CHAPTER 5

CLASSIFICATION ALGORITHMS

5.1 Classification Procedure

Classification is used to find out in which group each data instance is related within a given dataset. It is used for classifying data into different classes according to some constrains. Several major kinds of classification algorithms including C4.5, ID3, k-nearest neighbour classifier, Naive Bayes, Support Vector Machine, and Artificial Neural Network are used for classification. Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on the training set and class labels, and it can be used for classifying newly available data. Classification procedure is a recognized method for repeatedly making such decisions in new situations. Here it is assumed that problem is a concern with the construction of a procedure that will be applied to a continuing sequence of cases in which each new case must be assigned to one of the sets of predefined classes by observed features of data. Creation of a classification procedure from a set of data for which the exact classes are known in advance is termed as pattern recognition or supervised learning. Contexts in which a classification task is fundamental which includes, for example, assigning individuals to credit status by financial and other personal information, and the initial diagnosis of a patient's disease in order to select immediate treatment while awaiting perfect test results. Some of the most critical problems arising in science, industry, and commerce can be called classification or decision problems. Three main historical strands of research can be identified: statistical, machine learning and neural network. All groups have some objectives in common. They have all attempted to develop procedures that would be able to handle a wide variety of problems and to be extremely general used in practical settings with proven success. Classification refers to the task of assigning objects to one of the various predefined categories, is a determined problem that encompasses many different applications. The examples include detecting spam e-mail messages based upon the message header and content and classifying galaxies based on their shapes. Classifiers are used to enhance the performance of given data sets. To construct or to train a classifier is the process of creating a function or data structure that will be used for determining the missing value of the class attribute of the new unclassified instances. There are large numbers of learning schemes for classification and regression numeric prediction - like decision trees, Instance-based classifiers, support vector machines, Bayes decision schemes, neural networks and the like. Numerous attribute selection methods and evaluation methods exist like cross-validation and bootstrapping, and preprocessing techniques.

Two main phases of work on classification can be identified within the statistical community. The first "classical" phase concentrated on extension of Fisher's early work on linear discrimination. The second, "modern" phase concentrated on more flexible classes of models many of which attempt to provide an estimate of the joint distribution of the features within each class which can, in turn, provide a classification rule (*Michies et al.*, 1994). Statistical procedures are characterized by having a precise fundamental probability model which provides a probability of being in each class instead of just a classification. Also, it is usually assumed that the techniques will be used by statisticians and hence some human involvement is assumed about variable selection and transformation and overall structuring of the problem.

5.2 Classification techniques

(i) Machine Learning Based Approach

Machine Learning is covered automatic computing procedures based on logical or binary operations that learn a task from a series of examples. Here the concentration is on classification, and so attention has focused on decision-tree approaches in which classification results from a sequence of logical steps. These classification results are capable of representing the most complex problem given sufficient data. Other techniques such as genetic algorithms and Inductive Logic Procedures (ILP) are currently under active improvement, and its principle would allow us to deal with more general types of data including cases where the number and type of attributes may vary. Machine Learning approach aims to generate classifying expressions simple enough to be understood easily by the human and must mimic human reasoning sufficiently to provide insight into the decision process (*Michies et al.*, 1994). Like statistical approaches, background knowledge may be used in development but operation is assumed without human interference.

(ii) Neural Network

The field of Neural Networks has arisen from diverse sources ranging from understanding and emulating the human brain to broader issues of copying human abilities such as speech and can be used in various fields such as banking, legal, medical, news, in classification programme to categorize data as intrusive or normal. Neural networks consist of layers of interconnected nodes where each node producing a non-linear function of its input and input to a node may come from other nodes or directly from the input data. Also, some nodes are identified with the output of the network.

By this example, there are different applications for neural networks that involve recognizing patterns and making simple decisions about them. In airplanes, one can use a neural network as a basic autopilot where input units read signals from the various cockpit instruments and output units modifies the plane's controls appropriately to keep it safely on course. Inside a factory, we can use a neural network for quality control.

5.3 Classification Algorithms

Classification is one of the data mining techniques that is mainly used to analyze a given data set and takes each instance of it and assigns this instance to a particular class such that classification error will be least. It is used to extract models that accurately define important data classes within the given data set. Classification is a two step process. During the first step, the model is created by applying classification algorithm on training data set then in the second step the extracted model is tested against a predefined test data set to measure the model trained performance and accuracy. So classification is the process to assign class label from a data set whose class label is unknown.

(i) ID3 Algorithm

ID3 calculation starts with the original set as the root hub. On every cycle of the algorithm, it emphasizes through every unused attribute of the set and figures the entropy (or data pick up IG (A)) of that attribute. At that point chooses the attribute which has the smallest entropy (or biggest data gain) value. The set is S then split by the selected attribute (e.g. marks < 50, marks < 100, marks >= 100) to produce subsets of the information. The algorithm proceeds to recurse on each and every item in the subset and considering only items never selected before. Recursion on a subset may bring to a halt in one of these cases:

- Every element in the subset belongs to the same class (+ or -), then the node is turned into a leaf and labeled with the class of the examples.
- If there are no more attributes to be selected but the examples still do not belong to the same class (some are +, and some are -), then the node is turned into a leaf and labeled with the most common class of the examples in that subset.
- If there are no examples in the subset, then this happens when parent set found to be matching a specific value of the selected attribute. For example, if there was no example matching with marks >=100 then a leaf is created and is labeled with the most common class of the examples in the parent set.

Working steps of algorithm is as follows,

- Calculate the entropy for each attribute using the data set S.
- Split the set S into subsets using the attribute for which entropy is minimum (or, equivalently, information gain is maximum)
- Construct a decision tree node containing that attribute in a dataset.
- Recurse on each member of subsets using remaining attributes.

(ii) C4.5 Algorithm

C4.5 algorithm is used to produce a decision tree which is an expansion of prior ID3 calculation. It enhances the ID3 algorithm by managing both continuous and discrete properties, missing values and pruning trees after construction. The decision trees created by C4.5 can be used for grouping and often referred to as a statistical classifier. C4.5 creates decision trees from a set of the training data same way as the Id3 algorithm. As it is a supervised learning algorithm, it requires a set of training examples which can be seen as a pair: input object and the desired output value (class). The algorithm analyzes the training set and builds a classifier that must have the capacity to accurately arrange both training and test cases. A test example is an input object, and the algorithm must predict an output value. Consider the sample training data set $S = S_1, S_2 \ldots S_n$ which is already classified. Each sample S_i consists of feature vector $(x_{1,\ i}, x_{2,\ i} \ldots x_{n,\ i})$ where x_j represent attributes or features of the sample and the class in which S_i falls. At each node of the tree, C4.5 selects one attribute of the data that most efficiently splits its set of samples into subsets such that it results in one class or the other. The splitting condition is the normalized information gain (difference in entropy) which is a non-symmetric measure of the difference between two probability distributions P and Q. The attribute with the highest information gain is chosen to make the decision (Chandradeep Bhatt, 2014). General working steps of algorithm is as follows,

- Assume all the samples in the list belong to the same class. If it is true, it simply creates a leaf node for the decision tree, so that particular class will be selected.
- None of the features provide any information gain. If it is true, C4.5 creates a decision node higher up the tree using the expected value of the class.
- An instance of previously-unseen class encountered. Then, C4.5 creates a decision node higher up the tree using the expected value.

(iii) C.K Nearest Neighbors Algorithm

The closest neighbor (CN) rule distinguishes the classification of an unknown data point by its closest neighbor whose class is already known. M. Cover and P.E. Hart purpose k nearest neighbor (KNN) in which nearest neighbor is computed by estimation of k that indicates how many nearest neighbors are to be considered to characterize the class of a sample data point. It makes utilization of the more than one closest neighbor to determine the class in which the given data point belongs to and consequently it is called as K-nearest neighbor. These data samples are needed to be in the memory at the run time, and hence they are referred to as memory-based technique. The training points are assigned weights according to their distances from sample data point. However, at the same time, the computational complexity and memory requirements remain the primary concern dependably. To overcome memory limitation size of data set is reduced. For this the repeated patterns which don't include additional data are also eliminated from training data set. To further enhance the information focuses which don't influence the result are additionally eliminated from training data set. The K-nearest neighbor implementation can be done using ball tree, k-d tree, nearest feature line (NFL), principal axis search tree and orthogonal search tree. The tree-structured training data is further divided into nodes and techniques like nearest feature line and tunable metric divide the training data set according to planes. Using these algorithms, one can expand the speed of basic K-nearest neighbor algorithm. Consider that an object is sampled with a set of different attributes. Assuming its group can be determined from its attributes; different algorithms can be used to automate the classification process.

```
$K$←number of nearest neighbors
    For each object $X$ in the test set do
    calculate the distance D($X$, $Y$) between $X$ and every object $Y$ in the
    training set
    neighborhood ← the $k$ neighbors in the training set closest to $X$
    $X$.class ← SelectClass (neighborhood)
End for
```

(iv) Support Vector Machine Algorithm (SVM Algorithm)

Support Vector Machine has attracted a great deal of attention in the last decade and actively applied to various domain applications. Support Vector Machines are typically used for learning classification, regression or ranking function. Support Vector Machine is based on statistical learning theory and structural risk minimization principle and has the aim of determining the location of decision boundaries which is known as the hyper plane that produces the optimal separation of classes (*Han et al.*, 2011), (Vapnik *et al.*, 1995) (Burges, 1998). Maximizing the margin and thereby creating the largest possible distance between the separating hyper plane and the instances on either side of it has been proven to reduce an upper bound on the expected generalization error (Vapnik *et al.*, 1998). The efficiency of Support Vector Machine based classification does not directly depend on the dimension of classified entities. Though Support Vector Machine is the most robust and accurate classification technique, there are several problems. The data analysis in Support Vector Machine is based on convex quadratic programming, and it is computationally expensive, as solving quadratic programming methods require large matrix operations as well as time-consuming numerical computations (Bhavsar et al., 2012). Training time for Support Vector Machine scales quadratic ally in the number of examples, so researchers strive all the time for more efficient training algorithm (Wang, 1998) resulting in several variant based algorithm.

SVM can also be extended to learn nonlinear decision functions by first projecting the input data onto a high-dimensional feature space using kernel functions and formulating a linear classification problem in that feature space (*Bhavsar et al.,* 2012). The resulting feature space is much larger than the size of the dataset which is not possible to store in popular computers. Investigation on this issues leads to several decomposition based algorithms. The basic idea of decomposition method is to split the variables into two parts: a set of free variables called working set, which can be updated in each iteration and set of fixed variables, which are fixed at a particular value temporarily. This procedure is repeated until the termination conditions are met.

Support Vector Machines realize the following idea: they map x ∈ $\mathbb{R}^n$ into a high (possibly infinite) dimensional space and construct an optimal hyper plane in this space. Different mappings x → Φ(x) ∈ H construct different SVMs. The mapping Φ(·) is performed by a kernel function K(·, ·) which defines an inner product in H. The decision function given by an SVM is thus:

$$F(x) = w \cdot \Phi(x) + b = \sum_i \alpha_i^0 y_i K(x_i, x) + b \qquad (5.1)$$

The optimal hyper plane is the one with the maximal distance (in H-space) to the closed image $\Phi(x_i)$ from the training data (called the maximal margin). This reduces to maximizing

$$W^2(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \qquad (5.2)$$

Under constraints $\sum_{i=1}^{\ell} \alpha_i y_i = 0$ and $\alpha_i \geq 0$, i = 1, ..., ℓ . For the non-separable case, one can quadratic ally penalizes errors with the modified kernel K ← K + $\frac{1}{\lambda} I$ where I am the identity matrix and λ a constant penalizing the training errors.

The size of the maximal margin is M and the images $\Phi(x_1)$ ... $\Phi(x_\ell)$ of the training vectors are within a sphere of radius R. Then the following holds true (Vapnik *et al.*, 1998).

If images of training data of size ℓ belonging to a sphere of size R are separable with the corresponding margin M, then the expectation of the error probability has the bound

$$EP_{err} \leq \frac{1}{\ell} E\left\{\frac{R^2}{M^2}\right\} = \frac{1}{\ell} E\left\{R^2 W^2(\alpha^0)\right\} \qquad \text{... (5.3)}$$

Where the expectation is taken over sets of training data of size ℓ .

This justifies the idea that the performance depends on the ratio E{R^2/M^2} and not simply on the large margin M, where R is controlled by the mapping function $\Phi(\cdot)$. Under the assumption that the set of support vectors does not change when removing the probability of test error

$$EP_{err}^{\ell-1} \leq \frac{1}{\ell} E \sum_{p=1}^{\ell} \Psi\left(\frac{\alpha_p^0}{(K_{SV}^{-1})_{pp}} - 1\right) \qquad (5.4)$$

Where Ψ is the step function, K_{SV} is the matrix of dot products between support vectors, $p_{err}^{\ell-1}$ is the probability of test error for the machine trained on a sample of size $\ell - 1$, and the expectations are taken over by the random choice of the sample. The support vector method attempts to find the function from the set f(x, w, b) = w · Φ(x) + b that minimizes generalization error. It has first to enlarge the set of functions considered by the algorithm to f(x, w, b, σ) = w · Φ(x * σ) + b. Note that the mapping $\Phi_\sigma(x) = \Phi(x * \sigma)$ can be represented by choosing the kernel function K_σ in equations (5.2) and (5.3):

$$K_\sigma(x, y) = K((x * \sigma), (y * \sigma)) = (\Phi_\sigma(x) \cdot \Phi_\sigma(y)) \qquad \text{... (5.5)}$$

For any K. Using equation (4.4) one minimizes over σ:

$$R^2W^2(\sigma) = R^2(\sigma)\, W^2(\alpha^{0,}\ \sigma) \qquad \text{...(5.6)}$$

where the radius R for kernel K_σ can be computed by maximizing (Vapnik, 1998):

$$R^2(\sigma) = \max_\beta \sum_i \beta_i K_\sigma(x_i, x_i) - \sum_{i,j} \beta_i \beta_j K_\sigma(x_i, x_j) \qquad \text{... (5.7)}$$

subject to $\sum \beta_i = 1$, $\beta_i \geq 0$, $i = 1, ..., \ell$, and $W^2(\alpha^0, \sigma)$ is defined by the maximum of functional (5.2) using kernel (5.5). In a similar way, one can minimize the span bound over σ instead of equation (5.6).

Find the minimum of R^2W^2 over σ requires searching over all possible subsets of n features which are a combinatorial problem. To avoid this problem, classical methods of search include greedily adding or removing features (forward or backward selection) and hill climbing. All these methods are expensive to compute if n is large.

As an alternative to these approaches, it is suggested the following method: approximate the binary valued vector $\sigma \in \{0, 1\}^n$, with a real-valued vector $\sigma \in \mathbb{R}^n$. Then, to find the optimum value of σ one can minimize R^2W^2, or some other differentiable criterion, by gradient descent. As explained in (*Chapelle, et al.*, 2000) the derivative of our criterion is:

$$\frac{\partial R^2 W^2(\sigma)}{\partial \sigma_k} = R^2(\sigma) \frac{\partial W^2(\alpha^0, \sigma)}{\partial \sigma_k} + W^2(\alpha^0, \sigma) \frac{\partial R^2(\sigma)}{\partial \sigma_k} \qquad \text{... (5.8)}$$

$$\frac{\partial R^2(\sigma)}{\partial \sigma_k} = \sum_i \beta_i^0 \frac{\partial K_\sigma(x_i, x_j)}{\partial \sigma_k} - \sum_{i,j} \beta_i^0 \beta_j^0 y_i y_j \frac{\partial K_\sigma(x_i, x_j)}{\partial \sigma_k} \qquad \text{... (5.9)}$$

$$\frac{\partial W^2(\alpha^0, \sigma)}{\partial \sigma_k} = -\sum_{i,j} \alpha_i^0 \alpha_j^0 y_i y_j \frac{\partial K_\sigma(x_i, x_j)}{\partial \sigma_k} \qquad \text{... 5.10)}$$

The minimum of $\tau(\sigma,\alpha)$ can be estimated by minimizing equation (5.6) in space $\sigma \in \mathbb{R}^n$ using the gradients (5.8) with the following extra constraint which approximate integer programming:

$$R^2W^2(\sigma) + \lambda\sum_i(\sigma_i)^p \qquad \text{... (5.11)}$$

Subject to $\sum_i \sigma_i$ = m, $\sigma_i \geq 0$, i = 1, ...,l.

For large enough λ as $p \to 0$ only m elements of σ will be non-zero, approximating optimization problem $\tau(\sigma,\alpha)$. One can further simplify computations by considering a step-wise approximation procedure to find m features. To do this, one can minimize $R^2W^2(\sigma)$ with σ unconstrained. One then sets the $q < n$ smallest values of σ to zero, and repeats the minimization until only m nonzero elements of σ remain. This can mean repeatedly training an SVM just a few times, which can be fast.

(v) Artificial Neural Network Algorithm (ANN Algorithm)

Artificial neural networks (ANNs) are types of computer architecture inspired by biological neural networks (Nervous systems of the brain) and are used to approximate functions that can depend on a large number of inputs and are unknown. Artificial neural networks are presented as systems of interconnected “neurons” which can compute values from inputs and are capable of machine learning as well as pattern recognition due to their adaptive nature.

An artificial neural network operates by creating connections between many different processing elements each corresponding to a single neuron in a biological brain. These neurons may be constructed or simulated by a digital computer system. Each neuron takes many input signals then based on an internal weighting produces a single output signal that is sent as input to another neuron. The neurons are strongly interconnected and organized into different layers. The input layer receives the input, and the output layer produces the final output. In general, one or more hidden layers are sandwiched in between the two. This structure makes it impossible to forecast or know the exact flow of data.

Artificial neural networks typically start out with randomized weights for all their neurons. This means that initially they must be trained to solve the particular problem for which they are proposed. A Back-propagation Artificial Neural Network is trained by humans to perform specific tasks. During the training period, one can evaluate whether the Artificial Neural Networks' output is correct or not by observing the pattern. If it is correct, the neural weightings that produced that output are reinforced; if the output is incorrect, those weightings responsible can be diminished.

Implemented on a single computer, an artificial neural network is normally slower than more traditional solutions of algorithms. The Artificial Neural Networks' parallel nature allows it to be built using multiple processors giving it a great speed advantage at very little development cost. The parallel architecture allows Artificial Neural Networks to process enormous amounts of data very efficiently in less time. When dealing with large continuous streams of information such as speech recognition or machine sensor data Artificial Neural Networks can operate considerably faster as compared to other algorithms. An artificial neural network is useful in a variety of real-world applications such as visual pattern recognition and speech recognition that deal with complex often incomplete data. Also, recent programs for text-to-speech has utilized Artificial Neural Networks.

5.4 Decision Tree Based Classification

Decision Tree maps observation to a conclusion in the form of target values. Decision tree separates input space of a data set into mutually exclusive regions. Each of which is assigned to a label, value or an action to characterize its data points. Two variants: J48 and Random Tree are discussed in subsections.

(i) J48 Algorithm

A decision tree is a graphical representation which consists of internal and external nodes connected by branches. An internal node is responsible for implementing decision functions that determine which node to visit next. The external node has no child node, and it is associated with a value that characterizes the given data which leads to its being visited. Decision tree construction algorithms involve a two-step process. First- growing, second-pruning. The growing process sets up a very large decision tree, and pruning process reduces the large size and over fitting of the data. The pruned decision tree that is used for classification is called classification tree. A J48 decision tree is a predictive machine-learning model that decides the target value (dependent variable) of a new sample based on various attribute values of the available data. The nodes of a J48 decision tree denote the different attributes. Branches between the nodes tell us the possible values that these attributes can have in the observed samples. To build a decision tree, one needs to calculate the entropy and information gain.

Entropy: $E(S) = \Sigma - p_i \log_2 p_i$ (Where Σ varies from i=1 to c)

Information Gain: Gain (T, X) = Entropy (T) – Entropy (T,X).

(ii) Random Tree

It is the simplest tree algorithm that comes as a result of the stochastic process. Random binary tree refers system model that selects random values over time. Two different distributions are commonly used for Random Tree formation.

1) Inserting nodes one at a time according to a random permutation.
2) Choosing from a uniform discrete distribution. Repeated splitting is another distribution to form the random tree. Adding and removing nodes directly in a random binary tree will disrupt its random structure. To balance the nodes of Random Trees, a random permutation is used for dynamic insertion and deletion of nodes.

5.5 Function Based Classification

It is a mathematical classification (or statistical procedure) to classify an instance in a particular class.

(i) Logistics Regression

Logistic Regression predicts the probability of an outcome that can only have two values. It performs a least-square fit of a parameter vector β to a numeric target variable X. The logistic regression uses equation: F(X) = βT. X to formulate prediction model. Where X is the input vector (a constant term to accommodate the intercept), and β is parameter vector to a numeric target variable. It is possible to use this technique for classification by directly fitting logistic regression models to class indication variables. The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of the binary variable for two reasons.

1. A linear regression will predict values outside the acceptable range (e.g. predicting probabilities outside the range 0 to 1)
2. Since the two value experiments can only have one of two possible values for each experiment, the residuals will not normally be distributed about the predicted line.

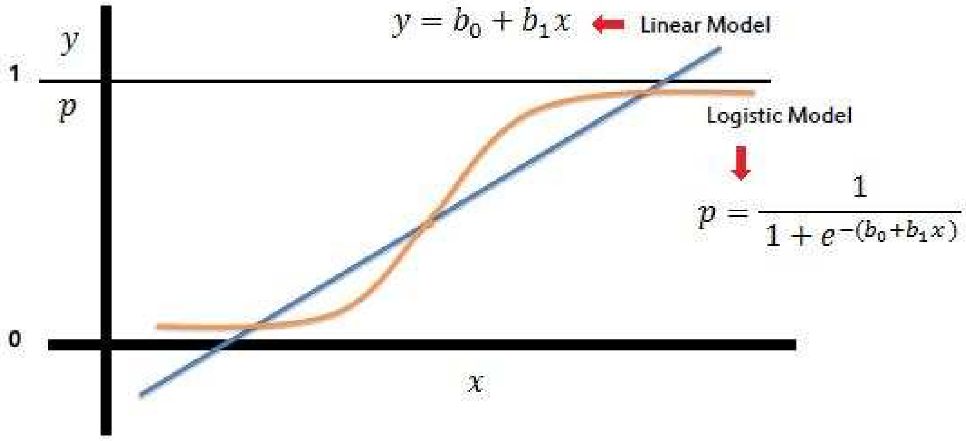

Figure 5.1 Logistic Regression

Figure 5.1 explains the logistic regression of two possible values of experiment.

Simply, the logistic regression equation can be written regarding an odds ratio-

$$P/1\text{-}p = \exp\,(b_0 + b_1 x) \qquad \ldots (5.12)$$

Here, b_0 is the constant responsible for moving the curve left and right while b_1 defines the steepness of the curve.

(ii) Multi-Layer Perceptron (MLP)

The simplest form of neural network needs to classify linearly separable patterns. While for non-linear patterns multi-layer Perceptron neural network model performs well. It maps set of input data onto a set of appropriate outputs. Multi-Layer Perceptron consists of multiple layers of nodes in a directed graph with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a non-linear Activation function. Multi-Layer Perceptron uses back propagation learning algorithm for training and widely used in pattern classification and recognition. The simplest form of MLP is shown in Fig. 5.2.

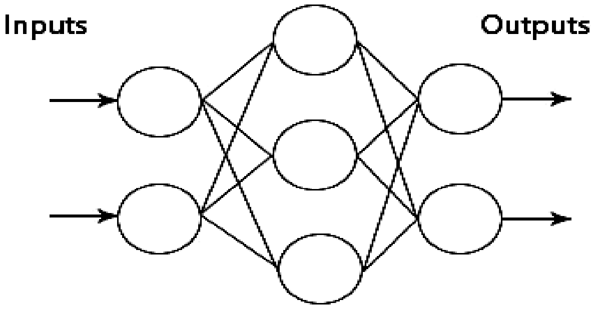

Figure 5.2 Multilayer Perceptron

Multi-layer Perceptron is a supervised learning algorithm that learns a function $f\ (\cdot)$: $R^m \rightarrow R^0$ by training on a dataset, where m is the number of dimensions for input and o is the number of dimensions for output. Given a set of features $X = x_1, x_2... x_m$ and a target y, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. The leftmost layer, known as the input layer, consists of a set of neurons $\{x_i \mid x_1, x_2... x_m\}$ representing the input features. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + ... + w_mx_m$, followed by a non-linear activation function $g(\cdot): R \rightarrow R$- like the hyperbolic tan function. The output layer receives the values from the last hidden layer and transforms them into output values.

5.6 Rule Based Classification

The rule-based classification is based upon rules like (if-else). Rules are mutually exclusive and exhaustive. Two variants are discussed in following subsections.

(i) ZeroR

ZeroR is the simplest method which only focuses on the class. It is a least accurate which predicts only the majority category (class). It is useful for determining a baseline as a benchmark for other classification methods. ZeroR algorithm ignores the predictors only constructs a frequency table for the target and select its most frequent value.

(ii) OneR

OneR is a rule-based classification that uses a single predictor. It generates one rule for each predictor in the data and constructs frequency table for each predictor against the target. OneR short for “One Rule” is an accurate classification algorithm, for each value of predictor, OneR makes the rule as follows.

1. Count how often each value of target (class) appears
2. Find the most frequent class

3. Make the rule assign that class to this value of predictors
4. Calculate the total error of the rules of each predictor
5. Choose the predictor with smallest total error.

OneR produces rules only slightly less accurate than state-of-the-art classification algorithms.

5.7 Bayesian-Based Classification

Bayesian classification is based upon Baye’s probability rules and depends on likelihood functions. Two variants are simple Bayes classification and Naïve Bayes classification.

(i) Simple Bayes Classification

Bayesian networks are a powerful probabilistic representation for classification. The Baye’s provides a rule for calculating posterior probability. If probability of item A is to be calculated over given item B then according to Baye’s theorem,

$$P\ (A/B) = P\ (B/A) * P\ (A) / P\ (B) \qquad (5.13)$$

P (A/B) is probability of A given that B is true (posterior probability).

P (B/A) is probability of B given that A is true (likelihood).

P (A) is prior probability of the class.

P (B) is prior probability of predictor.

(ii) Naive Bayes

The Naive Bayes classifier is based on Baye’s Theorem with independent assumptions between predictors. Naive Bayesian model is easy to build without complicated iterative parameter estimation. It analyzes all the attributes in the data individually, means the value of a predictor (X) on a given (C) is independent of the values of other predictors. This assumption is called class conditional independence. The working steps for Naive Baye’s classifier are as follows.

1. First calculate the posterior probability and construct the frequency table against the target
2. Transforming the frequency table into likelihood table and using the Naive Baye’s equation to calculate the posterior probability for each class
3. Class with highest probability is the outcome of prediction

   P (C/X) = P (X/C) * P (C) / P (X) (5.14)

   P (C/X) is posterior probability of class (target) given predictor (attribute)

   P (X/C) is likelihood which is the probability of predictor given class

   P (C) is prior probability of the class

   P (X) is prior probability of predictor

5.8 Measures of Classification Accuracy

(i) Execution Time

The time is taken by the tool to execute and evaluate an algorithm.

(ii) Accuracy

Accuracy defines the percentage of correctly classified instances from the test set by the classifier.

(iii) True Positive Rate (TPR)

The True Positive rate is the proportion of examples which were classified as a class X among all examples which truly have the X class. This shows how much the part of the class is truly captured.

(iv) False Positive Rate (FPR)

False Positive is the proportion of the examples which were incorrectly identified and belong to another class.

(v) True Negative Rate (TNR)

This is the state of incorrect instances that are classified or predicted as an incorrect class in the classification task and is known as True Negative.

(vi) False Negative Rate (FNR)

Incorrect instances are predicted or classified as a correct class in the classification. This is termed as False Negative. These four outcomes will be mapped in the contingency table.

They are depicted in Table 5.1.

Table 5.1 Contingency Table

Classification Instances	Actual Positive	Actual Negative
Positive outcome in the classification	True Positive [TP]	True Negative [TN]
Negative outcome in the classification	False Negative [FN]	False Positive [FP]

The above four outcomes are used to calculate the Precision, Recall, and F-measure. This will, in turn, be used to estimate the accuracy of the classification task.

(vii) Precision

Precision is the proportion of examples which truly has class X among all those which were classified as X. It is a measure of exactness. Positive predictive value (PPV) is called precision.

$$\text{PPV} = \frac{TP}{TP+FP} \quad \text{... (5.15)}$$

(viii) Recall

Recall is the proportion of examples which were classified as class X among all examples which truly have class X. It is a measure of completeness. Negative predictive value (NPV) is called recall.

$$\text{NPV} = \frac{TP}{TP+FN} \quad \text{... (5.16)}$$

(ix) F-Measure

F-measure is aggregate of precision and recall; the formula is defined as 2 *Precision * Recall / Precision + Recall (5.17)

(x) Receivers Operating Characteristics curve (ROC) Area

Receivers operating characteristics curve is a comparison of two operating characteristics TPR and FPR. It is also known as receivers operating characteristic curve. A receiver operating characteristic curve is a graphical measure which interprets the performance of a classifier as its discrimination threshold is varied. It is an outcome of plotting the true positive rate vs. false positive rate at various threshold settings. The true positive rate is a fraction of true positives out of the total actual positives while the fraction of false positives out of the total actual negatives indicates false positive rate. The point in threshold curve records various statistics such as true positive, false positive and the like. The curves are generated by sorting the prediction produced by the classifier in descending order of probability it assigns to the positive class.

The formula is defined as:

$$\text{ROC Area} = \frac{TP}{TP + FN * 100} \qquad \text{... (5.18)}$$

$$\text{FP Rate} = \frac{FP}{FP + TN} \qquad \text{... (5.19)}$$

(xi) Precision Recall characteristics curve (PRC) Area

The Precision-Recall Characteristics curve is known as precision-recall characteristics curve. It is a comparison of two operating characteristics (Positive Predictive Value and sensitivity) as the criterion changes. A precision-recall curve or PRC curve is a graphical plot which illustrates the performance of binary classifiers as its discrimination threshold is varied. PPV is a fraction of true positives out of test outcomes positive. While sensitivity is a fraction of true positive out of conditions positive.

(xii) Confusion Matrix

It is also called contingency table. In our case, the result is in two classes, so it is 2 X 2 confusion matrix. Numbers of correctly classified instances are diagonal in the matrix, and others are incorrectly classified.