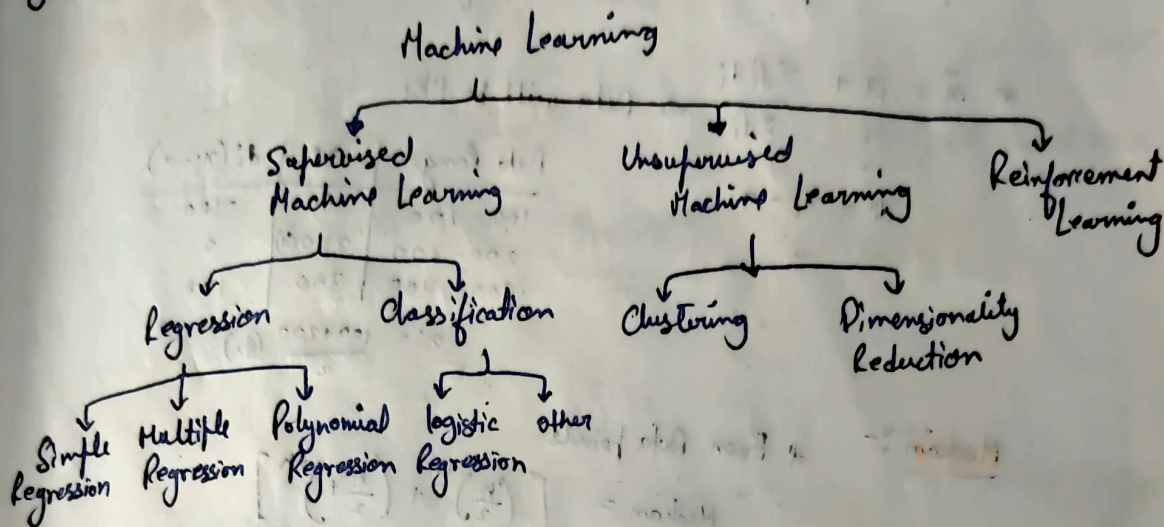


Statistical Machine Learning

SML

Vishnu Rame Chettyala

Types of Machine Learning



Types of Data

Quantitative Data :- Data like Numerical values, Profit, Quantity, Height, weight, Temperature etc.

- Discrete Values
- Continuous Values

Qualitative Data :- There are some data like Name, Category which cannot be expressed in numerical values.

- Structured Data
- Unstructured Data

* Types of Data Set →

Labeled Dataset

↳ Data Set of Images

Unlabeled Dataset

↳ Numerical Values & feature values

ADP

Basic Statistics

Mean :- $\frac{\sum f_i x_i}{\sum f_i}$

f_i = frequency (Grouped)
 x_i = Number

$\frac{\sum x_i}{N}$

x_i = Number
 N = number of element

(Normal data)

* $\bar{x} = a + \frac{\sum f_i d_i}{\sum f_i} \Rightarrow$ Data will be like

$d_i = x_i - a$

Data Range	f_i	$d_i (x_i - a)$
100 - 200	180	-100
200 - 300	250 (a)	0
300 - 400	350	100
$\bar{x} = \frac{100 + 200}{2}$ (En)		

Median :-

* Even Data points

Median = $\frac{\left[\left(\frac{n}{2} \right)^{th} + \left(\frac{n}{2} + 1 \right)^{th} \right]}{2}$

* Odd Data points

Median = $\left[\frac{(n+1)}{2} \right]^{th}$

* Grouped Data

Median = $\frac{l + \left(\frac{\frac{n}{2} - cf}{f} \right) h}{1}$

Ex:-

Class Interval || Frequency (f) | cf

0 - 20	17	17
20 - 40	28	45
40 - 60	32	77
60 - 80	24	101
80 - 100	19	120

$l = 40$
 $cf = 45$
 $f = 32$
 $h = 20$ Median will be at $\frac{120}{2} \Rightarrow 60$

Mode :- Highest repeating term

Range :- Minimum + Maximum

Ans

Range :- Maximum Value - Minimum Value

* What is Difference between Mean & Median??

Variance :- $\sigma^2 = \frac{1}{n} \sum (x_i - \mu)^2$ | $E(x^2) - (E(x))^2$

n = number of Data points

x_i = Data point

μ = Mean

σ = Standard Deviation

$E(x)$ = Expected Value of x .

$E(x^2)$ = Expected value of x^2 .

Standard Deviation :- $\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{n}}$

Data Pre-Processing

Crucial step in the data mining & other data related tasks.

- Data Cleaning.
- Data Transformation.
- Data Integration.

• Data Cleaning $\Rightarrow$

Missing Values $\Rightarrow$ Missing values are replaced by appropriate equivalent values.

Noisy Data $\Rightarrow$ Noisy Data removal. Outlier detection.

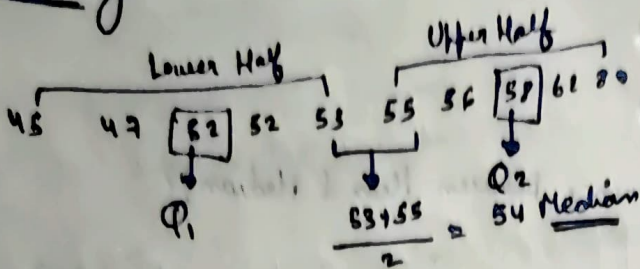
- Data Integrity $\Rightarrow$ Combining Data from different sources using record linkage and data fusion.

• Data Transformation $\Rightarrow$

Normalization

Standardization

Inter Quartile Range



$$Q_2 - Q_1 = \underline{\underline{IQR}}$$

Standardization

$$\text{Standard } X \Rightarrow Z = \frac{(X_i - \mu)}{\sigma}$$

X_i = Value of X .

μ = Mean.

σ = standard deviation.

Normalization

$$x = \frac{(x_i - x_{\min})}{(x_{\max} - x_{\min})}$$

Confidence Interval

A confidence interval is a range of values between which the observations is most likely to fall.

$$CI = \bar{n} \pm z \times \frac{\sigma}{\sqrt{n}}$$

z = Critical z -score calculated based on level of confidence

σ = standard deviation

n = Sample size

[Signature]

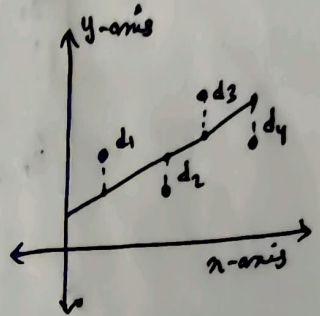
Linear Regression

Linear Regression is a statistical model that estimates the linear relationship between a dependent & some independent feature variables.

• Simple Linear Regression

$x \rightarrow$ Independent or feature variables

$y \rightarrow$ Dependent or output variable



$E \Rightarrow$ Error function

$$E = d_1^2 + d_2^2 + \dots + d_n^2$$

↳ after decomposing

$$E = (y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \dots + (y_n - \hat{y}_n)^2$$

$$E = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad [\hat{y} = y \text{ predicted at } x_i = x_i]$$

↳ after decomposing $\hat{y}_i$

$$E = \sum_{i=1}^n (y_i - (\beta_1 x_i + \beta_0))^2$$

$$E = \sum_{i=1}^n (y_i - \beta_1 x_i - \beta_0)^2$$

$E(\beta_1, \beta_0) \Rightarrow$ after differentiation

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \quad \text{--- (2)}$$

$$\beta_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad \text{--- (1)}$$

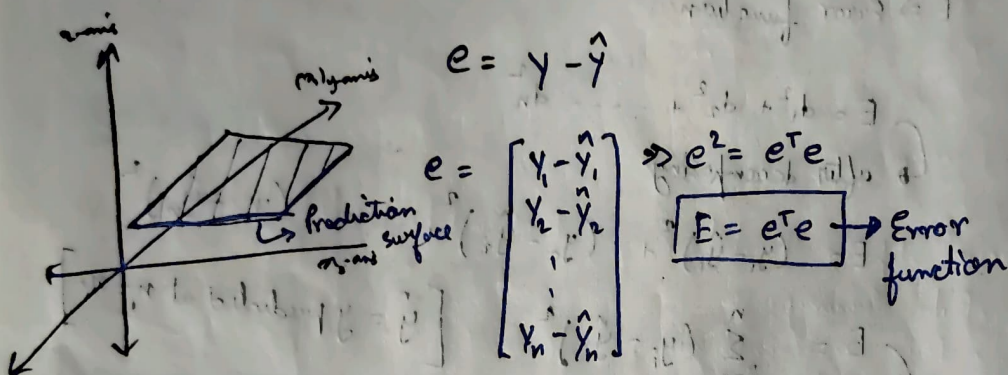
Handwritten signature

• Multiple Linear Regression

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_n x_n$$

$$\Rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1n} \\ 1 & x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{bmatrix}$$

$$\Rightarrow \hat{Y} = X\beta \text{ (Matrix Representation)}$$



$$E \Rightarrow (y - \hat{y})^T (y - \hat{y}) \Rightarrow (y - X\beta)^T (y - X\beta) \Rightarrow y^T y - 2y^T X\beta + \beta^T X^T X \beta$$

$$\Rightarrow y^T y - 2y^T X\beta + \beta^T X^T X \beta$$

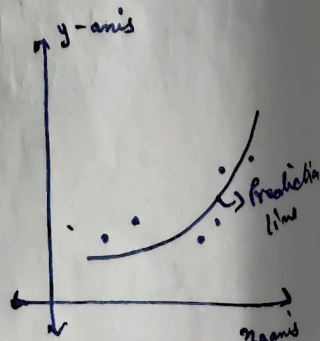
After derivating E function:

$$\Rightarrow \beta = (X^T X)^{-1} (X^T y) \rightarrow \text{final Eqn}$$

• Polynomial Linear Regression

Polynomial linear regression is a variation of linear regression where the relationship between the independent variable (x) and the dependent variable (y) is modeled as a polynomial equation. It helps capture non-linear patterns.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \dots + \beta_n x^n$$



Types of Error

$$\text{MAE} \Rightarrow \frac{\sum |y_i - \hat{y}_i|}{n}$$

$$\text{MSE} \Rightarrow \frac{\sum (y_i - \hat{y}_i)^2}{n}$$

$$\text{RMSE} = \sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}}$$

$$R^2 E = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

$$= \frac{\sum (y_i - \bar{y})^2}{\sum (y_i - \bar{y})^2} \Rightarrow \frac{\sum (\hat{y}_i - \bar{y})^2}{\sum (y_i - \bar{y})^2}$$

MAE (Mean Absolute Error) :- Measures the average magnitude of errors without considering their direction. It captures the absolute difference between predicted and actual values, making it less sensitive to outliers.

MSE (Mean Squared Error) :- Emphasizes larger errors by squaring them, capturing average squared difference. It is very sensitive for outliers.

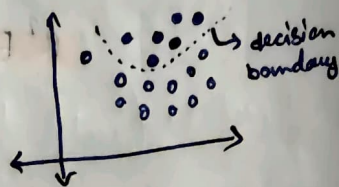
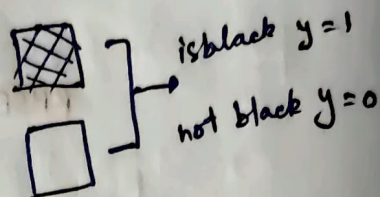
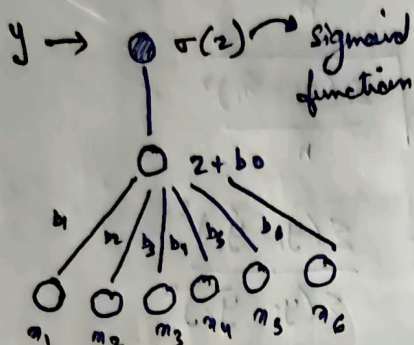
RMSE (Root Mean Squared Error) :- The squared root of MSE, providing errors in the same unit of target variable.

R^2 Error :- Indicates the proportion of variance in the dependent variable explained by the model. It captures how well the model fits the data, with value closer to 1 indicating a better fit.

[Signature]

Logistic Regression

Logistic Regression is a binary classifier algorithm which classifies between "is a class" or "not is a class".



$$m = [m_1, m_2, m_3, m_4, m_5, m_6] \rightarrow \text{features vector.}$$

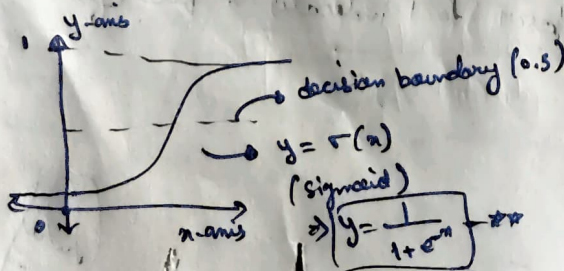
Features represent different types of characteristics which decide or affects the prediction of class.

$$b = [b_0, b_1, b_2, b_3, b_4, b_5, b_6] \rightarrow \text{weights vector.}$$

weights vector governs how much individual features affects the prediction of class. [b_0 is taken into account for features which doesn't affect or overall optimization].

$$Z = \sum_{i=1}^6 m_i b_i + b_0 \rightarrow \text{Real weighted value.}$$

$$y = \sigma(z) \rightarrow \text{Sigmoid function}$$



Sigmoid function is used to convert Real weighted value into probabilistic value for class decision.

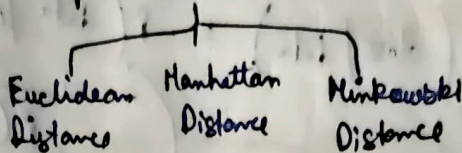
[Signature]

K-Nearest Neighbors (K-NN)

K-Nearest Neighbors (K-NN) is a simple algorithm that classifies or predicts data by finding the K closest points to a new point and using their majority class or average class value (for regression). It does not require model training but can be slow with large datasets. The choice of K is crucial for accuracy.

Algorithm :- $K = +ve$ Integer, $n =$ Input feature vector.

Step 1 :- Compute distance between x and all its neighbors



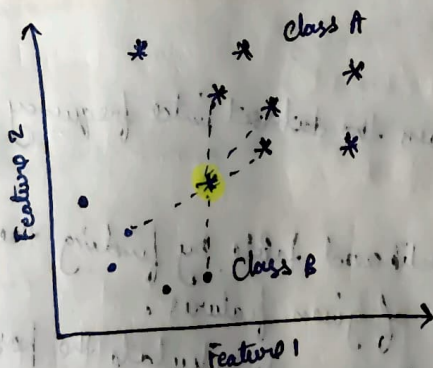
Step 2 :- Arrange all the distances in descending order and select the K smallest distance.

Step 3 :- Put x in our point class i which has the majority vote.

Euclidean distance :- $d(n, y) = \sqrt{\sum (n_i - y_i)^2}$

Manhattan distance :- $d(n, y) = \sum_{i=1}^n |n_i - y_i|$ ($n =$ dimensions)

Minkowski distance :- $d(n, y) = \left(\sum_{i=1}^n (n_i - y_i)^p \right)^{1/p}$ $p =$ some selected value



Too small value of K will make algo susceptible to noise

Too large value of K will make algo susceptible to inaccurate.

DDJ

Naive Bayes Classifier

Naive Bayes Theorem is foundationally based on Bayes Theorem where conditional probability is calculated.

$$P(A/B) = \frac{P\left(\frac{B}{A}\right) \cdot P(A)}{P(B)}$$

↪ This can be written (in our context) as

$$P(\text{class} / \text{data}) = \frac{P(\text{data} / \text{class}) P(\text{class})}{P(\text{data})}$$

↓
Probability of class
provided the given
data

↙
What is the probability that
the given data belongs to the
class.

• General formula for Naive Bayes :-

$$P(y | x_1, x_2, \dots, x_n) = \frac{P(x_1 | y) P(x_2 | y) \dots P(x_n | y) \cdot P(y)}{P(x_1) P(x_2) \dots P(x_n)}$$

x_i are features

y is the predictive class

• Steps ⇒

Step 1 :- Convert the given dataset into frequency table.

Step 2 :- Generate likelihood table by finding the probabilities of given features.

Step 3 :- Use Bayes theorem to calculate the posterior probability.

Signature

Example of frequency & likelihood table

⇒ example we have features :- Sunny, Overcast, Rainy
Class :- Yes or No

	Yes	No
Sunny	3	2
Overcast	4	0
Rainy	2	3
Total	9	5

Frequency Table

	Yes	No	
Sunny	3	2	5/14
Overcast	4	0	4/14
Rainy	2	3	5/14
Total	9/14	5/14	

Likelihood Table

• Gaussian Naive Bayes : This method is used for datasets with continuous numerical features. It assumes that the features follow a normal (Gaussian) distribution.

$$f(x|class) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad P(Pass=Yes) = \frac{\text{Count}(Yes)}{\text{Total Count}}$$

Dataset ⇒

Study hours	Exam Score	Class (Pass)
3	50	No
3	60	No
5	75	Yes
7	85	Yes
10	95	Yes
12		

• Multinomial Naive Bayes : This method is used for datasets where features represent discrete counts, often in text classification tasks.

Example :- Document Classification

Doc	word1 (count)	word2 (count)	Class
1	3	4	History
2	3	10	Science

$$P(\text{word1} | \text{History}) = \frac{3}{7}$$

$$P(\text{word2} | \text{History}) = \frac{4}{7}$$

$$P(\text{History}) = \frac{\text{Document in History}}{\text{Total Documents}} = \frac{1}{2}$$

$$\text{Prediction} \Rightarrow P(\text{class} | \text{Document}) = \frac{P(\text{class}) \cdot P(\text{word1} | \text{Document}) \cdot \dots}{P(\text{Document})}$$

Handwritten signature

- Bernoulli's Naive Bayes : The Bernoulli's Naive Bayes algorithm is designed for Binary/Boolean features, typically answering questions like "is this present or not". often used for text classification tasks where features are binary.

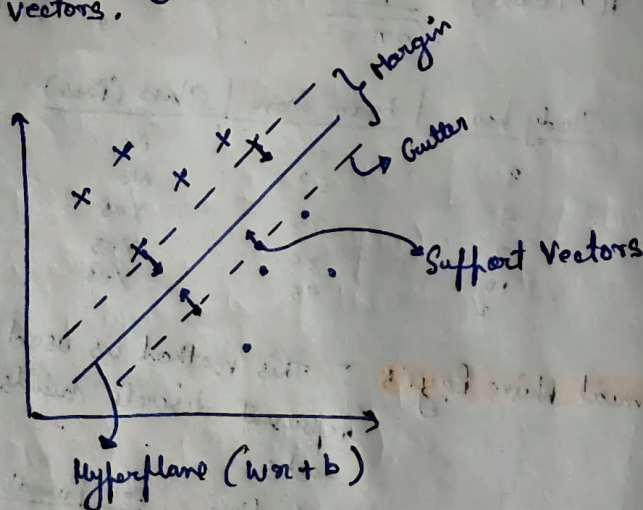
Document	"team"	"Government"	"win"	Class
Doc1	1	0	1	Sports
Doc2	1	1	1	Politics
Doc3	1	0	1	Sports
Doc4	0	1	0	Politics

$$P(\text{Sports}) = \frac{\text{Number of Sports Docs}}{\text{Total Docs}}$$

$$P(\text{team} = 1 | \text{Sports}) = \frac{\text{Number of Sports Docs with team} = 1}{\text{Total Sports Docs}}$$

Support Vector Machine

Support Vector Machine is a supervised Machine learning model which is used for classification and regression purposes (Avg. for regression). Goal is to find optimal Hyperplane with Maximum margin which is governed by support vectors.



Hyperplanes are separation lines which can be drawn to classify two classes.

$w \rightarrow$ weights vector

$x \rightarrow$ data point vector

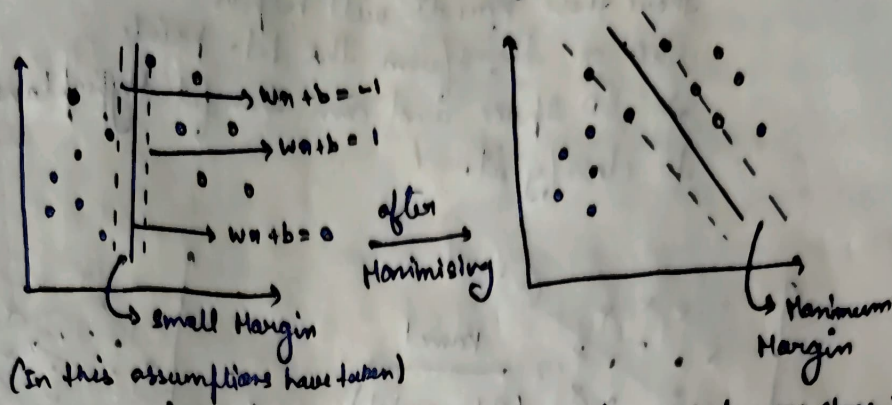
$b \rightarrow$ bias

if $w \cdot x + b > 0$ it is (x) class
if $w \cdot x + b < 0$ it is (o) class

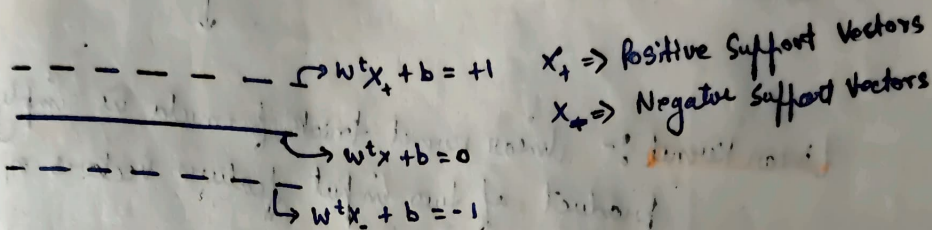
Decision Rule

Signature

Support vectors are data points that are closest to the hyperplane and they influence the position and orientation of the hyperplane. The same support vectors are used to minimize the margin.



$$y_i (w^T x_i + b) \geq 1 \quad \begin{cases} y_i = +1 \text{ for +ve class} \\ y_i = -1 \text{ for -ve class} \end{cases}$$



$$\bullet \text{ Margin} = \frac{w}{\|w\|} \cdot (x_+ - x_-) = \frac{w^T (x_+ - x_-)}{\|w\|} = \frac{2}{\|w\|} \rightarrow \text{cost function}$$

for Minimizing margin we need to minimize the cost function.
optimize

$$\min \left(\frac{1}{2} \|w\|^2 \right)$$

• Minimization of $\left(\frac{1}{2} \|w\|^2 \right)$ using Lagrange's Multipliers :-

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i [y_i (w^T x_i + b) - 1]$$

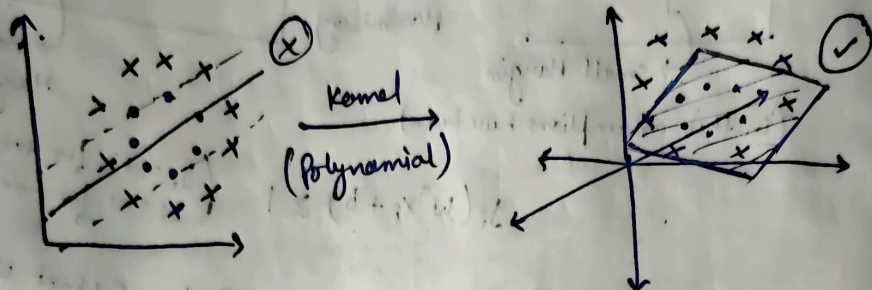
N = number of support vectors

w = weights vector

α_i = corresponding Lagrange Multiplier for each S vector
($\alpha_i > 0$)

[Signature]

- Kernel Tricks** :- There may be cases where datasets can not be classified with a linear regression relation, there these kernels will help us to scale or transform the data points in 3D space and can then fit hyperplane to classify it.



Linear Kernel :- Linear kernel trick formula is simply the dot product of two input feature vectors x_1 and x_2 .

$$K(x_1, x_2) = x_1 \cdot x_2$$

Polynomial Kernel :- Polynomial kernels represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables.

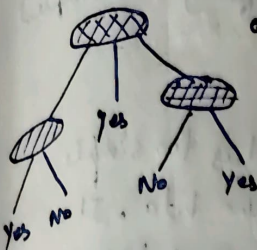
$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d$$

d = degree of polynomial

DD

Decision Tree Classifier

Decision Tree Classifier is a supervised machine learning algorithm used for classification tasks. It splits data into subsets based on feature values, forming a tree-like structure. At each node, the algorithm selects the feature that provides the highest Information Gain or Lowest Gini Impurity to partition the data. The goal is to create a tree where each leaf node represents a class label, and branches represent decision rules. It's intuitive, interpretable, and works well for categorical and numerical data.



$$\text{Entropy} = S = \sum_{i=0}^m -p_i \log_2(p_i), \quad \text{Information Gain}(S, A) = I - \sum_{i=0}^m -p_i \log_2(p_i)$$

$$\text{Gini Index} = 1 - \sum_{i=0}^m p_i^2$$

Information before split (Entropy)

m = total no. of classes

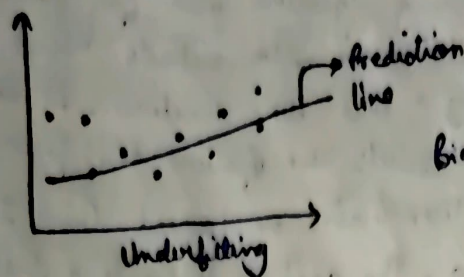
Problems with Decision Tree $\Rightarrow$

- Overfitting, Tends to overfit on the training dataset
- Instability, Small changes in data can alter the tree structure.
- Bias, favors features with more levels of dominant features.
- Limited Expressiveness, struggles with complex patterns.
- High Variance, sensitive to training data noise.
- Continuous Variables, splitting continuous features can be challenging.

Ensemble models like Random Forest and Gradient Boosting are based on decision trees, which increase accuracy and resilience by mixing several trees to improve generalization and reduce overfitting.

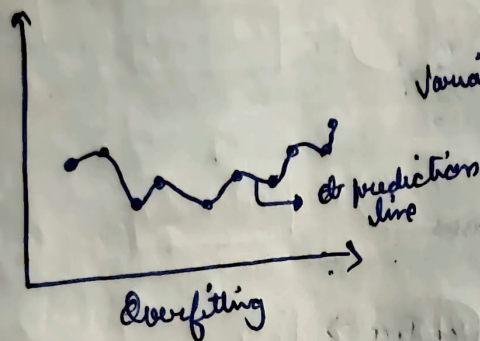
Q22

Bias Vs Variance



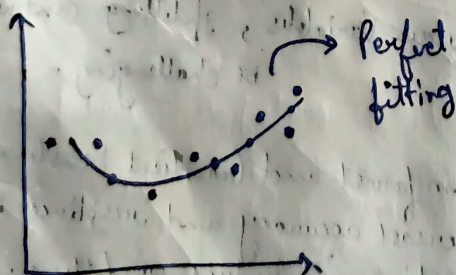
Bias = Underfitting (too simple)

Bias is the error that occurs when a model is too simple and doesn't capture the true patterns in the data. It's like trying to fit a straight line to data that has a curve. The model makes assumptions that are too strong and misses key details. High bias leads to underfitting.



Variance = Overfitting (too complex)

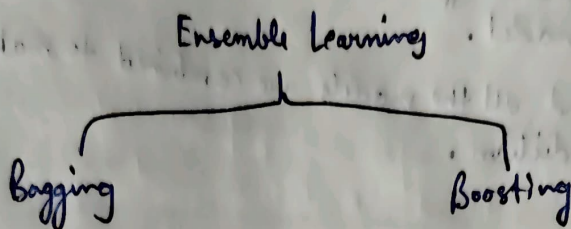
Variance is the error that occurs when a model is too complex and fits the training data too closely, even capturing random noise. It's like drawing a curve that bends exactly to every point in the data, but when you test it on new data it doesn't perform as well. High variance leads to overfitting.



Handwritten signature

Bagging Vs Boosting / Ensemble Learning

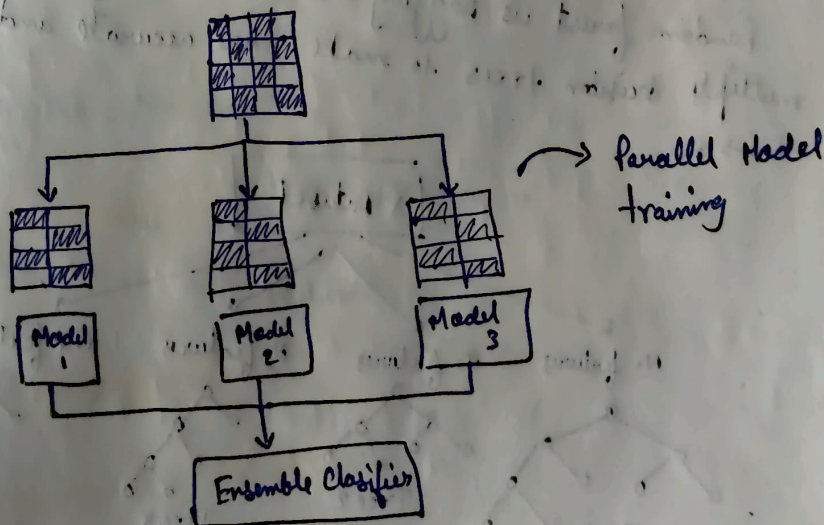
Ensemble Learning is a approach, we combine multiple model's ("weak learners") decision together to obtain final learning decision. General idea is set up same number of classifiers (experts) and allow them to vote.



Problems Ensemble Learning Solves:-

- Too little data
- Computational Limitation
- No Model perfect for the data

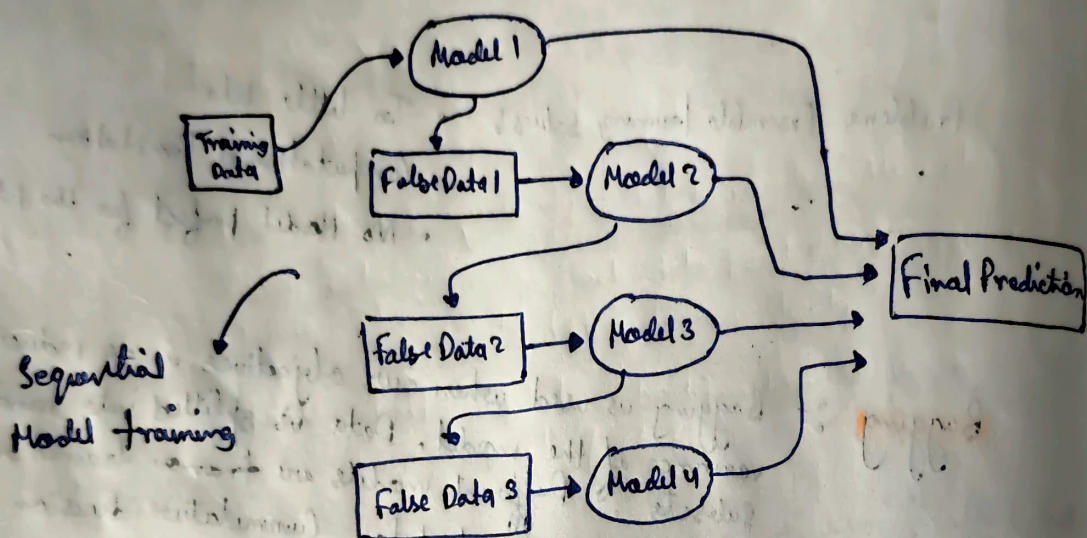
Bagging :- Bagging is used when our objective is to reduce the variance of the model. Data is splitted into random subsets. Different models are trained on that different subsets of data. Cumulative decision is taken from all that predictions of different model.



[Signature]

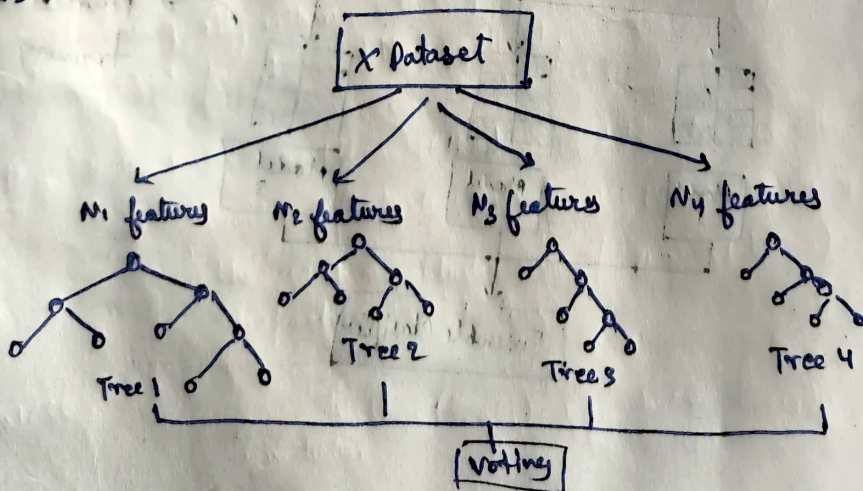
Boosting :- Boosting is a technique that combines multiple weak models (models that perform better than each other) to create a strong model overall. Model is used to reduce Bias.

- Train Model One by One
- Each new model focuses on the mistakes made by the previous model.
- At the End all the models are combined to make the final prediction.



Random Forest Classifier

Random forest is Bagging ensemble learning model that combines multiple decision trees to make more accurate and robust predictions.



Model creates multiple decision trees using random subsets of data and random features.

Combines output of all trees by majority vote or averages out for predictions.

Benefits :-

- Handles overfitting better than a single decision tree.

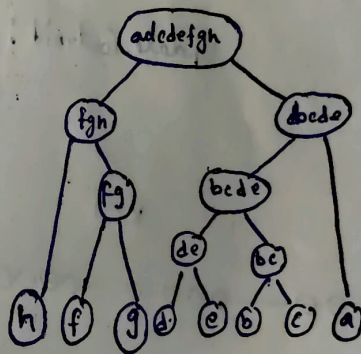
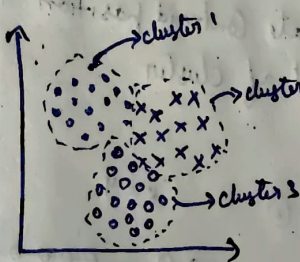
- Works well with large datasets and high dimensional data.

- Robust to noise and missing data;

Note:- Random forest assumes very low correlation between all trees.

Clustering

Clustering is an unsupervised learning technique used to group similar data samples based on specific criteria without predefined labels or guidance. It is used for data exploration to find hidden patterns or structures in the data which may hold valuable insight.



Types of Clustering Algorithms :-

Exclusive Clustering (Partition Methods) :-

K-Means

K-Medoids

Overlapping clustering :- Fuzzy / C-Means

Hierarchical Clustering :-

Agglomerative
divisive

Quf

Density Based Clustering :-

DBSCAN
Pamela

Grid Based Clustering :-

STING
WaveCluster

Similarity in clustering

depends on what we

want to emphasize

in the data, making

the choice of similarity

measure more crucial than the algo itself.

K-Means Clustering

K-Means clustering is a simple and popular clustering algorithm used to group or cluster data into K clusters based on similarity.

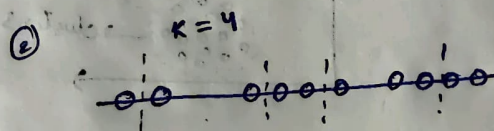
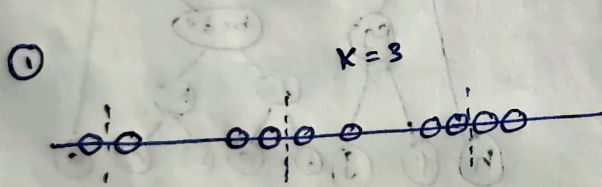
Steps of Algo $\Rightarrow$

Choose K : Decide number of K

Place centroids : Randomly select K points as the centroid

Assign Data : Assign each data point to the nearest centroid based on similarity.

Update Centroid : Update Centroid position to the mean point of cluster



As we can see different value of K can change our performance of our clustering Algorithm, selecting it is a mandatory task and method used for it is Elbow method.

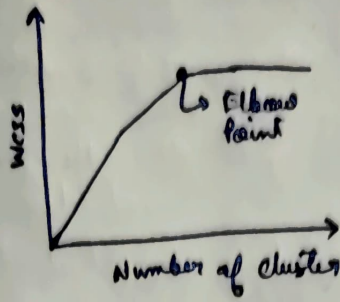
• **Elbow Method** :- The Elbow Method is used to determine the optimal number of clusters (K) for K-Means clustering by analyzing the within-cluster sum of squares (WCSS). WCSS measures compactness of clusters.

WCSS =

$$\sum_{P_i \in \text{cluster}_1} \text{distance}(P_i, C_1)^2 + \sum_{P_i \in \text{cluster}_2} \text{distance}(P_i, C_2)^2 + \dots$$

002

The elbow point is where the WSS decrease slows down, indicating the optimal number of clusters, as adding more clusters no longer improves the results significantly.



Qwq