MODULE III CLASSIFICATION AND PREDICTION 9

Classification and Prediction: - Issues Regarding Classification and Prediction - Classification by Decision Tree Introduction - Bayesian Classification - Rule Based Classification - Classification by Back propagation - Support Vector Machines - Associative Classification - Lazy Learners - Prediction - Model Evaluation and Selection Accuracy and Error Measures.

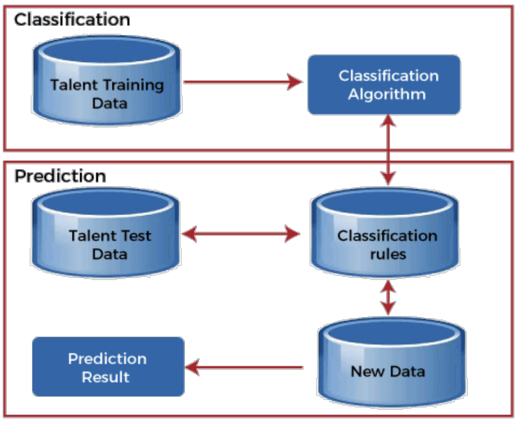
3.1 CLASSIFICATION AND PREDICATION IN DATA MINING

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:

1. Classification
2. 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?

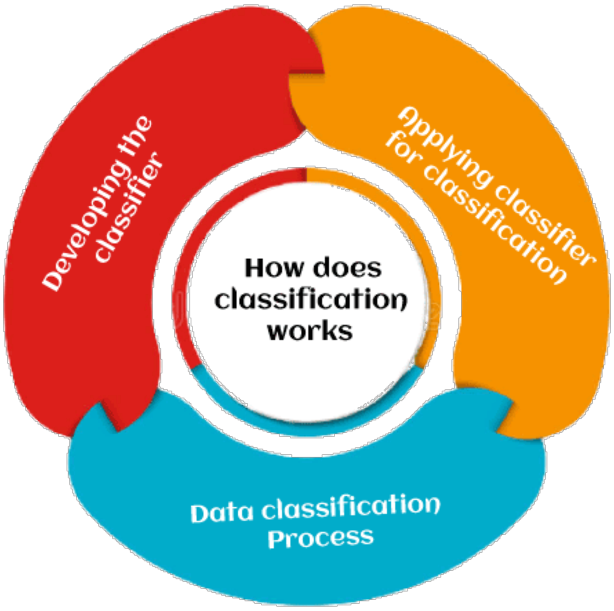
Classification is to identify the category or the class label of a new observation. First, a set of data is used as training data. The set of input data and the corresponding outputs are given to the algorithm. So, the training data set includes the input data and their associated class labels. Using 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

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
Applying classifier for classification
How does classification works
Data classification Process

1. **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.
2. **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.

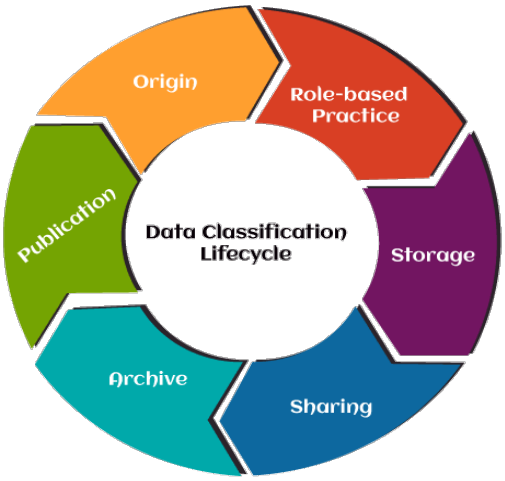
And with the help of machine learning classification algorithms, we can execute it automatically.

- **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.

3. **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:



1. **Origin:** It produces sensitive data in various formats, with emails, Excel, Word, Google documents, social media, and websites.
2. **Role-based practice:** Role-based security restrictions apply to all delicate data by tagging based on in-house protection policies and agreement rules.
3. **Storage:** Here, we have the obtained data, including access controls and encryption.
4. **Sharing:** Data is continually distributed among agents, consumers, and co-workers from various devices and platforms.
5. **Archive:** Here, data is eventually archived within an industry's storage systems.
6. **Publication:** Through the publication of data, it can reach customers. They can then view and download in the form of dashboards.

What is Prediction?

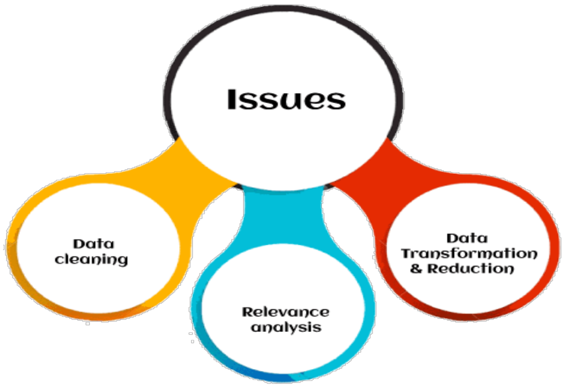
Another process of data analysis is prediction. It is used to find a numerical output. Same as in classification, the training dataset contains the inputs and corresponding numerical output values. The algorithm derives the model or a predictor according to the training dataset. The model should find 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:



1. **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.
2. **Relevance Analysis:** The database may also have irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
3. **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.

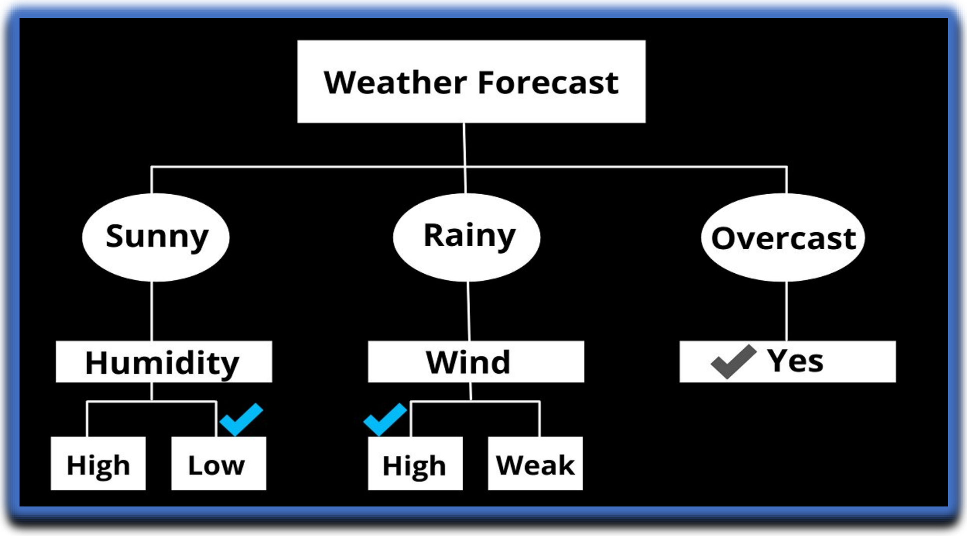
Difference between Classification and Prediction

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 IN DATA MINING

A decision tree is a type of algorithm that classifies information so that a tree-shaped model is generated. It is a schematic model of the information that represents the different alternatives and the possible results for each chosen alternative. Decision trees are a widely used model because they greatly facilitate understanding of the different options.



The above example of a decision tree helps to determine if one should play cricket or not. If the weather forecast suggests that it is overcast then you should definitely play cricket. If it is rainy, you should play only if the wind is weak and if it is sunny then you should play if the humidity is normal or low.

Decision Tree Components

The decision tree is made up of nodes and branches. There are different types of nodes and branches depending on what you want to represent. Decision nodes represent a decision to be made, probability nodes represent possible uncertain outcomes and terminal nodes that represent the final outcome.

On the other hand, the branches are differentiated into alternative branches, where each branch leads to a type of result and, the "rejected" branches, which represent the results that are rejected. The model is characterized in that the same problem can be represented with different trees.

Types of Decision Trees in Data Mining

Decision tree in data mining is mainly divided into two types –

Categorical Variable Decision Tree

A categorical variable decision tree comprises categorical target variables, which are further bifurcated categories, such as Yes or No. Categories specify that the stages of a decision process are categorically divided.

Continuous Variable Decision Tree

A continuous variable decision tree has a continuous target variable. One example to understand this could be – the unknown salary of an employee can be predicted bases on the available profile information, such as his/her job role, age, experience, and other continuous variables.

Functions of Decision Tree in Data Mining

For greater precision, multiple decision trees are combined with e 4 assembly methods.

Bagging or Assembly – This method creates several decision trees as a resampling of the source data, and then the tree that denotes that the best results should be used.

Random Jungle Sorter – Multiple decision trees are generated to increase the sort rate and efficiently separate data.

Expanded Trees – Multiple trees are created to correct the errors of the last one with respect to the first.

Random Forest or Rotation Forest – Decision trees created in this scenario are analyzed based on a series of main variables.

Decision Tree Algorithms

Although there are various algorithms used to create decision trees in data mining, the most relevant are the following:

ID3: decision trees with this algorithm are oriented towards finding hypotheses or rules in relation to the analyzed data.

C4.5: decision trees that use this algorithm focus on classifying data, in this way; they are associated with statistical classification.

ACR: the decision trees of this algorithm are focused on avoiding future problems, as they are used to detect the causes that generate the defects.

Advantages of Using Decision Trees in Data Mining

Decision trees in data mining provide us with various advantages to analyze and classify the data in your information base. However, experts highlight the following –

Ease of Understanding

Because data mining tools can visually capture this model in a very practical way, people can understand how it works after a short explanation. It is not necessary to have extensive knowledge of data mining or web programming languages.

Does Not Require Data Normalization

Most data mining techniques require the preparation of data for processing, that is, the analysis and discard of data in poor condition. This is not the case for decision trees in data mining, as they can start working directly.

Handling of Numbers and Categorized Data

One of the main differences between neural networks and decision trees is that the latter analyze a large number of variables.

While neural networks simply focus on numerical variables, decision trees encompass both numerical and nominal variables. Therefore, they will help you to analyze a large amount of information together.

“White Box” Model

In web programming and data mining, the white box model brings together a type of software test in which the variables are evaluated to determine what are the possible scenarios or execution paths based on a decision.

Uses of Statistics

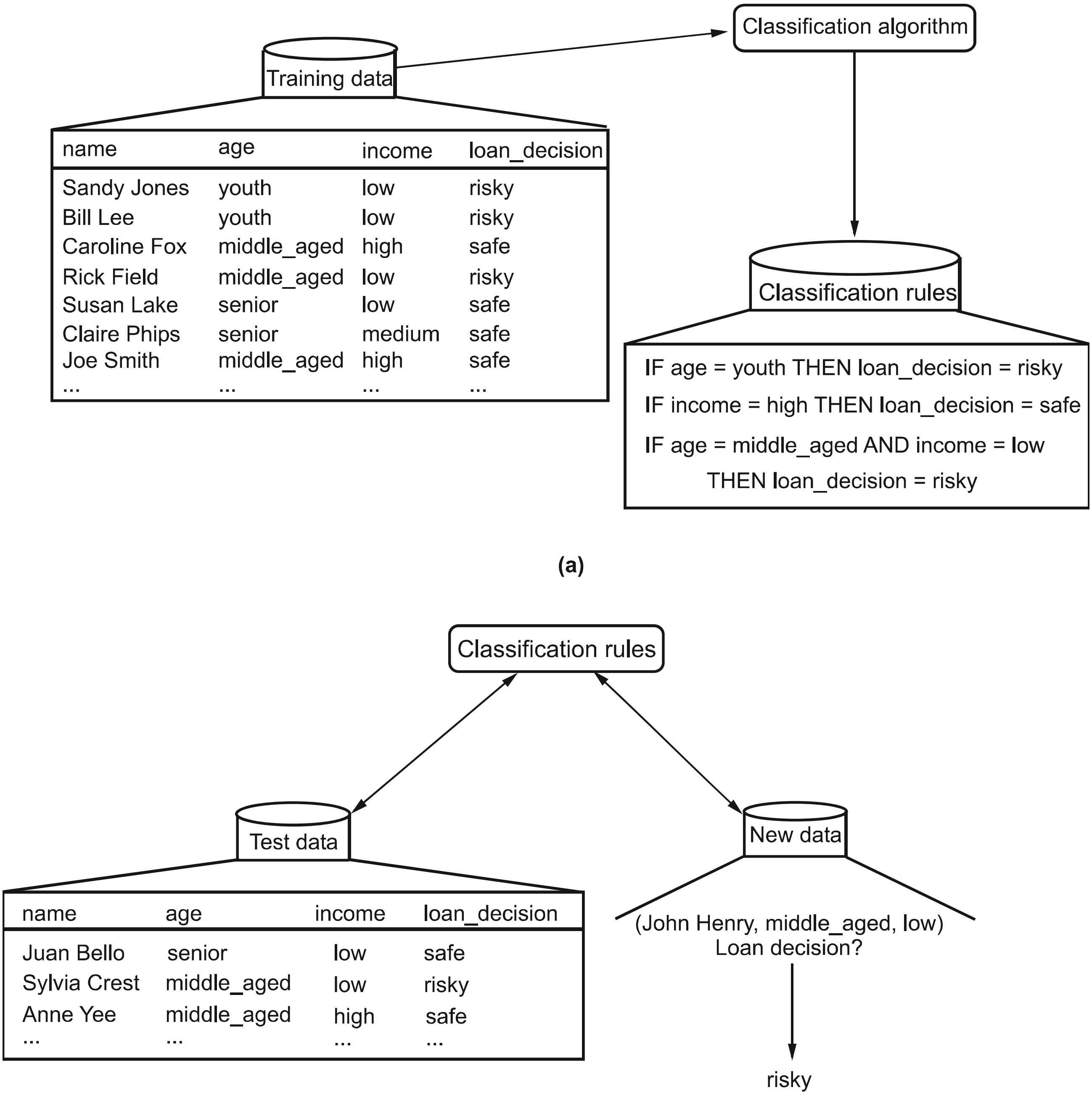
Decision trees and statistics work hand in hand to provide greater reliability to the model that is being developed. Since each result is supported by various statistical tests, the probability of any of the options analyzed can be known exactly.

3.2 CLASSIFICATION

4.Introduction to Classification

- Definition
  - Data classification is the process of organizing data into categories for its most effective and efficient use. A well-planned data classification system makes essential data easy to find and retrieve. Data classification is the process of sorting and categorizing data into various types, forms or any other distinct class.
  - Data classification enables the separation and classification of data according to data set requirements for various business or personal objectives.
  - Data Classification is a form of data analysis that extracts models describing important data classes. Such models, called classifiers, predict categorical (discrete, unordered) class labels.
- Data classification as a part of the Information Lifecycle Management (ILM) process can be defined as a tool for categorization of data to enable/help organizations to effectively answer the following questions :
  - What data types are available ?
  - Where are certain data located ?
  - What access levels are implemented ?
  - What protection level is implemented and does it adhere to compliance regulations ?
- For example,
  - A classification model can be built to categorize bank loan applications as either safe or risky. Such analysis helps us for a better understanding of the data at large
  - A medical researcher wants to analyse breast cancer data to predict which one of three
  - Specific treatments a patient should receive.
  - A marketing manager at Elexmart - an electronics outlet needs data analysis to help guess whether a customer with a given profile will buy a new laptop.
- Data Classification is mainly a data management process.
- Classification has many applications in various domains. These include fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.
- There are many classification methods used in machine learning, pattern recognition, and statistics. These algorithms are designed assuming a small data size. Most of these algorithms are memory resident.

- With high data volumes in the recent days, researchers have put efforts in developing scalable classification and prediction techniques capable of handling large amounts of disk-resident data.
- How does classification work ?
  - Data classification is a two-step process.
    - **A learning step :** In this step a classification model is constructed.
    - **A classification step :** In this step the model is used to predict class labels for given data.
  - Learning step
    - The learning step (or training phase), where a classification algorithm builds the classifier by analysing or "learning from" a training set made up of database tuples and their associated class labels.
    - A classifier is built describing a predetermined set of data classes or concepts.
    - This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from" a training set made up of database tuples and their associated class labels.
    - The learning process :
    - A tuple, X, is represented by an n-dimensional attribute vector, X_D .x_1, x_2, ……, x_n, depicting n measurements made on the tuple from n database attributes, respectively, A_1, A_2, ….., A_n.
    - Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete-valued and unordered.
    - It is categorical (or nominal) in that each value serves as a category or class.
    - The individual tuples making up the training set are referred to as training tuples and are randomly sampled from the database under analysis.
    - In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.
    - If the class label of each training tuple is provided, this step is also known as **Supervised Learning**. If the class label of each training tuple is not known, and the number or set of classes to be learned may not be known in advance, it is called **Unsupervised Learning**.
    - Refer Fig.(a) - Training data are analyzed by a classification algorithm. Here, the class label attribute is loan decision, and the learned model or classifier is represented in the form of classification rules.



name	age	income	loan_decision
Sandy Jones	youth	low	risky
Bill Lee	youth	low	risky
Caroline Fox	middle_aged	high	safe
Rick Field	middle_aged	low	risky
Susan Lake	senior	low	safe
Claire Phips	senior	medium	safe
Joe Smith	middle_aged	high	safe
...	...	...	...

IF age = youth THEN loan_decision = risky

IF income = high THEN loan_decision = safe

IF age = middle_aged AND income = low
THEN loan_decision = risky

(a)

name	age	income	loan_decision
Juan Bello	senior	low	safe
Sylvia Crest	middle_aged	low	risky
Anne Yee	middle_aged	high	safe
...	...	...	...

(John Henry, middle_aged, low)
Loan decision?

risky

(b)

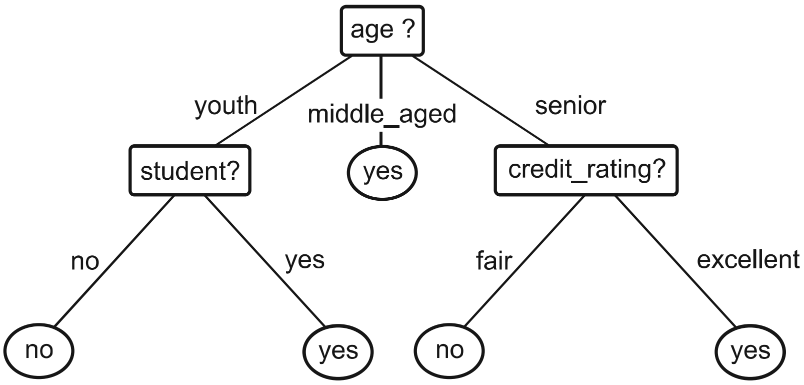
Learning and Classification

- **Classification step**
  - This first step of the classification process can also be viewed as the learning of a mapping or function, y = f(X) that can predict the associated class label y of a given tuple X.
  - Typically, this mapping is represented in the form of classification rules, decision trees, or mathematical formulae.

- In this step, the predictive accuracy of the classifier is estimated. For this purpose, a test set is used, made up of test tuples and their associated class labels. This test data is independent of the training tuples and not used for constructing the classifier tuple.
- The accuracy of a classifier on a given test set is the percentage of test set tuples that are correctly classified by the classifier.
- Refer Fig(b) - Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

3.2.1 Decision Tree Induction

- Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where
  - Each internal node (non-leaf node) denotes a test on an attribute
  - Each branch represents an outcome of the test
  - Each leaf node (or terminal node) holds a class label
  - The topmost node in a tree is the root node
- Example - Refer Fig. for a decision tree for the concept buys computer, indicating whether an Elexmart - an electronics outlet customer is likely to purchase a computer. Each internal (non-leaf) node represents a test on an attribute. Each leaf node represents a class (either buys computer = yes or buys computer = no).



A decision tree for Elexmart

- A decision tree represents the concept 'a customer buys computer'.

®

  - It predicts whether a customer at Elexmart is likely to purchase a computer. Internal nodes are denoted by rectangles, and leaf nodes are denoted by ovals.
  - Some decision tree algorithms produce only binary trees (where each internal node branches to exactly two other nodes), whereas others can produce nonbinary trees.
- How are decision trees used for classification ?
  - Given a tuple, X, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree.
  - A path is traced from the root to a leaf node, which holds the class prediction for that tuple.
  - Decision trees can easily be converted to classification rules.
- Why are decision tree classifiers so popular ?
  - The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery.
  - Decision trees can handle multidimensional data.
  - Their representation of acquired knowledge in tree form is intuitive and generally easy to assimilate by humans.
  - The learning and classification steps of decision tree induction are simple and fast.
  - In general, decision tree classifiers have good accuracy.
  - However, successful use may depend on the data at hand.
  - Decision tree induction algorithms have been used for classification in many application areas such as medicine, manufacturing and production, financial analysis, astronomy, and molecular biology.
  - Decision trees are the basis of several commercial rule induction systems.
- **Algorithm : Generate decision tree.**

  Generate a decision tree from the training tuples of data partition, D.

Input :

Data partition, D, which is a set of training tuples and their associated class labels;

attribute list, the set of candidate attributes;

Attribute selection method, a procedure to determine the splitting criterion that "best"

partitions the data tuples into individual classes. This criterion consists of a

splitting attribute and, possibly, either a *split-point* or *splitting subset*.

Output : A decision tree**.**

Method :

```
(1) create a node N;
(2) if tuples in D are all of the same class, C, then
(3)     return N as a leaf node labeled with the class C;
(4) if attribute list is empty then
(5)     return N as a leaf node labeled with the majority class in D; // majority voting
(6) apply Attribute selection method(D, attribute list) to find the "best" splitting criterion;
(7) label node N with splitting criterion;
(8) if splitting attribute is discrete-valued and multiway splits allowed then // not restricted to binary trees
(9)     attribute list ← attribute list - splitting attribute; //remove splitting attribute
(10) for each outcome j of splitting criterion
     // partition the tuples and grow subtrees for each partition
(11)     let Dj be the set of data tuples in D satisfying outcome j; // a partition
(12)     if Dj is empty then
(13)         attach a leaf labeled with the majority class in D to node N;
(14)     else attach the node returned by Generate decision tree(Dj ,attribute list) to node N;
         endfor
(15) return N;
```

- Applying algorithm for **generating decision tree** for Elexmart case.
  - Let A be the splitting attribute. A has v distinct values, fa1, a2, ….. , avg, based on the training data.
  - There are three possible scenarios, as illustrated in Fig. (4.1.2), These three possibilities for partitioning tuples based on the splitting criterion, each with examples.
  - Let A be the splitting attribute.

    (a) If A is discrete-valued, then one branch is grown for each known value of A.

    (b) If A is continuous-valued, then two branches are grown, corresponding to A <= split point and A > split point.

®

(c) If A is discrete-valued and a binary tree must be produced, then the test is of the form A € S_A , where S_A is the splitting subset for A.

- The algorithm uses the same process recursively to form a decision tree for the tuples at each resulting partition, Dj , of D (step 14).
  - The recursive partitioning stops only when any one of the following terminating conditions is true :
  - All the tuples in partition D (represented at node N) belong to the same class (steps 2 and 3).
  - There are no remaining attributes on which the tuples may be further partitioned (step 4). In this case, majority voting is employed (step 5).
  - This involves converting node N into a leaf and labeling it with the most common class in D.
  - Alternatively, the class distribution of the node tuples may be stored.
  - There are no tuples for a given branch, that is, a partition Dj is empty (step 12).
  - In this case, a leaf is created with the majority class in D (step 13).
  - The resulting decision tree is returned (step 15).
- Tree Pruning
  - Tree pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.
  - Tree Pruning Approaches -There are two approaches to prune a tree ?
- Pre-pruning - The tree is pruned by halting its construction early.
- Post-pruning - This approach removes a sub-tree from a fully-grown tree.
- Cost Complexity
  - The cost complexity is measured by the following two parameters ?
    - Number of leaves in the tree, and
    - Error rate of the tree.
- Strengths of Decision Tree approach
  - Decision trees are able to generate understandable rules.
  - Decision trees perform classification without requiring much computation.
  - Decision trees are able to handle both continuous and categorical variables.
  - Decision trees provide a clear indication of which fields are most important for prediction or classification.
- Weaknesses of Decision Tree approach
  - Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
  - Decision trees are prone to errors in classification problems with many class and relatively small number of training examples.

- ○ Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.

3.3. BAYESIAN CLASSIFICATION

- ○ 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.
- ○ Bayesian classifiers results with high accuracy and speed when applied to large databases.
- ○ Na¨Ive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes.
- ○ This assumption is called class conditional independence.
- ○ It is made to simplify the computations involved and, in this sense, is considered "na¨Ive."

- **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,

$$\boxed{P(H/X) = P(X/H)P(H) / P(X)}$$

- **Na¨Ive Bayesian classifiers -**
  - ○ Naive Bayes is a simple, yet effective and commonly-used, learning classifier.
  - ○ It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting.
  - ○ Na¨ive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence.
  - ○ It is made to simplify the computations involved and, in this sense, is considered "na¨Ive."

- It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.
- Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.
- There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle : all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable.
- For example, an object may be considered to be a ball if it is red, round, and about 7 cm in diameter.
- A naive Bayes classifier considers each of these features to contribute independently to the probability that this object is a ball , regardless of any possible correlations between the color, roundness, and diameter features.
- Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector

$$x = (x_1, \ldots.., x_n)$$

representing some n features (independent variables), it assigns to this instance probabilities

$$p(C_k | x_1, \ldots.., x_n)$$

for each of K possible outcomes or classes C_k.

- The problem with the above formulation is that if the number of features n is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible.
- The model is reformulated to make it more tractable.
- Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k | x) = \frac{p(C_k)\, p(x | C_k)}{p(x)}$$

- In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{Prior} \times \text{Likelihood}}{\text{Evidence}}$$

- Effectiveness of Bayesian classifiers -
  - Various empirical studies of this classifier in comparison to decision tree and neural network classifiers have found it to be comparable in some domains.

®

- In theory, Bayesian classifiers have the minimum error rate in comparison to all other classifiers.
- In practice this is not always the case. The inaccuracies are due to assumptions made for its use, such as class-conditional independence, and the lack of available probability data.
- Bayesian classifiers are also useful in theoretical justification for other classifiers that do not explicitly use Bayes' theorem.
- For example, under certain assumptions, many neural network and curve-fitting algorithm output the maximum posteriori hypothesis, like the na¨Ive Bayesian classifier.

3.4 RULE BASED CLASSIFICATION

- 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 R,

  R: IF age > 30 AND credit = good

  THEN buy_computer = yes

- Rule R can also be written as as follows -

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

- Points to remember while applying rule -
  - 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.
- Rule Extraction from a Decision Tree
  - Rule Extraction is to build a rule-based classifier by extracting IF-THEN rules from a decision tree.
  - 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.
    - In this algorithm, each rule for a given class covers many of the tuples of that class.

®

        - No decision tree is generated.
    - 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.
    - The Decision tree induction can be considered as learning a set of rules simultaneously.
- Algorithm for Sequential Covering
    - The Following is the sequential learning Algorithm where rules are learned for one class at a time.
    - While learning a rule from a class Ci, a rule should cover all the tuples from class C only and no tuple form any other class.

Algorithm : Sequential Covering

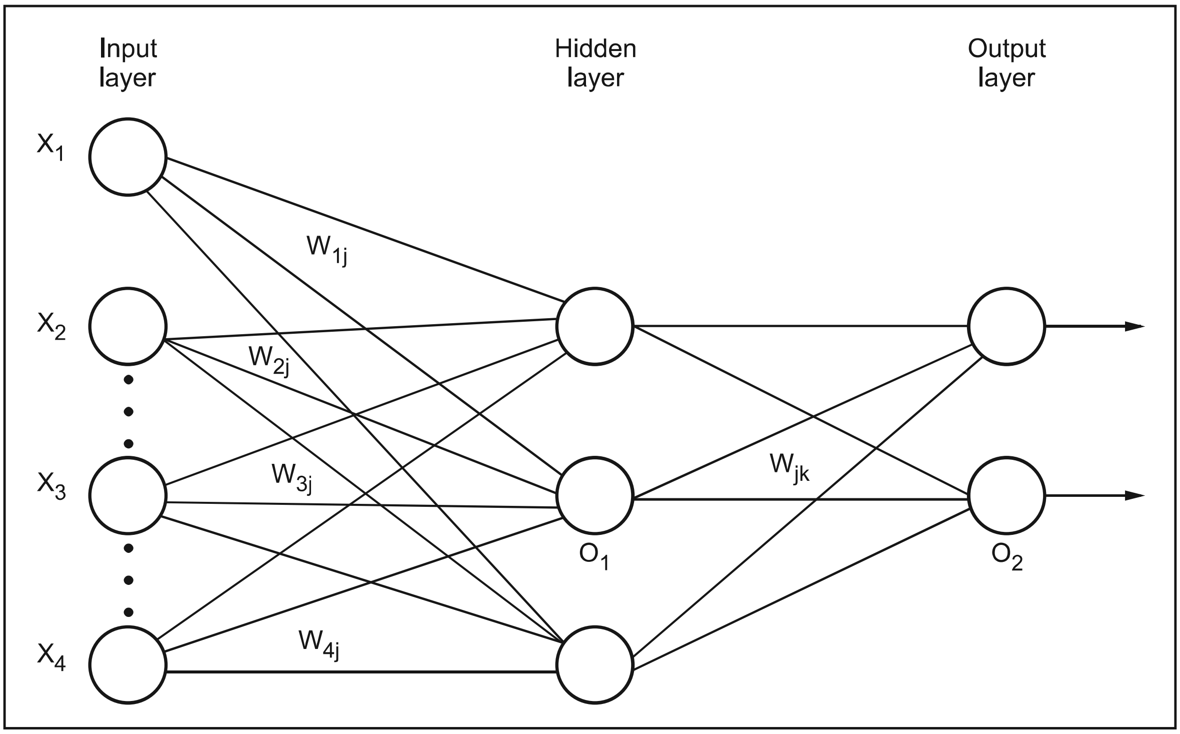
```
Input :
D, a data set class-labeled tuples,
Att_vals, the set of all attributes and their possible values.
Output :  A Set of IF-THEN rules.
Method :
Rule_set={ }; // initial set of rules learned is empty
for each class c do
   repeat
      Rule = Learn_One_Rule(D, Att_valls, c);
      remove tuples 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(First, Outer, Inner, Last) 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.
- ○ 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.

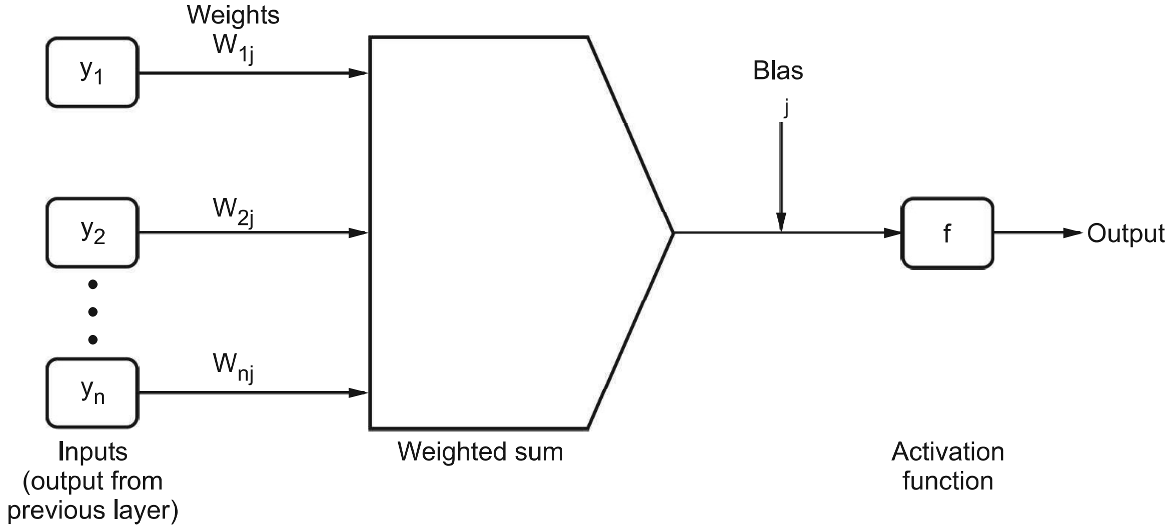
3.5 CLASSIFICATION BY BACKPROPOGATION

- The BackPropagation (BP) algorithm learns the classification model by training a multilayer feed-forward neural network.
- The generic architecture of the neural network for BP is shown in the following diagrams, with one input layer, some hidden layers, and one output layer.
- Each layer contains some units or perceptron. Each unit might be linked to others by weighted connections.
- The values of the weights are initialized before the training.
- The number of units in each layer, number of hidden layers, and the connections will be empirically defined at the very start.
- The back propagation algorithm performs learning on amultilayer fee-forward neural network.



Backpropagation using Multilayer feedforward Network

- The inputs correspond to the attributes measured for each raining sample. The inputs are fed simultaneously into layer of units making up the input layer.
- The weighted outputs of these units are, in turn, fed simultaneously to a second layer of neuron like units, known as a hidden layer.
- The hidden layer s weighted outputs can be input to another hidden layer, and so on. The number of hidden layers is arbitrary, although in practice, usually only one is used.
- The weighted outputs of the last hidden layer are input to units making up the output layer, which emits the network's prediction for given samples.
- The units in the hidden layers and output layer are sometimes referred to as neurodes, due to their symbolic biological basis, or as output units.
- Multilayer feed-forward networks of linear threshold functions, given enough hidden units, can closely approximate any function.
- Backpropagation
  - Back propagation learns by iteratively processing a set of training samples, comparing the network's prediction for each sample with the actual known class label.
  - For each training sample, the weights are modified so as to minimize the mean squared error between the network's prediction and the actual class.
  - These modifications are made in the "backwards" direction, that is , form the output layer through each hidden layer down to the first hidden layer (hence the name backpropagation).
  - Although it is not guaranteed in general the weights will eventually converge, and the learning process stops.



Back propagation

®

- The Backpropagation algorithm

  Initialize the weights.

  The weights in the network are initialized to small random number(e.g., ranging from – 1.0 to 1.0, or – 0.5 to 0.5).

  Initialize the biases to small random numbers.

  For Each training sample: X, is processed by the following steps.

    Perform Feed-forward computation to reduce error

    Back propagation to the output layer

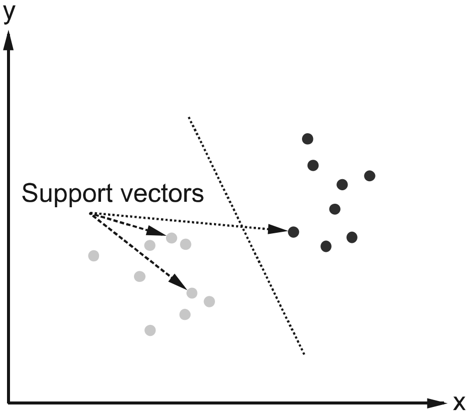
    Back propagation to the hidden layer

    Weights are adjusted / updated

  The algorithm is stopped when the value of the error function has become sufficiently small.

3.6 SUPPORT VECTOR MACHINES

- The Support-Vector Machines (SVMs) algorithm, created by Hava Siegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data,
- Support Vector Machines (SVMs), is one of the most widely used clustering algorithms in industrial applications.
- Support Vector Machines (SVMs) is a discriminative classifier formally defined by a separating hyperplane.
- In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.
- In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.
- Support Vector Machines (SVMs), a method for the classification of both linear and nonlinear data.
- Support Vector Machines (SVMs) is mostly used in classification problems.
- In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (Refer Fig. 4.1.5).



Support Vector Machines

- SVM searches for the hyperplane with the largest margin, that is, the Maximum Marginal Hyperplane (MMH). The associated margin gives the largest separation between classes.
- Support Vectors are simply the co-orinates of individual observation. Support Vector segregates the two classes (hyper-plane/ line).
- An SVM is an algorithm that works as follows.
  - SVM uses a nonlinear mapping to transform the original training data into a higher dimension.
  - Within this new dimension, it searches for the linear optimal separating hyperplane (i.e., a "decision boundary" separating the tuples of one class from another).
  - With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane.
  - The SVM finds this hyperplane using support vectors ("essential" training tuples) and margins (defined by the support vectors).
- In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.
- Parameter selection - The effectiveness of SVM depends on the selection of kernel, the kernel's parameters, and soft margin parameter C
- SVMs can be used to solve various real-world problems :
  - SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.
  - Classification of images can also be performed using SVMs. SVM is also used in image segmentation
  - Hand-written characters can be recognized using SVM.

  - The SVM algorithm has been widely applied in the biological and other sciences.
- Drawbacks of Support Vector Machines (SVMs)
  - It requires full labeling of input data
  - It has uncalibrated class membership probabilities-SVM avoids estimating probabilities on finite data
  - The SVM is only directly applicable for two-class tasks. Therefore, algorithms that reduce the multi-class task to several binary problems have to be applied
  - In SVMs, the parameters of a solved model are difficult to interpret.
- The SVMs are extended to achieve higher classification results. These are
  - Support-Vector Clustering (SVC)
    SVC is a similar method that also builds on kernel functions but is appropriate for unsupervised learning. It is considered a fundamental method in data science.
  - Multiclass SVM
    Multiclass SVM aims to assign labels to instances by using support-vector machines, where the labels are drawn from a finite set of several elements.

3.7 ASSOCIATIVE CLASSIFICATION IN DATA MINING

Data mining is an effective process that includes drawing insightful conclusions and patterns from vast amounts of data. Its importance rests in the capacity to unearth buried information, spot trends, and make wise judgments based on the information recovered.

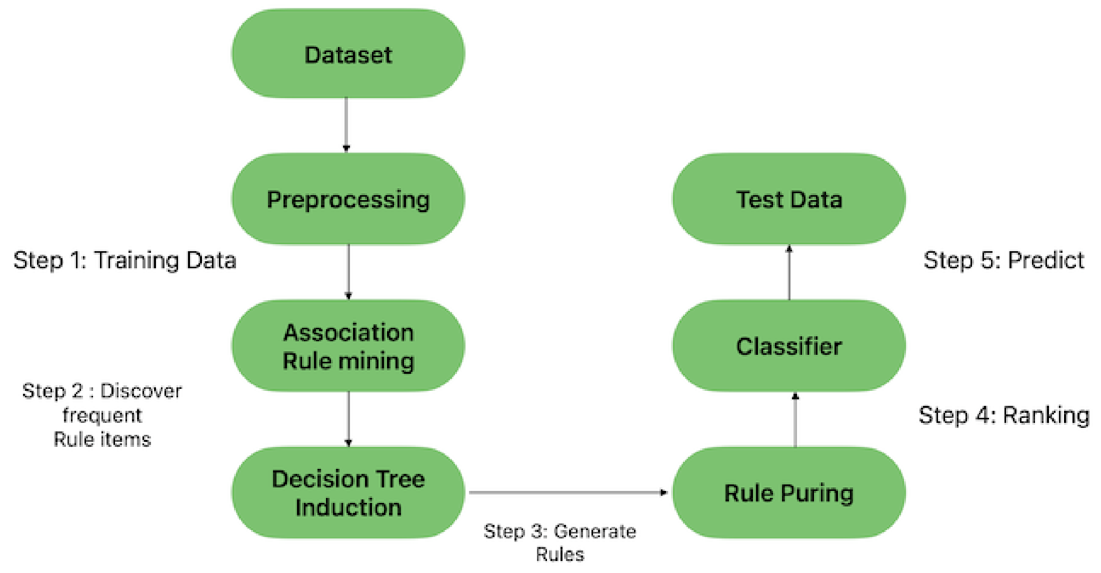
A crucial data mining approach called associative classification focuses on identifying connections and interactions between various variables in a dataset. Its goal is to find relationships and patterns among qualities so that future events can be predicted or new occurrences can be categorized. Associative categorization can be used to uncover useful patterns that help businesses and organizations better understand their data, make data−driven choices, and improve their operations.

This method offers a thorough framework to identify intricate linkages in data, resulting in insightful information and prospective advancements in a range of industries, including marketing, finance, healthcare, and more.

Understanding Associative Classification

Understanding associative classification is essential for realizing its full potential in data mining. Making prediction or classification jobs easier, entails identifying correlations and links between attributes in a collection. The fundamental goal of associative classification is to identify patterns connecting different variables by using association rule mining techniques.

Rule creation, rule assessment, and rule selection are generally the three main steps in the process. When rules are developed, they are based on the dataset, however, when rules are evaluated, they are evaluated for quality and importance. In order to improve the accuracy and relevance of the classification process, rule selection seeks to weed out unimportant or inapplicable rules. A few benefits of associative categorization are its capacity to manage complicated data linkages, manage high−dimensional datasets, and give comprehensible rules. The computational complexity of big datasets, sensitivity to noise and irrelevant features, and a possible trade−off between accuracy and interpretability are some of its drawbacks. Nevertheless, being aware of these factors enables data analysts to employ associative categorization efficiently and base choices on the discovered patterns.



Associative classification in Data mining

Techniques and Algorithms

Apriori Algorithm and Its Role in Associative Classification

In associative classification, the Apriori algorithm is a key method that is essential for identifying popular item sets. The method finds itemsets that meet a minimal support criterion via an iterative technique, creating strong correlations between qualities. Its main function in associative categorization is to produce a set of frequent item sets from which association rules may be derived.

Utilizing the "apriori property," which stipulates that any non−frequent itemset must have non−frequent subsets, the method effectively prunes the search space.

Fuzzy Association Rule Mining and Its Applications

A development of conventional association rule mining that addresses ambiguity and imprecision in data is fuzzy association rule mining. In datasets where characteristics include degrees of membership rather than binary values, it enables the discovery of relationships.

In fields like medical diagnosis or consumer behavior research, where ambiguity and vagueness are common, fuzzy association rule mining is very helpful. This method uses fuzzy logic to generate rules and identify correlations, allowing for more informed decision−making and the identification of patterns in large datasets.

Evaluation and Validation

Metrics of association rules

To assess the value and importance of association rules produced by associative classification, many metrics are used. The metrics lift, support, and interestingness are frequently employed. The potency of connections, the accuracy of forecasts, and the applicability of the patterns found are all quantified by these measures.

Different algorithms for associative classification.

1. CBA (Classification Based on Association) :

    It's a iterative approach of frequent itemset mining.

    Frequent itemsets are found after multiple passes.

    No. of passes made = length of longest rule obtained

    Uses heuristic method for construction of classifier.

    In case rules have same antecedent, highest confidence rule is selected.

    Decision list in formed for the set of rules forming a classifier.

    It shows more accuracy then C 4.5

2. CMAR (Classification Based on Multiple Association Rule)

    It uses variant of FP-growth algorithm for finding complete set of rules, satisfying minimum confidence and minimum support threshold.

    FP-tree is used to record all frequent itemset information is D in two scans.

    To find strongest group of rules it uses X 2 measure.

    Run time scalability and use of memory of CMAR is more efficient.

3. CPAR (Classification Based on Predictive Association Rules) :

    It uses classification called as FOIL, to build the rules to identify positive tuples from negative.

    In case of multi class problem, FOIL is applied to each class.

    Every time with generation of rule the positive tuples in data set are covered or removed.

    It uses best k rules of each group for prediction of class label of X.

3.8 LAZY LEARNERS (Learning from Your Neighbors)

Understanding Eager Learners

- The classification methods discussed in the earlier sections are examples of eager learners.
- Eager learners, when given a set of training tuples, construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify
- Examples of Eager Learners are
  - Decision tree induction,
  - Bayesian classification,
  - Rule-based classification
  - Classification by backpropagation,
  - Support vector machines
  - Classification based on association rule mining

Understanding Lazy Learners

- A lazy learner delays abstracting from the data until it is asked to make a prediction
- Lazy Learners
- Simply Stores the training data without doing any further processing on it, till it gets the next test set.
- It's slow as it calculates based on the current data set instead of coming up with an algorithm based on historic data
- It has large localized data so generalization takes time at every iteration
- Lazy learners do less work when a training tuple is presented and more work when making a classification or numeric prediction.
- Because lazy learners store the training tuples or "instances," they are also referred to as instance-based learners, even though all learning is essentially based on instances.
- While performing classification or numeric prediction, lazy learners can be computationally expensive.
- They require efficient storage techniques and are well suited to implementation on parallel hardware.
- They offer little explanation or insight into the data's structure. Lazy learners, however, naturally support incremental learning.
- They are able to model complex decision spaces having hyperpolygonal shapes that may not be as easily describable by other learning algorithms
- Examples of lazy learners :
  - K-nearest-neighbor classifiers
  - Case-based reasoning classifiers

Key differences in Eager Learners and Lazy Learners are

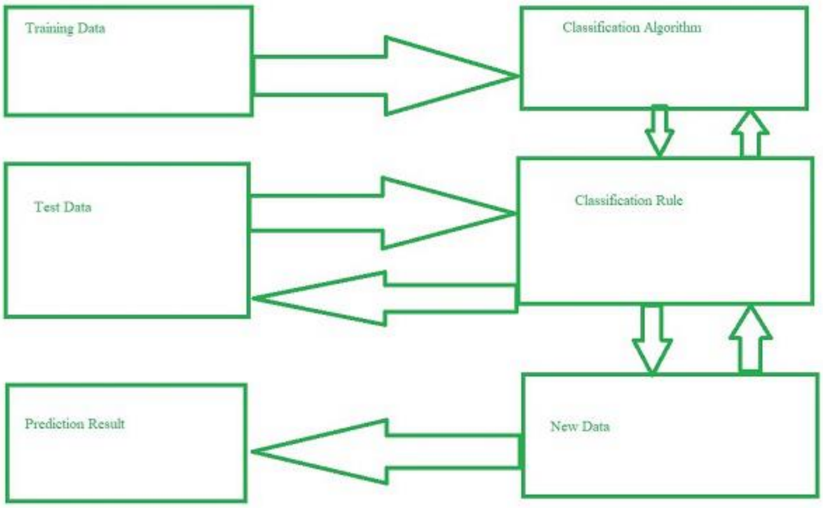
Aspect	Eager Learner	Lazy Learner
Technique	Eager learning (eg. Decision trees, SVM, NN) : Given a set of training set, constructs a classification model before receiving new (e.g., test) data to classify	Lazy learning (e.g., instance-based learning) : Simply stores training data (or only minor processing) and waits until it is given a test tuple
Accuracy	Eager : must commit to a single hypothesis that covers the entire instance space	Lazy method effectively uses a richer hypothesis space since it uses many local linear functions to form its implicit global approximation to the target function
Approach	1 k-nearest neighbour approach- Instances represented as points in a Euclidean space. 2 Locally weighted regression - Constructs local approximation	Store training examples and delay the processing ("lazy evaluation") until a new instance must be classified

3.9 Prediction in Data Mining

To find a numerical output, prediction is used. The training dataset contains the inputs and numerical output values. According to the training dataset, the algorithm generates a model or predictor. When fresh data is provided, the model should find a numerical output. This approach, unlike classification, does not have a class label. A continuous-valued function or ordered value is predicted by the model.

In most cases, regression is utilized to make predictions. For example: Predicting the worth of a home based on facts like the number of rooms, total area, and so on.

Consider the following scenario: A marketing manager needs to forecast how much a specific consumer will spend during a sale. In this scenario, we are bothered to forecast a numerical value. In this situation, a model or predictor that forecasts a continuous or ordered value function will be built.



Prediction Issues:

Preparing the data for prediction is the most pressing challenge. The following activities are involved in data preparation:

- **Data Cleaning:** Cleaning data include reducing noise and treating missing values. Smoothing techniques remove noise, and the problem of missing values is solved by replacing a missing value with the most often occurring value for that characteristic.
- **Relevance Analysis:** The irrelevant attributes may also be present in the database. The correlation analysis method is used to determine whether two attributes are connected.
- **Data Transformation and Reduction:** Any of the methods listed below can be used to transform the data.

  Normalization: Normalization is used to transform the data. Normalization is the process of scaling all values for a given attribute so that they lie within a narrow range. When neural networks or methods requiring measurements are utilized in the learning process, normalization is performed.

  Generalization: The data can also be modified by applying a higher idea to it. We can use the concept of hierarchies for this.

Other data reduction techniques include wavelet processing, binning, histogram analysis, and clustering.

3.10 MODEL EVALUATION AND SELECTION

Any project using machine learning must place a significant emphasis on selecting and model evaluation in order to be successful. We will assess how well our model works at this point in the procedure. In order to establish the further measures that need to be taken to improve this model, we conduct an analysis of more in-depth performance indicators. If you ignore this step, your model's performance will likely not be as good as it may be. To begin the process of increasing our model's accuracy on our dataset from 65% to 80% or 90%, we must first understand what our model predicts accurately and incorrectly. For better understanding, you can check out data science online certification courses to learn more about understanding and optimizing the model's features and parameters to achieve the desired accuracy.

Model Selection is the process of deciding which learning technique to use to model our data; for example, while attempting to solve a classification issue, we may consider using Logistic Regression, Support Vector Machines, trees, and other methods. It is also necessary to make choices on the degree of linear regression techniques while solving a regression problem.

Model Evaluation is a process to ascertain how well our model performs on a dataset it has not seen (its generalization capabilities). During the evaluation, a model's ability to perform well on various metrics such as accuracy, precision, recall, F1-score, and AUC-ROC is assessed by testing how well it can generalize to new data.

Model Selection

For regression and classification, we have spoken about how to fit models by minimizing error or maximizing likelihood given a dataset (also referred to as "*training data*"). This works well if we need to utilize our model for inferential or explanatory purposes. Or when we employ less adaptable models like linear regression or logistic regression.

However, when our focus moves from descriptive analysis to predictive modeling using methods like non-linear regression, Tree-based approaches, and Support Vector Machines, this is frequently no longer enough.

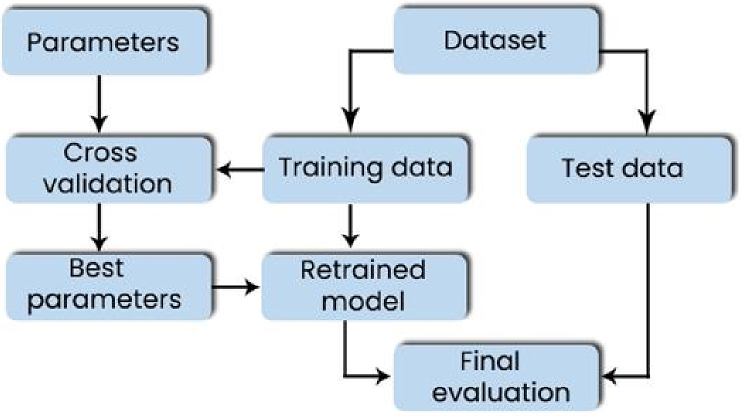
Our objective is to construct models that are generalizable beyond the available data set. It is easy for complex models to overfit our training data, in which case we don't get any insight into the population from which we sampled our training data. We state that our goal is to "*learn*" or "*train*" models that can extrapolate previously unknown data from the same population beyond what was included in the training set.

As you may see, this creates a little bit of a problem. What metrics can we use to evaluate a model's predictive capability when we can only use training data?

In-CrosValidation

Cross-validation is often used as a benchmark for measuring how well a model generalizes to new data. It plays a role in two crucial steps of data analysis: model selection and evaluation.

Model Complexity Selection is the process of deciding what kind of model to use. Take a look at this illustration of linear regression: What variables, interactions, and data transformations should I account for when fitting a linear regression model? Which depth of classification tree to employ is another illustration.



The error rate in a test may be estimated by cross-validation because it is a resampling technique (or any other performance measure on unseen data). Sometimes you'll have access to a sizable predetermined test dataset you should never use for training. If you don't have access to a dataset like this, cross-validation can help.The purpose of Model Assessment is to evaluate the model's overall performance. Take the linear regression models I constructed, where I used a limited set of variables as predictors. If we give it some data we haven't seen before, how well will it do? In the same vein, how can a classification tree answer the same query? (of specific depth).

Any given model can be evaluated with the validation set. However, this is for regular evaluations. As engineers specializing in machine learning, we use this information to adjust the model's hyperparameters. Therefore, the model occasionally encounters this information but never uses it to "*Learn*." Higher-level hyperparameters are updated based on the findings from the validation set. Therefore, a model is indirectly affected by the validation set. You may also hear the term "*Dev set*" or "*development set*" used to refer to the validation set. This makes sense, as the dataset is helpful during the model's "development" phase.

The Test dataset serves as the benchmark against which the model is tested. Only once a model has been fully trained is it put to use (using the train and validation sets). To compare several models, the "test set" is typically employed (For example, on many Kaggle competitions, the validation set is released initially along with the training set, and the actual test set is only released when the competition is about to close, and it is the result of the model on the Test set that decides the winner). However, utilizing the validation set as the test set is not recommended. There is a high standard of curation across the test set. As such, it includes representative samples of data from all relevant classes that the model would encounter in practice. All these continuous characteristics are a requirement for many different types of data mining project ideas in the real world.

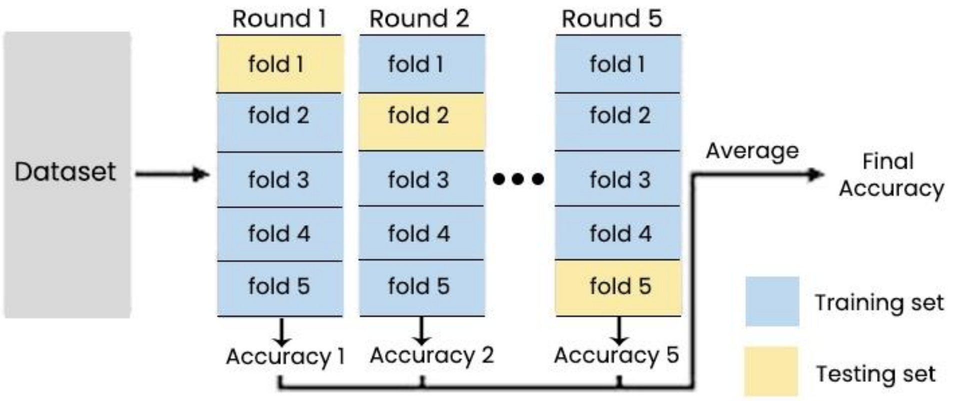
Now that you understand the functions of these datasets, you may seek guidance on how to divide your data into a Train, Validation, and Test set.
There are two primary factors at play here. First, the model you are training, and second, the total amount of samples in your data.
Some models require enormous amounts of data to train, in which case you should prioritize the more extensive training sets. For models with few hyperparameters, a smaller validation

set may suffice. Still, if your model has numerous hyperparameters, a larger validation set is recommended (although you should also consider cross-validation).
Further, a validation set is unnecessary if your model has no hyperparameters or ones that are difficult to adjust.
Overall, the train-test-validation split ratio, like many other aspects of machine learning, is very application-dependent, and it becomes simpler to make judgments as more and more models are trained and built.

There may be sampling problems with this method. Since the error rate is a random quantity that is influenced by the number of observations in both the training and validation sets, it can fluctuate widely. By averaging together several readings, we may get a more accurate approximation of the test error (remember the law of large numbers). To achieve this, we will run our validation resampling procedure 10 times (using new validation and training sets each time) and take the mean of the test errors.

This method is not without its flaws. In our validation strategy, we exaggerate errors since each training set only uses half of the data to train. This means that our models may not perform as well as those trained with the complete dataset. It is possible to rethink and apply our strategy to this problem thoroughly. Create individual validation sets for each training point.
Method: (a) for each observation I in the data set,
(b) predict the reaction to the i-th observation,
(c) calculate the prediction error.
The resulting error estimate from the cross-validation procedure is as follows.

CV(n)=1n1(i-i)2



The data set is shuffled and then divided into k groups at random to implement the cross-validation method. Iterating over each subset requires treating that subset as a test set while combining the remaining subsets into a single training set. A test group is used to validate the model, and this procedure is repeated k times.

By the end of the procedure, k distinct sets of data have been collected from k distinct test populations. Choosing the model with the greatest score makes it easy to determine which one is the best. Our Data science tutorial will help you to explore the world of data science and prepare to face the challenges.

Model Evaluation

A variety of measures may be used to assess the quality of a model. However, selecting an appropriate metric for assessment is typically difficult and is highly dependent on the nature of the problem being handled. The evaluator's ability to quickly and accurately match the issue statement with an acceptable statistic depends on his or her thorough familiarity with a wide variety of metrics.

Consider two hypotheses, h(x) and h', for performing a given task: (x). We wouldn't have any way to compare which one is superior. To put it simply, we could adopt the following measures at a high level:

1. Evaluate how well both ideas predict reality.
2. Find out if there is a statistically significant difference between the two outcomes.

If there are competing hypotheses, choose the one with the best results. If not, we can't conclude whether h(x) or h'(x) is superior using statistical methods.

When evaluating a model's performance on a classification job, we'll look at how well it can correctly place instances into classes. Think about it in binary terms. Our school has two different types of students: 1st and 0th. A successful prediction would be one in which the model correctly identifies an instance as belonging to either Class 1 or Class 0. A table outlining all the potential outcomes of our model may be constructed if we treat our 1 class as the "Positive class" and our 0 class as the "*Negative class*."

Data Science Training

- Personalized Free Consultation
- Access to Our Learning Management System
- Access to Our Course Curriculum
- Be a Part of Our Free Demo Class

Accuracy

The accuracy measure is the easiest to define; it is just the fraction of test instances that were labeled correctly. While it is applicable to a wide variety of general situations, its use is limited when dealing with imbalanced data sets. When investigating bank data for fraudulent activity, the ratio of fraudulent to non-fraudulent instances might be as high as 1:99. A 99% accurate model in this situation would correctly identify all test instances as not being fraudulent. There will be no utility for a model that is just 99% correct.A model will fail to capture 10 instances of fraud if it is inadequately trained to predict that all 1000 data points are not frauds. Measures of accuracy reveal that the model makes 990 out of 1000 correct predictions, giving it an accuracy of (990/1000)*100 = 99%.

Since the model failed to account for 10 key indicators of fraud, we need a metric that can zero in on these red flags. Because of this, accuracy is not a reliable measure of a model's efficacy.

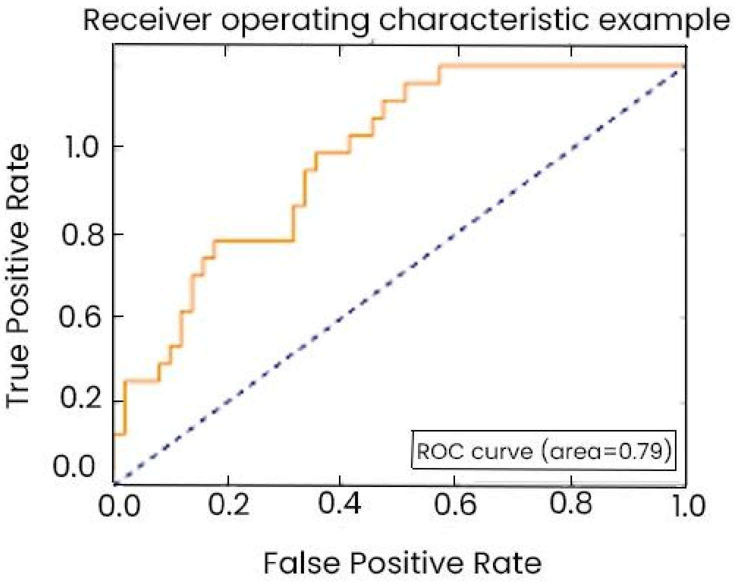
Precision

In order to determine whether or not a categorization is accurate, precision is employed as a measure.This formula is the proportion of true positive classifications relative to the sum of all true positive predictions. The higher the accuracy or the model's propensity to accurately identify the positive class, the higher the proportion.The necessity for exactness is seen in the predictive maintenance problem (determining when a machine will break down and arranging for repairs in advance). Incorrect projections may be costly for a business because of the high price of upkeep. The model's accuracy in identifying members of the positive class and its capacity to reduce false positives are of the utmost importance in such circumstances.

The percentage of positive instances that were properly detected, or recall, is expressed as a percentage of the total number of positive cases.Regarding the fraud issue, again, the recall value may prove to be rather helpful in situations of fraud, as a high recall value will imply that a large percentage of frauds were successfully discovered.

In order to strike a proper balance between the two factors, Recall and Precision, the F1 score is the harmonic mean of the two.It helps when you need to remember something and be precise about it, as when trying to figure out which plane pieces need fixing. Here, accuracy is needed to cut costs for the business (plane components are quite pricey), and recall is essential to ensure the machinery is safe and doesn't endanger people.

The ROC curve represents a relationship between the proportion of correct diagnoses (recall) and the proportion of incorrect diagnoses (false positives, or TN/(TN+FP)). Area Under the Receiver Operating Characteristics (AUC-ROC) measures how well a model predicts actual results.



The model predicts the output variable randomly if the curve is close to the 50% diagonal line.