UNIT-1
INTRODUCTION TO MACHINE LEARNING

1.1 Overview of Machine Learning

- Machine Learning is a field of artificial intelligence that allows systems to learn and improve from experience without being explicitly programmed. It is predicated on the notion that computers can learn from data, spot patterns, and make judgments with little human assistance.
- It is the study of making machines more human-like in their behaviour and decisions by giving them the ability to learn and develop their programs. This is done with minimum human intervention, i.e., no explicit programming. The learning process is automated and improved based on the experiences of the machines throughout the process.
- Good quality data is fed to the machines, and different algorithms are used to build ML models to train the machines on this data. The choice of algorithm depends on the type of data at hand and the type of activity that needs to be automated.

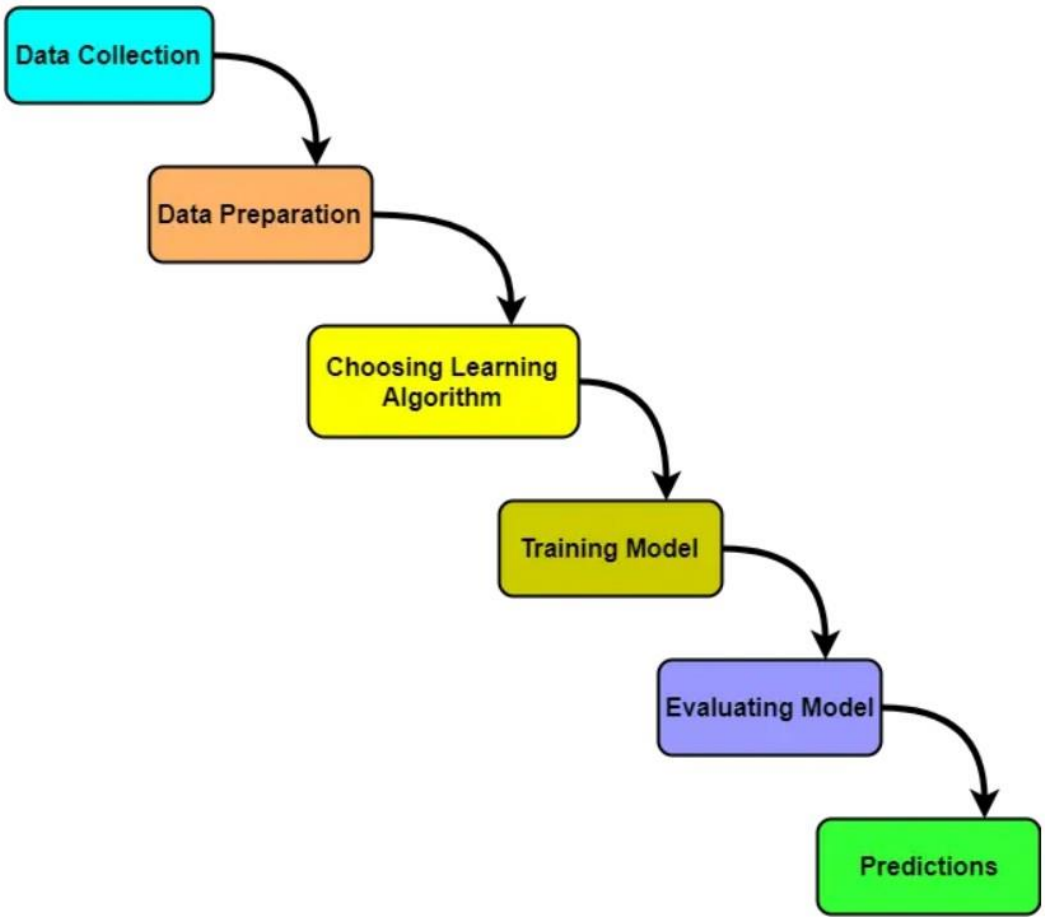
Human Learning vs. Machine Learning

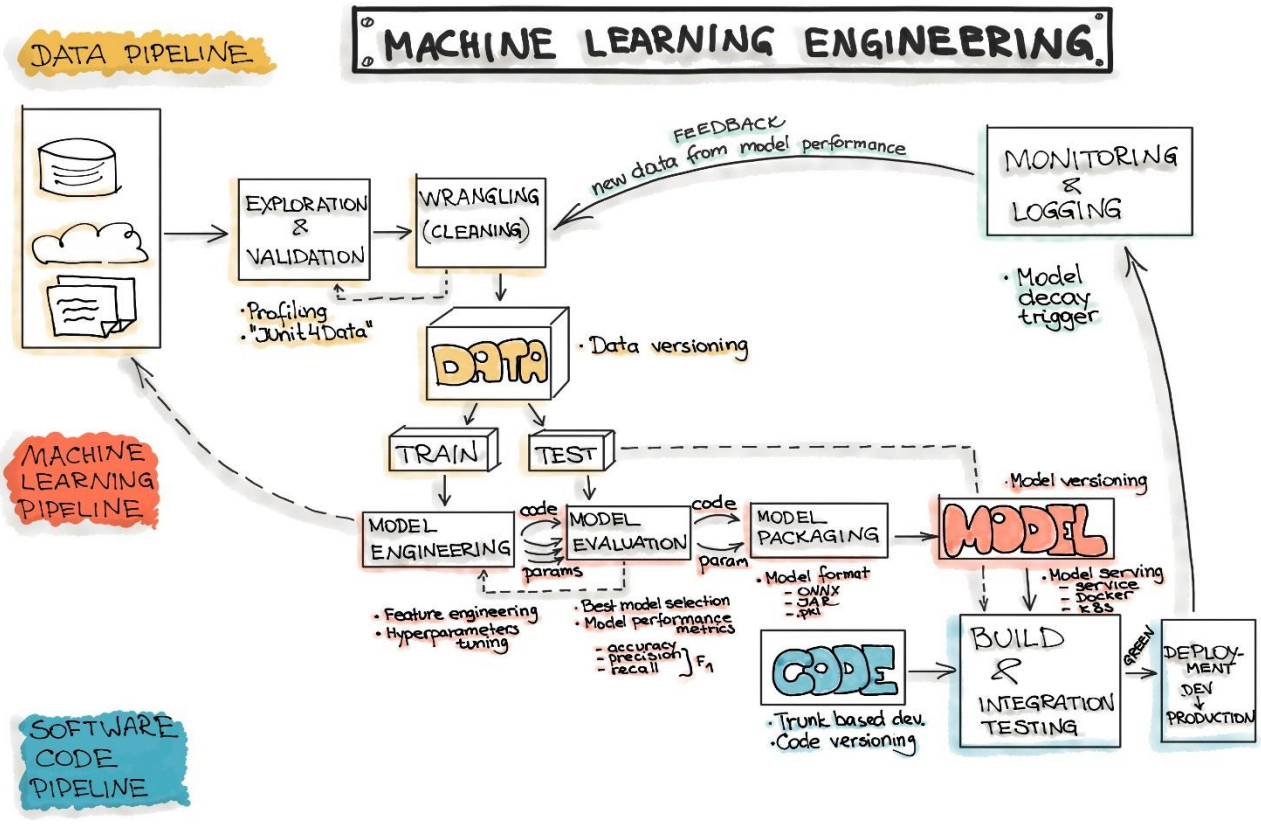
	Human	Machine
Cost	Low initial cost and high running cost.	High initial cost (in case of robots) and low running cost (work 24/7).
Creativity	Creative	Uninspired
Permanency of Intelligence	Human intelligence is perishable. We could not preserve Einstein's intelligence after his death.	Machine intelligence is permanent. It is easy to preserve intelligent tools like Siri and Watson.
Ease of duplication and dissemination of knowledge	Slow language-based communication process, some expertise can never be duplicated.	Knowledge can be copied from a machine and easily moved to another one.
Better in	• fusing data from multiple sources and interpreting the outside world • distinguishing faces • identifying objects • recognizing language sounds • learning from few examples. A kid can differentiate between a man and a tree just by showing him/her one example. • develop new concepts/ imagination and creative reasoning.	• faster at performing arithmetic and logical operations • dealing with multi-dimensional data • discovering complex patterns such as that exist in financial, scientific, or product data. • operations that require fast, precise, highly repeatable actions • working in harsh environments (in case of robots).

1.2 Machine Learning Terminology:

- **Model**: Also known as “hypothesis”, a Machine Learning model is the mathematical representation of a real-world process. A Machine Learning algorithm along with the training data builds a Machine Learning model.
- **Feature**: A feature is a measurable property or parameter of the dataset.
- **Feature Vector**: It is a set of multiple numeric features. We use it as an input to the Machine Learning model for training and prediction purposes.
- **Training**: An algorithm takes a set of data known as “training data” as input. The learning algorithm finds patterns in the input data and trains the model for expected results (target). The output of the training process is the Machine Learning model.
- **Prediction**: Once the Machine Learning model is ready, it can be fed with input data to provide a predicted output.
- **Target (Label)**: The value that the Machine Learning model has to predict is called the target or label.
- **Overfitting**: When a massive amount of data trains a Machine Learning model, it tends to learn from the noise and inaccurate data entries. Here the model fails to characterize the data correctly.
- **Underfitting**: It is the scenario when the model fails to decipher the underlying trend in the input data. It destroys the accuracy of the Machine Learning model. In simple terms, the model or the algorithm does not fit the data well enough.

Machine Learning Workflow





1. Data Collection-

- Data is collected from different sources.
- The type of data collected depends upon the type of desired project.
- Data may be collected from various sources such as files, databases, etc.
- The quality and quantity of gathered data directly affect the accuracy of the desired system.

2. Data Preparation-

In this stage,

- Data preparation is done to clean the raw data.
- Data collected from the real world is transformed into a clean dataset.
- Raw data may contain missing values, inconsistent values, duplicate instances, etc.
- So, raw data cannot be directly used for building a model.

Different methods of cleaning the dataset are-

- Ignoring the missing values
- Removing instances having missing values from the dataset.
- Estimating the missing values of instances using mean, median, or mode.
- Removing duplicate instances from the dataset.
- Normalizing the data in the dataset.

3. Choosing Learning Algorithm-

In this stage,

- The best-performing learning algorithm is researched.
- It depends upon the type of problem that needs to be solved and the type of data we have.
- If the problem is to classify and the data is labeled, classification algorithms are used.

- If the problem is to perform a regression task and the data is labeled, regression algorithms are used.
- If the problem is to create clusters and the data is unlabeled, clustering algorithms are used.

4. Training Model-

In this stage,

- The model is trained to improve its ability.
- The dataset is divided into a training dataset and a testing dataset.
- The training and testing split in order of 80/20 or 70/30.
- It also depends upon the size of the dataset.
- Training dataset is used for training purposes.
- Testing dataset is used for testing purposes.
- Training dataset is fed to the learning algorithm.
- The learning algorithm finds a mapping between the input and the output and generates the model.

5. Evaluating Model-

In this stage,

- The model is evaluated to test if the model is any good.
- The model is evaluated using the kept-aside testing dataset.
- It allows to test of the model against data that has never been used before for training.
- Metrics such as accuracy, precision, recall, etc are used to test the performance.
- If the model does not perform well, the model is re-built using different hyperparameters.
- The accuracy may be further improved by tuning the hyperparameters.

6. Predictions-

In this stage,

- The built system is finally used to do something useful in the real world.
- Here, the true value of machine learning is realized.

1.3 Artificial Intelligence vs. Machine Learning

Artificial Intelligence	Machine Learning
Artificial intelligence is the ability for a machine to mimic human behavior.	Using machine learning, a machine learns from past data without having to be explicitly programmed. It is a subset of artificial intelligence.
The goal is to increase the likelihood of success rather than accuracy.	The goal is to improve accuracy, but it is unconcerned about success.
Artificial intelligence aspires to create an intelligent system capable of performing a wide range of complex tasks.	Machine learning seeks to build machines that can only perform the tasks for which they have been trained.
Artificial intelligence is designed to solve complex problems by simulating natural intelligence.	Machine learning is designed to learn from data on a specific task in order to improve performance on that task.
A wide range of applications is possible with artificial intelligence.	Machine learning has limited scope.
Artificial intelligence can be classified into three broad categories based on its capabilities, namely, artificial narrow intelligence (ANI), artificial general intelligence (AGI), and artificial super intelligence (ASI).	Machine learning is also classified into three types,namely, supervised learning, unsupervised learning, and reinforcement learning.
Applications of artificial intelligence include Siri, customer service via expert systems, online gaming, intelligent humanoid robots, and so on.	Applications of machine learning include online recommendation systems, Google search algorithms, Facebook auto friend tagging suggestions, and so on.

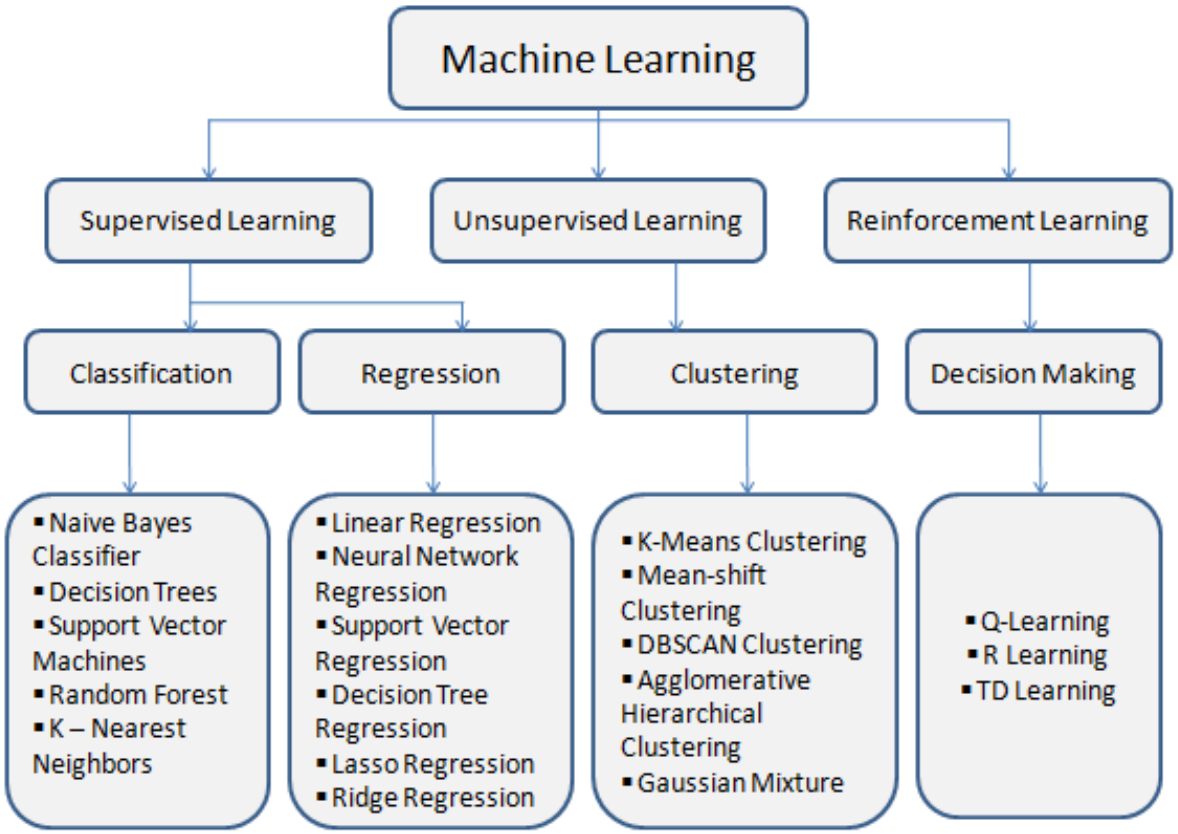
1.4. Types of Machine Learning

Machine Learning can be classified into three broad categories:

- **Supervised learning:**
    - Supervised learning is a class of problems that uses a model to learn the mapping between the input and target variables. Applications consisting of the training data describing the various input variables and the target variable are known as supervised learning tasks.
    - Let the set of input variable be (x) and the target variable be (y). A supervised learning algorithm tries to learn a hypothetical function which is a mapping given by the expression y=f(x), which is a function of x.
    - The learning process here is monitored or supervised. Since we already know the output the algorithm is corrected each time it makes a prediction, to optimize the results. Models are fit on training data which consists of both the input and the output variable and then it is used to make predictions on test data. Only the inputs are provided during the test phase and the outputs produced by the model are compared with the kept-back target variables and are used to estimate the performance of the model.
    - There are two types of supervised problems: Classification – which involves the prediction of a class label and Regression – which involves the prediction of a numerical value. 2. Unsupervised learning – Also called descriptive learning. A machine finds patterns in unknown objects by grouping similar objects.

- **Unsupervised learning:**
    - In an unsupervised learning problem, the model tries to learn by itself recognize patterns, and extract the relationships among the data. As in the case of supervised learning, there is no supervisor or teacher to drive the model. Unsupervised learning operates only on the input variables. There are no target variables to guide the learning process. The goal here is to interpret the underlying patterns in the data to obtain more proficiency in the underlying data.
    - There are two main categories in unsupervised learning: clustering – where the task is to find out the different groups in the data. And the next is Density Estimation – which tries to consolidate the distribution of data. These operations are performed to understand the patterns in the data. Visualization and Projection may also be considered unsupervised as they try to provide more insight into the data. Visualization involves creating plots and graphs on the data and Projection is involved with the dimensionality reduction of the data.
- **Reinforcement learning**
    - Reinforcement learning is type a of problem where there is an agent and the agent is operating in an environment based on the feedback or reward given to the agent by the environment in which it is operating. The rewards could be either positive or negative. The agent then proceeds in the environment based on the rewards gained.
    - The reinforcement agent determines the steps to perform a particular task. There is no fixed training dataset here and the machine learns on its own.

- Playing a game is a classic example of a reinforcement problem, where the agent's goal is to acquire a high score. It makes successive moves in the game based on the feedback given by the environment which may be in terms of rewards or penalization. Reinforcement learning has shown tremendous results in Google's AplhaGo of Google which defeated the world's number one Go player.



1.5. Tools and Technology for Machine Learning

Various tools and technologies support different stages of the Machine Learning workflow, from data preprocessing to model deployment. Here's a brief overview of some key tools and technologies in the field of Machine Learning:

- **Programming Languages:**
  - Python: Widely used for ML due to its extensive libraries (NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch).
  - R: Commonly used for statistical analysis and data visualization in ML.
- **Libraries and Frameworks:**
  - Scikit-learn: A simple and efficient tool for data analysis and modeling, built on NumPy, SciPy, and Matplotlib.
  - TensorFlow: Developed by Google, it's an open-source ML framework used for building and training deep learning models.
  - PyTorch: Developed by Facebook, it's another popular deep learning framework known for its dynamic computation graph.
  - Keras: High-level neural networks API running on top of TensorFlow or Theano, simplifying the process of building and training models.
  -
- **Data Processing and Analysis:**
  - NumPy and Pandas: Fundamental libraries for numerical operations and data manipulation in Python
- **Visualization Tools:**
  - Matplotlib and Seaborn: Python libraries for creating static, animated, and interactive visualizations.

- TensorBoard: A web-based tool provided with TensorFlow for visualizing Machine Learning experiments.

1.6. Application of Machine Learning

- **Facial recognition/Image recognition**
  - There are a lot of use cases of facial recognition, mostly for security purposes like identifying criminals, searching for missing individuals, aiding forensic investigations, etc. Intelligent marketing, diagnosing diseases, and tracking attendance in schools, are some other uses.
- **Automatic Speech Recognition**
  - Abbreviated as ASR, automatic speech recognition is used to convert speech into digital text. Its applications lie in authenticating users based on their voice and performing tasks based on human voice inputs. Speech patterns and vocabulary are fed into the system to train the model. Presently ASR systems find a wide variety of applications in the following domains:
    - Medical Assistance
    - Industrial Robotics
    - Forensic and Law enforcement
    - Defense & Aviation
    - Telecommunications Industry
    - Home Automation and Security Access Control
    - I.T. and Consumer Electronics
  - Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, Machine Learning algorithms are widely used in various applications of speech recognition. **Google**

Assistant, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow voice instructions.

- **Financial Services**
  - Machine Learning has many use cases in Financial Services. Machine Learning algorithms prove to be excellent at detecting fraud by monitoring the activities of each user and assessing that if an attempted activity is typical of that user or not. Financial monitoring to detect money laundering activities is also a critical security use case.
  - It also helps in making better trading decisions with the help of algorithms that can analyze thousands of data sources simultaneously. Credit scoring and underwriting are some of the other applications. The most common application in our day-to-day activities is the virtual personal assistants like Siri and Alexa.
- **Traffic predictions**
  - When you use Google Maps to map your commute to work or a new restaurant in town, it provides an estimated time of arrival. Google uses Machine Learning to build models of how long trips will take based on historical traffic data (gleaned from satellites). It then takes that data based on your current trip and traffic levels to predict the best route according to these factors.
- **Healthcare**
  - A vital application is in the diagnosis of diseases and ailments, which are otherwise difficult to diagnose. Radiotherapy is also becoming better.
  - Early-stage drug discovery is another crucial application that involves technologies such as precision medicine and next-generation sequencing. Clinical trials cost a lot of time and money to complete

and deliver results. Applying ML-based predictive analytics could improve on these factors and give better results.

  - These technologies are also critical to making outbreak predictions. Scientists around the world are using ML technologies to predict epidemic outbreaks.

- **Recommendation Systems**
  - Many businesses today use recommendation systems to effectively communicate with the users on their sites. It can recommend relevant products, movies, web series, songs, and much more. The most prominent use cases of recommendation systems are e-commerce sites like Amazon, Flipkart, and many others, along with Spotify, Netflix, and other web-streaming channels.
- **Credit card fraud detection**
  - Predictive analytics can help determine whether a credit card transaction is fraudulent or legitimate. Fraud examiners use AI and Machine Learning to monitor variables involved in past fraud events. They use these training examples to measure the likelihood that a specific event was fraudulent activity.

UNIT-2

Preparing the Model, Modelling, and Evaluation

2.1 Selecting a Model

- Input variables can be denoted by X, while individual input variables are represented as X1, X2, X3, ..., Xn, and output variables by symbol Y. The relationship between X and Y is represented in the general form:

  $Y = f(X) + e$,

  where ‘f ’ is the target function and ‘e’ is a random error term. Just like a target function concerning a machine learning model, some of these functions which are frequently tracked are
- A cost function (also called an error function) helps to measure the extent to which the model is going wrong in estimating the relationship between X and Y. In that sense, the cost function can tell how badly the model is performing. For example, R-squared (to be discussed later in this chapter) is a cost function of a regression model.
- The loss function is almost synonymous with the cost function – only the difference is that the loss function is usually a function defined by a data point, while the cost function is for the entire training data set.
- Machine learning is an optimization problem. We try to define a model and tune the parameters to find the most suitable solution to a problem. However, we need to have a way to evaluate the quality or optimality of a solution. This is done using an objective function. Objective means goal.
- The objective function takes in data and model (along with parameters) as input and returns a value. The target is to find values of model parameters to maximize or minimize the return value. When the objective is to minimize the value, it becomes synonymous with to cost function. Examples: maximize the reward function in reinforcement learning,

maximize the posterior probability in Naive Bayes, and minimize squared error in regression.

2.1.1 Predictive models

- Models for supervised learning or predictive models, as is understandable from the name itself, try to predict certain values using the values in an input data set. The learning model attempts to establish a relation between the target feature, i.e. the feature being predicted, and the predictor features. The predictive models have a clear focus on what they want to learn and how they want to learn. Predictive models, in turn, may need to predict the value of a category or class to which a data instance belongs. Below are some examples:

  1. Predicting a win/loss in a cricket match
  2. Predicting whether a transaction is fraud
  3. Predicting whether a customer may move to another product

  The models that are used for the prediction of target features of categorical value are known as classification models. The target feature is known as a class and the categories to which classes are divided are called levels. Some of the popular classification models include k-Nearest Neighbor (CNN), Naïve Bayes, and Decision Tree. Predictive models may also be used to predict numerical values of the target feature based on the predictor features. Below are some examples:

  1. Prediction of revenue growth in the succeeding year
  2. Prediction of rainfall amount in the coming monsoon
  3. Prediction of potential flu patients and demand for flu shots next winter

- The models that are used for the prediction of the numerical value of the target feature of a data instance are known as regression models. Linear Regression and Logistic Regression models are popular regression models.

2.1.2 Descriptive models

- Models for unsupervised learning or descriptive models are used to describe a data set or gain insight from a data set. There is no target feature or single feature of interest in the case of unsupervised learning. Based on the value of all features, interesting patterns or insights are derived about the data set.
- Descriptive models group together similar data instances, i.e. data instances having a similar value of the different features are called clustering models. Examples of clustering include

  1. Customer grouping or segmentation based on social, demographic, ethnic, etc. factors

  2. Grouping of music based on different aspects like genre, language, period, etc.

  3. Grouping of commodities in an inventory the most popular model for clustering is k-means.
- Descriptive models related to pattern discovery are used for market basket analysis of transactional data. In market basket analysis, based on the purchase pattern available in the transactional data, the possibility of purchasing one product based on the purchase of another product is determined. For example, transactional data may reveal a pattern that generally a customer who purchases milk also purchases biscuits at the same time. This can be useful for targeted promotions or in-store setups.
- Promotions related to biscuits can be sent to customers of milk products or vice versa. Also, in the store products related to milk can be placed close to biscuits.

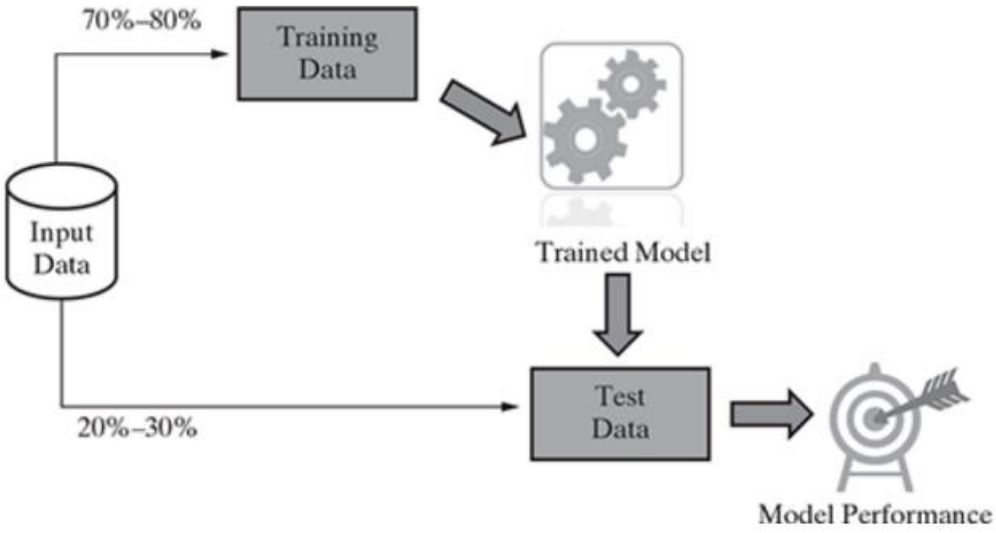
2.2 Training a Model for Supervised Learning

- Training a model simply means learning (determining) good values for all the weights and the bias from labeled examples. In supervised learning, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss; this process is called empirical risk minimization.

2.2.1 Cross-validation techniques

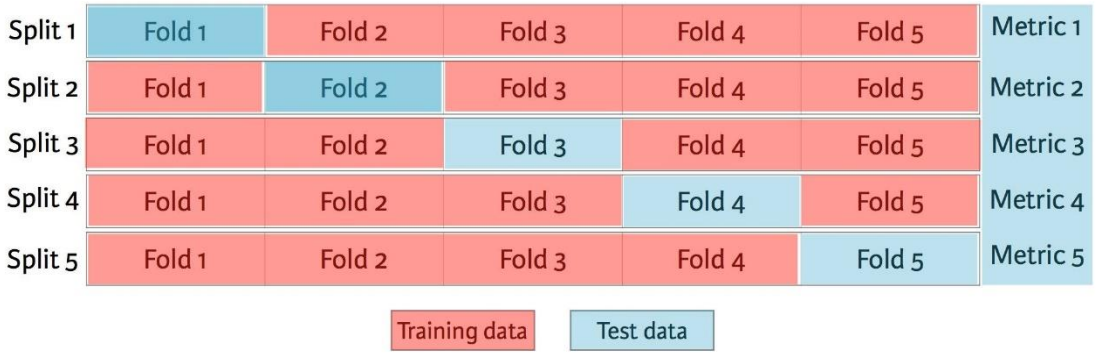
Holdout method

- In the case of supervised learning, a model is trained using the labeled input data. However, how can we understand the performance of the model? The test data may not be available immediately. Also, the label value of the test data is not known. That is the reason why a part of the input data is held back (that is how the name holdout originates) for evaluation of the model. This subset of the input data is used as the test data for evaluating the performance of a trained model.
- In general, 70%–80% of the input data (which is labeled) is used for model training. The remaining 20%–30% is used as test data for validation of the performance of the model. However, a different proportion of dividing the input data into training and test data is also acceptable. To make sure that the data in both buckets are similar, the division is done randomly.
- Random numbers are used to assign data items to the partitions. This method of partitioning the input data into two parts – training and test data, which is by holding back a part of the input data for validating the trained model is known as the holdout method.



K-fold Cross-validation Method

- Cross-validation or 'k-fold cross-validation' is when the dataset is randomly split up into 'k' groups. One of the groups is used as the test set and the rest are used as the training set. The model is trained on the training set and scored on the test set. Then the process is repeated until each unique group has been used as the test set.
- For example, for 5-fold cross-validation, the dataset would be split into 5 groups, and the model would be trained and tested 5 separate times so each group would get a chance to be the test set. This can be seen in the graph below.



2.3 Model Representation and Interpretability

- A machine learning model is interpretable if it's easy for humans to understand how it makes decisions. Interpretability requires a greater level of detail than explainability, which focuses on explaining the decisions made.

  Here are some types of interpretability in machine learning:

  - Model-agnostic: Uses tools that are applied after a model has been trained.
  - Surrogate models: Created by training a linear regression or decision tree on the original inputs and predictions of a complex model.
  - Modularity: A model is modular if a meaningful portion of its prediction-making process can be interpreted independently.

  Here are some types of machine learning models:

  - Descriptive: Helps understand what happened in the past.
  - Prescriptive: Automates business decisions and processes based on data.
  - Predictive: Predicts future business scenarios.

2.3 Evaluating the Performance of a Model

- Model evaluation is the process of using metrics to understand the performance of a machine learning model. It's important to assess the model's efficacy during the initial research phases.

2.4.1 Performance Metrics for Classification

- **Accuracy**

  It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to the total number of predictions made by the classifiers.

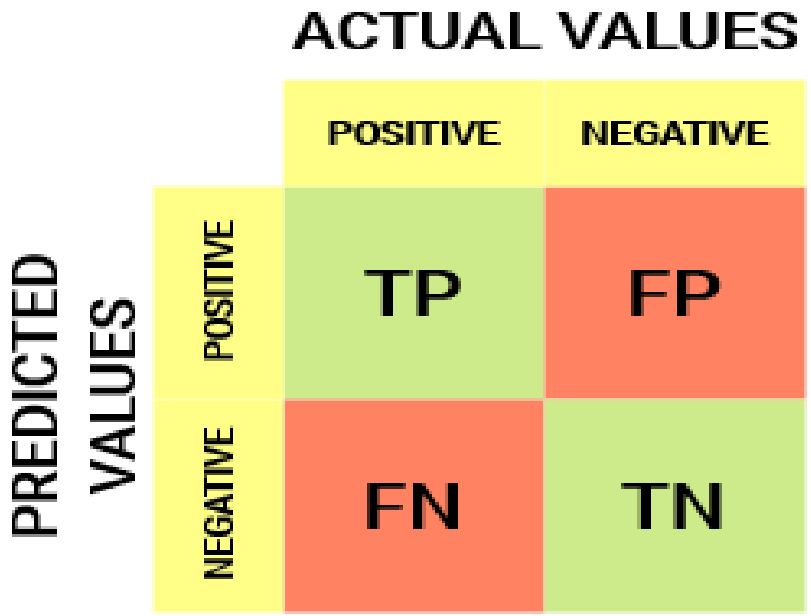
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Confusion Matrix**

  A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known.

  The confusion matrix is simple to implement, but the terminologies used in this matrix might be confusing for beginners.

  A typical confusion matrix for a binary classifier looks like the below image (However, it can be extended to use for classifiers with more than two classes).



1. True Positive (TP): In this case, the prediction outcome is true, and it is true in reality, too.
2. True Negative (TN): in this case, the prediction outcome is false, and it is false in reality, too.
3. False Positive (FP): In this case, prediction outcomes are true, but they are false in actuality.

4. False Negative (FN): In this case, predictions are false, and they are true in actuality.

- **Precision**

  The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was correct. It can be calculated as the True Positive or predictions that are true to the total positive predictions (True Positive and False Positive).

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- **Recall**

  The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

- **F-Score**

  F-score or F1 Score is a metric to evaluate a binary classification model based on predictions that are made for the positive class. It is calculated with the help of Precision and Recall So, the F1 Score can be calculated as the harmonic mean of both precision and Recall, assigning equal weight to each of them.

  The formula for calculating the F1 score is given below:

$$F1 - score = 2 * \frac{precision * recall}{precision + recall}$$

When to use F-Score?

As the F-score makes use of both precision and recall, it should be used if both of them are important for evaluation, but one (precision or recall) is slightly more important to consider than the other. For example, when False negatives are comparatively more important than false positives, or vice versa.

2.4.2 Evaluation Metrics for Regression Task

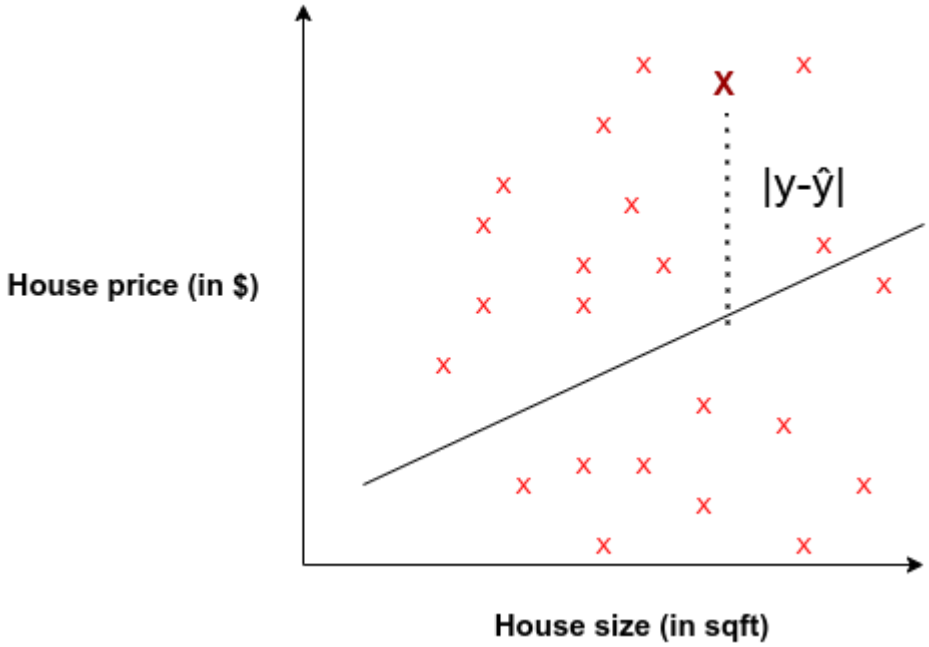
Mean Absolute Error (MAE)

Mean Absolute Error is the average of the difference between the ground truth and the predicted values. Mathematically, it is represented as :

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \check{y}_j|$$

Where:

- y_j: ground-truth value
- y_hat: predicted value from the regression model
- N: number of datums



A few key points for MAE

- It's more robust towards outliers than MAE since it doesn't exaggerate errors.
- It gives us a measure of how far the predictions were from the actual output. However, since MAE uses the absolute value of the residual, it doesn't give us an idea of the direction of the error, i.e. whether we're under-predicting or over-predicting the data.
- Error interpretation needs no second thoughts, as it perfectly aligns with the original degree of the variable.
- MAE is non-differentiable as opposed to MSE, which is differentiable.

Similar to MSE, this metric is also simple to implement.

```
mae = np.abs(y-y_hat)
print(f"MAE: {mae.mean():0.2f} (+/- {mae.std():0.2f})")
```

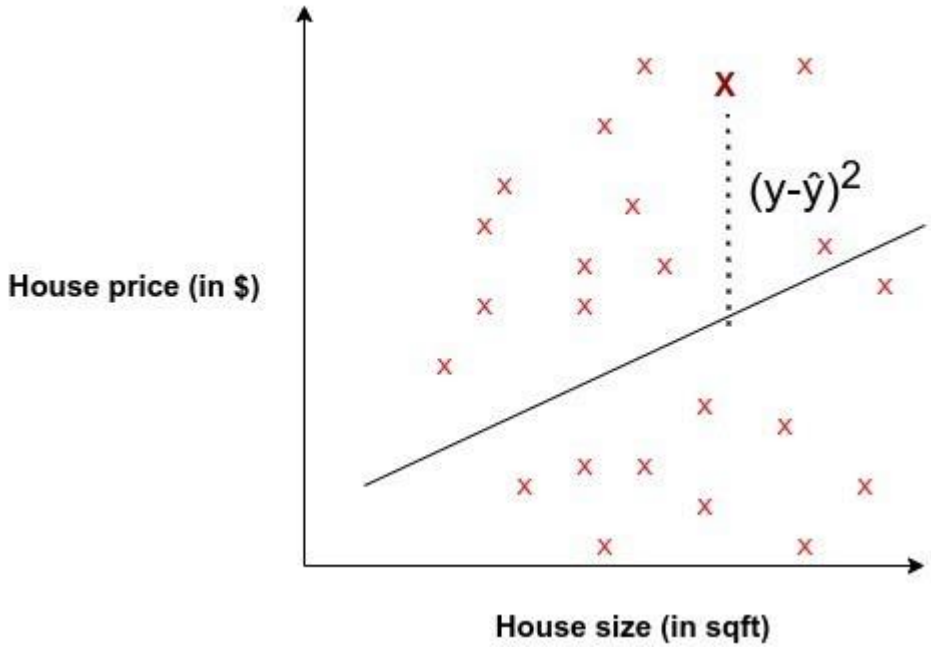
Mean Squared Error (MSE)

Mean squared error is perhaps the most popular metric used for regression problems. It essentially finds the average of the squared difference between the target value and the value predicted by the regression model.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(y_j - \check{y}_j)^2$$

Where:

- y_j: ground-truth value
- y_hat: predicted value from the regression model
- N: number of datums



A few key points related to MSE:

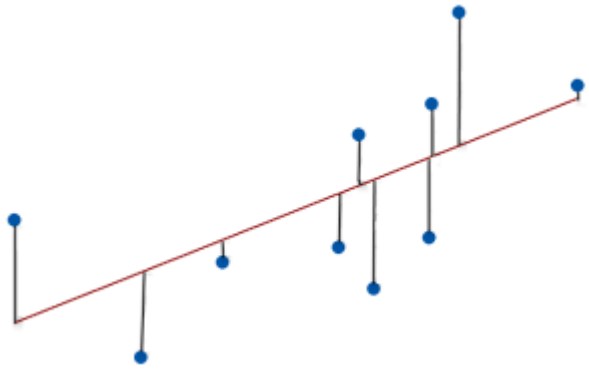
- It's differentiable, so it can be optimized better.
- It penalizes even small errors by squaring them, which essentially leads to an overestimation of how bad the model is.
- Error interpretation has to be done with the squaring factor(scale) in mind. For example, in our Boston Housing regression problem, we got MSE=21.89 which primarily corresponds to $(\text{Prices})^2$.
- Due to the squaring factor, it's fundamentally more prone to outliers than other metrics.

This can be implemented simply using NumPy arrays in Python.

```
me = (y-y_hat)**2
print(f"MSE: {mse.mean():0.2f} (+/- {mse.std():0.2f})")
```

Root Mean Squared Error (RMSE)

- The root mean square error (RMSE) measures the average difference between a statistical model's predicted values and the actual values. Mathematically, it is the standard deviation of the residuals. Residuals represent the distance between the regression line and the data points.
- RMSE quantifies how dispersed these residuals are, revealing how tightly the observed data clusters around the predicted values.



- As the data points move closer to the regression line, the model has less error, lowering the RMSE. A model with less error produces more precise predictions.
- RMSE values can range from zero to positive infinity and use the same units as the dependent (outcome) variable.
- Use the root mean square error to assess the amount of error in a regression or other statistical model. A value of 0 means that the predicted values perfectly match the actual values, but you'll never see that in practice. Low RMSE values indicate that the model fits the data well and has more precise predictions. Conversely, higher values suggest more error and less precise predictions.
- The root mean square error is a non-standardized goodness-of-fit assessment corresponding to its standardized counterpart—R-squared.
- The RSME formula should look familiar because it is essentially the standard deviation formula. That makes sense because the root mean

square error is the standard deviation of the residuals. It measures the scatter of the observed values around the predicted values.

- The RSME formula for a sample is the following:

$$RSME = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{N - P}}$$

yi is the actual value for the ith observation.

ŷi is the predicted value for the ith observation.

N is the number of observations.

P is the number of parameter estimates, including the constant.

Mean Absolute Percentage Error (MAPE)

Mean absolute percentage error (MAPE) is a metric that measures the accuracy of a forecasting method. It's also known as mean absolute percentage deviation (MAPD).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - F_i|}{A_i}$$

A_i is the actual value

F_i is the forecast value

n is total number of observations

- MAPE is calculated by averaging the absolute percentage errors of each entry in a dataset. It's expressed as a ratio, where At is the actual value and Ft is the forecast value.
- MAPE is one of the most commonly used KPIs to measure forecast accuracy. A MAPE value of 20% means that the average absolute percentage difference between the predictions and the actuals is 20%.

- MAPE has some limitations:
    - It can't be used when the actual values have instances of zero.
    - MAPE can sometimes favor models that under-forecast.

R2 Score

- The R2 score, also known as the coefficient of determination, is a statistical measure that evaluates how well a model fits a dataset. It is used to evaluate the performance of a linear regression model.
- The R2 score is calculated by:
    - Subtracting the average actual value from each of the actual values
    - Squaring the results
    - Summing the results
    - Dividing the first sum of errors (unexplained variance) by the second sum (total variance)
    - Subtracting the result from one
- The R2 score takes a value between 0 and 1. A value of 1 indicates that the model perfectly fits the data.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

- The R2 score can also be defined as:
    - The proportion of the variance in the dependent variable that is predictable from the independent variable(s)
    - (total variance explained by model) / total variance

2.4 Improving the Performance of a Model.

Here are some ways to improve the performance of a model:

- Training, testing, and data validation: These are essential steps in evaluating model performance.

- Choose a robust algorithm: Algorithms are the key factor used to train machine learning (ML) models.
- Improve data: Improving the quality and quantity of training data can provide a more robust model performance.
- Feature selection: Use the Correlation Feature Selection method to determine which features influence the output.
- Benchmarking: Compare models on common datasets and evaluation metrics to identify the most suitable model for a particular task.
- Interpret your model accuracy score: Identify what your model struggles to extract.

Other ways to improve model performance include:

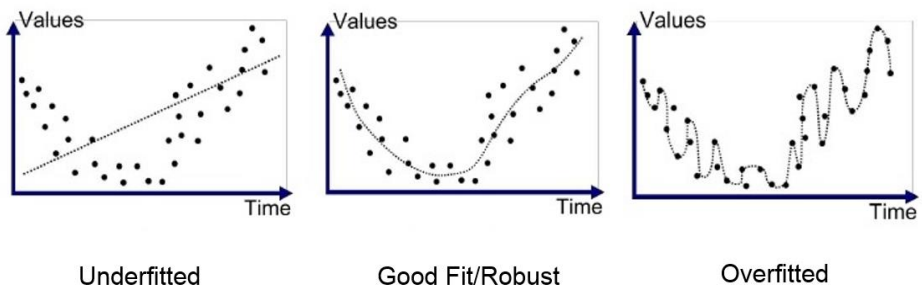
- Operations workflow
- Ensemble learning
- Supervised or unsupervised ML
- Anonymizing data
- Sampling data from large data sets
- Reducing dimensions
- Randomly shifting and rotating existing images

2.4.1 Under-fitting and Over-fitting

- **Under-fitting**

  A data model that can't accurately capture the relationship between input and output variables. Under-fitting causes a high error rate on both the training set and unseen data.

- **Over-fitting**

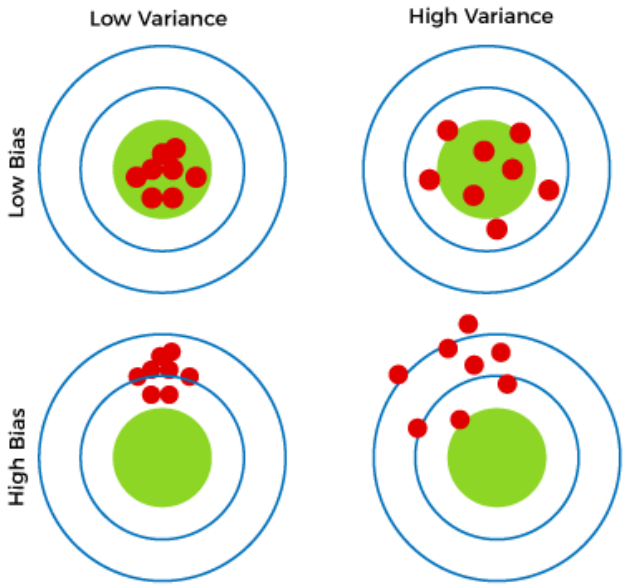


- A modeling error that occurs when a function is too closely aligned to a limited set of data points. Over-fitting causes the model to only be useful in reference to its initial data set.

2.6 Bias–variance tradeoff

What is bias?

- Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.



What is variance?

- Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

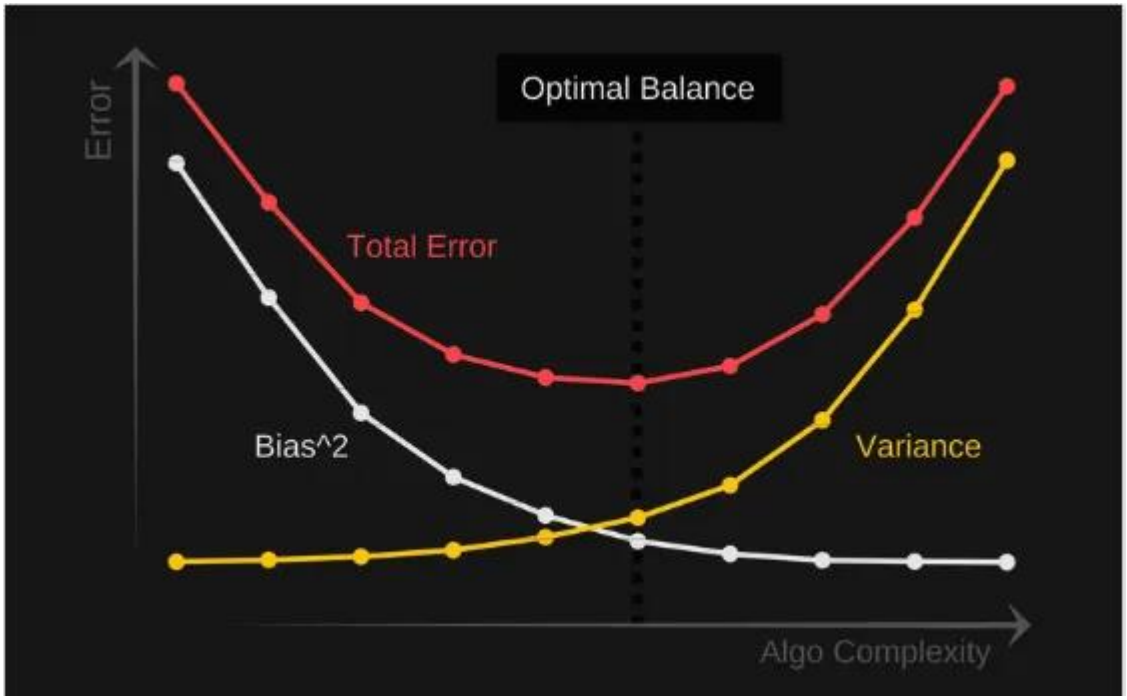
Why is Bias Variance Tradeoff?

- If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand, if our model has large

number of parameters then it’s going to have high variance and low bias. So we need to find the right/good balance without over-fitting and under-fitting the data.

- 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.
- Total Error

  To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.



UNIT-3

PROBABILITY AND STATISTICS

3.1 Overview of Probability

- Probability and statistics are both the most important concepts for Machine Learning. Probability is about predicting the likelihood of future events, while statistics involves the analysis of the frequency of past events.

  Nowadays, Machine Learning has become one of the first choices for most freshers and IT professionals. But, to enter this field, one must have some pre-specified skills, and one of those skills is Mathematics. Mathematics is very important to learning ML technology and developing efficient applications for the business.

- Probability can be calculated by the number of times the event occurs divided by the total number of possible outcomes. Let's suppose we tossed a coin, then the probability of getting head as a possible outcome can be calculated as below formula:

  P (H) = Number of ways to head occur/ total number of possible outcomes

  P (H) = ½

  P (H) = 0.5

  Where;

  P (H) = Probability of occurring Head as outcome while tossing a coin.

3.2 Statistical Tools in Machine Learning

Statistics is a branch of mathematics that deals with collecting, analyzing, interpreting, and visualizing empirical data. Descriptive statistics and inferential statistics are the two major areas of statistics. Descriptive statistics are for describing the properties of sample and population data (what has happened). Inferential statistics use those properties to test hypotheses, reach conclusions, and make predictions (what can you expect).

3.3 Descriptive Statistics

It helps in understanding the basic features of the data by summarizing them numerically or graphically. Facts regarding the data involved can be presented by descriptive analysis, however, any kind of generalization or conclusion is not possible.

Descriptive statistics provide a summary of the data, such as the mean, median, standard deviation, and variance. Univariate descriptive statistics are used to describe data containing only one variable. On the other hand, bivariate and multivariate descriptive statistics are used to describe data with multiple variables.

The marks of students in two classes are {70, 85, 90, 65} and {60, 40, 89, 96}. The average marks for each class are 77.5 and 71.25, respectively.

Descriptive statistics can be broadly classified into two categories - measures of central tendency and measures of dispersion.

Types of Descriptive Statistics:

Descriptive statistics are methods used to summarize and describe the main features of a dataset. They provide a way to organize and simplify data, making it easier to understand and interpret. Here are the major types of descriptive statistics, along with examples and visualizations:

Measures of Central Tendency

Mean The average value of a dataset, calculated by adding all values and dividing by the number of values.

Median: The middle value in a dataset when the values are arranged in order.

Mode: The most frequent value in a dataset.

Example:

Consider the following dataset of scores: 85, 92, 78, 95, 82.

Mean = (85 + 92 + 78 + 95 + 82) / 5 = 86.4

Median = 85 (arranged in order: 78, 82, 85, 92, 95)

Mode = no mode (no value occurs more than once)

Measures of Variability

Range: The difference between the highest and lowest values in a dataset.

Variance: The average of the squared differences from the mean.

Standard Deviation: The square root of the variance, measuring how spread out the values are from the mean.

Example:

Using the same dataset of scores:

Range = 95 - 78 = 17

Variance = 35.36

Standard Deviation = 5.95

3.4 Inferential Statistics

It is simply used for explaining the meaning of descriptive stats. It is simply used to analyze, interpret results, and draw conclusions.

Inferential statistics can be classified into hypothesis testing and regression analysis. Hypothesis testing also includes the use of confidence intervals to test the parameters of a population. Given below are the different types of inferential statistics.

Types of Inferential Statistics:

Hypothesis testing is a part of statistics in which we make assumptions about the population parameter. So, hypothesis testing mentions a proper procedure by analyzing a random sample of the population to accept or reject the assumption.

Z-test

Z-test is mainly used when the data is normally distributed. We find the Z-statistic of the sample means and calculate the z-score. Z-score is given by the formula,

$$\textbf{Z-score} = (x - \mu) / \sigma$$

Z-test is mainly used when the population mean and standard deviation are given.

Confidence interval:

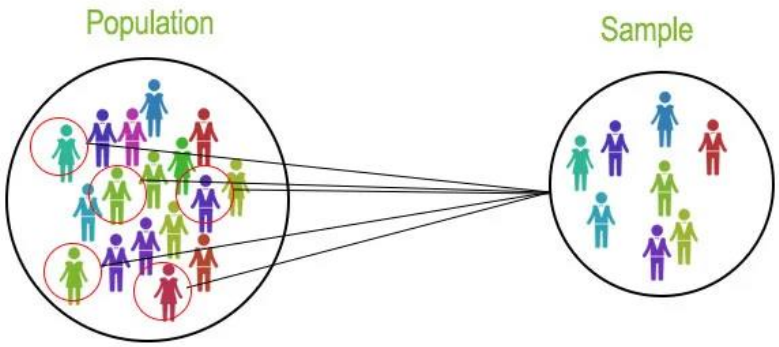
A confidence interval is a range of values that is likely to contain the true population parameter. It is used to estimate the range of values in which the population parameter lies. The confidence interval is calculated from the sample data and is often used in hypothesis testing.

Population

It refers to the collection that includes all the data from a defined group being studied. The size of the population may be either finite or infinite.

Sample

The study of the entire population is always not feasible, instead, a portion of data is selected from a given population to apply the statistical methods. This portion is called a Sample. The size of the sample is always finite



	Descriptive Statistics	Inferential Statistics
Purpose	Describe and summarize the data	Make inferences and draw conclusions about a population based on sample data
Data Analysis	Analyzes and interprets the characteristics of a dataset	Uses sample data to make generalizations or predictions about a larger population
Population vs Sample	Focuses on the entire population or dataset	Focuses on a subset of the population (sample) to draw conclusions about the entire population
Measurements	Provides measures of central tendency and dispersion	Estimates parameters, test hypotheses, and determines the level of confidence or significance in the results
Examples	Mean, median, mode, standard deviation, range, frequency tables	Hypothesis testing, confidence intervals, regression analysis, ANOVA (analysis of variance), chi-square tests, t-tests, etc.
Goal	Summarize, organize, and present data	Generalize findings to a larger population, make predictions, test hypotheses, evaluate relationships, and support decision-making
Population Parameters	Not typically estimated	Estimated using sample statistics (e.g., sample mean as an estimate of population mean)

Sample Representativeness	Not required	Crucial; that the sample should be representative of the population to ensure accurate inferences

3.5 Concept of Probability

Probability means possibility. It is a branch of mathematics that deals with the occurrence of a random event. The value is expressed from zero to one.

Experiment as a process that generates well-defined outcomes. On any single repetition of an experiment, one and only one of the possible experimental outcomes will occur.

Examples: Hitting a target, checking the boiling point of a liquid, taking an examination for a student, conducting interviews for some jobs, tossing a coin, rolling a die, hitting a ball with a batsman, sale of products, chemical reaction of elements, are few examples of experiments.

The **sample space** for an experiment is the set of all experimental outcomes.

Example: In the experiment of hitting a target, sample space can be hitting a target, missing the target.

An **Event** is one or more of the possible outcomes of an experiment.

Example: If we toss a coin, getting a head will be one event, and getting a tail will be another event.

For example:- when we toss a coin, either we get Head OR Tail, only two possible outcomes are possible (H, T). But when two coins are tossed then there will be four possible outcomes, i.e. {(H, H), (H, T), (T, H), (T, T)}.

Joint Probability

When the probability of two more events occurring together and at the same time is measured it is marked as Joint Probability. For two events A and B, it is denoted by joint probability is denoted as, $P(A \cap B)$ intersection of two or more events.

Formula: $P(A \cap B) = P(A) * P(B)$

Example: Find the probability that the number three will occur twice when two dice are rolled at the same time.

Solution: Number of possible outcomes when a die is rolled = 6

i.e. {1, 2, 3, 4, 5, 6}

Let A be the event of occurring 3 on first die and B be the event of occurring 3 on the second die.

Both the dice have six possible outcomes, the probability of a three occurring on each die is 1/6.

P(A) =1/6

P(B)=1/6

P(A, B) = 1/6 x 1/6 = 1/36

Marginal Probability

Probability of a single event occurring, independent of other events. It's found by summing the probabilities of the event across all possible outcomes of the other variable(s).

Now we have to calculate these probabilities by using a two-way table.

If you are given a pmf = pXY(x,y) , and we will calculate the marginal probability pY(y).

To calculate the marginal probability we will use the formula $py(y)=\sum_i p(xi,y)$.

Let's draw a table to calculate these probabilities.

$p(x, y)$	$X = 3$	$X = 4$
$Y = 2$	0.2	0.1
$Y = 3$	0.1	0.2
$Y = 4$	0.1	0.3

Now if we wish to calculate the marginal pY(3)

Now by using the formula of marginal py(y)=∑ip(xi,y) at Y=3

⇒pY(3)=P(Y=3)

⇒pY(3)=P(Y=3,X=3) + P(Y=3,Y=4)

Now from the table, if we look at the values as mentioned in the above expression then we get

⇒pY(3)=0.1+0.2

⇒pY(3)=0.3

Here we get the marginal probability of the taken example.

Conditional Probability

The probability of an event A based on the occurrence of another event B is termed conditional Probability. It is denoted as P(A|B) and represents the probability of A when event B has already happened.

Here:

P(A | B) = The probability of A given B (or) the probability of A which happens after B

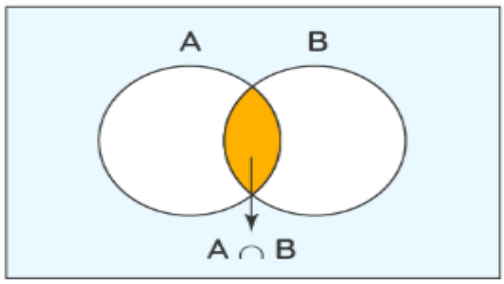
P(B | A) = The probability of B given A (or) the probability of B which happens after A

P(A ∩ B) = The probability of happening of both A and B

P(A) = The probability of A

P(B) = The probability of B

Conditional Probability Formula



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

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

Example: A bag contains 3 red and 7 black balls. Two balls are drawn at random without replacement. If the second ball is red, what is the probability that the first ball is also red?

Solution:

Let A: event of selecting a red ball in first draw

B: event of selecting a red ball in the second draw

P(A ∩ B) = P(selecting both red balls) = 3/10 × 2/9 = 1/15

P(B) = P(selecting a red ball in the second draw) = P(red ball and rad ball or black ball and red ball)

= P(red ball and red ball) + P(black ball and red ball)

= 3/10 × 2/9 + 7/10 × 3/9 = 3/10

∴ P(A|B) = P(A ∩ B)/P(B) = 1/15 ÷ 3/10 = 2/9.

Example: Two dice are rolled, if it is known that atleast one of the dice always shows 4, find the probability that the numbers appeared on the dice have a sum 8.

Solution:

Let,

A: one of the outcomes is always 4

B: sum of the outcomes is 8

Then, A = {(1, 4), (2, 4), (3, 4), (4, 4), (5, 4), (6, 4), (4, 1), (4, 2), (4, 3), (4, 5), (4, 6)}

B{(4, 4), (5, 3), (3, 5), (6, 2), (2, 6)}

$n(A) = 11$, $n(B) = 5$, $n(A \cap B) = 1$

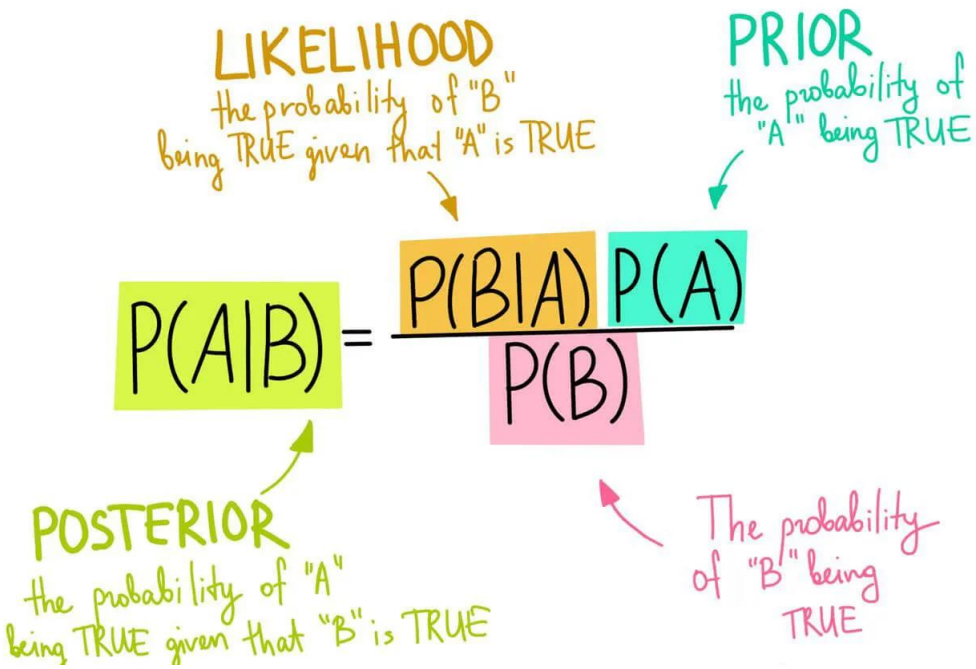
$P(B|A) = n(A \cap B)/n(A) = 1/11$.

Actually the basic difference between them is that the joint probability is the probability of two events occurring simultaneously, and in the marginal probability is the probability of an event irrespective of the outcome of another variable, and conditional probability is the probability of one event occurring in the presence of a second event.

Bayes' Theorem

Bayes' theorem is also known as **Bayes' rule, Bayes' law**, or **Bayesian reasoning**, which determines the probability of an event with uncertain knowledge.

In probability theory, it relates the conditional probability and marginal probabilities of two random events.



P(A|B) is known as **posterior**, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the **prior probability**, probability of hypothesis before considering the evidence

P(B) is called **marginal probability**, pure probability of an evidence.

Example:

There are two urns containing colored balls. The first urn contains 50 red balls and 50 blue balls. The second urn contains 30 red balls and 70 blue balls. One of the two urns is randomly chosen (both urns have a probability of 50% of being chosen) and then a ball is drawn at random from one of the

two urns. If a red ball is drawn, what is the probability that it comes from the first urn?

Solution

In probabilistic terms, what we know about this problem can be formalized as follows:

$$P(\text{red}|\text{urn } 1) = \frac{1}{2}$$
$$P(\text{red}|\text{urn } 2) = \frac{3}{10}$$
$$P(\text{urn}1) = \frac{1}{2}$$
$$P(\text{urn } 2) = \frac{1}{2}$$

The unconditional probability of drawing a red ball can be derived using the law of total probability:

$$\begin{aligned} P(\text{red}) &= P(\text{red}|\text{urn } 1)P(\text{urn } 1) + P(\text{red}|\text{urn } 2)P(\text{urn } 2) \\ &= \frac{1}{2} \cdot \frac{1}{2} + \frac{3}{10} \cdot \frac{1}{2} \\ &= \frac{1}{4} + \frac{3}{20} \\ &= \frac{5+3}{20} = \frac{2}{5} \end{aligned}$$

By using Bayes' rule, we obtain

$$\begin{aligned} P(\text{urn } 1|\text{red}) &= \frac{P(\text{red}|\text{urn } 1)P(\text{urn } 1)}{P(\text{red})} \\ &= \frac{\frac{1}{2} \cdot \frac{1}{2}}{\frac{2}{5}} \\ &= \frac{1}{4} \cdot \frac{5}{2} = \frac{5}{8} \end{aligned}$$

3.6 Random Variables

A random variable is a variable which represents the outcome of a trial, an experiment, or an event. It is a specific number which is different each time the trial, experiment, or event is repeated.

- A random variable is a variable whose value is unknown or a function that assigns values to each of an experiment's outcomes.

- A random variable can be either discrete (having specific values) or continuous (any value in a continuous range).
- The use of random variables is most common in probability and statistics, where they are used to quantify outcomes of random occurrences.
- Risk analysts use random variables to estimate the probability of an adverse event occurring.

Types of Random Variables

Continuous random variable

Continuous random variables take up an infinite number of possible values which are usually in a given range. Typically, these are measurements like weight, height, the time needed to finish a task, etc.

To give you an example, the life of an individual in a community is a continuous random variable. Let's say that the average lifespan of an individual in a community is 110 years.

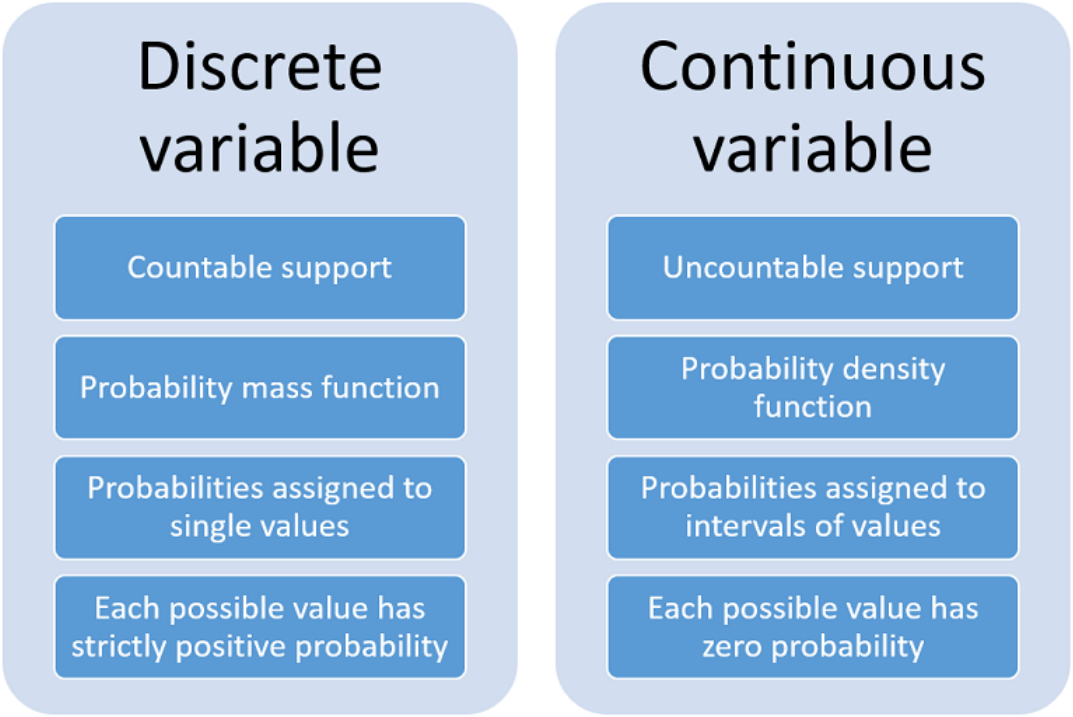
Therefore, a person can die immediately on birth (where life = 0 years) or after he attains an age of 110 years. Within this range, he can die at any age. Therefore, the variable 'Age' can take any value between 0 and 110.

Hence, continuous random variables do not have specific values since the number of values is infinite. Also, the probability at a specific value is almost zero.

Discrete random variable

Discrete random variables take on only a countable number of distinct values. Usually, these variables are counts (not necessarily though). If a random variable can take only a finite number of distinct values, then it is discrete.

Number of members in a family, number of defective light bulbs in a box of 10 bulbs, etc. are some examples of discrete random variables.



3.7 Probability Distribution

Sampling Distribution

A sampling distribution is a probability distribution of a statistic that is based on random samples from a population. It describes the range of possible outcomes for a statistic, such as the mean or mode of a variable.

Discrete Distribution

A discrete probability distribution is a type of probability distribution that shows all possible values of a discrete random variable along with the associated probabilities. In other words, a discrete probability distribution gives the likelihood of occurrence of each possible value of a discrete random variable.

Such a distribution will represent data that has a finite countable number of outcomes

A discrete probability distribution counts occurrences that have countable or finite outcomes.

In finance, discrete distributions are used in options pricing and forecasting market shocks or recessions.

Represented by bars or points, such as in a histogram or probability mass function plot.

Examples: binomial distribution, Poisson distribution, geometric distribution

Continuous Distribution

Continuous Probability Distributions. A continuous distribution describes the probabilities of a continuous random variable's possible values. A continuous random variable has an infinite and uncountable set of possible values (known as the range).

Involves continuous random variables that can take any value within a range. Examples include height, weight, temperature, and time.

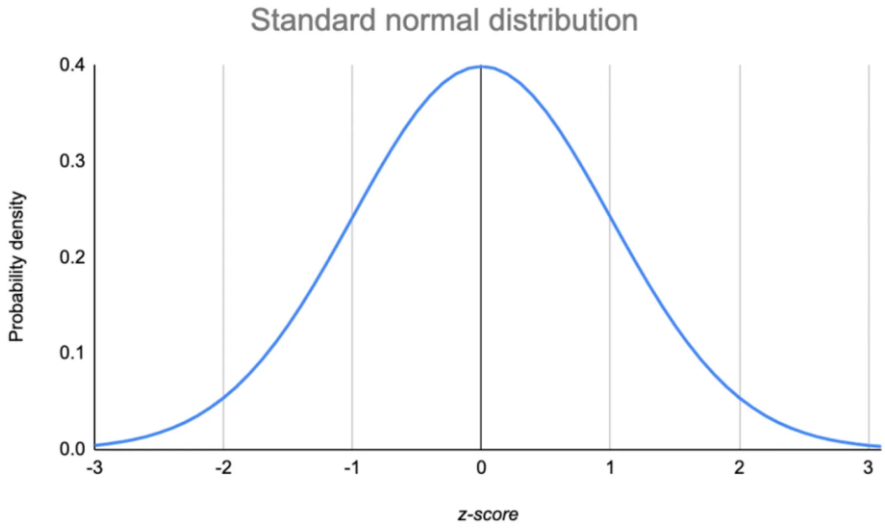
Represented by smooth curves, such as the bell curve of the normal distribution.

Examples: normal distribution, exponential distribution, beta distribution.

Normal Distribution

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean.

The normal distribution appears as a "bell curve" in graphical form.



3.8 Central Limit Theorem

The **central limit theorem,** which is a statistical theory, states that when a large sample size has a finite variance, the samples will be normally distributed, and the mean of samples will be approximately equal to the mean of the whole population.

In other words, the central limit theorem states that for any population with mean and standard deviation, the distribution of the sample mean for sample size N has mean μ and standard deviation σ/√n.

Central limit theorem formula

Fortunately, you don’t need to actually repeatedly sample a population to know the shape of the sampling distribution. The parameters of the sampling distribution of the mean are determined by the parameters of the population:

The standard deviation of the sampling distribution is the standard deviation of the population divided by the square root of the sample size.

$$\mu_{\overline{x}} = \mu$$

$$\sigma_{\overline{x}} = \frac{\sigma}{\sqrt{n}}$$

Where,

N is the normal distribution

μ is the mean of the population

σ is the standard deviation of the population

n is the sample size

3.9 Monte Carlo Approximation

A Monte Carlo simulation is used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables. It is a technique used to understand the impact of risk and uncertainty.

A Monte Carlo simulation is used to tackle a range of problems in many fields including investing, business, physics, and engineering. It is also referred to as a multiple probability simulation.

A Monte Carlo simulation requires assigning multiple values to an uncertain variable to achieve multiple results and then averaging the results to obtain an estimate.

Monte Carlo simulations assume perfectly efficient markets.

History of the Monte Carlo Simulation

The Monte Carlo simulation was named after the gambling destination in Monaco because chance and random outcomes are central to this modeling technique, as they are to games like roulette, dice, and slot machines.

The technique was initially developed by Stanislaw Ulam, a mathematician who worked on the Manhattan Project, the secret effort to create the first atomic weapon. He shared his idea with John Von Neumann, a colleague at the Manhattan Project, and the two collaborated to refine the Monte Carlo simulation.

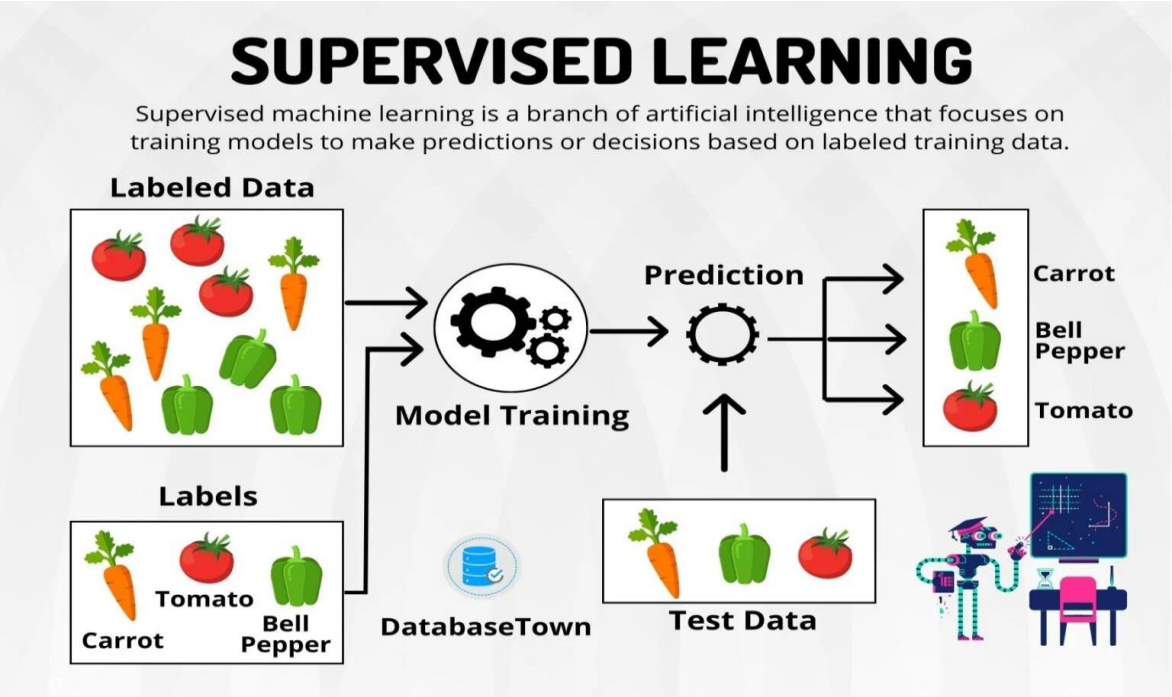
UNIT-4

Classification & Regression Algorithms

4.1 Supervised learning

- Supervised learning is the type of machine learning in which machines are trained using well "labeled" training data, and based on that data, machines predict the output. The labeled data means some input data is already tagged with the correct output.
- Examples of supervised learning are as follows:

  How does supervised learning work?



4.2 Types of Supervised Machine Learning Algorithms

- There are two types of supervised learning algorithms:
  1. Classification
  2. Regression

4.2.1 Classification

- Classification algorithms are used when the output variable is categorical, which means there are two classes Yes-No, Male-Female, True-false, etc.

- For example:

  A bank may have a customer dataset containing credit history, loans, investment details, etc. and they may want to know if any customer will default. In the historical data, we will have Features and Targets.

  - Features will be attributes of a customer such as credit history, loans, investments, etc.
  - Target will represent whether a particular customer has defaulted in the past (normally represented by 1 or 0 / True or False / Yes or No.)

- Classification algorithms are used for predicting discrete outcomes, if the outcome can take two possible values such as True or False, Default or No, or Yes or No, it is known as Binary Classification.
- When the outcome contains more than two possible values, it is known as Multiclass Classification.

  Many machine learning algorithms can be used for classification tasks. Some of them are:

  - Logistic Regression
  - Decision Tree Classifier
  - K Nearest Neighbour Classifier
  - Random Forest Classifier
  - Neural Networks

- **Binary Classifier:** If the classification problem has only two possible outcomes, it is called a Binary Classifier.

  Example: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.

- **Multi-class Classifier:** If a classification problem has more than two outcomes, then it is called a Multi-class Classifier.

  Example: Classifications of types of crops, Classification of types of music.

- Classification Algorithms can be further divided into the Mainly two main categories:
  - **Linear Models**
    - Logistic Regression
    - Support Vector Machines
  - **Non-linear Models**
    - K-Nearest Neighbours
    - Naïve Bayes
    - Decision Tree Classification
    - Random Forest Classification

4.2.2 Learner in Classification

- **Lazy Learners:** Lazy Learner first stores the training dataset and waits until it receives the test dataset.

  In the Lazy learner case, classification is done based on the most related data stored in the training dataset.

  It takes less time in training but more time for predictions. **Example:** K-NN algorithm, Case-based reasoning
- **Eager Learners:** Eager Learners develop a classification model based on a training dataset before receiving a test dataset.

  Opposite to Lazy learners, Eager Learner takes more time in learning and less time in prediction.

  Example: Decision Trees, Naïve Bayes, ANN.

4.2.3 Regression

- Regression algorithms are used if there is a relationship between the input variable and the output variable.
- It is used for the prediction of continuous variables, such as Weather forecasting, Market Trends, etc.

Types of Regression in Supervised Learning.

- **Linear Regression:** Here, we have only one independent variable to predict the output, i.e., the dependent variable.
- **Multiple Regression:** Here, we have more than one independent variable to predict the output, i.e., the dependent variable.
- **Polynomial Regression:** The graph between the dependent and independent variables follows a polynomial function.

Regression	Classification
Regression is the task of predicting a continuous quantity.	Classification is the task of predicting a discrete class label.
Regression Means to predict the output value using training data.	Classification means to group the output into a class.
A regression problem requires the prediction of a quantity.	In a classification problem data is labelled into one of two or more class.
If it is a real number or continuous then it is regression problem.	If it is discrete or categorical variable, then it is classification problem.
A regression problem with multiple input variables is called a multivariable regression problem.	A classification problem with two classes is called binary, more than 2 classes is called as multi-classification problem.
Example : Predict the house prices.	**Example** : Is that E-mail is Spam or not a Spam.

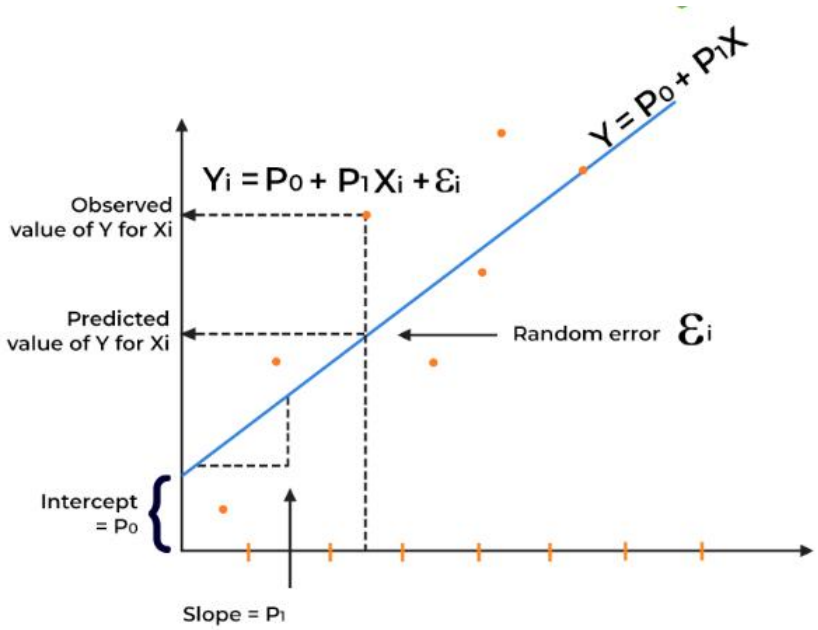
4.3. Supervised Machine Learning Algorithms:

- Linear Regression
- Support Vector Machines
- Decision Trees
- Random Forest
- Logistic Regression
- K-NN
- Naïve Bayes.

4.3.1 Linear Regression Algorithm

- Linear regression is a statistical method that predicts the relationship between two variables and is used for predictive analysis
- Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

  Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression.



- Mathematical equation for linear regression:

- $y = p0 + p1x + \varepsilon$ is the formula used for simple linear regression.

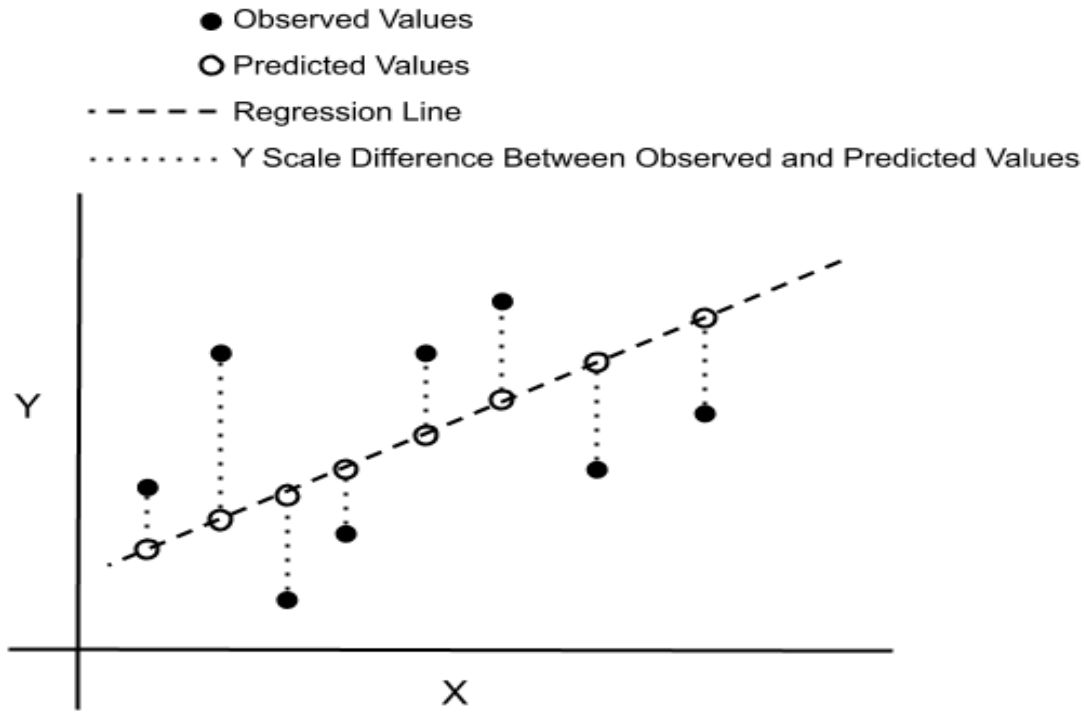
  y is the predicted value of the dependent variable (y) for any given value of the independent variable (x).

  p0 is the intercept, the predicted value of y when the x is 0.

  p1 is the regression coefficient – how much we expect y to change as x increases.

  x is the independent variable (the variable we expect is influencing y).

  ε (Random error) is the error of the estimate, or how much variation there is in our regression coefficient estimate.



- **Assumption for Linear Regression Model**

  Linear regression is a powerful tool for understanding and predicting the behavior of a variable, however, it needs to meet a few conditions to be accurate and dependable solutions.

- **Linearity**: The independent and dependent variables have a linear relationship with one another. This implies that changes in the dependent variable follow those in the independent variable(s) linearly.
- **Independence**: The observations in the dataset are independent of each other. This means that the value of the dependent variable for one observation does not depend on the value of the dependent variable for another observation.
- **Homoscedasticity**: Across all levels of the independent variable(s), the variance of the errors is constant. This indicates that the amount of the independent variable(s) has no impact on the variance of the errors.
- **Normality**: The errors in the model are normally distributed.
- **No multicollinearity**: There is no high correlation between the independent variables. This indicates that there is little or no correlation between the independent variables.

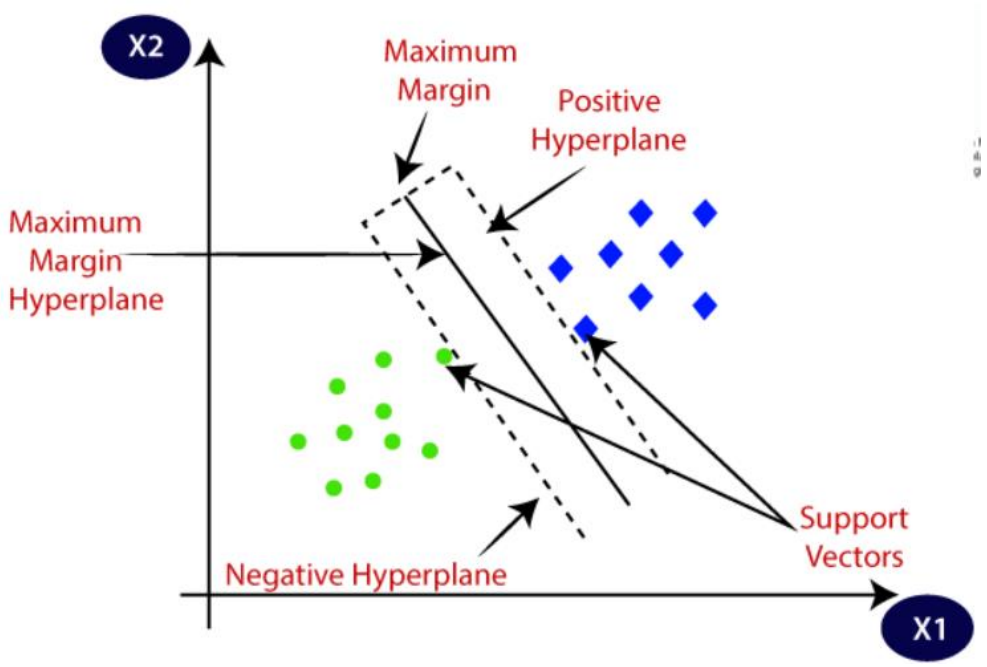
4.3.2 Support Vector Machines Algorithm

- Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. it is mostly used in classification problems.
- The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

Support Vector Machine Terminology

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. $wx+b = 0$.

The **hyperplane** with the maximum margin is called the **optimal hyperplane**.

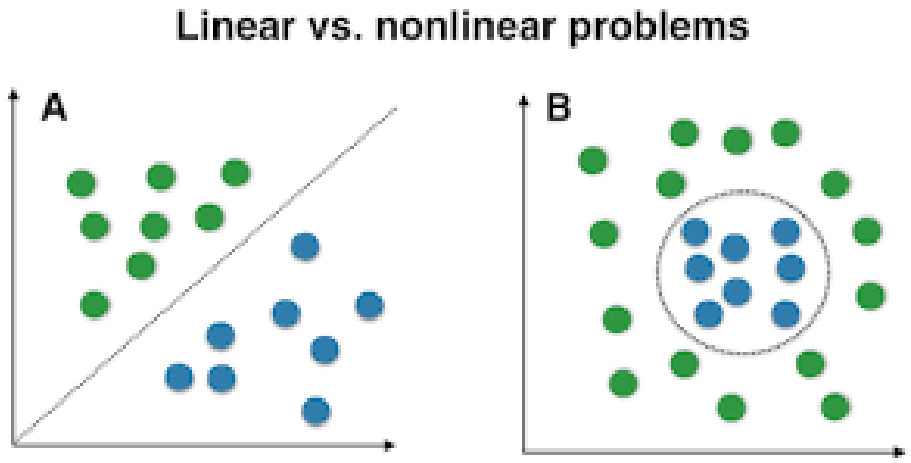


- **Support Vectors:** Support vectors are the closest data points to the hyperplane, which plays a critical role in deciding the hyperplane and margin.
- **Margin**: Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize 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.

SVM can be of two types:

- **Linear SVM**: Linear SVM is used for linearly separable data, which means 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 Linear SVM classifier.
- **Non-linear SVM**: Non-linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data, and the classifier used is called a Non-linear SVM classifier



- **Strengths of SVM**
  - SVM can be used for both classification and regression.
  - It is robust, i.e. not much impacted by data with noise or outliers.
  - The prediction results using this model are very promising.
- **Weaknesses of SVM**
  - SVM is applicable only for binary classification, i.e. when there are only two classes in the problem domain.

- The SVM model is very complex – almost like a black box when it deals with a high-dimensional data set. Hence, it is very difficult and close to impossible to understand the model in such cases.
- It is slow for a large dataset, i.e. a data set with either a large number of features or a large number of instances.

4.3.3 K-Nearest Neighbours Algorithm

K-nearest neighbors (K-NN) algorithm is a type of supervised ML algorithm that can be used for both classification as well as regression.

How does K-NN work?

The K-NN working can be explained based on the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step 2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step 3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, 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 neighbors is maximum.
- **Step 6:** Our model is ready.

Why the K-NN algorithm is called a lazy learner?

Eager learners follow the general steps of machine learning, i.e. perform an abstraction of the information obtained from the input data and then follow it through by a generalization step. However, as we have seen in the case of the K-NN algorithm, these steps are completely skipped. It stores the training data and directly applies the philosophy of nearest neighborhood finding to arrive at the classification. So, for K-NN, there is no learning happening in the real sense. Therefore, K-NN falls under the category of lazy learner.

- **Strengths of the k-NN algorithm**
  - Extremely simple algorithm – easy to understand

- Very effective in certain situations, eg. for recommender system design
- Very fast or almost no time is required for the training phase

- **Weaknesses of the k-NN algorithm**

  Classification is done completely based on the training data. So, it has a heavy reliance on the training data. If the training data does not represent the problem domain comprehensively, the algorithm fails to make an effective classification. The classification process is very slow. Also, a large amount of computational space is required to load the training data for classification.

4.3.4 Naïve Bayes Algorithm

- The Naïve Bayes algorithm is a supervised learning algorithm, which is based on the **Bayes theorem** and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- **It is a probabilistic classifier, which means it predicts based on the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentiment analysis, and classifying articles**.
- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified based on color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identifying that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Example: Customer Purchase Prediction

Table with data on the past purchases of customers in a store:

No.	Age	Student	Income	Credit	Buys
1	Young	Yes	High	Fair	No
2	Senior	No	High	Excellent	Yes
3	Middle	Yes	Medium	Fair	Yes
4	Young	Yes	Low	Fair	No
5	Middle	Yes	Low	Excellent	Yes
6	Senior	No	Medium	Excellent	No
7	Young	No	Medium	Excellent	Yes
8	Young	Yes	Medium	Fair	Yes
9	Middle	Yes	High	Excellent	Yes
10	Senior	No	Low	Fair	No

- The training set

  Each row in the table contains the age of the customer, whether they are a student or not, their level of income, their credit rating, and whether or not they have purchased the product.

  A new customer with the following properties arrives at the store: <Age = Young, Student = Yes, Income = Low, Credit = Excellent>

  You need to predict whether this customer will buy the product or not.

  We first compute the class prior probabilities by counting the number of rows that have Buys = Yes (6 out of 10) and the number of rows that have Buys = No (4 out of 10):

$$P(\text{Buys} = \text{Yes}) = 6/10 = 0.6$$
$$P(\text{Buys} = \text{No}) = 4/10 = 0.4$$

$$P(\text{Age} = \text{Young}|\text{Buys} = \text{Yes}) = 2/6 = 0.333$$
$$P(\text{Age} = \text{Young}|\text{Buys} = \text{No}) = 2/4 = 0.5$$

$$P(\text{Student} = \text{Yes}|\text{Buys} = \text{Yes}) = 4/6 = 0.667$$
$$P(\text{Student} = \text{Yes}|\text{Buys} = \text{No}) = 2/4 = 0.5$$

$$P(\text{Income} = \text{Low}|\text{Buys} = \text{Yes}) = 1/6 = 0.167$$
$$P(\text{Income} = \text{Low}|\text{Buys} = \text{No}) = 2/4 = 0.5$$

$$P(\text{Credit} = \text{Excellent}|\text{Buys} = \text{Yes}) = 4/6 = 0.667$$
$$P(\text{Credit} = \text{Excellent}|\text{Buys} = \text{No}) = 1/4 = 0.25$$

Then, we compute the likelihood of the features in each class:

Therefore, the class posterior probabilities are:

α is the normalization factor ($\alpha = 1 / P(\mathbf{x})$).

Since $P(\text{Buys} = \text{Yes}|\mathbf{x}) > P(\text{Buys} = \text{No}|\mathbf{x})$, we predict that the customer will buy the product.

If we want to get the actual probability that the customer will buy the product, we can first find the normalization factor using the fact that the two posterior probabilities must sum to 1:

$$0.0148\alpha + 0.0125\alpha = 1 \Rightarrow \alpha = \frac{1}{0.0148 + 0.0125} = 36.63$$

Then, we can plug it in the posterior probability for Buy = Yes:

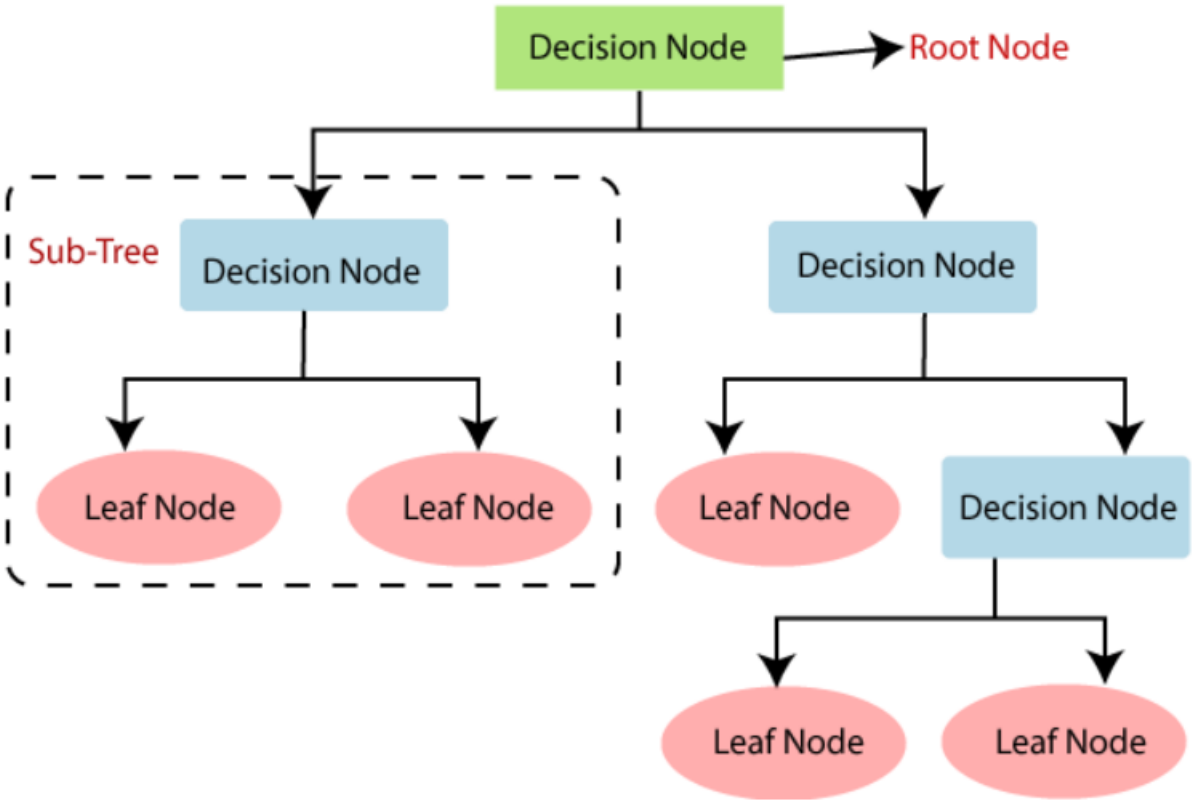
$$\begin{aligned} P(\text{Buys} = \text{Yes}|\mathbf{x}) &= P(\text{Yes}) \cdot P(\text{Young}|\text{Yes}) \cdot P(\text{Student}|\text{Yes}) \cdot P(\text{Low}|\text{Yes}) \cdot P(\text{Excellent}|\text{Yes}) \\ &= 0.6 \cdot 0.333 \cdot 0.667 \cdot 0.167 \cdot 0.667\alpha = 0.0148\alpha \end{aligned}$$

$$\begin{aligned} P(\text{Buys} = \text{No}|\mathbf{x}) &= P(\text{No}) \cdot P(\text{Young}|\text{No}) \cdot P(\text{Student}|\text{No}) \cdot P(\text{Low}|\text{No}) \cdot P(\text{Excellent}|\text{No}) \\ &= 0.4 \cdot 0.5 \cdot 0.5 \cdot 0.5 \cdot 0.25\alpha = 0.0125\alpha \end{aligned}$$

The probability that the customer will buy the product is 54.21%.

4.3.5 Decision Trees Algorithm

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the tests are performed based on features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- To build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further splits the tree into subtrees.



Decision Tree Terminologies

- **Root Node:** The root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.
- **Pruning:** Pruning is the process of removing unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

- **Attribute Selection Measures**

  While implementing a Decision tree, the main issue arises as to how to select the best attribute for the root node and sub-nodes. So, to solve such problems there is a technique which is called an **Attribute selection measure or ASM.** By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

  - **Information Gain**
  - **Gini Index**
  - **Information Gain:**
  - Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
  - It calculates how much information a feature provides us about a class.
  - According to the value of information gain, we split the node and built the decision tree.
  - A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:
    Information Gain= Entropy(S)- [(Weighted Avg) *Entropy (each feature)]

  - **Entropy:** Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:
    Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

    Where,

    - S= Total number of samples
    - P(yes)= probability of yes
    - P(no)= probability of no

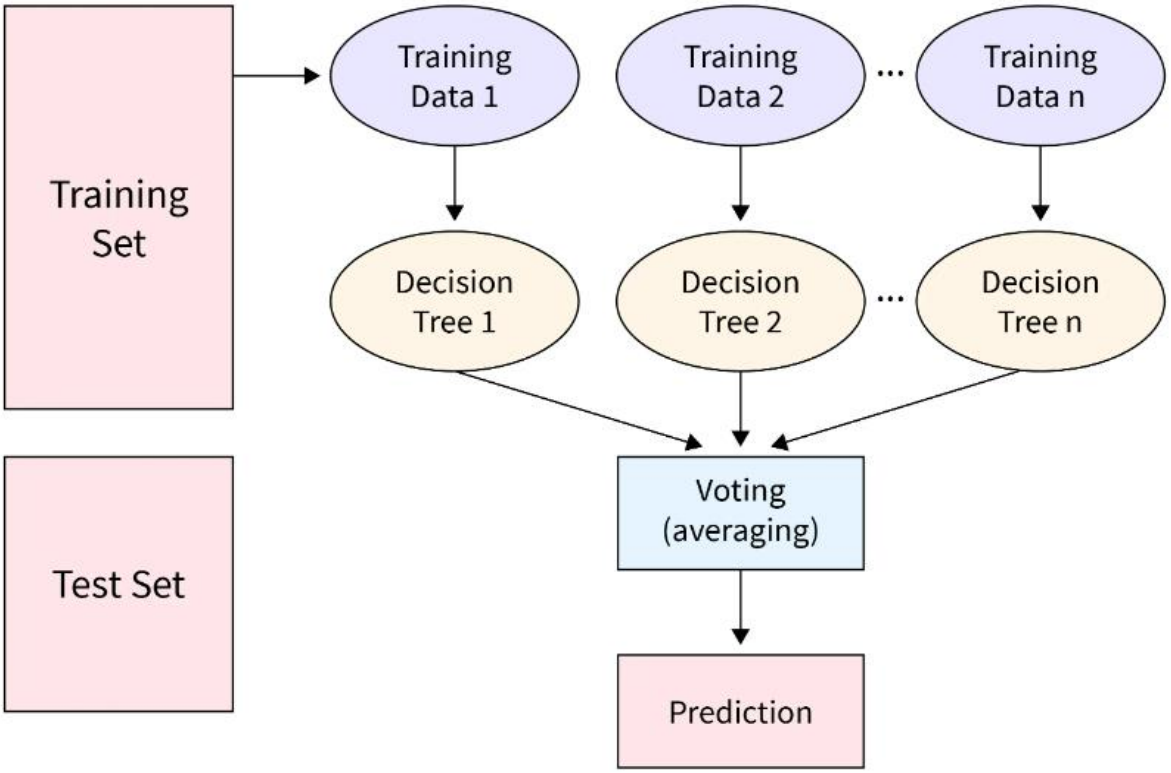
- **Gini Index:**
  - The Gini index is a measure of impurity or purity used while creating a decision tree in the CART (Classification and Regression Tree) algorithm.
  - An attribute with the low Gini index should be preferred as compared to the high Gini index.
  - It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
  - The Gini index can be calculated using the below formula:

$$\text{Gini Index} = 1 - \sum_j P_j^2$$

- **Strengths of decision tree**
  - It produces very simple understandable rules. For smaller trees, not much Mathematical and computational knowledge is required to understand this Model.
  - Works well for most of the problems.
  - It can handle both numerical and categorical variables.
  - It can work well both with small and large training data sets.
  - Decision trees provide a definite clue of which features are more useful for classification.
- **Weaknesses of decision tree**
  - Decision tree models are often biased towards features having more number of possible values, i.e. levels.
  - This model gets over-fitted or under-fitted quite easily.
  - Decision trees are prone to errors in classification problems with many classes and a relatively small number of training examples.
  - A decision tree can be computationally expensive to train.
  - Large trees are complex to understand.

4.3.6 Random Forest Algorithm

- Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting



- **Use of Random Forest**

  Below are some points that explain why we should use the Random Forest algorithm:

  - It takes less training time as compared to other algorithms.
  - It predicts output with high accuracy, even for a large dataset it runs efficiently.
  - It can also maintain accuracy when a large proportion of data is missing

- **Working of Random Forest Algorithm**

  Random Forest works in two phases. first is to create the random forest by combining N decision trees, and the second is to make predictions for each tree created in the first phase.

  The Working process can be explained in the below steps and diagram:

  Step 1: Select random K data points from the training set.

  Step 2: Build the decision trees associated with the selected data points (Subsets).

  Step 3: Choose the number N for decision trees that you want to build.

  Step 4: Repeat Steps 1 & 2.

  Step 5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

- **Strengths of random forest**
  - It runs efficiently on large and expansive data sets.
  - It has a robust method for estimating missing data and maintains precision when a large proportion of the data is absent.
  - It has powerful techniques for balancing errors in a class population of unbalanced data sets.
  - It gives estimates (or assessments) about which features are the most important ones in the overall classification.
  - It generates an internal unbiased estimate (gauge) of the generalization error as the forest generation progresses.
  - Generated forests can be saved for future use on other data.
  - Lastly, the random forest algorithm can be used to solve both classification and regression problems.
- **Weaknesses of random forest**
  - This model, because it combines several decision tree models, is not as easy to understand as a decision tree model.
  - It is computationally much more expensive than a simple model like a decision tree.

- **Advantages of Supervised learning:**
  - With the help of supervised learning, the model can predict the output based on prior experiences.
  - It performs classification and regression tasks.
  - In supervised learning, we can have an exact idea about the classes of objects.
  - The supervised learning model helps us to solve various real-world problems such as fraud detection, spam filtering, etc.
- **Disadvantages of supervised learning:**
  - Supervised learning models are not suitable for handling complex tasks.
  - Supervised learning cannot predict the correct output if the test data is different from the training dataset.
  - Training requires lots of computation time.
  - In supervised learning, we need enough knowledge about the classes of objects.

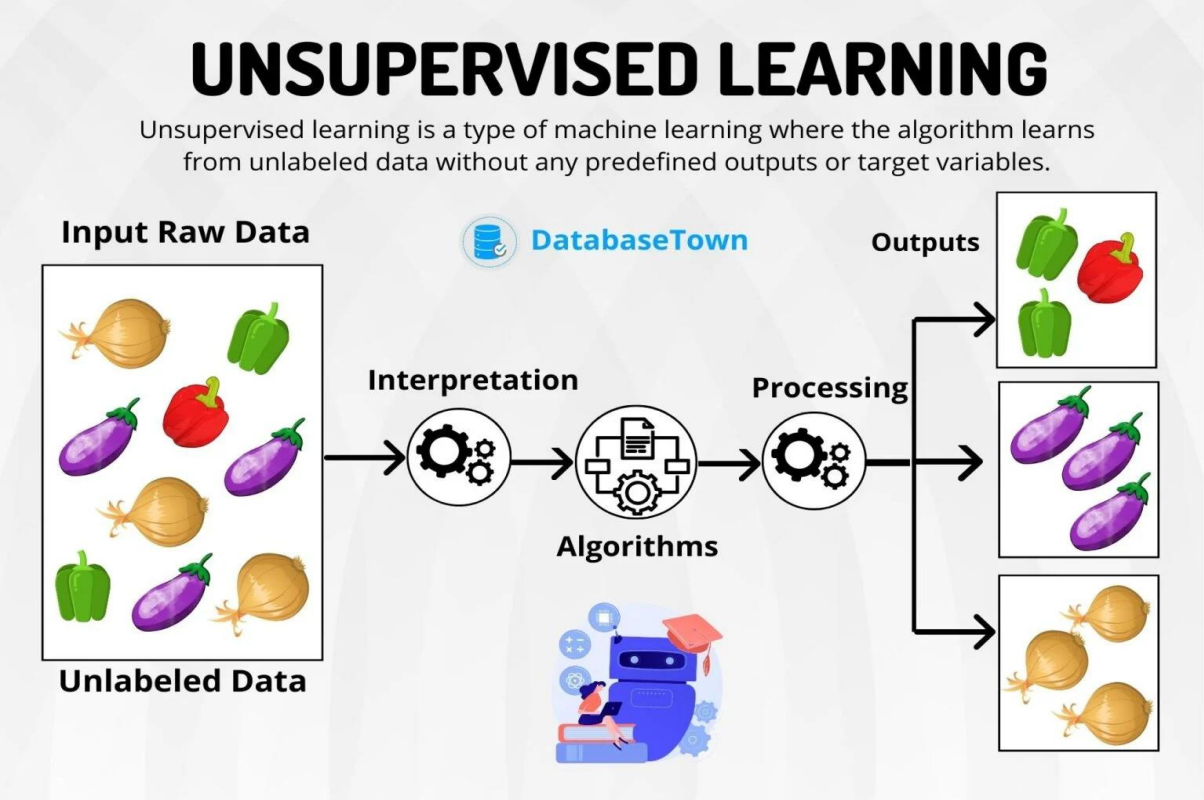
UNIT-5

Clustering and Association Rules

5.1 Unsupervised learning

- Unsupervised learning is a type of machine learning where the algorithm learns from unlabelled data without any predefined outputs or target variables.
- Unsupervised learning finds patterns, similarities, or groupings within the data to get insights and make data-driven decisions.
- It is particularly useful when dealing with large datasets.

5.2 Working of Unsupervised Learning



5.3 Distance Measure

- Distance measure determines the similarity between two elements and it influences the shape of the clusters.
- Some of the ways we can calculate distance measures include:
    - Euclidean distance measure
    - Manhattan distance measure
    - Cosine distance measure

5.3.1 Euclidean distance measure

- The Euclidean distance formula helps to find the distance of a line segment. Let us assume two points, such as (x_1, y_1) and (x_2, y_2) in the two-dimensional coordinate plane.
- Thus, the Euclidean distance formula is given by:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

- **Euclidean Distance Examples**

Find the distance between two points P(0, 4) and Q(6, 2).

Solution:

Given:

P(0, 4) = (x_1, y_1) , Q(6, 2) = (x_2, y_2)

The distance between the point PQ is

PQ = $\sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]}$

PQ = $\sqrt{[(6 - 0)^2 + (2 - 4)^2]}$

PQ = $\sqrt{[(6)^2 + (-2)^2]}$

PQ = $\sqrt{(36+4)}$

PQ = $\sqrt{40}$ and PQ = $2\sqrt{10}$.

Therefore, the distance between two points P(0,4) and Q(6, 2) is $2\sqrt{10}$.

5.3.2 Manhattan distance

- **Manhattan Distance**: The distance measured by calculating the sum of absolute differences of two points or vectors.
- Suppose we have to find the Manhattan distance between points A (x_1, y_1) and
- B (x_2, y_2). Then, it is given by Distance, $d = |x_1 - x_2| + |y_1 - y_2|$
- The Manhattan distance in a 2-dimensional space is given as:

$$d = |p_1 - q_1| + |p_2 - q_2|$$

- The generalized formula for an n-dimensional space is given as:

$$D_m = \sum_{i=1}^{n} |p_i - q_i|$$

- **Example:** Calculate the Manhattan distance between Point P1(4,4) and P2(9,9).

- **Solution:** The Manhattan distance between P_1 and P_2 is given by,

  Given: First point, $P_1 = (4, 4)$

  Second point, $P_2 = (9, 9)$

  $P_1P_2 = |x_1 - x_2| + |y_1 - y_2|$

  $P_1P_2 = |4 - 9| + |4 - 9|$

  $= |-5| + |-5|$

  $= 5 + 5$

  $= 10$ units

Hence, the Manhattan distance between point P_1 (4,4) and P_2 (9,9) is 10 units.

5.3.3 Cosine distance measure

- Cosine similarity is a metric that measures the cosine of the angle between two vectors projected in a multi-dimensional space.
- The cosine similarity is described mathematically as the division between the dot product of vectors and the product of the euclidean norms or magnitude of each vector.
- Cosine similarity measures how similar two vectors are, and Cosine distance measures how different they are. In real applications, it depends on the task and what function to choose. You can use cosine similarity as a loss function or as a measure for clustering

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

$$\left\|\vec{a}\right\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\left\|\vec{b}\right\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

Where, A and B are vectors in a multidimensional space.

Since the $cos(\theta)$ value is in the range $[-1,1]$:

- -1 value will indicate strongly opposite vectors i.e. no similarity
- 0 indicates independent (or orthogonal) vectors
- 1 indicates a high similarity between the vectors

- **Cosine Distance:** Usually, people use the cosine similarity as a similarity metric between vectors. Now, the cosine distance can be defined as follows:

Cosine Distance = 1 — Cosine Similarity

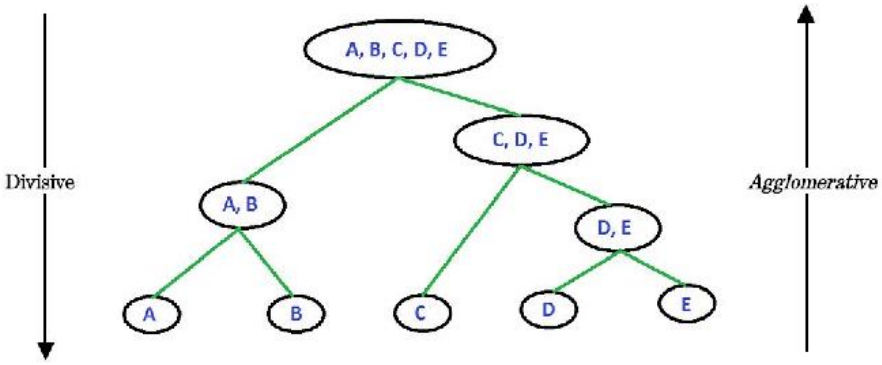
5.4. Types of Unsupervised Learning Algorithm

5.4.1 Clustering Algorithms

- Clustering is a type of unsupervised learning wherein data points are grouped into different sets based on their degree of similarity.
- The various types of clustering are:
  - Hierarchical clustering
  - Partitioning clustering
- Hierarchical clustering is further subdivided into:
  - Agglomerative clustering (bottom-up approach)
  - Divisive clustering (top-down approach)
- Partitioning clustering is further subdivided into:
  - K-Means clustering
  - Fuzzy C-Means clustering

5.4.2 Hierarchical clustering

- Hierarchical clustering is a method of grouping similar objects into clusters in a tree-like structure.
- **Agglomerative clustering:** is a bottom-up approach. We begin with each element as a separate cluster and merge them into successively more massive clusters.
- **Divisive clustering:** is a top-down approach. We begin with the whole set and proceed to divide it into successively smaller clusters.



5.5 Unsupervised Learning Algorithms:

5.5.1 K-means clustering

- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabelled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.
- It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm below shows the simple algorithm of K-means

Step 1: Select K points in the data space and mark them as initial centroids loop

Step 2: Assign each point in the data space to the nearest centroid to form K clusters

Step 3: Measure the distance of each point in the cluster from the centroid

Step 4: Calculate the Sum of Squared Error (SSE) to measure the quality of the clusters.

Step 5: Identify the new centroid of each cluster based on the distance between points

Step 6: Repeat Steps 2 to 5 to refine until the centroids do not change the end loop.

Choosing the Optimal Number of Clusters

The number of clusters that we choose for the algorithm shouldn't be random. Each and every cluster is formed by calculating and comparing the mean distances of each data point within a cluster from its centroid.

We can choose the right number of clusters with the help of the Within-Cluster-Sum-of-Squares (WCSS) method. WCSS stands for the sum of the squares of distances of the data points in each and every cluster from its centroid.

The main idea is to minimize the distance (e.g., euclidean distance) between the data points and the centroid of the clusters. The process is iterated until we reach a minimum value for the sum of distances.

Elbow Method

Here are the steps to follow in order to find the optimal number of clusters using the elbow method:

Step 1: Execute the K-means clustering on a given dataset for different K values (ranging from 1-10).

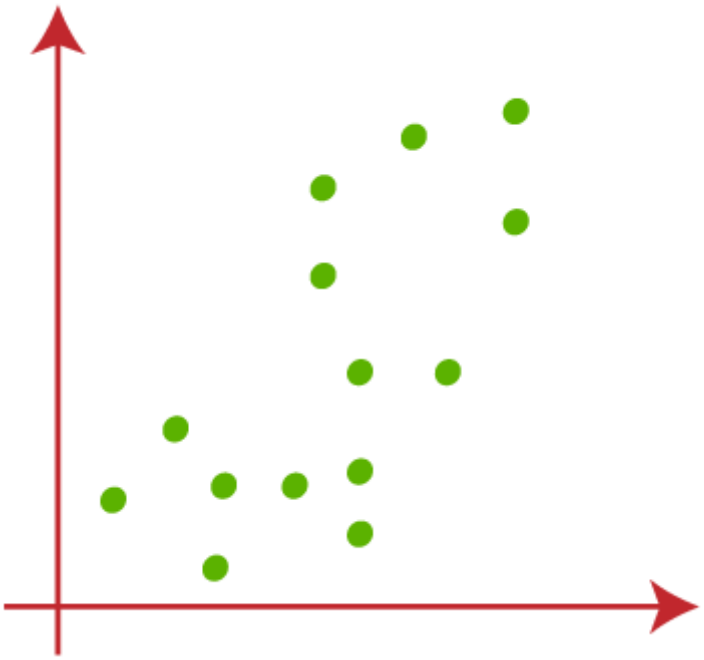
Step 2: For each value of K, calculate the WCSS value.

Step 3: Plot a graph/curve between WCSS values and the respective number of clusters K.

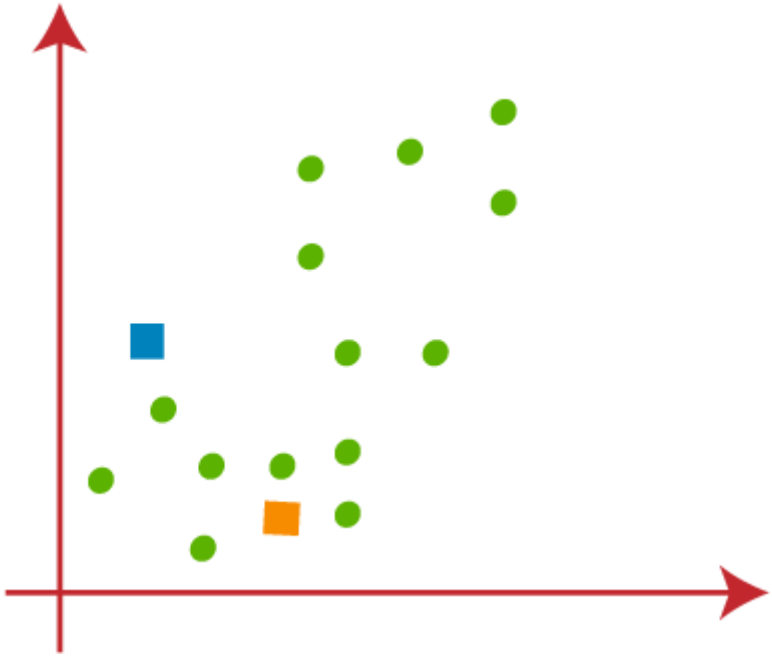
Step 4: The sharp point of bend or a point (looking like an elbow joint) of the plot, like an arm, will be considered as the best/optimal value of K.

Let's understand the above steps by considering the visual plots:

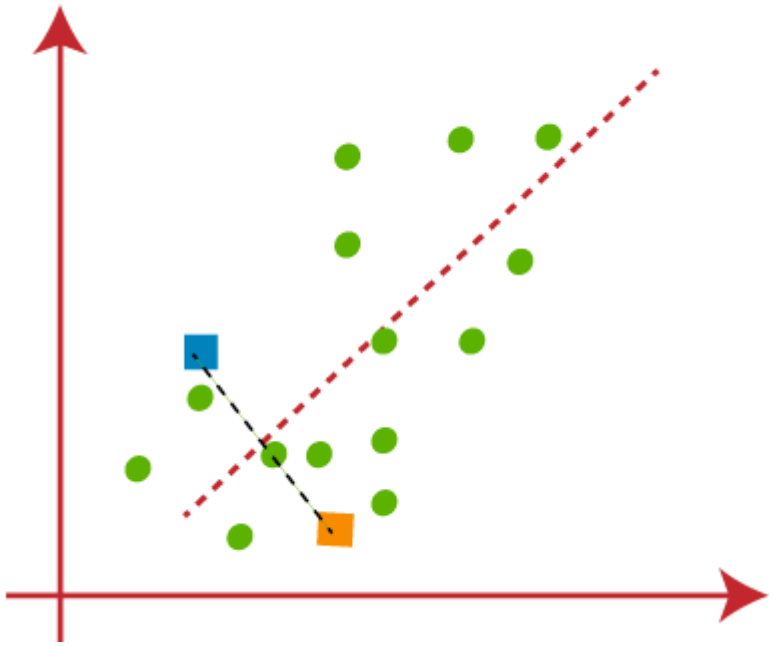
Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



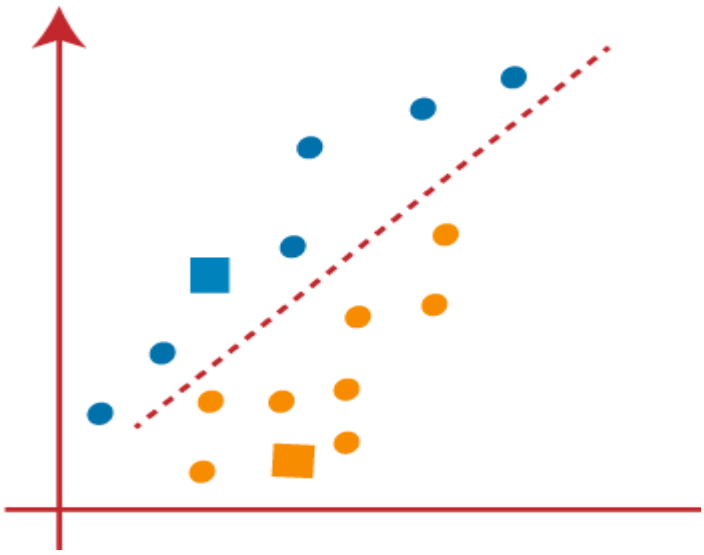
- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



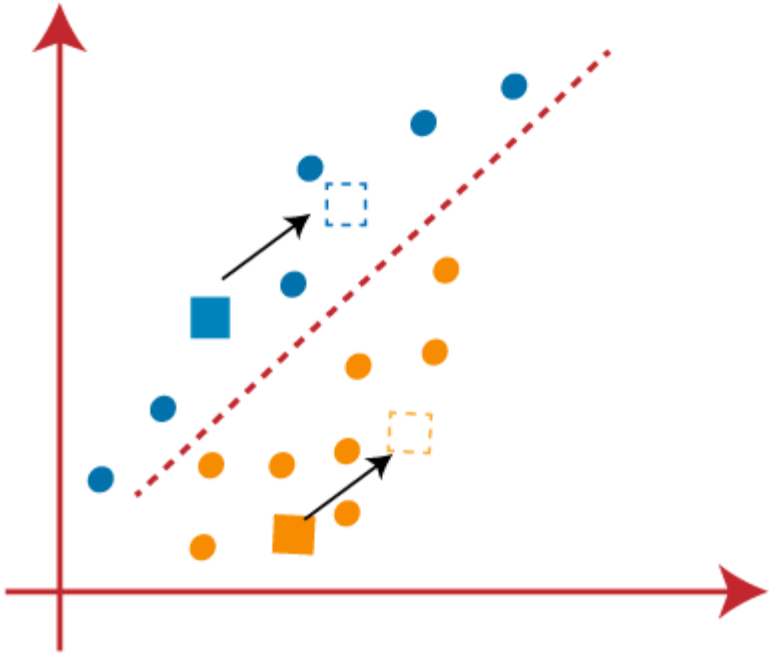
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



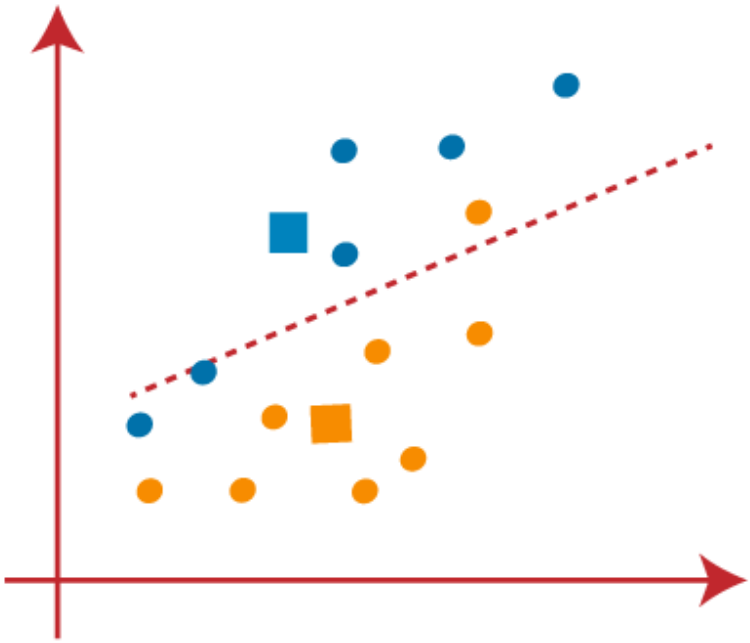
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's colour them as blue and yellow for clear visualization.



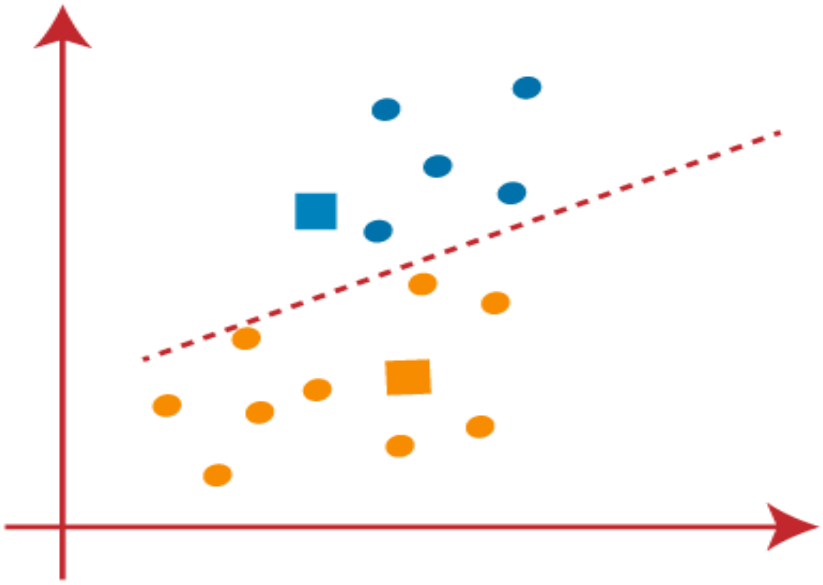
- As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids:



- Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like

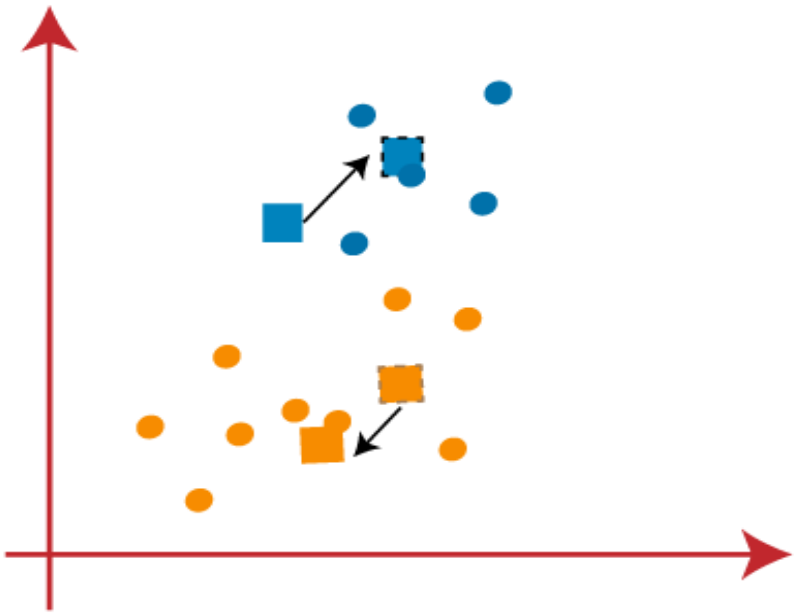


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

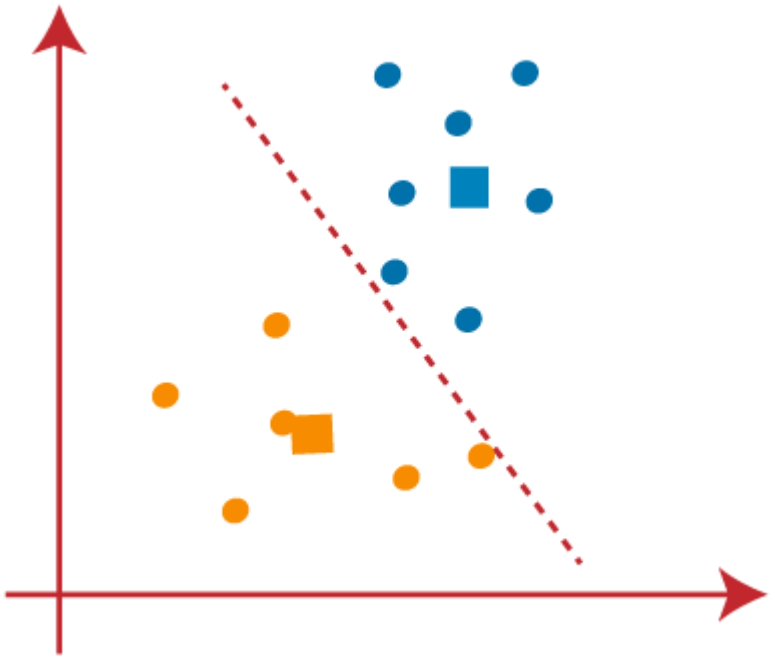


reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

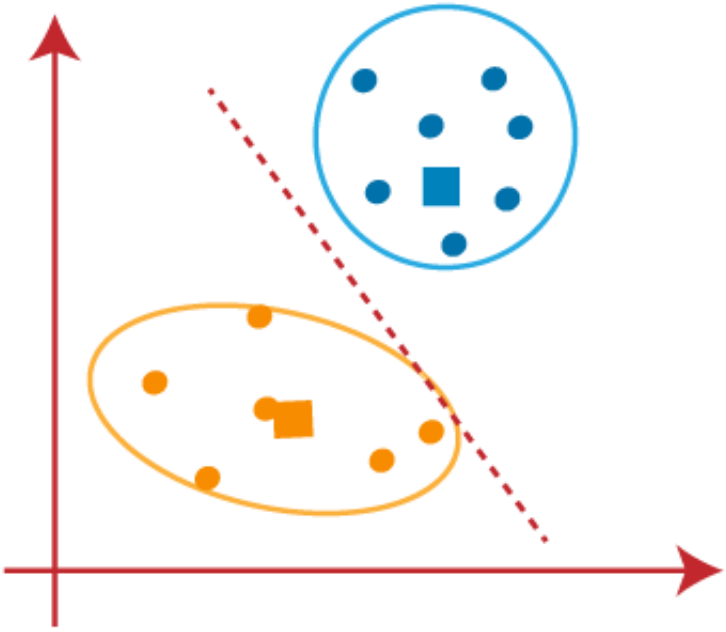
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



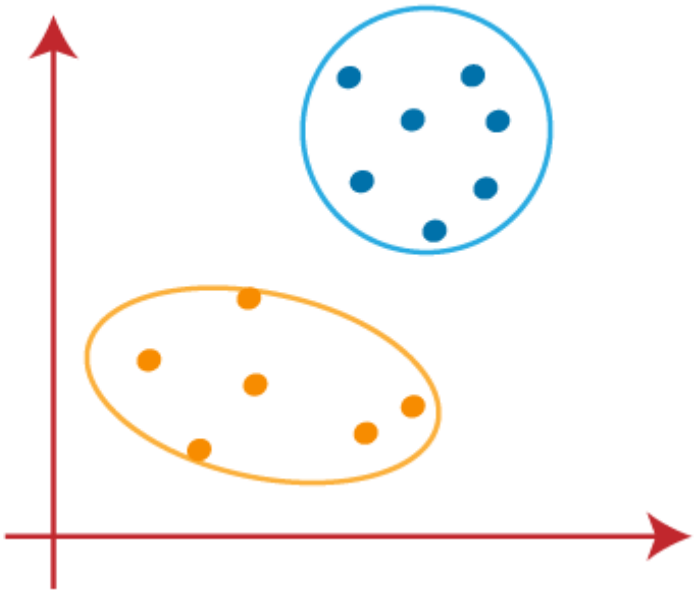
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



- We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



- **Challenges of Unsupervised Learning**

  While unsupervised learning has many benefits, some challenges can occur when it allows machine learning models to execute without any human intervention. Some of these challenges can include:

  - Computational complexity due to a high volume of training data
  - Longer training times
  - Higher risk of inaccurate results
  - Human intervention to validate output variables
  - Lack of transparency into the basis on which data was clustered

5.6. Advantages and Disadvantages Unsupervised Learning

- **Advantages Unsupervised Learning**

- **Use of Unlabelled Data**

  Unsupervised learning helps us to find hidden patterns or structures in data that don't have any labels. It gives us valuable insights and knowledge by uncovering meaningful connections and information that we may not have noticed before.

- **Scalability**

  Unsupervised learning algorithms handle large-scale datasets without manual labelling and make them more scalable than supervised learning in certain scenarios. Unsupervised learning algorithms handle large-scale datasets without manual labelling and make them more scalable than supervised learning in certain scenarios. Unsupervised learning algorithms handle large-scale datasets without manual labeling and make them more scalable than supervised learning in certain scenarios.

- **Anomaly Detection**

  Unsupervised learning can effectively detect anomalies or outliers in data, which is particularly useful for fraud detection, network security, or identifying rare events.

- **Data Preprocessing**

  Unsupervised learning techniques like dimensionality reduction can help preprocess data by reducing noise, removing irrelevant features, and improving efficiency in subsequent supervised learning tasks.

- **Disadvantages of Unsupervised Learning**

  Unsupervised learning has some limitations and challenges:

- **Lack of Ground Truth**

  Since unsupervised learning deals with unlabelled data, there is no definitive measure of correctness or accuracy. Evaluation and interpretation of results become subjective and rely heavily on domain expertise.

- **Interpretability**

  Unsupervised learning algorithms often provide clusters or patterns without explicit labels or explanations. Interpreting and understanding the meaning of these clusters can be challenging and subjective.

- **Overfitting and Model Selection**

  Unsupervised learning models are susceptible to overfitting and choosing the optimal model or parameters can be challenging due to the absence of a labeled validation set.

- **Limited Guidance**

  Unlike supervised learning, where the algorithm learns from explicit feedback, unsupervised learning lacks explicit guidance, which can result in the algorithm discovering irrelevant or noisy patterns.

5.7. Difference between Supervised and Unsupervised learning

Supervised Learning	Unsupervised Learning
Supervised learning algorithms are trained using labelled data.	Unsupervised learning algorithms are trained using unlabelled data.
The supervised learning model takes direct feedback to check if it is predicting the correct output or not.	The unsupervised learning model does not take any feedback.

The supervised learning model predicts the output.	The unsupervised learning model finds the hidden patterns in data.
In supervised learning, input data is provided to the model along with the output.	In unsupervised learning, only input data is provided to the model.
The goal of supervised learning is to train the model so that it can predict the output when it is given new data.	The goal of unsupervised learning is to find hidden patterns and useful insights from the unknown dataset.
Supervised learning needs supervision to train the model.	Unsupervised learning does not need any supervision to train the model.
Supervised learning can be categorized into **Classification** and **Regression** p roblems.	Unsupervised Learning can be classified in **Clustering** and A**ssociation** pr oblems.
Supervised learning can be used for those cases where we know the input as well as corresponding outputs.	Unsupervised learning can be used for those cases where we have only input data and no corresponding output data.
A supervised learning model produces an accurate result.	Unsupervised learning models may give less accurate results as compared to supervised learning.

It includes various algorithms such as Linear Regression, Logistic Regression, Support Vector Machine, Multi-class Classification, Decision tree, Bayesian Logic, etc.	It includes various algorithms such as Hierarchical Clustering, KNN, and Apriori algorithm.

5.8 Association rule

- The Association rule is a learning technique that helps identify the dependencies between two data items. Based on the dependency, it then maps accordingly so that it can be more profitable. The association rule furthermore looks for interesting associations among the variables of the dataset. It is undoubtedly one of the most important concepts of Machine Learning and has been used in different cases such as association in data mining and continuous production, among others. However, like all other techniques, association in data mining, too, has its own set of disadvantages.
- An Association rule has 2 parts:
    - an antecedent (if) and
    - a consequent (then)
- An antecedent is something that's found in data, and a consequent is an item that is found in combination with the antecedent. Have a look at this rule for instance:

    "If a customer buys bread, he's 70% likely to buy milk."

    In the above association rule, bread is the antecedent and milk is the consequent. Simply put, it can be understood as a retail store's association rule to target their customers better. If the above rule is a result of a

thorough analysis of some data sets, it can be used to not only improve customer service but also improve the company's revenue.

- Association rules are created by thoroughly analyzing data and looking for frequent if/then patterns. Then, depending on the following two parameters, the important relationships are observed:
    - **Support:** Support indicates how frequently the if/then relationship appears in the database.
    - **Confidence:** Confidence tells about the number of times these relationships are true.

5.8.1 Types of Association Rules

There are typically four different types of association rules in data mining. They are

- Multi-relational association rules
- Generalized Association rule
- Quantitative Association Rules

- **Multi-Relational Association Rule**

  Also known as MRAR, the multi-relational association rule is defined as a new class of association rules that are usually derived from different or multi-relational databases. Each rule under this class has one entity with different relationships that represent the indirect relationships between entities.

- **Generalized Association Rule**
    - Moving on to the next type of association rule, the generalized association rule is largely used for getting a rough idea about the interesting patterns that often tend to stay hidden in data.

- **Quantitative Association Rules**
  - This particular type is one of the most unique kinds of all the four association rules available. What sets it apart from the others is the presence of numeric attributes in at least one attribute of quantitative association rules. This is in contrast to the generalized association rule, where the left and right sides consist of categorical attributes.

- **Algorithms Of Associate Rule**

  Apriori Algorithm
  - The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties, as we shall see later.
  - **Support**

    The rule A→B holds in the transaction set D with supports, where s is the percentage of transactions in D that contain A U B (i.e. the union of sets A and B say, or, both A and B).

    Support (A=>B) = P(A ∪ B).
  - **Confidence**

    The rule A→B has confidence c in the transaction set D, where c is the percentage of transactions in D containing A that also contains B.

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

Example Let's look at a concrete example, based on the All Electronics transaction database, D, of the Table below.

Consider the following dataset and we will find frequent itemsets and generate association rules for them.

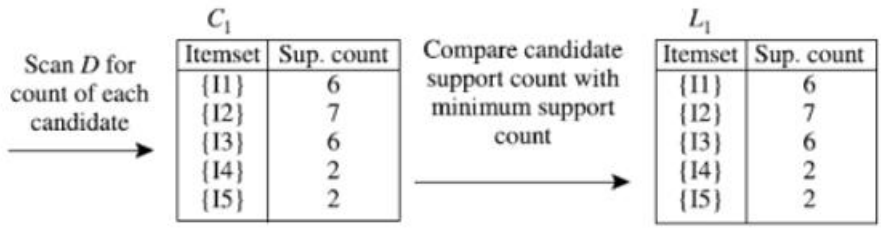
minimum support count is 2

minimum confidence is 60%

Transactional Data for an *AllElectronics* Branch

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Create a table containing support count of each item present in dataset – Called **C1(candidate set)**

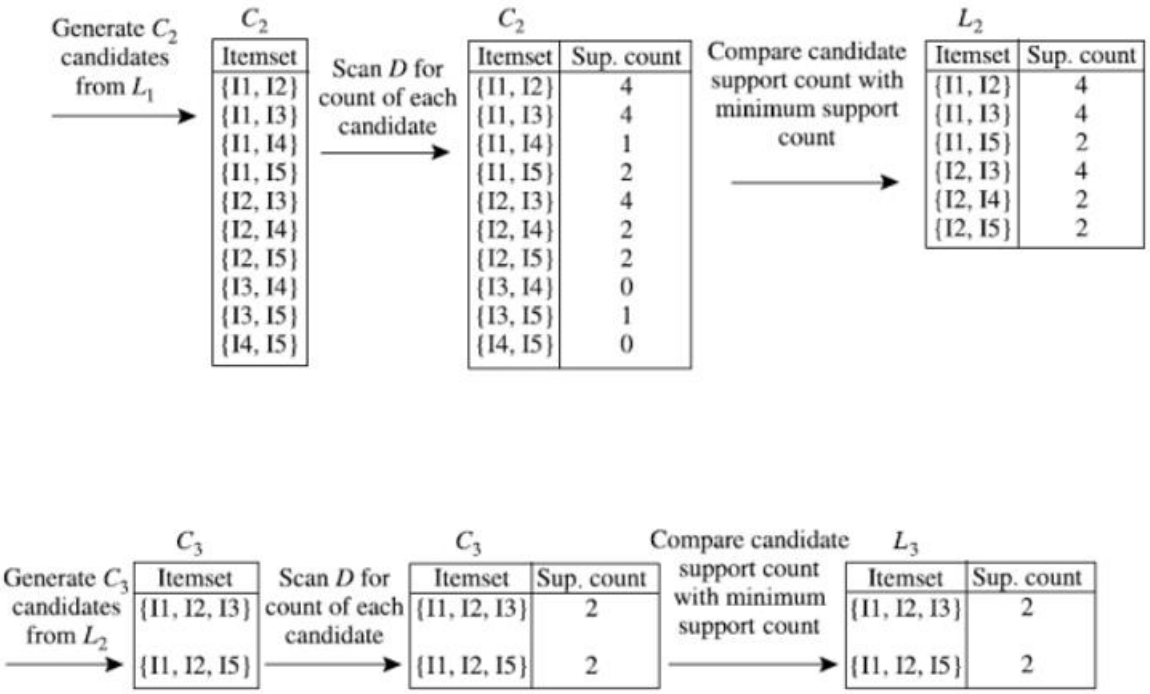


Step-2: K=2

Generate candidate set C2 using L1 (this is called join step). Condition of joining Lk-1 and Lk-1 is that it should have (K-2) elements in common.

Check all subsets of an itemset are frequent or not and if not frequent remove that itemset.(Example subset of{I1, I2} are {I1}, {I2} they are frequent.Check for each itemset)

Now find support count of these itemsets by searching in dataset.



Generation of the candidate itemsets and frequent itemsets, where the minimum support count is 2.

Step-3:

Step 2: Generating Association Rules

$\{I1, I2\} \Rightarrow I5$, $confidence = 2/4 = 50\%$
$\{I1, I5\} \Rightarrow I2$, $confidence = 2/2 = 100\%$
$\{I2, I5\} \Rightarrow I1$, $confidence = 2/2 = 100\%$
$I1 \Rightarrow \{I2, I5\}$, $confidence = 2/6 = 33\%$
$I2 \Rightarrow \{I1, I5\}$, $confidence = 2/7 = 29\%$
$I5 \Rightarrow \{I1, I2\}$, $confidence = 2/2 = 100\%$

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules are output, because these are the only ones generated that are strong. Note that, unlike conventional classification rules, association rules can contain more than one conjunct in the right side of the rule. ■