ML-ESE

Machine Learning:

Arthur Samuel, an early American leader in the field of computer gaming and artificial intelligence, coined the term “Machine Learning” in 1959 while at IBM and defined machine learning as “the field of study that gives computers the ability to learn without being explicitly programmed “.

The field of study known as machine learning is concerned with the question of how to construct computer programs that automatically improve with experience. Machine learning is a subfield of artificial intelligence that involves the development of algorithms and statistical models that enable computers to improve their performance in tasks through experience.

Learning:

A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Classification of Machine Learning:

Supervised Learning:

- Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- The given data is labelled.
- Basically, supervised learning is when we teach or train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer.
- Both classification and regression problems are supervised learning problems.
- Types: Regression, Logistic Regression, k-NN, Naïve Bayes classifier, Decision tree

Advantages:

- Allows collecting data and produces data output from previous experiences.
- Helps to optimize performance criteria with the help of experience.
- Helps to solve various types of real-world computation problems.
- It performs classification and regression tasks.
- It allows estimating or mapping the result to a new sample.
- We have complete control over choosing the number of classes we want in the training data.

Disadvantages:

- Classifying big data can be challenging.
- It requires a lot of time because of training
- Supervised learning cannot handle all complex tasks in Machine Learning.
- Computation time is vast for supervised learning.
- It requires a labelled data set.
- It requires a training process.

Unsupervised Learning:

- Unsupervised learning is the training of a machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance.

- Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.
- Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore, the machine is restricted to find the hidden structure in unlabelled data by itself.
- Two categories: Clustering and Association
- Clustering types: Hierarchal, K-means, PCA,

Advantages:

- It does not require training data to be labelled.
- Dimensionality reduction can be easily accomplished using unsupervised learning.
- Capable of finding previously unknown patterns in data.
- Flexibility: It can be applied to a wide variety of problems, including anomaly detection, and association rule mining.
- Exploration: Allows for the exploration of data and the discovery of novel and potentially useful patterns that may not be apparent from the outset.
- Low cost

Disadvantages:

- Difficult to measure accuracy or effectiveness due to lack of predefined answers during training.
- The results often have lesser accuracy.
- The user needs to spend time interpreting and label the classes following that classification.
- Lack of guidance: It lacks the guidance and feedback provided by labelled data, which can make it difficult to know whether the discovered patterns are relevant or useful.
- Sensitivity to data quality: Unsupervised learning can be sensitive to data quality, including missing values, outliers, and noisy data.
- Scalability: Unsupervised learning can be computationally expensive, particularly for large datasets or complex algorithms, which can limit its scalability.

Reinforcement Learning:

- Reinforcement Learning (RL) is the science of decision making.
- It is about learning the optimal behaviour in an environment to obtain maximum reward.
- In RL, the data is accumulated from machine learning systems that use a trial-and-error method. Data is not part of the input that we would find in supervised or unsupervised machine learning.
- Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next.
- After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect.
- It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.
- Reinforcement learning is an autonomous, self-teaching system that essentially learns by trial and error.
- It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes.

Main points in Reinforcement learning –

Input: The input should be an initial state from which the model will start

Output: There are many possible outputs as there are a variety of solutions to a particular problem

Training: The training is based upon the input. The model will return a state and the user will decide to reward or punish the model based on its output.

The model keeps continues to learn and the best solution is decided based on the maximum reward.

Types: Positive, Negative

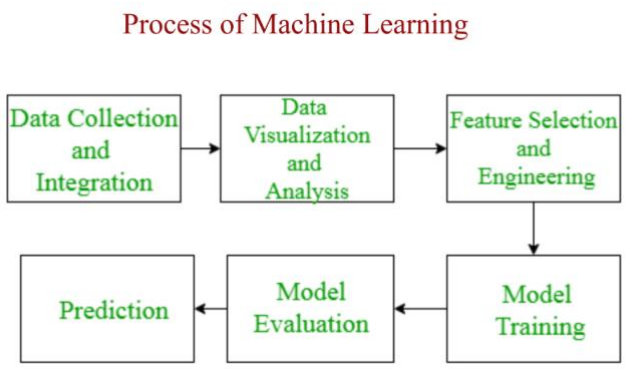
Advantages:

- Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
- The model can correct the errors that occurred during the training process.
- In RL, training data is obtained via the direct interaction of the agent with the environment
- Reinforcement learning can handle environments that are non-deterministic, meaning that the outcomes of actions are not always predictable. This is useful in real-world applications where the environment may change over time or is uncertain.
- Reinforcement learning can be used to solve a wide range of problems, including those that involve decision making, control, and optimization.

Disadvantages:

- Reinforcement learning is not preferable to use for solving simple problems.
- Reinforcement learning needs a lot of data and a lot of computation
- Reinforcement learning is highly dependent on the quality of the reward function. If the reward function is poorly designed, the agent may not learn the desired behaviour.
- Reinforcement learning can be difficult to debug and interpret. It is not always clear why the agent is behaving in a certain way, which can make it difficult to diagnose and fix problems.

<u>Process of Machine Learning:</u>



1. Data Collection and Integration:
    - Data Collected acts as input (features) to model
    - More data, better the model is
    - Integration – placing related data together
    - After this data preparation phase starts

2. Exploratory Data Analysis and Visualization:
    - We have better understanding and notice unseen patterns
    - Helps developers identify outliers and missing data
    - Can be done by plotting histograms, scatter plots etc.

3. Feature Selection and Engineering:
    - Selecting features developers want to use in the model
    - Should be selected such that min correlation b/w them and max correlation b/w selected features and output.
    - Feature engineering is converting raw data into useful data or getting the maximum out of the original data – deals with accuracy and precision.

4. Model Training:
    - Data split into 3 parts – Training, Validation and Test data
    - 70%-80% of data goes into the training data set which is used in training the model.
    - Validation data is used to avoid overfitting or underfitting situations (10-15%)
    - Rest 10%-15% of data goes into the test data set.

5. Model Evaluation:
    - To get the most accurate predictions to test data
    - A confusion matrix is created after model evaluation to calculate accuracy and precision numerically.

6. Prediction:
    - Developer deploys the model.
    - After model deployment, it becomes ready to make predictions.
    - Made on training data and test data to have better understanding of build model.

Bias: Difference between the prediction of the values by the ML model and the correct/actual value.

Variance: Variability of model prediction for a given data point which tells us spread of our data is called the variance of the model.

Bias-Variance Trade Off:

- If the algorithm is too simple then it may be on high bias and low variance condition and thus is error-prone.
- If algorithms fit too complex then it may be on high variance and low bias.
- In the latter condition, the new entries will not perform well. Well, there is something between both of these conditions, known as Trade-off or Bias Variance Trade-off.

- This trade-off in complexity is why there is a trade-off between bias and variance. An algorithm can't be more complex and less complex at the same time.

Underfitting:

A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data.

Reasons:

- High bias and low variance
- The size of the training dataset used is not enough.
- The model is too simple.
- Training data is not cleaned and also contains noise in it.

Techniques to reduce underfitting:

- Increase model complexity
- Increase the number of features, performing feature engineering
- Remove noise from the data.
- Increase the number of epochs or increase the duration of training to get better results.

Overfitting:

A statistical model is said to be over fitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance.

Reasons:

- High variance and low bias
- The model is too complex
- The size of the training data

Techniques to reduce overfitting:

- Increase training data.
- Reduce model complexity.
- Early stopping during the training phase
- Have an eye over the loss over the training period as soon as loss begins to increase stop training).
- Use dropout for neural networks to tackle overfitting.

Curse of Dimensionality: The curse of dimensionality in machine learning is defined as follows, As the number of dimensions or features increases, the amount of data needed to generalize the machine learning model accurately increases exponentially.

Naïve Bayes Classifier Algorithm:

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Bayes Theorem: Used to determine the probability of a hypothesis with prior knowledge.

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

	Outlook	Play
0	Rainy	Yes
1	Sunny	Yes
2	Overcast	Yes
3	Overcast	Yes
4	Sunny	No
5	Rainy	Yes
6	Sunny	Yes
7	Overcast	Yes
8	Rainy	No
9	Sunny	No
10	Sunny	Yes
11	Rainy	No
12	Overcast	Yes
13	Overcast	Yes

1. Frequency table for the Weather Conditions:

Weather	Yes	No
Overcast	5	0
Rainy	2	2
Sunny	3	2
Total	10	5

2. Likelihood table weather condition:

Weather	No	Yes	
Overcast	0	5	5/14= 0.35
Rainy	2	2	4/14=0.29
Sunny	2	3	5/14=0.35
All	4/14=0.29	10/14=0.71	

Applying Bayes' theorem:

P(Yes|Sunny) = P(Sunny|Yes) * P(Yes)/P(Sunny)
P(Sunny|Yes) = 3/10= 0.3
P(Sunny)= 0.35
P(Yes)=0.71
P(Yes|Sunny) = 0.3*0.71/0.35= 0.60

Advantages of Naïve Bayes Classifier:

- Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
- It can be used for Binary as well as Multi-class Classifications.
- It performs well in Multi-class predictions as compared to the other Algorithms.
- It is the most popular choice for text classification problems.

Disadvantages of Naïve Bayes Classifier:

- Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

Applications: Credit Scoring, Medical data classification, Real time predictions, Text classification: Spam filtration, Sentiment Analysis, Classifying Articles.

Weight Space: Parameter space in artificial neural networks, where the parameters are weights on graph edges

Linear Basis Function Models:

Class of machine learning algorithms used for various tasks like regression and classification. These models involve transforming the input data using a set of basis functions to create a linear combination of features, which then forms the foundation for making predictions or classifications.

Key Components:

- Basis functions: They are functions that transform the input features into a new representation. Eg: polynomial basis functions, Gaussian radial basis functions, Fourier basis functions etc. The choice of basis functions impacts how well the model can capture the underlying patterns in the data.
- Weights and Parameters: Linear models include weights or coefficients associated with each basis function. These weights determine the contribution of each basis function to the final prediction. Adjusting these weights during the learning process is essential for the model to fit the data effectively.
- Linear Combination: Linear models combine the basis functions and their associated weights in a linear manner to produce the model's output. The transformed features are linearly weighted to generate predictions.

Aspect	Discriminant Function	Probabilistic Generative Models	Probabilistic Discriminative Models
Main Purpose	Compute decision boundary directly	Model joint distribution of data	Model conditional distribution of classes
Output	Scalar value indicating the class	Probabilities for classes and data	Probabilities for classes given data
Example	Linear discriminant function (LDA), Support Vector Machines (SVMs)	Gaussian Mixture Models (GMMs), Naive Bayes	Logistic Regression, Neural Networks
Training	Learns decision boundary directly from data	Models data generation process	Models decision boundary for classification
Data Generation	Does not generate new data	Can generate new samples	Does not generate new data
Probability Estimation	Does not provide explicit probability estimates	Provides explicit class and data probabilities	Provides class probabilities given data
Use Case	Binary classification, multi-class classification	Image generation, data augmentation	Image classification, sentiment analysis
Performance	Good for simple data distributions, can work well with limited data	May not be optimal for complex distributions, but provides insights into data generation process	Often achieves strong classification performance
Example	Given a new data point, computes a score that determines the class	Given data and classes, models how the data is generated	Given data, models how classes are distributed
Examples	Linear Discriminant Analysis (LDA), Support Vector Machines (SVMs)	Gaussian Mixture Models (GMMs), Naive Bayes	Logistic Regression, Neural Networks

Sr.No	Linear Regression	Logistic Regression
1	Linear regression is used to predict the continuous dependent variable using a given set of independent variables.	Logistic regression is used to predict the categorical dependent variable using a given set of independent variables.
2	Linear regression is used for solving Regression problem.	It is used for solving classification problems.
3	In this we predict the value of continuous variables	In this we predict values of categorical varibles
4	In this we find best fit line.(Linear)	In this we find Sigmoid-Curve .
5	Least square estimation method is used for estimation of accuracy.	Maximum likelihood estimation method is used for Estimation of accuracy.
6	The output must be continuous value,such as price,age,etc.	Output is must be categorical value such as 0 or 1, Yes or no, etc.
7	It required linear relationship between dependent and independent variables.	It not required linear relationship.
8	There may be collinearity between the independent variables.	There should not be collinearity between independent variable.

Linear Regression:

Types of Linear Regression:

Simple Linear Regression (SLR):

- It has only one Independent Variable.
- Example: No. of litres of petrol required and Kilometres driven.

Multiple Linear Regression (MLR):

- It has more than one Independent (or Input) Variables.
- Example: Number of litres of petrol, age of vehicle, speed and kilometres driven.

Algorithm Steps:

- Data Collection
- Data Preprocessing
- Data Splitting
- Model Initialization
- Cost Function
- Gradient Descent
- Model Training
- Making Predictions
- Model Evaluation
- Visualization

Hours Studied (x)	Exam Score (y)
2	50
3	65
4	75
5	80
6	90

y = ax + b

y is the predicted exam score
x is the number of hours studied
m is the slope (coefficient) of the line
b is the y-intercept

Step 1: Calculate the means of x and y:

mean(x) = (2 + 3 + 4 + 5 + 6) / 5 = 4
mean(y) = (50 + 65 + 75 + 80 + 90) / 5 = 72

Step 2: Calculate the slope (a):

a = Σ((x - mean(x)) * (y - mean(y))) / Σ((x - mean(x))^2)

= ((2-4)*(50-72) + (3-4)*(65-72) + (4-4)*(75-72) + (5-4)*(80-72) + (6-4)*(90-72)) / ((2-4)^2 + (3-4)^2 + (4-4)^2 + (5-4)^2 + (6-4)^2) = 485 / 10 = 48.5

Step 3: Calculate the y-intercept (b):

b = mean(y) - m * mean(x) = 72 - 48.5 * 4 = -18

So, the equation for the linear regression line is: y = 48.5x – 18

Logistic Regression:

$$\hat{P} = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

A team scored 285 runs in a cricket match. Assuming regression coefficients to be 0.3548 and 0.00089 respectively, calculate its probability of winning the match.

The logistic regression coefficients are given to be

$\beta_0 = 0.3548$ and $\beta_1 = 0.00089$ and the value of the independent variable (match score) $X_1 = 285$. Putting these values in the logistic regression formula.

$$\hat{P} = \frac{e^{\beta_0 + \beta_1 x_1}}{1 + e^{\beta_0 + \beta_1 x_1}}$$

$$\hat{P} = \frac{e^{0.3548 + 0.00089 \times 285}}{1 + e^{0.3548 + 0.00089 \times 285}} = \frac{1.8375}{1 + 1.8375}$$

$$= 0.6475$$

Hence, the probability of the team winning the match is 0.6475 or 64.75%.

Bayesian Linear Regression:

It is an extension of the traditional linear regression that incorporates Bayesian principles to estimate the parameters of the linear regression model. It combines prior knowledge (prior distribution) about the parameters with observed data to obtain a posterior distribution, allowing for uncertainty quantification and better inference about the model's parameters. It provides a probabilistic framework for modelling uncertainty in parameter estimates, making it particularly useful when dealing with limited data or when you want to quantify uncertainty in your predictions.

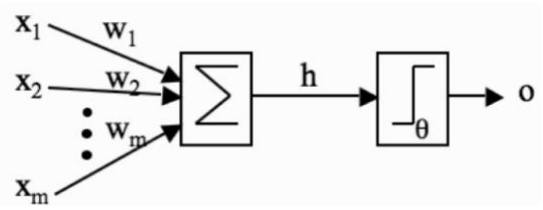
- Model Assumption
- Parameter estimation in standard linear regression
- Bayesian Approach
- Bayes' Theorem
- Posterior distribution
- Parameter Estimation with Bayesian Linear Regression
- Prediction and Uncertainty

Hebb's Rule:

Hebb's rule says that the changes in the strength of synaptic connections are proportional to the correlation in the firing of the two connecting neurons. If two neurons consistently fire simultaneously, then any connection between them will change in strength, becoming stronger. However, if the two neurons never fire simultaneously, the connection between them will die away. The idea is that if two neurons both respond to something, then they should be connected.

"Hebb's Rule states that when two neurons are activated at the same time, the connection between them gets stronger."

McCulloch and Pits Neurons:

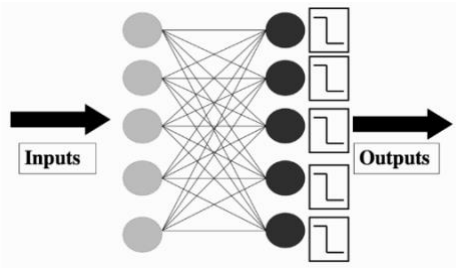


A picture of McCulloch and Pitts' mathematical model of a neuron. The inputs xi are multiplied by the weights wi, and the neurons sum their values. If this sum is greater than the threshold then the neuron fires; otherwise, it does not.

Limitations:

- Binary Activation: The model uses a simple on/off activation, unlike the gradual activation in real neurons.
- No Weights: It doesn't consider the strengths of connections between inputs and the neuron.
- Fixed Threshold: The threshold for activation is fixed, not adaptable.
- Limited Complexity: It can only handle linear decision boundaries, not complex patterns.
- Can't Learn: The model can't learn from data; real neural networks can.
- No Adaptability: It can't adapt to changes or new experiences.
- Oversimplified Biology: It doesn't fully represent how real neurons work.
- Single Output: It produces only one output; real problems often need multiple outputs.
- No Hidden Layers: It lacks the concept of hidden layers for complex tasks.
- Limited Generalization: It struggles to apply learning to new, unseen data.

Perceptron:



The Perceptron is nothing more than a collection of McCulloch and Pitts neurons together with a set of inputs and some weights to fasten the inputs to the neurons.

The Perceptron network, consisting of a set of input nodes (left) connected to McCulloch and Pitts neurons using weighted connections.

The neurons in the Perceptron are completely independent of each other.

Perceptron Structure:

- A perceptron consists of three main components: Inputs (x1, x2, ..., xn), Weights (w1, w2, ..., wn): Associated with inputs, Activation Function: This function determines whether the neuron should fire or not based on the weighted sum of inputs.

Perceptron Operation:

The operation of a perceptron involves these steps:

- Multiply each input by its corresponding weight.
- Sum up the weighted inputs.
- Pass the sum through an activation function.

Activation Function:

The activation function is typically a step function or a similar function that converts the sum of weighted inputs into a binary output (0 or 1), representing a decision.

Use in Binary Classification:

Perceptron's are often used for binary classification tasks, where the output represents a class prediction (e.g., 0 or 1, yes or no).

Learning and Adaptation:

Perceptron's can learn from data using a learning algorithm like the perceptron learning rule. This algorithm adjusts the weights based on misclassifications to improve the accuracy of predictions.

Neural Network:

A perceptron (fundamental unit in neural networks) is a single neuron, and multiple perceptron's are typically combined in layers to create a neural network capable of solving more complex problems.

Linear Separability:

It refers to the property of data points being separable by a straight line. Linear separability is most commonly associated with binary classification problems, where you're trying to separate data points into two classes.

Linear Decision Boundary: The dividing line (or hyperplane) is called a linear decision boundary. It's determined by the weights and bias in the linear equation.

Non-Linear Separability: If it's not possible to draw a straight line to separate the data points, the data is not linearly separable. In this case, you might need more complex models or techniques to classify the data accurately.

Handling Non-Linear Data: For data that is not linearly separable, more advanced techniques, such as kernel methods, neural networks with non-linear activation functions, and decision trees, can be used to capture complex patterns and classify the data accurately.

Linear Equation for a Decision Boundary: In a binary classification problem, you're trying to classify data points into two classes: positive (1) and negative (0). A linear equation for a decision boundary can be written as: $w0+w1*x1+w2*x2 = 0$

Determining Classes: Given the equation, you can determine the class of a data point based on where it lies with respect to the decision boundary: If $w_o + w_1 * x_1 + w_2 * x_2$ is positive or zero, the data point is classified as positive (1). If $w_o + w_1 * x_1 + w_2 * x_2$ is negative, the data point is classified as negative.

Visual Representation: In a 2D space, the decision boundary is a straight line. If the data points can be perfectly separated by this line, the data is linearly separable.

Linearly Separable Data: For linearly separable data, there exists a set of weights (w0, w1, w2) that allows you to draw a line that separates the positive and negative classes without any misclassifications.

Linear Discriminant Analysis (LDA):

- Linear Discriminant Analysis (LDA) is a dimensionality reduction and classification technique used in machine learning and statistics.
- Its primary purpose is to find a linear combination of features that best characterizes or discriminates between classes or groups in a dataset.
- It's a supervised learning algorithm, meaning it uses labelled data to learn patterns and make predictions.

Steps:

- Collect data with labels
- Compute class means
- Compute Scatter matrices
- Compute Eigenvectors and Eigenvalues
- Sort by Eigenvalues
- Select discriminant functions
- LDA function
- Projection onto lower dimensional space

Applications: Face Recognition, Text Classification, Image Analysis

Advantages	Disadvantages
Reduces data dimensions while retaining information	Assumes data follows a Gaussian distribution
Enhances class separability	Sensitive to outliers
Effective with small data sets	Limited to linear transformations
Facilitates classification	Requires labeled data
Computationally efficient	Prone to overfitting, particularly with small datasets

Principle Component Analysis (PCA):

- Principal Component Analysis (PCA) is a widely used technique in machine learning and statistics for dimensionality reduction and feature extraction.
- It aims to find a new set of uncorrelated variables called principal components that capture the maximum variance in the original data.

Steps:

- Standardize the data
- Calculate covariance matrix
- Compute Eigenvectors and Eigenvalues
- Sort by Eigenvalues
- Choose principal components
- Form feature matrix
- Projection onto lower dimensional space
- Optional Reconstruction

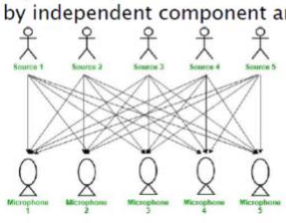
Applications: Dimensionality Reduction, Pattern Recognition, Signal Processing, Image Compression

Advantages	Disadvantages
Simplifies complex data structures	May result in information loss due to dimension reduction
Reduces overfitting risk	Assumes linear relationships within the data
Speeds up learning algorithms	Interpretability can be challenging in high dimensions
Enhances visualization	Sensitive to feature scaling
Handles multicollinearity	Requires careful handling of missing data

Independent Component Analysis (ICA):

- Independent Component Analysis (ICA) is a statistical technique used to separate a multivariate signal into additive subcomponents that are statistically independent or as independent as possible.
- It's widely used in signal processing, image analysis, and various other domains.
- A simple application of ICA is the "cocktail party problem", where the underlying speech signals are separated from a sample data consisting of people talking simultaneously in a room.
- Usually, the problem is simplified by assuming no time delays or echoes.

Consider *Cocktail Party Problem* or *Blind Source Separation* problem to understand the problem which is solved by independent component analysis.



Here, There is a party going into a room full of people. There is 'n' number of speakers in that room and they are speaking simultaneously at the party. In the same room, there are also 'n' number of microphones placed at different distances from the speakers which are recording 'n' speakers' voice signals.

Applications: Blind Source Separation, Image Deblurring and denoising, Speech and Audio Processing

Advantages	Disadvantages
Unmixes mixed signals into statistically independent components	Assumes statistical independence, which might not always hold
Useful in separating sources in signal processing	Sensitive to noise in the data
Can discover underlying hidden factors	Non-unique solutions may occur
Widely used in various domains such as image processing	Computational complexity increases with the number of components
Effective in feature extraction	Requires a large amount of data for accurate results

Principal Component Analysis	Independent Component Analysis
It reduces the dimensions to avoid the problem of overfitting.	It decomposes the mixed signal into its independent sources' signals.
It deals with the Principal Components.	It deals with the Independent Components.
It focuses on maximizing the variance.	It doesn't focus on the issue of variance among the data points.
It focuses on the mutual orthogonality property of the principal components.	It doesn't focus on the mutual orthogonality of the components.
It doesn't focus on the mutual independence of the components.	It focuses on the mutual independence of the components.

Expectation Maximization (EM) Algorithm:

- Used for latent variables (not directly observable and actually inferred from values of other observed variables) too in order to predict their values with condition that general form of probability distribution governing those latent variables is known to us.
- Base of many unsupervised clustering algos
- Uses available observed data of dataset to estimate missing data and use this data to update values of parameters

Steps:

- Expectation - using observed available data of dataset to estimate values of missing data
- Maximization – Complete data generated after expectation to update parameters

Advantages of EM algorithm –

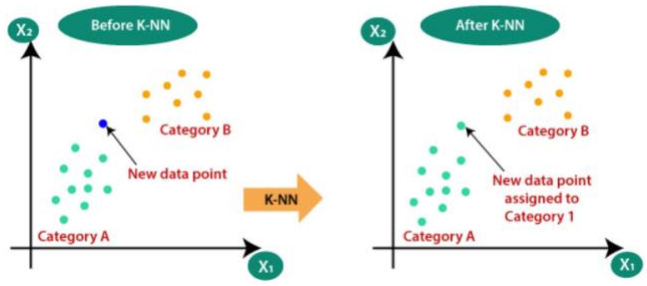
- It is always guaranteed that likelihood will increase with each iteration.
- The E-step and M-step are often pretty easy for many problems in terms of implementation.
- Solutions to the M-steps often exist in the closed form.

Disadvantages of EM algorithm –

- It has slow convergence.
- It makes convergence to the local optima only.
- It requires both the probabilities, forward and backward (numerical optimization requires only forward probability).

K - Nearest Neighbour (KNN):

Step-1: Select the number K of the neighbours
Step-2: Calculate the Euclidean distance of K number of neighbours
Step-3: Take the K nearest neighbours as per the calculated Euclidean distance.
Step-4: Among these k neighbours, count the number of the data points in each category.
Step-5: Assign the new data points to that category for which the number of the neighbour is maximum.



There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
Very low value can be noisy and lead to effect of outliers
Large values of K are good but may find some difficulties

Advantages:

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

Disadvantages:

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

Cover's Theorem on the Separability of Patterns:

Cover states that a pattern classification problem cast in a nonlinear high-dimensional space is more likely to be linearly separable than in a low-dimensional space. This statement is called Cover's Theorem on separability of patterns.

https://vtechworks.lib.vt.edu/bitstream/handle/10919/36847/Ch3.pdf

Radial-Basis-Function Networks:

Radial basis function (RBF) networks are a commonly used type of artificial neural network for function approximation problems. Radial basis function networks are distinguished from other neural networks due to their universal approximation and faster learning speed. An RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer.

The input layer receives input data and passes it into the hidden layer, where the computation occurs. The hidden layer of Radial Basis Functions Neural Network is the most powerful and very different from most Neural networks. The output layer is designated for prediction tasks like classification or regression.

https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-are-radial-basis-functions-neural-networks

K-Means Clustering:

K-means clustering is a method for grouping n observations into K clusters. It uses vector quantization and aims to assign each observation to the cluster with the nearest mean or centroid, which serves as a prototype for the cluster. Originally developed for signal processing, K-means clustering is now widely used in machine learning to partition data points into K clusters based on their similarity. The goal is to minimize the sum of squared distances between the data points and their corresponding cluster centroids, resulting in clusters that are internally homogeneous and distinct from each other.

https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

Recursive Least-Squares Estimation of the Weight Vector:

Adaptive filter algorithm that recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This approach is in contrast to other algorithms such as the least mean squares (LMS) that aim to reduce the mean square error. In the derivation of the RLS, the input signals are considered deterministic, while for the LMS and similar algorithms they are considered stochastic. Compared to most of its competitors, the RLS exhibits extremely fast convergence. However, this benefit comes at the cost of high computational complexity.

https://en.wikipedia.org/wiki/Recursive_least_squares_filter

Hybrid Learning Procedure for RBF Networks:

The hybrid learning algorithm (HLA), combining the gradient paradigm and the linear least square (LLS) paradigm, can be used to adjust the centres and the widths. This algorithm includes two passes, namely, forward pass and backward pass. In the forward pass, we supply input data and functional signals to calculate the hidden output R. Then, the weight W is modified by the LLS method. In the backward pass, the errors propagate from the output end towards the input end. Keeping the weight fixed, the centres and widths of the RBF nodes are modified.

https://www.sciencedirect.com/science/article/pii/S0307904X06000965

The Support Vector Machine Viewed as a Kernel Machine:

- Gaussian Kernel: It is used to perform transformation when there is no prior knowledge about data.
- Gaussian Kernel Radial Basis Function (RBF): Same as above kernel function, adding radial basis method to improve the transformation.

- Sigmoid Kernel: this function is equivalent to a two-layer, perceptron model of the neural network, which is used as an activation function for artificial neurons.
- Polynomial Kernel: It represents the similarity of vectors in the training set of data in a feature space over polynomials of the original variables used in the kernel.
- Linear Kernel: used when data is linearly separable.

https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/

Design of Support Vector Machines:

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well it's best suited for classification. The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Hyperplane: Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. $wx+b = 0$.

Support Vectors: Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.

Margin: Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin. The wider margin indicates better classification performance.

Kernel: Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.

Hard Margin: The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.

Soft Margin: When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.

C: Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.

Hinge Loss: A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.

Dual Problem: A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

https://www.geeksforgeeks.org/support-vector-machine-algorithm/