# UNIT III

# CLASSIFICATION

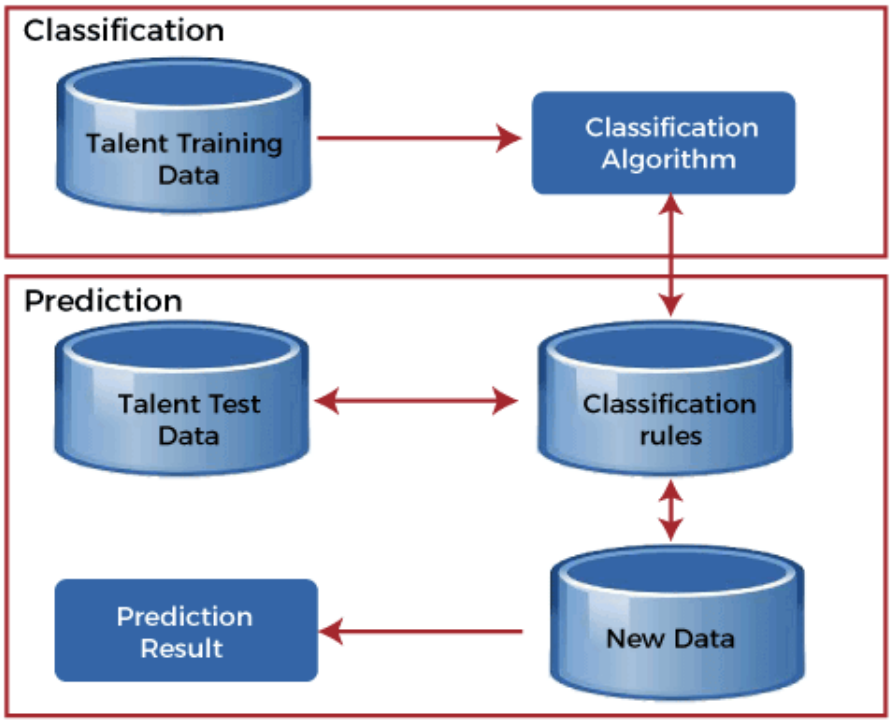
## CLASSIFICATION AND PREDICATION IN DATA MINING

There are two forms of data analysis that can be used to extract models describing important classes or predict future data trends. These two forms are as follows:

- Classification
- Prediction

We use classification and prediction to extract a model, representing the data classes to predict future data trends. Classification predicts the categorical labels of data with the prediction models. This analysis provides us with the best understanding of the data at a large scale.

Classification models predict categorical class labels, and prediction models predict continuous-valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.




Classification and Prediction Process

## WHAT IS CLASSIFICATION?

Classification is to identify the category or the class label of a new observation.

First, a set of data is used as training data.

The set of input data and the corresponding outputs are given to the algorithm.

So, the training data set includes the input data and their associated class labels.

Using the training dataset, the algorithm derives a model or the classifier.

The derived model can be a decision tree, mathematical formula, or a neural network.

In classification, when unlabeled data is given to the model, it should find the class to which it belongs.

The new data provided to the model is the test data set.

Classification is the process of classifying a record.

One simple example of classification is to check whether it is raining or not.

The answer can either be yes or no.

So, there is a particular number of choices. Sometimes there can be more than two classes to classify. That is called **multiclass classification**.

The bank needs to analyze whether giving a loan to a particular customer is risky or not.

**For example**, based on observable data for multiple loan borrowers, a classification model may be established that forecasts credit risk.

The data could track job records, homeownership or leasing, years of residency, number, type of deposits, historical credit ranking, etc.

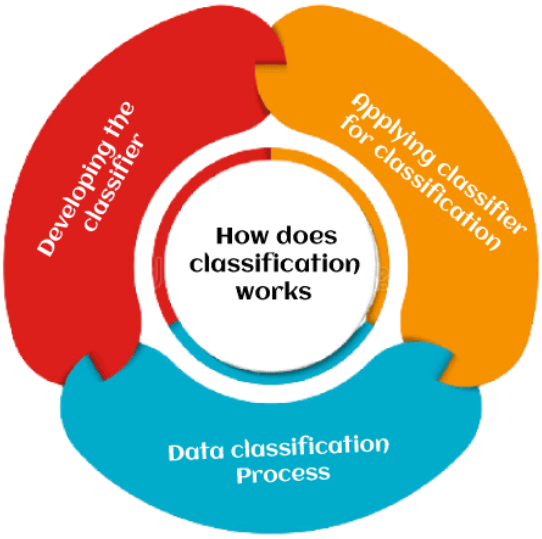
The goal would be credit ranking, the predictors would be the other characteristics, and the data would represent a case for each consumer.

In this example, a model is constructed to find the categorical label. The labels are risky or safe.

## HOW DOES CLASSIFICATION WORKS?

The functioning of classification with the assistance of the bank loan application has been mentioned above. **There are two stages in the data classification system:**

- classifier or model creation and
- classification classifier.




- **Developing the Classifier or model creation:** This level is the learning stage or the learning process. The classification algorithms construct the classifier in this stage. A classifier is constructed from a training set composed of the records of databases and their corresponding class names. Each category that makes up the training set is referred to as a category or class. We may also refer to these records as samples, objects, or data points.

- **Applying classifier for classification:** The classifier is used for classification at this level. The test data are used here to estimate the accuracy of the classification algorithm. If the consistency is deemed sufficient, the classification rules can be expanded to cover new data records. It includes:

    - **Sentiment Analysis:** Sentiment analysis is highly helpful in social media monitoring. We can use it to extract social media insights. We can build sentiment analysis models to read and analyze misspelled words with advanced machine learning algorithms. The accurate trained models provide consistently accurate outcomes and result in a fraction of the time.

    - **Document Classification:** We can use document classification to organize the documents into sections according to the content. Document classification refers to text classification; we can classify the words in the entire document. And with the help of machine learning classification algorithms, we can execute it automatically.

    - **Image Classification:** Image classification is used for the trained categories of an image. These could be the caption of the image, a statistical value, a theme. You can tag images to train your model for relevant categories by applying supervised learning algorithms.

    - **Machine Learning Classification:** It uses the statistically demonstrable algorithm rules to execute analytical tasks that would take humans hundreds of more hours to perform.

- **Data Classification Process:** The data classification process can be categorized into five steps:

    - Create the goals of data classification, strategy, workflows, and architecture of data classification.

    - Classify confidential details that we store.

- Using marks by data labelling.
- To improve protection and obedience, use effects.
- Data is complex, and a continuous method is a classification.

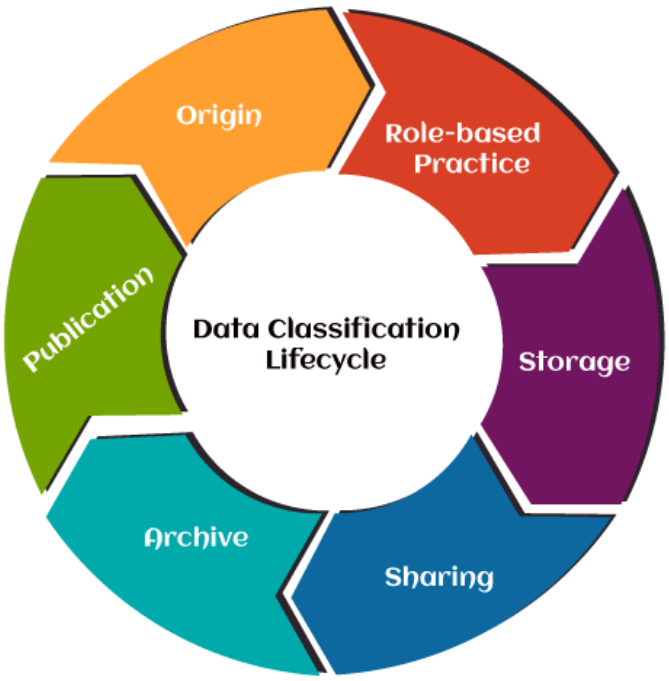
## WHAT IS DATA CLASSIFICATION LIFECYCLE?

The data classification life cycle produces an excellent structure for controlling the flow of data to an enterprise.

Businesses need to account for data security and compliance at each level.

With the help of data classification, we can perform it at every stage, from origin to deletion.

The data life-cycle has the following stages, such as:




- **Origin:** It produces sensitive data in various formats, with emails, Excel, Word, Google documents, social media, and websites.
- **Role-based practice:** Role-based security restrictions apply to all delicate data by tagging based on in-house protection policies and agreement rules.
- **Storage:** Here, we have the obtained data, including access controls and encryption.
- **Sharing:** Data is continually distributed among agents, consumers, and co-workers from various devices and platforms.
- **Archive:** Here, data is eventually archived within an industry's storage systems.

- **Publication:** Through the publication of data, it can reach customers. They can then view and download in the form of dashboards.

## WHAT IS PREDICTION?

Another process of data analysis is prediction. It is used to find a numerical output.

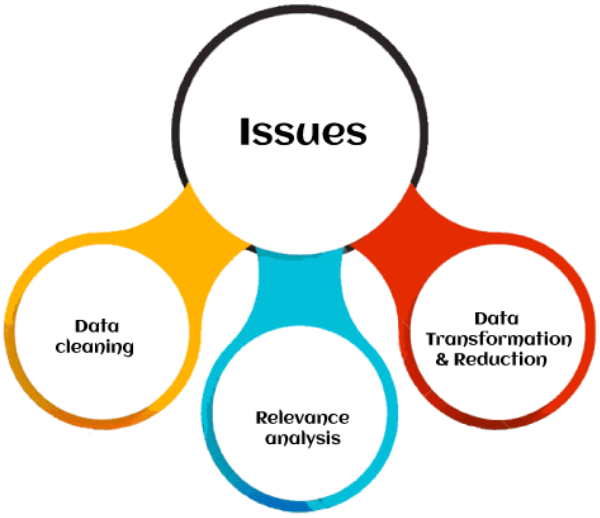
Same as in classification, the training dataset contains the inputs and corresponding numerical output values. The algorithm derives the model or a predictor according to the training dataset. The model should find a numerical output when the new data is given. Unlike in classification, this method does not have a class label. The model predicts a continuous-valued function or ordered value.

Regression is generally used for prediction. Predicting the value of a house depending on the facts such as the number of rooms, the total area, etc., is an example for prediction.

For example, suppose the marketing manager needs to predict how much a particular customer will spend at his company during a sale. We are bothered to forecast a numerical value in this case. Therefore, an example of numeric prediction is the data processing activity. In this case, a model or a predictor will be developed that forecasts a continuous or ordered value function.

## CLASSIFICATION AND PREDICTION ISSUES

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities, such as:



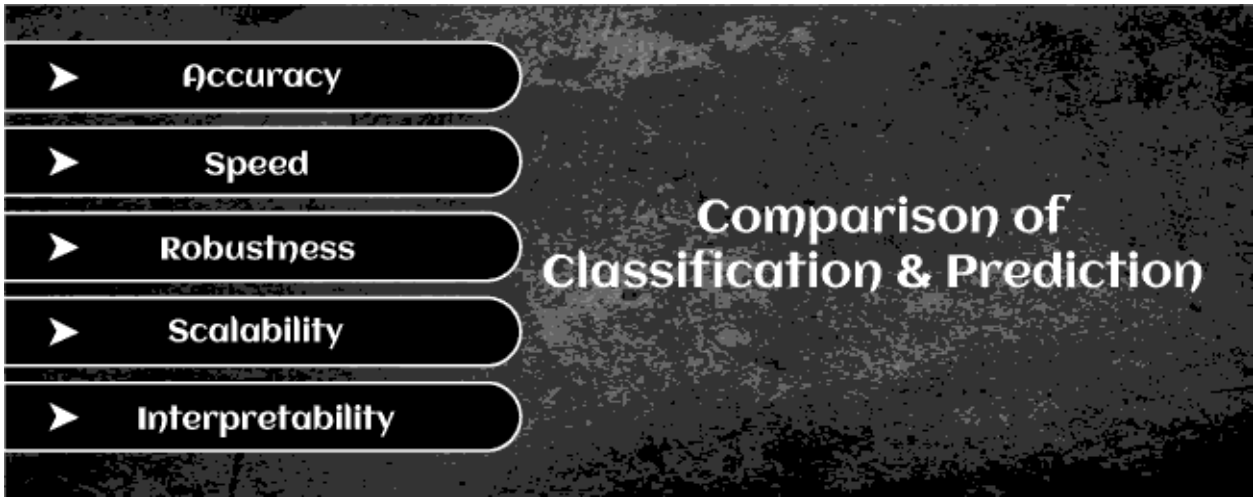

- **Data Cleaning:** Data cleaning involves removing the noise and treatment of missing values. The noise is removed by applying smoothing techniques, and the problem of missing values is solved by replacing a missing value with the most commonly occurring value for that attribute.
- **Relevance Analysis:** The database may also have irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
- **Data Transformation and reduction:** The data can be transformed by any of the following methods.

- **Normalization:** The data is transformed using normalization. Normalization involves scaling all values for a given attribute to make them fall within a small specified range. Normalization is used when the neural networks or the methods involving measurements are used in the learning step.
- **Generalization:** The data can also be transformed by generalizing it to the higher concept. For this purpose, we can use the concept hierarchies.

NOTE: Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis, and clustering.

**COMPARISON OF CLASSIFICATION AND PREDICTION METHODS**

Here are the criteria for comparing the methods of Classification and Prediction, such as:




- **Accuracy:** The accuracy of the classifier can be referred to as the ability of the classifier to predict the class label correctly, and the accuracy of the predictor can be referred to as how well a given predictor can estimate the unknown value.
- **Speed:** The speed of the method depends on the computational cost of generating and using the classifier or predictor.
- **Robustness:** Robustness is the ability to make correct predictions or classifications. In the context of data mining, robustness is the ability of the classifier or predictor to make correct predictions from incoming unknown data.
- **Scalability:** Scalability refers to an increase or decrease in the performance of the classifier or predictor based on the given data.
- **Interpretability:** Interpretability is how readily we can understand the reasoning behind predictions or classification made by the predictor or classifier.

## DIFFERENCE BETWEEN CLASSIFICATION AND PREDICTION

The decision tree, applied to existing data, is a classification model. We can get a class prediction by applying it to new data for which the class is unknown. The assumption is that the new data comes from a distribution similar to the data we used to construct our decision tree. In many instances, this is a correct assumption, so we can use the decision tree to build a predictive model. Classification of prediction is the process of finding a model that describes the classes or concepts of information. The purpose is to predict the class of objects whose class label is unknown using this model. Below are some major differences between classification and prediction.

| Classification | Prediction |
|---|---|
| Classification is the process of identifying which category a new observation belongs to based on a training data set containing observations whose category membership is known. | Predication is the process of identifying the missing or unavailable numerical data for a new observation. |
| In classification, the accuracy depends on finding the class label correctly. | In prediction, the accuracy depends on how well a given predictor can guess the value of a predicated attribute for new data. |
| In classification, the model can be known as the classifier. | In prediction, the model can be known as the predictor. |
| A model or the classifier is constructed to find the categorical labels. | A model or a predictor will be constructed that predicts a continuous-valued function or ordered value. |
| **For example**, the grouping of patients based on their medical records can be considered a classification. | **For example**, We can think of prediction as predicting the correct treatment for a particular disease for a person. |

**DECISION TREE INDUCTION**

Decision Tree is a supervised learning method used in data mining for classification and regression methods. It is a tree that helps us in decision-making purposes.

The decision tree creates classification or regression models as a tree structure.

It separates a data set into smaller subsets, and at the same time, the decision tree is steadily developed.

The final tree is a tree with the decision nodes and leaf nodes.

A decision node has at least two branches.

The leaf nodes show a classification or decision.

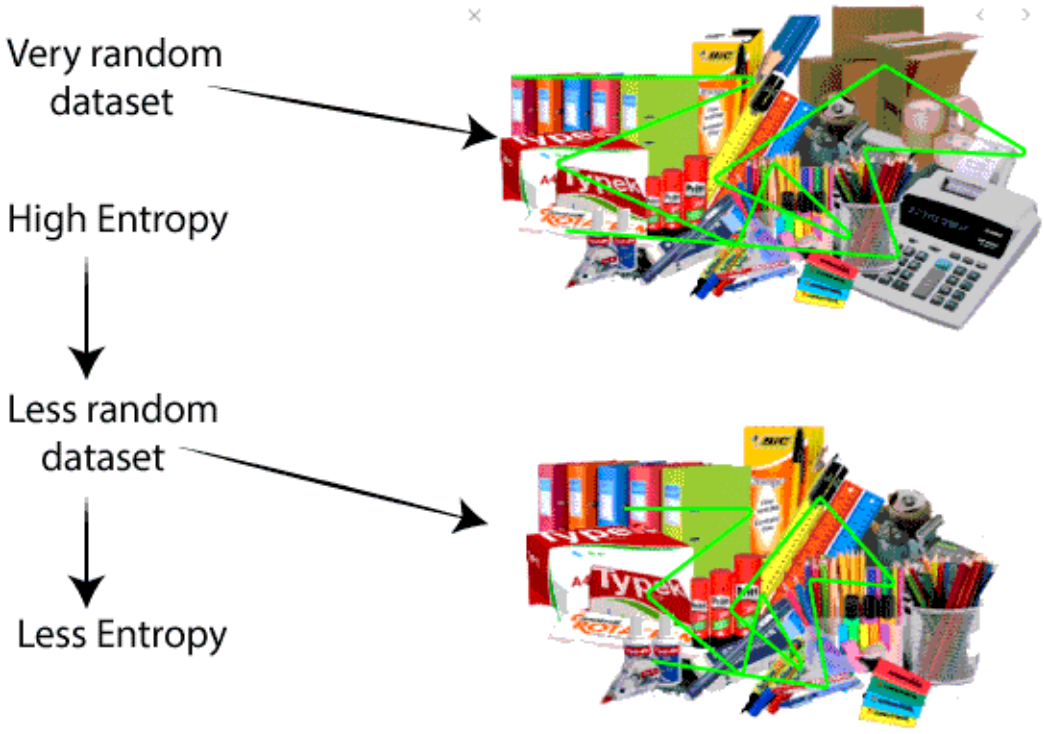
We can't accomplish more split on leaf nodes-The uppermost decision node in a tree that relates to the best predictor called the root node.

Decision trees can deal with both categorical and numerical data.

**KEY FACTORS:**

**ENTROPY:**

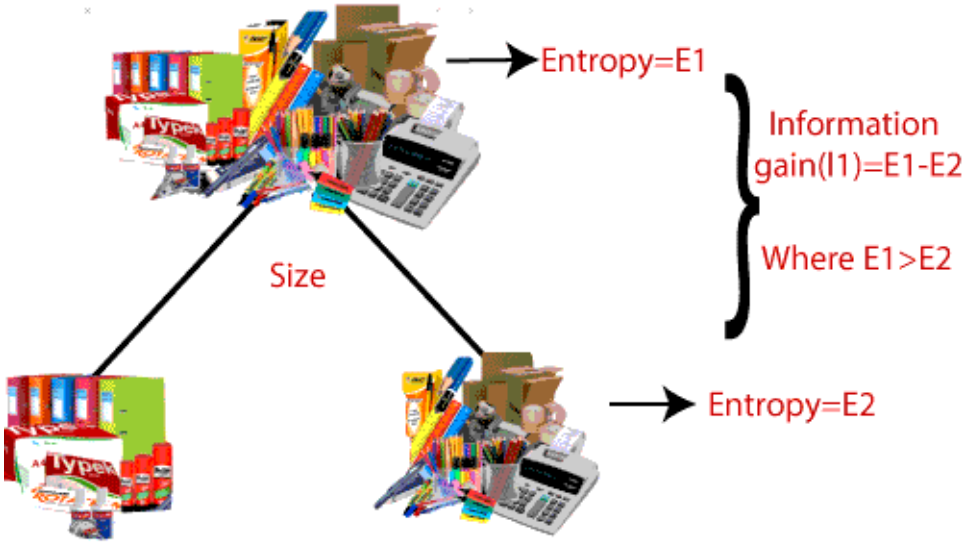
Entropy refers to a common way to measure impurity. In the decision tree, it measures the randomness or impurity in data sets.



**INFORMATION GAIN:**

Information Gain refers to the decline in entropy after the dataset is split.

It is also called **Entropy Reduction**. Building a decision tree is all about discovering attributes that return the highest data gain.




In short, a decision tree is just like a flow chart diagram with the terminal nodes showing decisions. Starting with the dataset, we can measure the entropy to find a way to segment the set until the data belongs to the same class.

Why are decision trees useful?

It enables us to analyze the possible consequences of a decision thoroughly.

It provides us a framework to measure the values of outcomes and the probability of accomplishing them.

It helps us to make the best decisions based on existing data and best speculations.

**In other words**, we can say that a decision tree is a hierarchical tree structure that can be used to split an extensive collection of records into smaller sets of the class by implementing a sequence of simple decision rules.

A decision tree model comprises a set of rules for portioning a huge heterogeneous population into smaller, more homogeneous, or mutually exclusive classes.

The attributes of the classes can be any variables from nominal, ordinal, binary, and quantitative values, in contrast, the classes must be a qualitative type, such as categorical or ordinal or binary.

In brief, the given data of attributes together with its class, a decision tree creates a set of rules that can be used to identify the class.

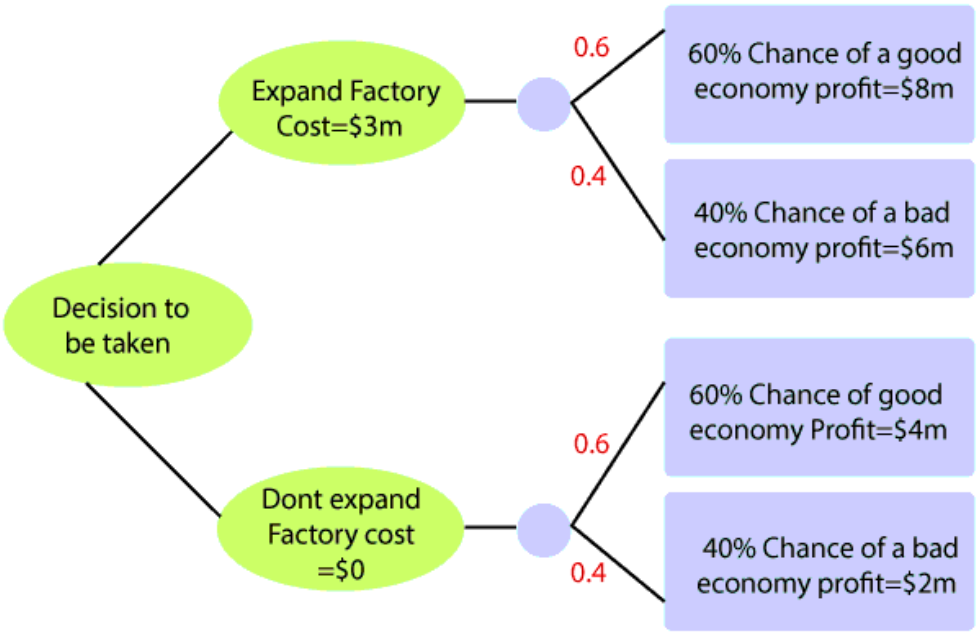
One rule is implemented after another, resulting in a hierarchy of segments within a segment.

The hierarchy is known as the **tree**, and each segment is called a **node**.

With each progressive division, the members from the subsequent sets become more and more similar to each other.

Hence, the algorithm used to build a decision tree is referred to as recursive partitioning. The algorithm is known as **CART** (Classification and Regression Trees)

Consider the given example of a factory where




Expanding factor costs $3 million, the probability of a good economy is 0.6 (60%), which leads to $8 million profit, and the probability of a bad economy is 0.4 (40%), which leads to $6 million profit.

Not expanding factor with 0$ cost, the probability of a good economy is 0.6(60%), which leads to $4 million profit, and the probability of a bad economy is 0.4, which leads to $2 million profit.

The management teams need to take a data-driven decision to expand or not based on the given data.

Net Expand = ( 0.6 *8 + 0.4*6 ) - 3 = $4.2M
Net Not Expand = (0.6*4 + 0.4*2) - 0 = $3M
$4.2M > $3M,therefore the factory should be expanded.

Decision tree Algorithm:

The decision tree algorithm may appear long, but it is quite simply the basis algorithm techniques is as follows:

The **algorithm** is based on three parameters**: D, attribute_list, and Attribute _selection_method.**

Generally, we refer to **D** as a **data partition.**

Initially, **D** is the entire set of **training tuples** and their related **class levels** (input training data).

The parameter **attribute_list** is a set of **attributes** defining the tuples.

**Attribute_selection_method** specifies a **heuristic process** for choosing the attribute that "best" discriminates the given tuples according to **class**.

**Attribute_selection_method** process applies an **attribute selection measure**.

**ADVANTAGES OF USING DECISION TREES:**

A decision tree does not need scaling of information.

Missing values in data also do not influence the process of building a choice tree to any considerable extent.

A decision tree model is automatic and simple to explain to the technical team as well as stakeholders.

Compared to other algorithms, decision trees need less exertion for data preparation during pre-processing.

A decision tree does not require a standardization of data.

Data Mining Bayesian Classifiers

In numerous applications, the connection between the attribute set and the class variable is non- deterministic. In other words, we can say the class label of a test record cant be assumed with certainty even though its attribute set is the same as some of the training examples. These circumstances may emerge due to the noisy data or the presence of certain confusing factors that influence classification, but it is not included in the analysis. For example, consider the task of predicting the occurrence of whether an individual is at risk for liver illness based on individuals eating habits and working efficiency. Although most people who eat healthly and exercise consistently having less probability of occurrence of liver disease, they may still do so due to other factors. For example, due to consumption of the high-calorie street foods and alcohol abuse. Determining whether an individual's eating routine is healthy or the workout efficiency is sufficient is also subject to analysis, which in turn may introduce vulnerabilities into the leaning issue.

**BAYESIAN CLASSIFICATION:**

Bayesian classification uses Bayes theorem to predict the occurrence of any event.

Bayesian classifiers are the statistical classifiers with the Bayesian probability understandings.

The theory expresses how a level of belief, expressed as a probability.

Bayes theorem came into existence after Thomas Bayes, who first utilized conditional probability to provide an algorithm that uses evidence to calculate limits on an unknown parameter.

Bayes's theorem is expressed mathematically by the following equation that is given below.

$$P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}$$

Where X and Y are the events and P (Y) $\neq$ 0

P(X/Y) is a conditional probability that describes the occurrence of event X is given that Y is true.

P(Y/X) is a conditional probability that describes the occurrence of event Y is given that X is true.

P(X) and P(Y) are the probabilities of observing X and Y independently of each other. This is known as the marginal probability.

**BAYESIAN INTERPRETATION:**

In the Bayesian interpretation, probability determines a "degree of belief."

Bayes theorem connects the degree of belief in a hypothesis before and after accounting for evidence.

For example, Lets us consider an example of the coin.

If we toss a coin, then we get either heads or tails, and the percent of occurrence of either heads and tails is 50%.

If the coin is flipped numbers of times, and the outcomes are observed, the degree of belief may rise, fall, or remain the same depending on the outcomes.

P(X), the prior, is the primary degree of belief in X

P(X/Y), the posterior is the degree of belief having accounted for Y.

The quotient $\frac{P(Y/X)}{P(Y)}$ represents the supports Y provides for X.

Bayes theorem can be derived from the conditional probability:

$$P(X/Y) = \frac{P(X \cap Y)}{P(Y)}, \text{ if } P(Y) \neq 0$$

$$P(Y/X) = \frac{P(Y \cap X)}{P(X)}, \text{ if } P(X) \neq 0$$

Where P (X⋂Y) is the joint probability of both X and Y being true, because

$$P(Y \cap X) = P(X \cap Y)$$

$$\text{or, } P(X \cap Y) = P(X/Y)P(Y) = P(Y/X)P(X)$$

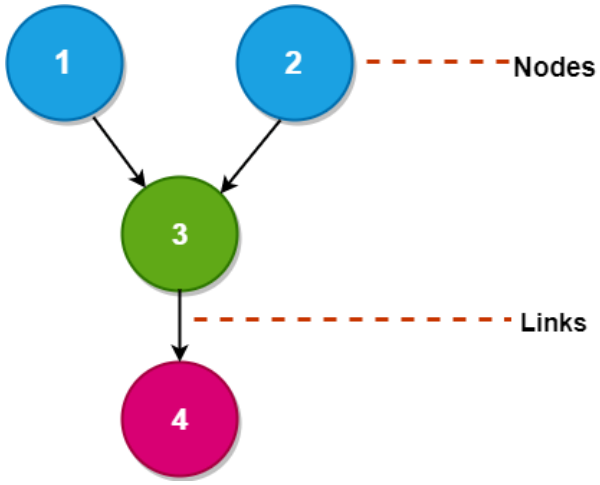
$$\text{or, } P(X/Y) = \frac{P(Y/X)P(X)}{P(Y)}, \text{ if } P(Y) \neq 0$$

**BAYESIAN NETWORK**

A Bayesian Network falls under the classification of Probabilistic Graphical Modelling (PGM) procedure that is utilized to compute uncertainties by utilizing the probability concept.

Generally known as Belief Networks, Bayesian Networks are used to show uncertainties using Directed Acyclic Graphs (DAG)

A Directed Acyclic Graph is used to show a Bayesian Network, and like some other statistical graph, a DAG consists of a set of nodes and links, where the links signify the connection between the nodes.




The nodes here represent random variables, and the edges define the relationship between these variables.

A DAG models the uncertainty of an event taking place based on the Conditional Probability Distribution (CDP) of each random variable.

A Conditional Probability Table (CPT) is used to represent the CPD of each variable in a network.

Bayesian classification is based on Bayes' Theorem.

Bayesian classifiers are the statistical classifiers.

Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

**BAYE'S THEOREM**

Bayes' Theorem is named after Thomas Bayes. There are two types of probabilities −

- Posterior Probability [P(H/X)]
- Prior Probability [P(H)]

where X is data tuple and H is some hypothesis.

According to Bayes' Theorem,

$$P(H/X)= P(X/H)P(H) / P(X)$$

## BAYESIAN BELIEF NETWORK

Bayesian Belief Networks specify joint conditional probability distributions.

They are also known as Belief Networks, Bayesian Networks, or Probabilistic Networks.

- A Belief Network allows class conditional independencies to be defined between subsets of variables.
- It provides a graphical model of causal relationship on which learning can be performed.
- We can use a trained Bayesian Network for classification.

There are two components that define a Bayesian Belief Network −

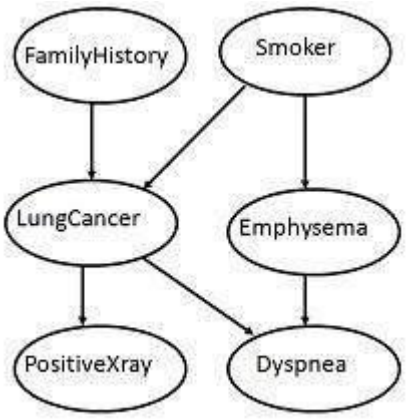
- Directed acyclic graph
- A set of conditional probability tables

## DIRECTED ACYCLIC GRAPH

- Each node in a directed acyclic graph represents a random variable.
- These variable may be discrete or continuous valued.
- These variables may correspond to the actual attribute given in the data.

## DIRECTED ACYCLIC GRAPH REPRESENTATION

The following diagram shows a directed acyclic graph for six Boolean variables.




## ACYCLIC GRAPH

The arc in the diagram allows representation of causal knowledge.

For example, lung cancer is influenced by a person's family history of lung cancer, as well as whether or not the person is a smoker.

It is worth noting that the variable PositiveXray is independent of whether the patient has a family history of lung cancer or that the patient is a smoker, given that we know the patient has lung cancer.

**CONDITIONAL PROBABILITY TABLE**

The conditional probability table for the values of the variable LungCancer (LC) showing each possible combination of the values of its parent nodes, FamilyHistory (FH), and Smoker (S) is as follows −

**PROBABILITY TABLE**

| | FH,S | FH,-S | -FH,S | -FH,S |
|---|---|---|---|---|
| LC | 0.8 | 0.5 | 0.7 | 0.1 |
| -LC | 0.2 | 0.5 | 0.3 | 0.9 |

**RULE BASED CLASSIFICATION**

**IF-THEN RULES**

Rule-based classifier makes use of a set of IF-THEN rules for classification.

We can express a rule in the following from −

**IF condition THEN conclusion**

Let us consider a rule R1,

**R1: IF age = youth AND student = yes**

**THEN buy_computer = yes**

**POINTS TO REMEMBER −**

- The IF part of the rule is called rule antecedent or precondition.
- The THEN part of the rule is called rule consequent.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.
- The consequent part consists of class prediction.

Note − We can also write rule R1 as follows −

**R1: (age = youth) ^ (student = yes))(buys computer = yes)**

If the condition holds true for a given tuple, then the antecedent is satisfied.

**RULE EXTRACTION**

Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

**POINTS TO REMEMBER −**

To extract a rule from a decision tree −

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent.

**RULE INDUCTION USING SEQUENTIAL COVERING ALGORITHM**

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data.

We do not require to generate a decision tree first.

In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER.

As per the general strategy the rules are learned one at a time.

For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples.

This is because the path to each leaf in a decision tree corresponds to a rule.

**Note**: The Decision tree induction can be considered as learning a set of rules simultaneously.

The Following is the sequential learning Algorithm where rules are learned for one class at a time.

When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

**ALGORITHM: SEQUENTIAL COVERING**

**Input:**

D, a data set class-labeled tuples,

Att_vals, the set of all attributes and their possible values.

**Output:** A Set of IF-THEN rules.

**Method:**

```
Rule_set={ }; // initial set of rules learned is empty
for each class c do
    repeat
    Rule = Learn_One_Rule(D, Att_valls, c);
    remove tuples covered by Rule form D;
  until termination condition;
```

Rule_set=Rule_set+Rule; // add a new rule to rule-set

end for

return Rule_Set;

**RULE PRUNING**

The rule is pruned is due to the following reason −

The Assessment of quality is made on the original set of training data.

The rule may perform well on training data but less well on subsequent data.

That's why the rule pruning is required.

The rule is pruned by removing conjunct.

The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.

FOIL is one of the simple and effective method for rule pruning.

For a given rule R,

**FOIL_Prune = pos - neg / pos + neg**

where pos and neg is the number of positive tuples covered by R, respectively.

**Note** : This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

**LAZY LEARNER**

Lazy learning is a type of machine learning that doesn't process training data until it needs to make a prediction. Instead of building models during training,

lazy learning algorithms wait until they encounter a new query.

This method stores and compares training examples when making predictions.

It's also called instance-based or memory-based learning.

The Machine Learning systems which are categorized as instance-based learning are the systems that learn the training examples by heart and then generalizes to new instances based on some similarity measure.

It is called instance-based because it builds the hypotheses from the training instances. It is also known as memory-based learning or lazy-learning (because they delay processing until a new instance must be classified).

The time complexity of this algorithm depends upon the size of training data.

Each time whenever a new query is encountered, its previously stores data is examined. And assign to a target function value for the new instance.

The worst-case time complexity of this algorithm is O (n), where n is the number of training instances.

For example, If we were to create a spam filter with an instance-based learning algorithm, instead of just flagging emails that are already marked as spam emails, our spam filter would be programmed to also flag emails that are very similar to them.

This requires a measure of resemblance between two emails.

A similarity measure between two emails could be the same sender or the repetitive use of the same keywords or something else.

**ADVANTAGES:**

Instead of estimating for the entire instance set, local approximations can be made to the target function.

This algorithm can adapt to new data easily, one which is collected as we go .

**DISADVANTAGES:**

- Classification costs are high
- Large amount of memory required to store the data, and each query involves starting the identification of a local model from scratch.

**Some of the instance-based learning algorithms are:**

- K Nearest Neighbor (KNN)
- Self-Organizing Map (SOM)
- Learning Vector Quantization (LVQ)
- Locally Weighted Learning (LWL)
- Case-Based Reasoning

**LAZY LEARNING EXPLAINED**

Lazy learning algorithms work by memorizing the training data rather than constructing a general model.

When a new query is received, lazy learning retrieves similar instances from the training set and uses them to generate a prediction.

The similarity between instances is usually calculated using distance metrics, such as Euclidean distance or cosine similarity.

One of the most popular lazy learning algorithms is the k-nearest neighbors (k-NN) algorithm. In k-NN, the k closest training instances to the query point are considered, and their class labels are used to determine the class of the query.

Lazy learning methods excel in situations where the underlying data distribution is complex or where the training data is noisy.

## EXAMPLES OF REAL-WORLD LAZY LEARNING APPLICATIONS

Lazy learning has found applications in various domains. Here are a few examples:

**RECOMMENDATION SYSTEMS.** Lazy learning is widely used in recommender systems to provide personalized recommendations. By comparing user preferences to similar users in the training set, lazy learning algorithms can suggest items or products of interest, such as movies, books, or products.

**MEDICAL DIAGNOSIS.** Lazy learning can be employed in medical diagnosis systems. By comparing patient symptoms and medical histories to similar cases in the training data, lazy learning algorithms can assist in diagnosing diseases or suggesting appropriate treatments.

**ANOMALY DETECTION.** Lazy learning algorithms are useful for detecting anomalies or outliers in datasets. For example, an algorithm can detect credit card fraud by comparing a transaction to nearby transactions based on factors like location and history. If the transaction is unusual, such as being made in a faraway location for a large amount, it may be flagged as fraudulent.

## LAZY LEARNING VS EAGER LEARNING MODELS

Lazy learning stands in contrast to eager learning methods, such as decision trees or neural networks, where models are built during the training phase. Here are some key differences:

**TRAINING PHASE.** Eager learning algorithms construct a general model based on the entire training dataset, whereas lazy learning algorithms defer model construction until prediction time.

**COMPUTATIONAL COST.** Lazy learning algorithms can be computationally expensive during prediction since they require searching through the training data to find nearest neighbors. In contrast, eager learning algorithms typically have faster prediction times once the model is trained.

**INTERPRETABILITY.** Eager learning methods often provide more interpretability as they produce explicit models, such as decision trees, that can be easily understood by humans. Lazy learning methods, on the other hand, rely on the stored instances and do not provide explicit rules or models.

## BENEFITS OF LAZY LEARNING

Lazy learning offers several advantages:

**ADAPTABILITY.** Lazy learning algorithms can adapt quickly to new or changing data. Since the learning process happens at prediction time, they can incorporate new instances without requiring complete retraining of the model.

**ROBUSTNESS TO OUTLIERS.** Lazy learning algorithms are less affected by outliers compared to eager learning methods. Outliers have less influence on predictions because they are not used during the learning phase.

**FLEXIBILITY.** When it comes to handling complex data distributions and nonlinear relationships, lazy learning algorithms are effective. They can capture intricate decision boundaries by leveraging the information stored in the training instances.

## LIMITATIONS OF LAZY LEARNING

Despite its benefits, lazy learning has certain limitations that should be considered:

**HIGH PREDICTION TIME.** Lazy learning can be slower at prediction time compared to eager learning methods. Since they require searching through the training data to find nearest neighbors, the computational cost can be significant, especially with large datasets.

**STORAGE REQUIREMENTS.** Lazy learning algorithms need to store the entire training dataset or a representative subset of it. This can be memory-intensive, particularly when dealing with large datasets with high-dimensional features.

**SENSITIVITY TO NOISE.** Noise or irrelevant features in the training data can significantly impact the accuracy of lazy learning model predictions, because they rely on direct comparison with stored instances.

**OVERFITTING**. Lazy learning algorithms are prone to overfitting when the training dataset is small or when there are too many stored instances. Overfitting occurs when the model memorizes the training instances, including their noise or outliers, leading to poor generalization on unseen data.

**LACK OF TRANSPARENCY.** Lazy learning methods do not provide explicit models or rules that can be easily interpreted. This lack of transparency makes it challenging to understand the reasoning behind specific predictions or to extract actionable insights from the model.