N Venkatesh

nvk.vicky.23@gmail.com

What is Machine Learning:

Machine learning is a subset of AI, which uses algorithms that learn from data to make predictions. These predictions can be generated through supervised learning and unsupervised learning.

What is AI:

AI (Artificial Intelligence) in machine learning is defined the development of algorithms and models that enable computers to learn from data, make decisions, and improve performance over time without explicit programming.

What is Algorithm:

Algorithm is a set of mathematical instructions used to train a model by finding patterns in data and making predictions.

What are the types of in Machine Learning:

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Reinforecement Learning

What is Supervised Learning:

Supervised learning in ML is a type of learning where a model is trained on labeled data (input-output pairs) to make predictions on new, unseen data.

What is Unsupervised Learning:

Unsupervised learning in ML is a type of learning where a model is trained on unlabeled data to find patterns, structures, or relationships in the data without explicit output labels.

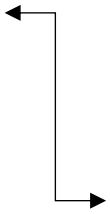
What is Semi-Supervised Learning:

Semi-supervised learning in ML is a technique that uses a small amount of labeled data along with a large amount of unlabeled data to improve model accuracy, bridging the gap between supervised and unsupervised learning.

What is Reinforcement Learning:

Reinforcement learning in ML is when a computer learns to make decisions by trying different actions and learning from the results. It gets rewards when it does something good and penalties when it makes a mistake, and over time, it gets better at making decisions to get the most rewards.

- Environment
- Agent
- Action
- Reward



We will try to make an agent that acts in an environment to increase its chance of winning.
Example: Playing Chess in Computer

What are the types of Supervised Learning:

Classification:

Classification in machine learning is the task of predicting the category or class of an input based on its features.

Example: Email Spam Detection, Medical Diagnosis, Image Recognition

Types of classification:

Type	Definition	Example
Binary Classification	Classifies data into 2 categories.	Spam vs. Not Spam
Multi-Class Classification	Classifies data into 3 or more categories.	Digit Recognition (0–9)
Multi-Label Classification	Data points belong to multiple classes.	Image (Cat, Dog, Outdoor)
Ordinal Classification	Classes have a natural order.	Customer Satisfaction (1–5 scale)
Imbalanced Classification	Class distribution is uneven.	Fraud Detection
Hierarchical Classification	Classes are organized hierarchically.	Species Classification

Binary Classification:

A type of machine learning where the goal is to classify data into **two categories** (like Yes/No or True/False).
Examples:

- Spam detection: Spam or Not Spam.
- Disease diagnosis: Has Disease or Does Not Have Disease.
- Fraud detection: Fraudulent or Legitimate.

Algorithms Used:

- **Logistic Regression**
- **Decision Trees**
- **Support Vector Machines (SVM)**
- **Neural Networks (for complex tasks)**

Multi Class Classification:

Multi-Class Classification in machine learning involves categorizing data into three or more classes or categories

Examples:

- Classifying fruits: Apple, Banana, or Orange.
- Recognizing handwritten digits: 0, 1, 2, ..., 9.
- Sorting news articles: Sports, Politics, Technology, etc..

Algorithms Used:

- **Decision Trees**
- **Random Forest**
- **Logistic Regression (Softmax for multi-class)**
- **Neural Networks**

Classification Metrics:

Metric	Description	Use Case	Pros	Cons	Python Code
Accuracy	Fraction of correctly classified samples over total samples.	Balanced datasets.	Simple to compute and interpret.	Misleading for imbalanced datasets.	accuracy_score(y_true, y_pred)
Precision	Fraction of true positives over all predicted positives.	Reducing false positives.	Useful for imbalanced datasets.	Ignores false negatives.	precision_score(y_true, y_pred)
Recall (Sensitivity)	Fraction of true positives over all actual positives.	Reducing false negatives.	Critical for safety-critical systems.	Ignores false positives.	recall_score(y_true, y_pred)
F1 Score	Harmonic mean of precision and recall.	Balancing precision and recall.	Useful for imbalanced datasets.	May not reflect class distribution.	f1_score(y_true, y_pred)
ROC-AUC	Area under the Receiver Operating Characteristic curve.	Evaluating classifier performance.	Considers all thresholds, good for binary classification.	May not generalize to multi-class directly.	roc_auc_score(y_true, y_scores)
Log Loss	Measures the uncertainty of predictions.	Probabilistic classifiers.	Penalizes overconfident wrong predictions.	Sensitive to probability calibration.	log_loss(y_true, y_probas)
Confusion Matrix	Tabulates true positives, false positives, true negatives, and false negatives.	Diagnosing classifier performance.	Provides a detailed breakdown of results.	Does not summarize performance in a single metric.	confusion_matrix(y_true, y_pred)

Regression

Regression in machine learning is the task of predicting a continuous numeric value based on input features.

Example: Predicting house prices, Sales Prices,Stock Market

Regression Type	Description	Use Case	Algorithm
Linear Regression	Models the relationship between input features and a continuous target variable using a linear approach.	Predicting house prices, sales, etc.	Ordinary Least Squares (OLS)
Polynomial Regression	Extends linear regression by fitting a polynomial relationship between input and target variables.	Modeling non-linear relationships (e.g., sales over time).	Polynomial Regression
Ridge Regression (L2)	Regularized linear regression with a penalty on the sum of squared coefficients to avoid overfitting.	Predicting with many features.	Regularized Linear Regression
Lasso Regression (L1)	Regularized linear regression with a penalty on the sum of absolute coefficients, performing feature selection.	Automatic feature selection, reducing complexity.	Regularized Linear Regression
Elastic Net Regression	Combines penalties of both Ridge and Lasso regression for more balanced regularization.	When there are highly correlated features.	Regularized Linear Regression
Support Vector Regression (SVR)	Uses SVM principles to find a function that fits data within a tolerance margin.	Non-linear data modeling (e.g., stock prices).	Support Vector Machines (SVM)
Decision Tree Regression	Uses a decision tree to predict continuous values based on feature splits.	Non-linear relationships with clear decisions.	Decision Tree Regressor
Random Forest Regression	An ensemble method that combines multiple decision trees to improve prediction accuracy and reduce overfitting.	High accuracy with complex, non-linear data.	Random Forest Regressor
Gradient Boosting Regression	Builds trees sequentially, where each tree corrects the errors of the previous one.	High-performance, minimizing errors.	Gradient Boosting (XGBoost, LightGBM)
K-Nearest Neighbors (KNN) Regression	Predicts target values by averaging the target of the k-nearest neighbors in the feature space.	Predicting based on similarity (e.g., student performance).	K-Nearest Neighbors (KNN)

Type of Model	Model Name	Key Algorithms/Techniques
Linear Models	**Linear Regression**	Used for regression tasks; models the relationship between input and output as a linear function.
	Logistic Regression	Used for classification tasks; models probability using a sigmoid function.
	Linear Discriminant Analysis (LDA)	Supervised technique for classification; assumes normal distribution of features within classes.
	Ridge Regression	Regularized version of linear regression to prevent overfitting.
	Lasso Regression	Regularized linear regression with L1 penalty, promoting sparsity in feature selection.
	Elastic Net	Combines L1 and L2 regularization for more robust regression models.
Non-Linear Models	**Decision Trees**	A tree-based model that splits data based on feature values; works for both classification and regression tasks.
	Random Forests	Ensemble of decision trees that improves accuracy by averaging predictions and reducing overfitting.
	Support Vector Machines (SVM)	Can be linear or non-linear (with kernels); finds the optimal hyperplane for classification.
	k-Nearest Neighbors (KNN)	A non-parametric model where predictions are based on the majority of the 'k' nearest data points.
	Neural Networks (ANN)	Deep learning model with multiple layers, highly flexible in capturing non-linear patterns.
	Gradient Boosting Machines (GBM)	Ensemble technique that builds trees sequentially; commonly used variants include XGBoost, LightGBM, and CatBoost.
	Polynomial Regression	Extension of linear regression where polynomial terms are added to capture non-linear relationships.
	Isolated Forest	An algorithm for anomaly detection that works well for high-dimensional data by isolating outliers.
	Gaussian Processes	A non-linear regression technique based on probability theory, used in applications like time-series forecasting.
	t-SNE	Non-linear dimensionality reduction technique, mainly used for visualization of high-dimensional data.
	Autoencoders	A type of neural network for unsupervised learning, used for data compression and noise reduction.

Regression Metrics:

Metric	Description	Use Case	Python Code
Mean Absolute Error (MAE)	Measures the average absolute difference between predicted and actual values.	Use when you want interpretability and to avoid over-penalizing large errors.	from sklearn.metrics import mean_absolute_error mae = mean_absolute_error(y_true, y_pred)
Mean Squared Error (MSE)	Measures the average squared difference between predicted and actual values.	Use when larger errors should be penalized more.	from sklearn.metrics import mean_squared_error mse = mean_squared_error(y_true, y_pred)
Root Mean Squared Error (RMSE)	Square root of MSE, providing error in the same units as the target variable.	Use when interpretability in the same scale as the target variable is important.	import numpy as np rmse = np.sqrt(mean_squared_error(y_true, y_pred))
R^2 Score (Coefficient of Determination)	Measures the proportion of variance explained by the model, ranging from 0 to 1.	Use to evaluate how well the model explains variability in the data.	from sklearn.metrics import r2_score r2 = r2_score(y_true, y_pred)
Adjusted R^2	Adjusts R^2 for the number of predictors, penalizing for overfitting.	Use when comparing models with different numbers of features.	n, p = len(y_true), X.shape[1] adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)
Mean Absolute Percentage Error (MAPE)	Measures percentage error, ignoring scale, but can over-penalize small actual values.	Use when scale invariance is needed but with caution for small actual values.	mape = np.mean(np.abs((y_true - y_pred) / y_true)) * 100
Mean Squared Log Error (MSLE)	Measures the squared difference of log-transformed predictions and actuals.	Use when penalizing under-predictions less is preferred, such as in growth modeling.	from sklearn.metrics import mean_squared_log_error msle = mean_squared_log_error(y_true, y_pred)

What are the types of Unsupervised Learning:

Clustering:

Clustering in machine learning is defined as grouping the data points into clusters based on their similarities and differences. It works on unlabelled data.

Example: Grouping customers based on purchasing behavior or grouping news articles based on topics.

Clustering Type	Description	How It Works	Use Case	Pros	Cons
K-Means Clustering	Partitions data into K clusters based on the similarity of data points.	Assigns points to centroids and recalculates centroids iteratively.	Market segmentation, image compression.	Simple, fast, works well for spherical clusters.	Requires K to be defined, sensitive to initial centroids.
Hierarchical Clustering	Builds a tree-like structure (dendrogram) by merging or splitting clusters.	Agglomerative: merges closest clusters; Divisive: splits clusters.	Gene expression, hierarchical data.	Does not require the number of clusters to be set.	Computationally expensive, sensitive to noise.
DBSCAN	Groups points based on density and marks outliers as noise.	Identifies dense regions and marks isolated points as noise.	Geospatial data, anomaly detection.	Can find arbitrarily shaped clusters, no need for K.	Sensitive to parameter choices, not good for clusters of varying densities.
K-Medoids	Similar to K-means, but uses actual data points as centers (medoids).	Selects data points as cluster centers instead of centroids.	Robust clustering with outliers.	More robust to outliers than K-means.	Computationally more expensive than K-means.
Mean Shift Clustering	Identifies high-density regions and assigns points to those regions (modes).	Moves data points towards the mode (high-density regions).	Image segmentation, object tracking.	Can find arbitrarily shaped clusters, no need for K.	Computationally expensive.
Gaussian Mixture Models	Models data as a mixture of multiple Gaussian distributions.	Uses Expectation-Maximization (EM) to estimate Gaussian parameters.	Anomaly detection, density estimation.	Can model complex distributions, soft clustering.	Assumes Gaussian distribution, computationally expensive.
Spectral Clustering	Uses eigenvalues of a similarity matrix to reduce dimensionality before applying another algorithm.	Builds a similarity graph, then applies K-means on eigenvectors.	Image segmentation, social network analysis.	Can handle complex cluster structures.	Computationally expensive, needs similarity matrix.
Affinity Propagation	Identifies exemplars (representative points) for each cluster and forms clusters around them.	Messages exchanged between points to determine exemplars.	Customer segmentation, speech recognition.	No need to specify K, can find cluster centers automatically.	Computationally expensive.
Agglomerative Clustering	A type of hierarchical clustering that merges clusters iteratively until all points are in one.	Merges closest clusters bottom-up.	Text clustering, document grouping.	Does not require the number of clusters to be set.	Computationally expensive for large datasets.

Clustering Metrics:

Metric	Description	Purpose
Silhouette Score	Measures how similar a point is to its cluster compared to other clusters.	Evaluates cluster separation.
Davies-Bouldin Index	Measures average similarity ratio of each cluster with the cluster most similar to it.	Assesses clustering quality.
Adjusted Rand Index (ARI)	Compares similarity of cluster assignments to a ground truth, adjusted for chance.	Evaluates external clustering.
Normalized Mutual Information (NMI)	Measures information shared between cluster assignments and ground truth.	Evaluates external clustering.
Within-Cluster Sum of Squares (WCSS)	Measures the total variance within clusters (used in K-means).	Assesses compactness.
Elbow Method (WCSS Curve)	Plots WCSS for different cluster counts to identify the optimal number of clusters.	Determines optimal K.

Dimensionality Reduction:

Dimensionality Reduction in machine learning is the process of reducing the number od features in a dataset while retaining its essential information.

Technique	Type	Use Case	Main Advantage
PCA	Linear	Data visualization, noise reduction	Fast, simple, and effective for linear data
LDA	Supervised Linear	Classification problems, feature reduction	Maximizes class separability
t-SNE	Non-linear	High-dimensional data visualization	Preserves local structure and pairwise distances
Autoencoders	Non-linear (Neural Net)	Complex data (e.g., images, time-series)	Learns non-linear feature representations
Isomap	Non-linear	Non-linear manifold data	Captures global structure in low dimensions
ICA	Non-linear	Signal processing, source separation	Separates independent signals
SVD	Linear (Matrix factorization)	Text analysis, recommendation systems	Decomposes large sparse matrices
UMAP	Non-linear	Visualization, clustering, scalable data	Faster and scalable alternative to t-SNE

Feature Agglomeration	Clustering-based	Correlated features merging	Reduces redundancy by merging similar features
NMF	Non-negative	Text mining, image processing	Suitable for parts-based representations

Principal Component Analysis (PCA):

- It is a dimensionality reduction technique.
- PCA transforms high dimensional data into a lower dimensional space by identifying the directions that maximize variance.

Labeled data:

Labeled data in supervised learning is data that comes with both input and the correct output.

For example:

- **Input (Feature):** A picture of a cat.
- **Label (Output):** "Cat."

Unlabeled data:

Unlabeled data in unsupervised learning means data that doesn't have predefined answers or categories

For example:

A collection of images without tags describing what they show

Outliners:

Outliers are data points that significantly differ from the majority of the dataset, often due to variability, measurement errors, or anomalies.

By using KNN Imputation we can fill null values:

From sklearn.impute import KNNImputer
impute = KNNImputer()

For i in **df.select_dtypes** (include='number').**columns:**
    df[i] = impute.**fit_transform**(df[i])

Normalization:

Normalization is a technique in machine learning that scales data to fit within a specific range, like 0 to 1, to ensure consistency and improve model performance.

Standardization:

Standardization is the process of scaling data to have a mean of 0 and a standard deviation of 1, ensuring all features are on the same scale.

Difference between Normalization and Standardization :

Normalization	Standardization
Rescales values to a range between 0 and 1	Centers data around the mean and scales to a standard deviation of 1
Useful when the distribution of the data is unknown or not Gaussian	Useful when the distribution of the data is Gaussian or unknown
Sensitive to outliers	Less sensitive to outliers
Retains the shape of the original distribution	Changes the shape of the original distribution
May not preserve the relationships between the data points	Preserves the relationships between the data points
Equation: (x – min)/(max – min)	Equation: (x – mean)/standard deviation

Error:

In machine learning, an error is defined as the difference between the predicted output of a model and the actual observed output from the dataset.

Types of Errors:

- **Training Error**: The error calculated on the training dataset.
- **Validation Error**: The error calculated on the validation dataset to evaluate the model's performance during training.
- **Testing Error**: The error calculated on the test dataset to assess the model's generalization ability.

Gradient Descent:

Gradient Descent is an optimization algorithm used in machine learning and deep learning to minimize the error of a model by iteratively adjusting its parameters.

Assumptions of Linear Regression:

Linear regression relies on the following key assumptions:

- **Linearity**:
  - The data should be Linear
  - There should be Linear relationship between the dependent variable and independent variable.
- **Independence**:
  - The errors are should not be dependent on one another.
  - There should be no correlation between the residual terms
  - This phenomenon is known as **Auto-Correlation.**
- **Homoscedasticity**:
  The variance of residuals (errors) is constant across all levels of the independent variables.
- **Normality of Residuals**:
  Residuals (errors) are normally distributed, particularly for hypothesis testing and confidence interval estimation.
- **No Multicollinearity**:
  Independent variables should not be highly correlated with each other.
- **Fixed Independent Variables**:
  The values of the independent variables are fixed or measured without error.
- **No Endogeneity**:
  Independent variables are not correlated with the error term.

Multicollinearity:

If more than two features are highly correlated that will be called as Multi-Collinearity.

Pairwise Correlations:

A pairwise correlation measures the degree to which two variables are linearly related, usually quantified by the correlation coefficient (r), which ranges from -1 to 1.

- **+1**: Perfect positive correlation (both variables increase together).
- **-1**: Perfect negative correlation (one variable increases as the other decreases).
- **0**: No linear correlation.

Variance Inflation Factor (VIF):

Variance Inflation Factor (VIF) is a measure used to detect multicollinearity in regression analysis. It quantifies how much the variance of a regression coefficient is inflated due to the correlation between predictor variables.

- If VIF is greaterthan 10 then we will be said as there is multicollinearity.
- If VIF is lessthan 10 then that will be consider as a good value.

Issues Caused by Multicollinearity

- Unstable Coefficients
- Inflated Standard Errors
- Reduced Model Interpretability
- Overfitting

Solutions to Multicollinearity:

- Remove One of the Correlated Variables
- Principal Component Analysis (PCA)
- Regularization (Ridge or Lasso Regression)
- Increase Sample Size
- Combine Correlated Features

Over Fitting:

If Training accuracy is high and test accuracy is low then it is called as Overfitting.

Under Fitting:

If Training accuracy is low and test accuracy is low then it is called as Underfitting.

Parameters:

Parameters is a type of variable which is used during the training process to optimize the model performance on a given task and to make predictions.

Sigmoid Function:

Sigmoid Function is used to convert the prediction values into the range of 0 to 1.

What is the difference between bagging and boosting:

Bagging and boosting are both ensemble learning techniques, but they work in different ways.

Bagging	Boosting
Various training data subsets are randomly drawn with replacement from the whole training dataset.	Each new subset contains the components that were misclassified by previous models.
Bagging attempts to tackle the over-fitting issue.	Boosting tries to reduce bias.
If the classifier is unstable (high variance), then we need to apply bagging.	If the classifier is steady and straightforward (high bias), then we need to apply boosting.
Every model receives an equal weight.	Models are weighted by their performance.
Objective to decrease variance, not bias.	Objective to decrease bias, not variance.
It is the easiest way of connecting predictions that belong to the same type.	It is a way of connecting predictions that belong to the different types.
Every model is constructed independently.	New models are affected by the performance of the previously developed model.

Classification Notes:

- Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.
- Classification can be performed on structured or unstructured data.
- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. Eg: Gender classification (Male / Female)
- **Multi-class classification:** Classification with more than two classes. In multi class classification each sample is assigned to one and only one

target label. Eg: An animal can be cat or dog but not both at the same time

- **Multi-label classification:** Classification task where each sample is mapped to a set of target labels (more than one class). Eg: A news article can be about sports, a person, and location at the same time.

Logistic Regression Classifier:

- A logistic regression classifier is a statistical model used for binary or multi-class classification that predicts the probability of class membership using a logistic (sigmoid) function.

When should we use logistic regression?

- When our data is binary: 0/1, True/False, Yes/No
- When we need probabilistic results
- When our data is linearly separable
- When we need to understand the impact of the feature.

The types of logistic regression:

- Binary (eg. Tumor Malignant or Benign)
- Multi-linear functions failsClass (eg. Cats, dogs or Sheep's)

What is the Cost Function J(θ)?

The Cost Function J(θ) represents optimization objective i.e. we create a cost function and minimize it so that we can develop an accurate model with minimum error.

What is Gradient Descent?

The main goal of Gradient descent is to minimize the cost value. i.e. min J(θ). Now to minimize our cost function we need to run the gradient descent function on each parameter.

Naive Bayes Classifier:

Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Types Of Naive Bayes Algorithms:

- **Gaussian Naive Bayes:**
  Best suited for continuous data where features are assumed to follow a normal distribution.

- **Multinomial Naive Bayes:**
  Works well with categorical data or frequency-based data (like word counts in text classification).

- **Bernoulli Naive Bayes:**
  Ideal for binary/boolean data, where each feature represents the presence or absence of a specific event.

- **Complement Naive Bayes:**
  A variation of Multinomial Naive Bayes, which is designed to improve performance on imbalanced datasets.

Applications of Naive Bayes Algorithms:

- Real time Prediction
- Multi class Prediction
- Text classification/ Spam Filtering/ Sentiment Analysis
- Recommendation System

Stochastic Gradient Descent (SGD):

Stochastic Gradient Descent (SGD) is an optimization algorithm used to minimize a loss function in machine learning, particularly in the context of training models like linear regression, logistic regression, and neural networks.

Or

Stochastic Gradient Descent is a fast and efficient optimization technique commonly used in training machine learning models, especially when handling large-scale data.

Gradient Descent:

Gradient Descent is an optimization algorithm used to minimize the loss function in machine learning and deep learning models. The goal is to find the optimal model parameters (weights) that minimize the error between the model's predictions and the actual data.

Key Points:

- **Concept**: It iteratively adjusts the parameters of the model in the direction of the **negative gradient** (the steepest descent) of the loss function, aiming to reach the global minimum or a local minimum.

- **Learning Rate**: The size of each step taken towards the minimum is controlled by the **learning rate**. A smaller learning rate takes smaller steps, while a larger learning rate takes larger steps.

- **Steps**:
  - Calculate the gradient (derivative) of the loss function with respect to the model parameters.
  - Update the parameters by subtracting the gradient multiplied by the learning rate.

- **Types**:
  - **Batch Gradient Descent**: Computes the gradient using the entire dataset (slow for large datasets).
  - **Stochastic Gradient Descent (SGD)**: Uses a single data point to compute the gradient (faster but noisier).
  - **Mini-batch Gradient Descent**: Uses small batches of data for gradient computation (a compromise between batch and SGD).

The libraries for neural networks:

- Adam
- Adagrad
- Adadelta
- RMSProp

K Neighbors Classifier:

- K-Neighbors Classifier is a supervised machine learning algorithm that classifies a data point based on the majority class of its K nearest neighbors in the feature space.

Support Vector Classification:

- Support Vector Classification (SVC) is a supervised machine learning algorithm that finds the hyperplane that best separates data into different classes by maximizing the margin between them.

- What is Hyperplane?
  Hyperplane is a decision boundary that separates different classes or groups in the data. It is a flat affine subspace of one dimension less than the data's feature space.

- In **2D** (two features), a hyperplane is a line.
- In **3D** (three features), a hyperplane is a plane.

  In classification the hyperplane is used to separate data points from different classes in the best possible way.

Support Vector Machines (SVMs) have several types based on how they handle data and the problem they solve. Here are the main types:

- **Linear SVM**
  - If a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
  - Used when the data is linearly separable.
  - Finds a straight hyperplane to separate the classes.
- **Non-linear SVM**
  - If a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.
  - Used when data is not linearly separable.
  - Applies the **kernel trick** to transform the data into a higher-dimensional space where it becomes linearly separable.
- **Binary SVM**
  - Handles classification tasks with only two classes.
- **Multi-class SVM**
  - Extends SVM for problems with more than two classes.
  - Common approaches:
    - **One-vs-One (OvO):** Builds one SVM for each pair of classes.
    - **One-vs-Rest (OvR):** Builds one SVM for each class against all other classes.
- **Support Vector Regression (SVR)**
  - Adaptation of SVM for regression problems.
  - Predicts continuous values rather than class labels.
- **What is the difference between linear separable data and non linear separable data?**
  - **Linearly separable data**: Simple boundary, solvable with linear models.
  - **Non-linearly separable data**: Complex boundary, requires transformations or advanced methods.

DecisionTreeClassifier:

- DecisionTreeClassifier is a machine learning algorithm that builds a model by recursively splitting the data into subsets based on feature values, aiming to classify data points by creating a tree-like structure of decisions.

Random Forest Classifier:

- RandomForestClassifier is an ensemble learning algorithm that constructs a multitude of decision trees and combines their results to improve classification accuracy and control overfitting.

X G Boost classifier:

- XGBoost is a powerful, scalable, and efficient gradient boosting algorithm that uses decision trees as base learners to solve regression, classification, and ranking problems.

Differentiate between feature selection and feature extraction summary?

- **Feature Selection** simplifies the dataset by choosing *important features*.
- **Feature Extraction** creates *new, compact features* while preserving the underlying information.