$\text{Gini (outlook = Rain)} = 1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 = 0.48$$

$$\text{gini (outlook)} = (5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48$$

$$= 0.171 + 0 + 0.171$$

$$= 0.342$$

Temperature	yes	NO	No of instances
Hot	2	2	4
cool	3	1	4
Mild	4	2	6

$\text{Gini}(\text{Tem} = \text{Hot}) = 1 - (2/4)^2 - (2/4)^2 = 0.5$

$\text{gini}(\text{tem} = \text{cool}) = 1 - (3/4)^2 - (1/4)^2 = 1 - 0.562 - 0.0625 = 0.375$

$\text{Gini}(\text{tem} = \text{mild}) = 1 - (4/6)^2 - (2/6)^2 = 0.439$

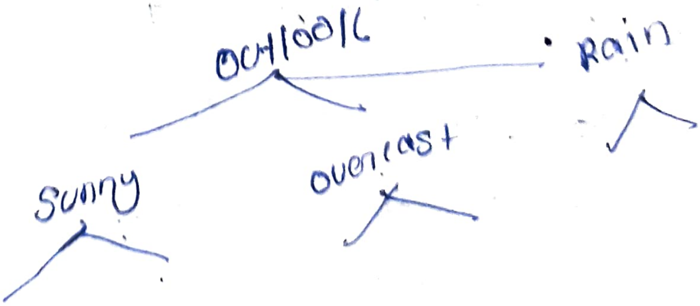
$\text{Gini}(\text{tem}) = (4/14) \times 0.5 + (4/14) \times 0.375 + (6/14) \times 0.439$

$= 0.439$

Similarly we calculate for all coloums.

Feature	Gini index
outlook	0.342
temp	0.439
Humidity	0.367
wind.	0.428

The winner is outlook because it cost is minim

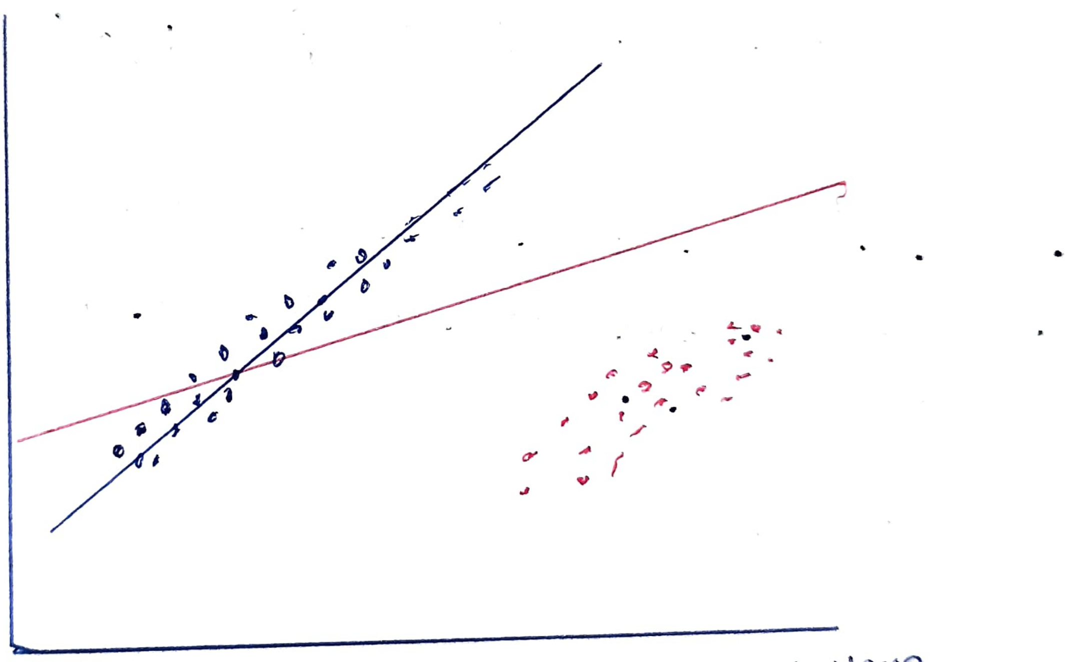


Robust linear model estimation using RANSAC

We can see how robustly fit a linear model to faulty data using RANSAC algorithm.

The ordinary linear regressor is sensitive to outliers, and the fitted line can easily be skewed away from the truth underlying relationship of data.

The RANSAC regressor automatically split the data into inliers and outliers, and the fitted line is determined only by the identified inliners.



RANSAC = (Random sample consensus) fit a model from random subsets of inliers from the complex data set.

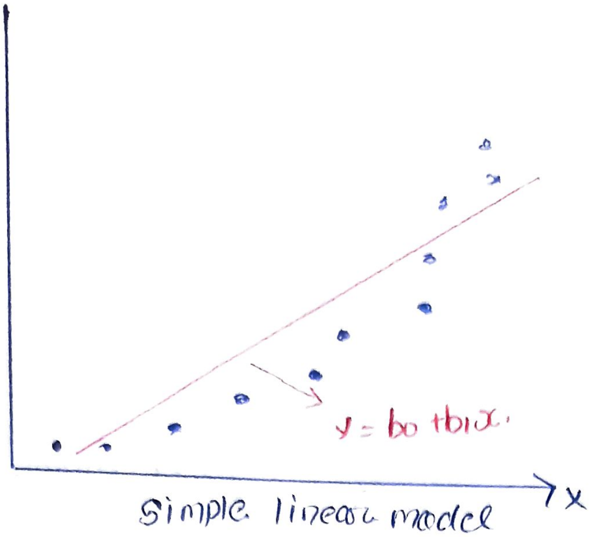
RAN ministic algorithm producing only a reasonable result with a certain, probability, which is dependent on the number of iterations (max-trials parameter). it is typically used for linear ~~for~~ and non-linear regression problems and is especially popular in the field of photogrammetic computer vision.

Polynomial Regression

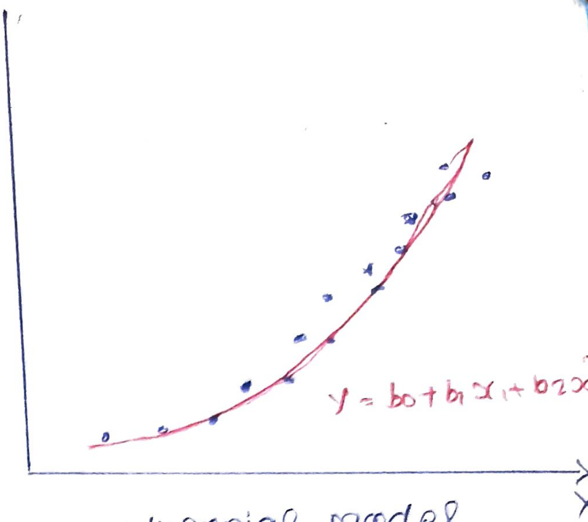
Polynomial Regression is a regression algorithm that models the relationship between a dependent (y) and independent variable as nth degree polynomial.

Equation= $b_0 + b_1 x_1 + b_2 x_1^2 + b_3 x_1^3 - - b_n x_1^n$

- It is also called the special case of multiple Linear Regres
- it is a linear model with some modification in order to inclease the accuracy.
- it work well on non-linear data
- In polynomial regression, the original features are converted into polynomial feature of required degree (n and then modeled using a linear model.



simple linear model



polynomial model

Equation of the polynomial Regression model

simple Linear equation : $y = b_0 + b_1x$

multiple -- -- : $y = b_0 + b_1x + b_2x_2 + b_3x_3 --$

polynomial -- -- : $y = b_0 + b_1x_i + b_2x_i^2 + b_3x_i^3 -$

ARIMA

Autoregressive integrated moving Average is a model use to analyze and forcast time series data.

Time series data is a sequence of observations record over time, such as stock prices, temperature reading, or sale

i) Autoregressive (AR) component : This component looks at the relationship between an observation and a certain number of past observation.

ii) Integrate (I) : it deals with transforming the data to make it stationary

iii) Moving Average (MA) it is the influence of past error term in predecting future values.

it consider the average of the error and their relationship to previous error.

By combining these components, ARIMA helps cature the patterns, trends and dependencies present in the time series data.

SARIMA : Seasonal (ARIMA) is a extension of ARIMA that incorporates seasonality into the analysis.

Regularized methods for Regression

Regulirized method foor regression are techniques used to mitigate (reduce) overfitting and improve the generalization performance of regression model. These methods add a regularization term to the loss function during model training, which help control the complexity of the model.

There are two commonly used regulized method foor regression

1) Ridge Regression (L2 Regularization)

ii) Lasso Regression (L1 Regularization)

iii) Elastic net (Ridge + Lessco)

i) Ridge Regularization: Ridge Regression adds a penalty term to the loss function, which is proportional to the sum of square coefficient of the regression model.

This penalty term discourages large coefficient values, making the model less sensitive to individual datapoint and reducing the likelihood of overfiting.

Loss function of ridge = Loss + alpha or lemda * (sum of Square of cofficients)

$$\text{Loss} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

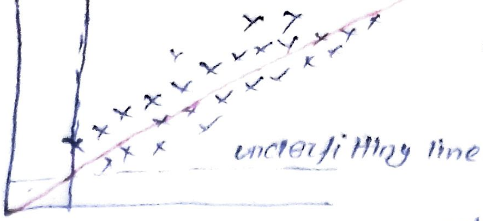
$$\text{Loss function of ridge} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda (m^2)$$

YN

alpha is a hyperparameter which control the strength of regularization alph↑ → more Shrinking of cofficient to zero, reducing model complexity.

y = mx + b

coefficient (wait)



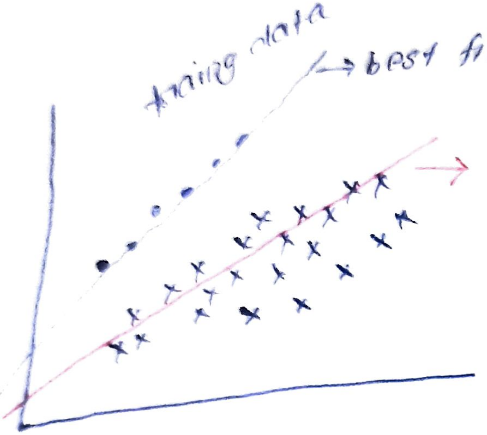
$y = mx + b$

weight

when $m >>> b$ then overfitting

when $b >>> m$ then underfitting

when $m \simeq b$ then best fit.



→ best fit line according to traing data but overfitting, not perform good on testing data

→ Ridge Regularization line reduce overfitting problem

● → traing data

x → testing data

idge LOSS = $$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \boxed{\lambda ||W||^2}$$

→ shrinkage cofficient

$\lambda(\omega_1^2 + \omega_2^2 + \omega_3^2 - - -)^2$

$\lambda \to 0$ to ∞

Regularization):

Lasso regression also add a penalty term to the loss function, but instead of the sum of squared coefficient, it uses the sum of the absoulte values of the coeffients.

This encourages sparsity in the model, meaning that some coeffients may become exactly zero.

Lasso regression is useful for feature selection, as it tends to set less important feature to zero, effectively removing them from model.

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda ||w||$$

3) Elastic Net

$$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda ||w||^2 + \lambda ||w||$$

[Relationship using correlation matrix]

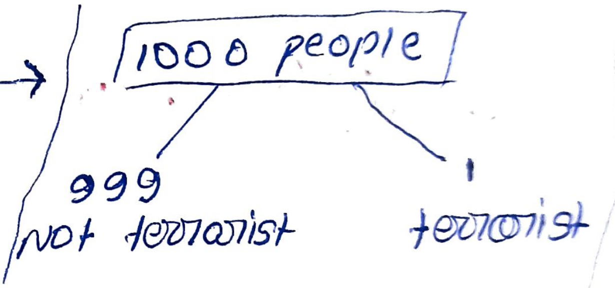
it is used to calculate the relaship between diffirend features in the dataset.

-1 to 1

i) positive corrolation (>0)
ii) negative correlation (<0)
iii) no correlation (0)

confusion Matrix

imbalanced dataset →



it i
imbalen
data

→Predicted

	1	0
1	0	1
0	0	999

add
total

$$\text{Accuracy} = \frac{999}{999+1} \approx 99.9\% \text{ accuracy}$$

here accuracy is 99.9 % according to it but it is not g
way to mesure accuracy because data is imbalance

i) Precision: What proportion of predicted positive is truly positive

(Jitne chijho ko positive bola hai usme se kitne chijhe sach me positive hai)

or

predicted positive value is what percent *really positive

		predicted negative	 positive
actual	negative	True negative	False positive
	positive	False negative	True positive

recall

precision

$$\text{precesion} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

$$= \frac{\text{True positive}}{\text{Total predicted positive}}$$

ii) Recall: what proportion of actual positive is corre classified

jitne logo ko sach me cancer tha usme se kitno logo k detect kiya gaya.

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}$$

$$= \frac{\text{True positive}}{\text{Total actual positive}}$$

$$\left[F_1 = 2 \times \frac{precision * Recall}{precision + Recall} \right]$$

No Free Lunch (NFL) theorem is a fundamental result in the field of machine learning and optimizatio or essentially states that there is no on universal algorithm that performs optimally accross all possible problem domain or data distributions.

it argues that when averaging over all problem instances, every algorithm will have the same performance on average. on other word, no algorithms can outperform any other algorithm when considering the entire space of possible probl

The NFL theorem also highlights the important of understanding the problem at hand, exploring different algorithms and adapting them to specific problem.

Error decomposition

Error decomposition refers to the process of breaking down the total error in a prediction or estimation task into different component of understanding. their individual contribution.

1) Bais :

2) variance :

3) irreducible Error : The irreducible error represents the inherent noise or uncertainty in the data that cannot be reduced, even with perfect model.

performance measure

The ultimate goal of performance measure is to reduce errors.

1) MAE (Mean absolute error (MAE)

y_i - target value $\hat{y}_i$ = predicted valu

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

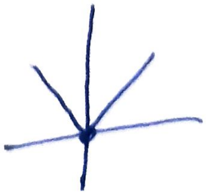
• Advantage

1) Same unit as y-axis

2) Robust to outliers

• Disadvantage

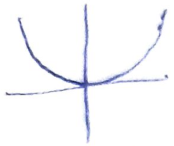
it is modulous it is not differencia at 0



2) MSE (mean square error)

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

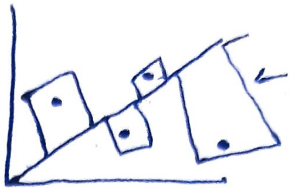
adv

- 

Differentiable at all points

dis

- not good for outliers



covering more area outliers

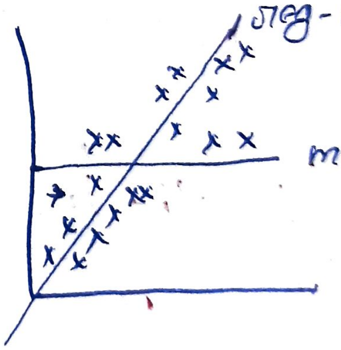
- unit of error = square of y - axis unit

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

- same unit as y.

> coefficient of determination (R^2) or goodness of fit

$$R^2 = 1 - \frac{SS_R}{SS_m}$$



SS_R = sum of square error in the regression line

SS_m = sum of square error in mean line

$$R^2 = 1 - \frac{\left[\sum_{i=1}^{n} (y_i - \hat{y}_i)^2\right]_{reg}}{\left[\sum_{i=1}^{n} (y_i^{\wedge} - \hat{y}_i)^2\right]_{mean}}$$

5> MSLE (mean square logarithimic error)

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}\left(\log(1+\hat{y}_i) - \log(1+y_i)\right)^2$$

6> MAPE (mean absolute percentage error) also known as MAPD

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

A_t → Actual value

F_t → Forecast value

VC Dimension

VC → Vapnik - chervonenkin

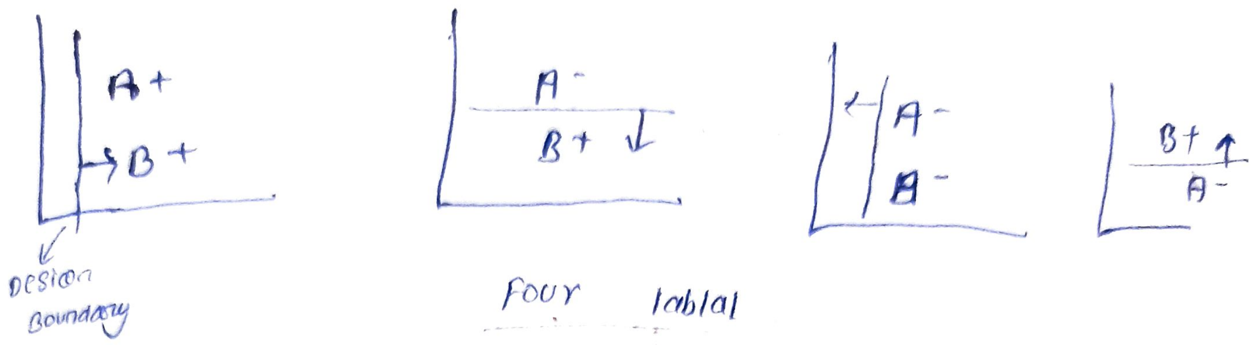
VC- dimension represent the maximum number of data point that can be shattered or perfectly separated by hypothesic class

For two dat point = n = 2, total 4 label (2^2)

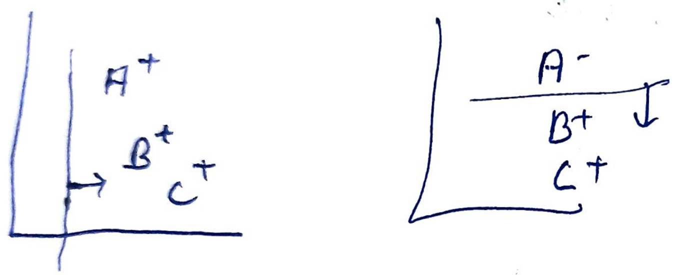
" three " " (n=3) total 8 (label) (2^3)

So, (2^n)

Let take two class of posetive and negative n= 2



n= 3



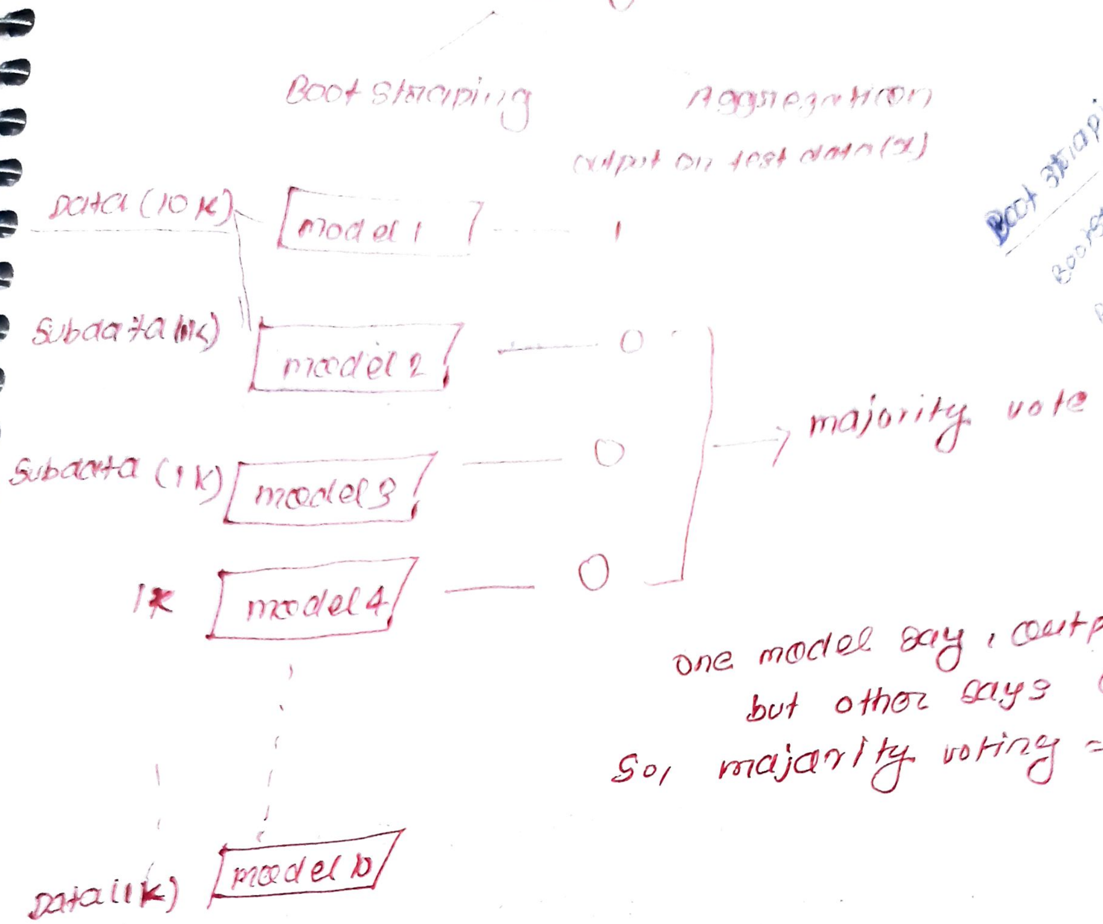
- - total 8 posibility

VC concept apply in various ml, like, svm, neural network decision tree etc

Bagging and boosting

Bagging and boosting are both ensemble learning technique that aim to improve the performance of individual base classifiers by combining predictions.

Bagging



Boot straping + Aggregation
Bootstraping + Aggregation
Bootstraping + Aggregation

one model say, output should 1
but other says 0
So, majarity voting = 0 So, output = 0

we dive our data and give to each model induvidualy for training.

Bagging involves training multiple base classifiers independently on different subset of the training data. Each base classifier is trained on a random selected subset of the original training data.

The final prediction is then made by aggregating the predictions of all base classifiers typically through majority voting.

Ex, Random Forest
Bagged Decision tree

Boosting

Boosting is an ensemble learning technique that combines multiple weak learner on base model to create a strong predictive model.

It is a sequential learning process where each subsequent model in the ensemble is trained to correct mistake made by previous models.

Ex,
- AdaBoost , Gradient Boosting

Ensemble learning is a machine learning technique that combines the prediction of multiple individual model to make final prediction

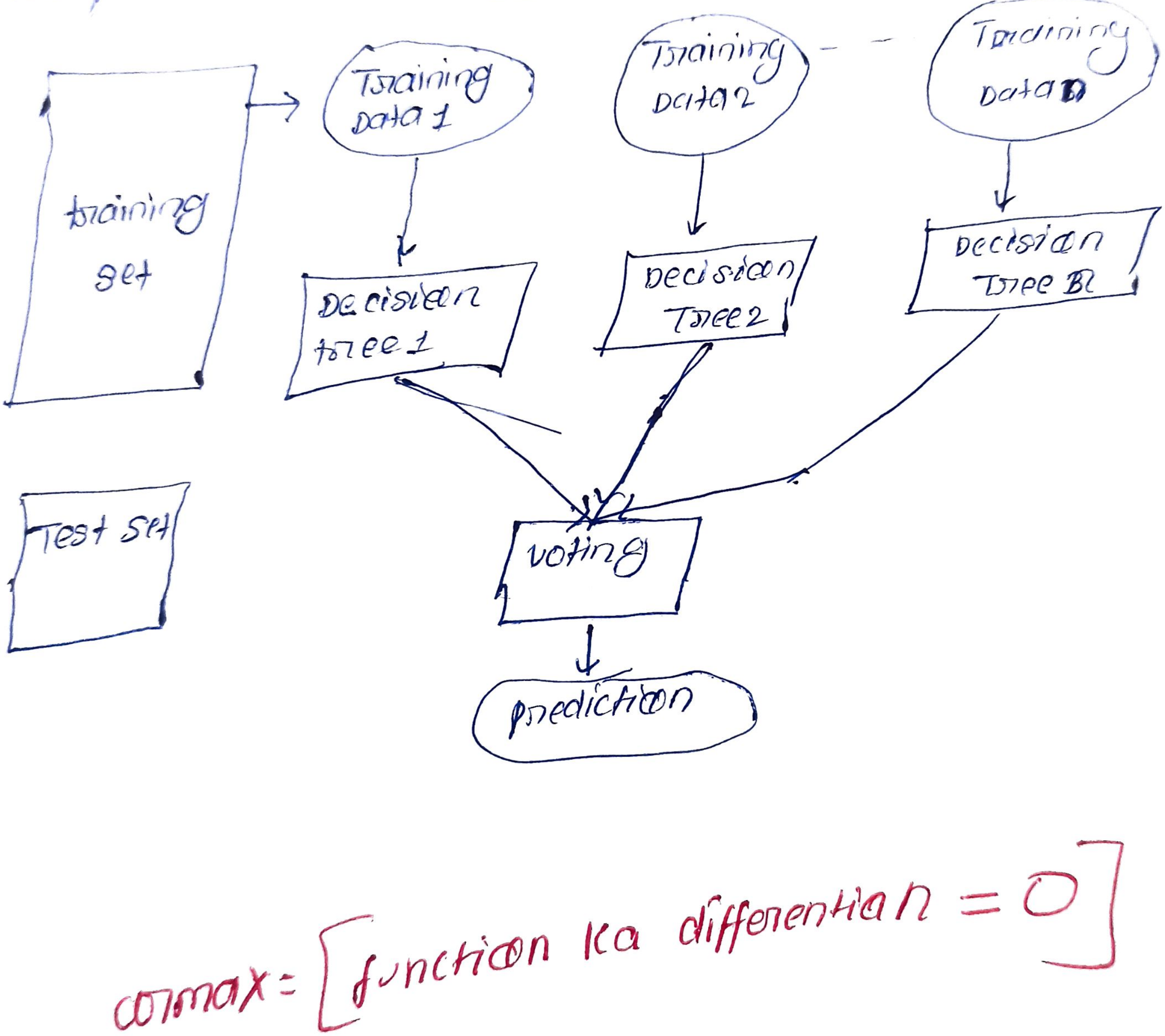
Ex Bagging, Boosting , Random Forest

= Random Forest = Bagging + Decision tree.

in Bagging we use any algo.. if we use Decision tree in bagging then it called Random Forest.

The majority voting classifier is a simple ensemble learning technique that combine the predictions of multiple individual classifiers or model by majority voting to make the final prediction.

Random Forest



argmax = [function ka differentiation = 0]

(1) well-posed learning problem consist of

⇒ inpute data, outpute data, problem statement.

2) primary goal of empirical risk minimization

Ans= To minimize risk associated with ml model

3) what is the role of inductive bias in empirical risk minimizatio

Ans - To bias the model towards certain type of data

4) PAC → probably Approximately correct learning

5) Inductive bias helps models generalize well to unseen data

⇒ True

6) what does the "approximately correct" aspect in PAC learning

Ans The model is approximately accurate in its Generalization to unseen data.

7) VC dimension in PAC refer to ?

Ans The capacity of hypothesis space

8) why is data preprocessing an essential step

⇒ it helps improve the performance and reliability of