Chapter 2

Literature review on data mining research

A literature survey on the research and developments in the data mining domain is given in this chapter. The chapter is organised as individual sections for each of the popular data mining models and respective literature is given in each section.
2.1 Data mining concepts

Data mining is a collection of techniques for efficient automated discovery of previously unknown, valid, novel, useful and understandable patterns in large databases. The patterns must be actionable so that they may be used in an enterprise’s decision making process. It is usually used by business intelligence organizations, and financial analysts, but it is increasingly used in the sciences to extract information from the enormous data sets generated by modern experimental and observational methods [6].

A typical example for a data mining scenario may be “In the context of a super market, if a mining analysis observes that people who buy pen tend to buy pencil too, then for better business results the seller can place pens and pencils together.”

Data mining strategies can be grouped as follows:

- **Classification**- Here the given data instance has to be classified into one of the target classes which are already known or defined [19, 20]. One of the examples can be whether a customer has to be classified as a trustworthy customer or a defaulter in a credit card transaction data base, given his various demographic and previous purchase characteristics.

- **Estimation**- Like classification, the purpose of an estimation model is to determine a value for an unknown output attribute. However, unlike classification, the output attribute for an estimation problem are numeric rather than categorical. An example can be “Estimate the salary of an individual who owns a sports car?”

- **Prediction**- It is not easy to differentiate prediction from classification or estimation. The only difference is that rather than determining the
current behaviour, the predictive model predicts a future outcome. The output attribute can be categorical or numeric. An example can be “Predict next week’s closing price for the Dow Jones Industrial Average”. [53, 65] explains the construction of a decision tree and its predictive applications.

- **Association rule mining** - Here interesting hidden rules called association rules in a large transactional data base is mined out. For e.g. the rule {milk, butter->biscuit} provides the information that whenever milk and butter are purchased together biscuit is also purchased, such that these items can be placed together for sales to increase the overall sales of each of the items [2, 40].

- **Clustering** - Clustering is a special type of classification in which the target classes are unknown. For e.g. given 100 customers they have to be classified based on certain similarity criteria and it is not preconceived which are those classes to which the customers should finally be grouped into.

The main application areas of data mining are in Business analytics, Bioinformatics [33, 34, and 64], Web data analysis, text analysis, social science problems, biometric data analysis and many other domains where there is scope for hidden information retrieval [42]. Some of the challenges in front of the data mining researchers are the handling of complex and voluminous data, distributed data mining, managing high dimensional data and model optimization problems.

In the coming sections the various stages occurring in a typical data mining problem are explained. The various data mining models that are
commonly applied to various problem domains are also discussed in detail in the coming sections.

2.2 Data mining stages

![Data mining stages](image)

*Figure 2.1: Data mining stages*

Any data mining work may involve various stages as shown in Figure 2.1. Business understanding involves understanding the domain for which the data mining has to be performed. The various domains can be financial domain, educational data domain and so on. Once the domain is understood properly, the domain data has to be understood next as shown in figure 2.1. Here relevant data in the needed format will be collected and understood.

Data preparation or pre-processing is an important step in which the data is made suitable for processing. This involves cleaning data, data transformations, selecting subsets of records etc. When data is prepared there
are two stages, namely selection and transformation. Data selection means selecting data which are useful for the data mining purpose. It is done by selecting required attributes from the database by performing a query. Data transformation or data expression is the process of converting the raw data into required format which is acceptable by data mining system. For e.g., Symbolic data types are converted into numerical form or categorical form.

Data modelling involves building the models like decision tree, neural network etc from above pre-processed data.

### 2.3 Data mining models

There are many popular models that can be effectively used in different data mining problems. Decision trees, neural networks, Naive Bayes classifier, Lazy learners, Support vector machines, and regression based classifiers are few among them. Depending upon the type of application, nature of data and attributes, one can decide which can be the most suited model. Still there is no clear cut answer to the question of which is the best data mining model. One can only say for a particular application one model is better than the other.

#### 2.3.1 Decision trees

The decision tree is a popular classification method. It is a tree like structure where each internal node denotes a decision on an attribute value. Each branch represents an outcome of the decision and the tree leaves represent the classes [54, 56]. Decision tree is a model that is both predictive and descriptive. A decision tree displays relationships found in the training data.
In data mining and machine learning, a decision tree is a predictive model; that is, a mapping from observations about an item to conclusions about its target value [62]. More descriptive names for such tree models are classification tree (discrete outcome) or regression tree (continuous outcome). In these tree structures, leaves represent classifications and branches represent conjunctions of features that lead to those classifications. The machine learning technique for inducing a decision tree from data is called decision tree learning.

Given a set of examples (training data) described by some set of attributes (ex. Sex, rank, background) the goal of the algorithm is to learn the decision function stored in the data and then use it to classify new inputs. The discriminative power of each attribute that can best split the dataset is done either using the concept of information gain or Gini index. Popular decision tree algorithms like ID3, C4.5, C5 etc use information gain to select the next best attribute whereas popular packages like CART, IBM intelligent miner etc use the Gini index concept.

**Information gain**

A decision tree can be constructed top-down using the information gain in the following way:

1. Let the set of training data be S. Continuous attributes if any, should be made discrete, before proceeding further. Once this is done put all of S in a single tree node.
2. If all instances in S are in same class, then stop
3. Split the next node by selecting an attribute A , for which there is maximum information gain
4. Split the node according to the values of A
5. Stop if either of the following conditions is met, otherwise continue with step 3:

(a) If this partition divides the data into subsets that belong to a single class and no other node needs splitting.

(b) If there are no remaining attributes on which the sample may be further divided.

In order to build a decision tree, it is needed to be able to distinguish between 'important' attributes, and attributes which contribute little to the overall decision process. Intuitively, the attribute which will yield the most information should become our first decision node. Then, the attribute is removed and this process is repeated recursively until all examples are classified. This procedure can be translated into the following steps:

\[ I(p(v_1), \ldots, p(v_i)) = \sum_{i=1}^{\ell} -p(v_i) \log_2 p(v_i) \] \hspace{1cm} - (2.1)

\[ \text{Gain(Attributes)} = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} \cdot I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right) \] \hspace{1cm} - (2.2)

First, the formula calculates the information required to classify the original dataset (\(p\) - # of positive examples, \(n\) - # of negative examples). Then, the dataset is split on the selected attribute (with \(v\) choices) and the information gain is calculated. This process is repeated for every attribute, and the one with the highest information gain is chosen to be the decision node.

Even though algorithms like ID3, C4.5, C5 etc uses information gain concepts; there are few differences between them. [6] Gives a comparison of various decision tree algorithms and their performances.

C4.5 handles both continuous and discrete attributes. In order to handle continuous attributes, C4.5 creates a threshold and then splits the list into those...
whose attribute value is above the threshold and those that are less than or equal to it. In order to handle training data with missing attribute values, C4.5 allows attribute values to be marked as “?” for missing. Missing attribute values are simply not used in gain or entropy calculations. C4.5 uses pruning concepts. The algorithm goes back through the tree once it's been created and attempts to remove branches that do not help by replacing them with leaf nodes. C5.0 is significantly faster and memory efficient than C4.5. It also uses advanced concepts like boosting, which is later discussed in the thesis.

Algorithms based on information gain theory tend to favour those attributes that have more values where as those based on Gini index tend to be weak when number of target classes are more. Hence Nowadays, researchers are trying to optimize the decision tree performances by using techniques like pre-pruning (removing useless branches of the tree as the tree is built), post pruning (removing useless branches of the tree after the tree is built) etc.

Other variants of decision tree algorithms include CS4 [34, 35], as well as Bagging [9], Boosting [17, 47], and Random forests [10]. A node is removed only if the resulting pruned tree performs no worse than the original, over the cross validation set [63]. Since the performance is measured on validation set, this pruning strategy suffers from the disadvantage that the actual tree is based on less data. However, in practice, C4.5 makes some estimate of error based on training data itself, using the upper bound of a confidence interval (by default is 25%) on the re-substitution error. The estimated error of the leaf is within one standard deviation of the estimated error of the node. Besides reduced error pruning, C4.5 also provides another pruning option known as sub tree raising. In sub tree raising, an internal node might be replaced by one of nodes below and samples will be redistributed. A
detailed illustration on how C4.5 conducts its post-pruning is given in [44, 55].
Other algorithms for decision tree induction include ID3 (predecessor of C4.5) [42], C5.0 (successor of C4.5), CART (classification and regression trees) [8], LMDT (Linear Machine Decision Trees) [63], OC1 (oblique classifier) and so on.

![Figure 2.2: A Sample decision tree-Partial view](image)

Figure 2.2 shows the partial view of a sample decision tree. One of the decision rule it provides is, if Outlook is sunny and humidity is normal, one can go ahead to play.

Usually, when a decision tree is built from the training set, it may be over fitted, which means that the tree performs well for training data only. Its performance will not be that good with unseen data. So one can “Shave off” nodes and branches of a decision tree, essentially replacing a whole sub tree by a leaf node, if it can be established that the expected error rate in the sub tree is greater than that in a single leaf. This makes the classifier simpler.
Pruning is a technique to make an over fitted decision tree simpler and more general. In post pruning, after the tree is fully grown, with some test data, some branches are removed and smaller trees are derived. Now the subset tree with minimum error and simplicity is selected as the final tree.

Another approach is called pre-pruning in which the tree construction is halted early. Essentially a node is not split if this would result in the goodness measure of tree falling below a threshold. It is however, quite difficult to choose an appropriate threshold. The classic decision tree algorithm named C4.5 was proposed by Quinlan [44]. Majority of the research works in decision trees are concerned with the improvement in the performance using optimization techniques such as pruning. [20] Reports a work dealing with understanding student data using data mining. Here decision tree algorithms are used for predicting graduation, and for finding factors that lead to graduation.

[22] Provides an overview about how data mining and knowledge discovery in databases are related to each other and to other fields, such as machine learning, statistics, and databases. [8] Suggests methods to classify objects or predict outcomes by selecting from a large number of variables, the most important ones in determining the outcome variable. [31] Discusses how performance evaluation of a model can be done by using confusion matrix, which contains information about actual and predicted classifications done by a classification system.

In a typical classification task, data is represented as a table of samples, which are also known as instances. Each sample is described by a fixed number of features which are known as attributes and a label that indicated its
class [27]. [52] Describes a special technique that uses genetic algorithms for attribute analysis.

Data mining can extract implicit, previously unknown and potentially useful information from data [32, 62]. It is a learning process, achieved by building computer programs to seek regularities or patterns from data automatically. Machine learning provides the technical basis of data mining. Classification learning is a generalization of concept learning [63]. The task of concept learning is to acquire the definition of a general category given a set of positive and negative training examples of the category [37]. Thus, it infers a Boolean-valued function from the training instances. As a more general form of concept learning, classification learning can deal with more than two class instances. In practice, the learning process of classification is to find models that can separate instances in the different classes using the information provided by training instances.

2.3.2 Neural networks

Neural networks offer a mathematical model that attempts to mimic the human brain [5, 15]. Knowledge is represented as a layered set of interconnected processors, which are called neurons. Each node has a weighted connection to other nodes in adjacent layers. Individual nodes take the input received from connected nodes and use the weights together with a simple function to compute output values. Learning in neural networks is accomplished by network connection weight changes while a set of input instances is repeatedly passed through the network. Once trained, an unknown instance passing through the network is classified according to the values seen at the output layer. [51,57] surveys existing work on neural network construction, attempting to identify the important issues involved, directions
the work has taken and the current state of the art. Typically, a neural network model is having a configuration as shown in figure 2.3 in its basic form. Neurons only fire when input is bigger than some threshold. It should, however, be noted that firing doesn't get bigger as the stimulus increases, it is an all or nothing arrangement [28].

Figure 2.3: A neural network configuration

Suppose a firing rate is there at each neuron. Also suppose that a neuron connects with \( m \) other neurons and so receives \( m \)-many inputs \( x_1 , \ldots , x_m \). This configuration is actually called a Perceptron.

In 1962, Rosenblatt proposed the perceptron model. It was one of the earliest neural network models. A Perceptron models a neuron by taking a weighted sum of inputs and sending the output 1, if the sum is greater than some adjustable threshold value otherwise it sends 0. This is the all or nothing spiking described in the previous paragraph. It is also called an activation function.

The inputs \( (x_1, x_2, x_3 \ldots x_m) \) and connection weights \( (w_1, w_2, w_3\ldots w_m) \) in Figure 2.4 are typically real values, both positive (+) and negative (-).
feature of some $x_i$ tends to cause the perceptron to fire, the weight $w_i$ will be positive; if the feature $x_i$ inhibits the perceptron, the weight $w_i$ will be negative. The perceptron itself consists of weights, the summation processor, and an activation function, and an adjustable threshold processor (called bias). For convenience the normal practice is to treat the bias, as just another input. Figure 2.4 illustrates the revised configuration with bias.

The bias can be thought of as the propensity (a tendency towards a particular way of behaving) of the perceptron to fire irrespective of its inputs. The perceptron configuration network shown in Figure 2.4 fires if the weighted sum > 0, or in mathematical terms, it can be represented as in (2.3)

$$\text{Weighted sum} = \sum_{i=1}^{m} \text{bias} + (w^i x^i) \quad \text{(2.3)}$$

**Activation function:** The activation usually uses one of the following functions.

**Sigmoid function:** The stronger the input is, the faster the neuron fires. The sigmoid is also very useful in multi-layer networks, as the sigmoid curve allows for differentiation (which is required in Back Propagation training of multi layer networks). In mathematical terms, it can be represented as in (2.4)

$$f(x) = \frac{1}{1+e^{-x}} \quad \text{(2.4)}$$

**Step function:** A step function is a basic on/off type function, if $0>x$ then 0, else if $x>=0$ then 1. Hence depending on the type of input, output and problem domain, suited functions are adopted at respective layers.

Learning can be of two types. Supervised and unsupervised. As an example of supervised learning one can consider a real world example of a
baby learning to recognise a chair. He is taught with many objects that are chairs and that are not chairs. After this training, when a new object is shown, he can correctly identify it as a chair or not. This is exactly the idea behind the perceptron. As an example of unsupervised learning, one can consider a six months old baby recognising his mother. Here a supervisor does not exist. All classification algorithms are examples of supervised learning. But clustering is unsupervised learning, where a model does not exist based on which classification has to be performed.

**Figure 2.4:** Artificial Neuron configurations, with bias as additional Input

**Perceptron learning:** The Perceptron is a single layer neural network whose weights and biases are trained to produce a correct target vector when presented with the corresponding input vector. The training technique used is called the perceptron learning rule. The perceptron generated great interest due to its ability to generalize from its training vectors and work with randomly distributed connections. Perceptrons are especially suited for simple problems in pattern classification. Suppose the data can be separated perfectly into two
groups using a hyper plane, it is said to be linearly separable [67]. If the data is linearly separable, the perceptron learning rule can be applied which is given below.

**The Learning rule:** The perceptron is trained to respond to each input vector with a corresponding target output of either 0 or 1. The learning rule converges on a solution in finite time if a solution exists.

The learning rule can be summarized in the following two equations:

\[
b = b + [T - A] \quad \text{- (2.5)}
\]

For all inputs i:

\[
W(i) = W(i) + [T - A] * P(i) \quad \text{- (2.6)}
\]

Where \(W\) is the vector of weights, \(P\) is the input vector presented to the network, \(T\) is the correct result that the neuron should have shown, \(A\) is the actual output of the neuron, and \(b\) is the bias.

**Training:** Vectors from a training set are presented to the network one after another. If the network's output is correct, no change is made. Otherwise, the weights and biases are updated using the perceptron learning rule (as shown above). When each epoch (an entire pass through all of the input training vectors is called an epoch) of the training set has occurred without error, training is complete.

When the training is completed, if any input training vector is presented to the network and it will respond with the correct output vector. If a vector, \(P\), not in the training set is presented to the network, the network will tend to exhibit generalization by responding with an output similar to target vectors for input vectors close to the previously unseen input vector \(P\). The
transfer function used in the Hidden layer is Log-Sigmoid while that in the output layer is Pure Linear.

Neural networks are very good in classification and regression tasks where the attributes have missing values and also when the attribute values are categorical in nature [28, 67]. The accuracy observed is very good, but the only bottle neck is the extra training time and complexity in the learning process when the number of training set examples seems very high. [15, 51] Describe how neural networks can be applied in data mining. There are some algorithms for extracting comprehensible representations from neural networks. [5] Describes research to generalize and extend the capabilities of these algorithms. The application of the data mining technology based on neural network is vast. One such area of application is in the design of mechanical structure. [57] Introduces one such application of the data mining based on neural network to analyze the effects of structural technological parameters on stress in the weld region of the shield engine rotor in a submarine.

[70] Explains an application of neural networks in study of proteins. In that work, global adaptive techniques from multiple alignments are used for prediction of Beta-turns. This also introduces global adaptive techniques like Conjugate gradient method, Preconditioned Conjugate gradient method etc. An approach to discover symbolic classification rules using neural networks is discussed in [67]. Here, first the network is trained to achieve the required accuracy rate, and then activation values of the hidden units in the network are analyzed. Classification rules are generated using the result of this analysis.

**Back propagation in multi layer perceptrons:** Among the several neural network architectures, for supervised classification, feed forward multilayer
network trained with back propagation algorithm is the most popular. Neural networks, in which signal flows from input to output (forward direction) are called, feed forward neural networks. A single layer neural network can solve linearly separable problems only. But when the problem to be solved is more complex, a multilayer feed forward neural network can be used. Here there can be hidden layers other than input and output layers. There is a layer of weights between two adjacent levels of units (input, hidden or output). Back propagation is the training method most often used with feed-forward multi layer networks, typically when the problem to be solved is a non linear classification problem. In this algorithm, the error at the output layer is propagated back to adjust the weights of network connections. It starts by making weight modifications at the output layer and then moving backward through the hidden layers. Conjugate gradient algorithms are one of the popular back propagation algorithms which are explained in section 4.3.

[28] Proposes a 3 layer feed forward network to select the input attributes that are most useful for discriminating classes in a given set of input patterns. This is particularly helpful in feature selection. [60, 61] describe practical applications of neural networks in patient data analysis and [69] describes application of radial basis functions neural networks in protein sequence classification.

### 2.3.3 Naive Bayes classifier

This classifier offers a simple yet powerful supervised classification technique. The model assumes all input attributes to be of equal importance and independent of one another. Naive Bayes classifier is based on the classical Bayes theorem presented in 1763 which works on the probability theory. In simple terms, a naive Bayes classifier assumes that the presence (or
absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. Even though these assumptions are likely to be false, Bayes classifier still works quite well in practice.

Depending on the precise nature of the probability model, Naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for Naive Bayes model uses the method of maximum likelihood.

Despite its simplicity, Naive Bayes can often outperform more sophisticated classification methods. An advantage of the Naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

The classifier is based on Bayes theorem, which is stated as:

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2.7) \]

Where

- \( P(A) \) is the prior probability or marginal probability of \( A \). It is "prior" in the sense that it does not take into account any information about \( B \).
- \( P(A|B) \) is the conditional probability of \( A \), given \( B \). It is also called the posterior probability because it is derived from or depends upon the specified value of \( B \).
- \( P(B|A) \) is the conditional probability of \( B \) given \( A \).
- \( P(B) \) is the prior or marginal probability of \( B \), and acts as a normalizing constant. Bayes' theorem in this form gives a mathematical representation of how the conditional probability of event \( A \) given \( B \) is related to the converse conditional probability of \( B \) given \( A \). Similarly numeric data
can be dealt with in a similar manner provided that the probability density function representing the distribution of the data is known. For example, suppose the training data contains a continuous attribute, $x$. First the data has to be segmented by the class, and then compute the mean and variance of $x$ in each class. Let $\mu_c$ be the mean of the values in $x$ associated with class $c$, and let $\sigma^2_c$ be the variance of the values in $x$ associated with class $c$. Then, the probability of some value given a class, $P(x = v | c)$, can be computed by plugging $v$ into the equation for a Normal distribution parameterized by $\mu_c$ and $\sigma^2_c$. Equation (2.8) is used for numeric data assuming numeric data follows normal distribution.

$$P(x = v | c) = \frac{1}{\sqrt{2\pi\sigma^2_c}} e^{-\frac{(v - \mu_c)^2}{2\sigma^2_c}}$$

(2.8)

The Naive Bayes classifier is illustrated with an example in section 2.3.3.1.

2.3.3.1 Example

Training Examples

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>

Figure 2.5: Training example for Naive Bayes classifier
Consider the data in figure 2.5 and assume one want to compute whether tennis can be played? (This is called as hypothesis H). Given the following weather conditions:
Outlook = “sunny” Temp = “cool” Humidity = “high” Wind = “Strong”
The above attribute conditions together can be called as evidence E. 
It has to be found out whether tennis can be played. Hence one has to compute P(Playtennis = Yes|E).
Using the formulae in (2.7),
P(Playtennis = Yes|E) = P(E|Playtennis = yes) * P(Playtennis = yes)/P(E)
P(E|Playtennis = yes) is computed as follows:
P(Outlook = sunny|Playtennis = Y) * P(Temp = cool|palytennis = yes) *
Also P(Playtennis = yes) = 9/14 = 0.642
So P(Playtennis = Yes|E) = 0.008 * 0.642 / P(E) = 0.005 / P(E)
Similarly P(Playtennis = No|E), by applying values becomes 0.021 / P(E)
HENCE ONE CAN NOT PLAY TENNIS!!!!!!

The same technique may be applied to different data sets and there are many packages like SPSS, WEKA etc that support Bayes classifier. Bayes classifier is so popular owing to its simplicity and efficiency. The classifier is explained in detail in [62]. [38] Describes the possibilities of real world applications of ensemble classifiers that involve Bayesian classifiers. In [16] a hybrid approach of classifiers that involves Naive Bayes classifier is also discussed.

Usually there are two active research areas in this comparison of classifier performances in a domain. One is optimizing a single classifier
performance by improvement of single classifier algorithm. The second approach is combining classifiers to improve accuracy.

Another important application area of data mining is association rule mining. It is all about finding interesting patterns usually purchase patterns in a sales transactional data bases. [4] Represents an application of association rules techniques in the data mining domain.

Another approach in improving classifier performances was in studies of applying special algorithms like genetic algorithms [52] and fuzzy logic [12] concepts into classifiers, which were found to be successful in improving accuracies. The section 2.3.4 presents a literature survey on the research that has happened in the domain of ensemble of classifiers.

2.3.4 Ensemble of classifiers

An ensemble of classifiers is an approach in which several classifiers are combined together to improve the overall classifier performance. It can be done in two ways, homogenous way in which same classifiers are combined and heterogeneous or hybrid in which different classifiers are combined.

“Whether an ensemble of homogenous or heterogeneous classifiers yields good performance” is always been a debatable question. [36] Proposes that depending on a particular application, an optimal combination of heterogeneous classifiers seems to perform better than homogenous classifiers.

[13, 50, 58] explain the possibilities of combining data mining models to get better results. In this work, the classifier performance is improved using the stacking approach. There are many strategies for combining classifiers like voting [7], bagging and boosting each of which may not involve much learning in the meta or combining phase. Stacking is a parallel combination of classifiers in which all the classifiers are executed parallel and learning takes
place at the meta level. Decision on which model or algorithm performs best at the meta level for a given problem is an active research area.

When only the best classifier among the base level classifiers is selected, the valuable information provided by other classifiers is ignored. In classifier ensembles which are also known as combiners or committees, the base level classifier performances are combined in some way such as voting or stacking.

It is found that stacking method is particularly suitable for combining multiple different types of models. Instead of selecting one specific model out of multiple ones, the stacking method combines the models by using their output information as inputs into a new space. Stacking then generalizes the guesses in that new space. The outputs of the base level classifiers are used to train a meta classifier. In this next level, it is ensured that the training data has accurately completed the learning process. For example, if a classifier consistently misclassifies instances from one region as a result of incorrectly learning the feature space of that region, the meta classifier may be able to discover this problem. This is the clear advantage of stacking over other methods like voting where no such learning takes place from the output of base level classifiers. Using the learned behaviours of other classifiers, it can improve such training deficiencies. Other bright future research areas in ensemble methods can be design and development of distributed, parallel and agent based ensemble methods for better classifier performances as pointed out in [16].

**Ensemble of decision trees:** Ensemble methods are learning algorithms that construct a set of classifiers and then classify new samples by taking a vote of their predictions [17]. Generally speaking, an ensemble method can increase
predictive performance over a single classifier. In [18], Dietterich explains why ensemble methods are efficient compared to a single classifier within the ensemble. Besides, plenty of experimental comparisons are performed to show significant effectiveness of ensemble methods in improving the accuracy of single base classifiers [3, 7, 9, 24, 43 and 46].

The original ensemble method is Bayesian averaging [17]. But bagging (bootstrap aggregation) and boosting are two of most popular techniques for constructing ensembles. [24] Explains the principle of ensemble of decision trees.

**Bagging of decision trees:** The technique of bagging (derived from bootstrap aggregation) was coined by Breiman [9], who investigated the properties of bagging theoretically and empirically for both classification and numeric prediction. Breiman had also proposed a classification algorithm namely random forest, as a variant of conventional decision tree algorithm, which is included in the WEKA data mining package [10].

Bagging of trees combines several tree predictors trained on bootstrap samples of the training data and gives prediction by taking majority vote. In bagging, given a training set with samples, a new training set is obtained by drawing samples uniformly with replacement. When there is a limited amount of training samples, bagging attempts to neutralize the instability of single decision tree classifier by randomly deleting some samples and replicating others. The instability inherent in learning algorithms means that small changes to the training set cause large changes in the learned classifier.

**Boosting of decision trees:** Unlike bagging, in boosting, every new tree that is generated, are influenced by the performance of those built previously. Boosting encourages new trees to become “experts” for samples handled
incorrectly by earlier ones [55]. When making classification, boosting weights a tree’s contribution by its performance, rather than giving equal weight to all trees which is adopted by bagging. There are many variants on the idea of boosting. The version introduced below is called AdaBoostM1 which was developed by Freund and Schapire [23] and designed specifically for classification. The AdaBoostM1 algorithm maintains a set of weights over the training data set and adjusts these weights after iterations of the base classifier. The adjustments increase the weight of samples that are misclassified and decrease the weight of samples that are properly classified. By weighting samples, the decision trees are forced to concentrate on those samples with high weight. There are two ways that AdaBoostM1 manipulates these weights to construct a new training set to feed to the decision tree classifier [55]. One way is called boosting by sampling, in which samples are drawn with replacement using probability proportional to their weights. Another way is boosting by weighting, in which the presence of sample weights changes the error calculation of tree classifier. That is, using the sum of the weights of the misclassified samples divided by the total weight of all samples, instead of the fraction of samples that are misclassified. [48, 49] give introduction and applications of boosting. In [43], C4.5 decision tree induction algorithm is implemented to deal with weighted samples

**CS4— a new method of ensemble of decision trees:** CS4 stands for cascading-and-sharing for ensemble of decision trees. It is a newly developed classification algorithm based on an ensemble of decision trees. The main idea of this method is to use different top-ranked features as the root node of a decision tree in an ensemble (also named as a committee) [35]. Different from bagging or boosting which uses bootstrapped data, CS4 always builds decision
trees using exactly the same set of training samples. The difference between this algorithm and Dietterich’s randomization trees is also very clear. That is, the root node features of CS4 induced trees are different from each other while every member of a committee of randomized trees always shares the same root node feature (the random selection of the splitting feature is only applied to internal nodes). On the other hand, compared with the random forests method which selects splitting features randomly, CS4 picks up root node features according to their rank order of certain measurement (such as entropy, gain ratio). Thus, CS4 is claimed as a novel ensemble tree method. Breiman noted in [9] that most of the improvement from bagging is evident within ten replications. Therefore, 20 is set (default value is 10) as the number of bagging iterations for Bagging classifier, the number of maximum boost iterations for AdaBoostM1, and the number of trees in the forest for Random forests algorithm.

**Example applications:** Classifier ensembles have wide applications ranging from simple applications to remote sensing. [25, 26] explain ensemble of classifiers which classifies very high resolution remote sensing images from urban areas.

[11] Describes a practical application of ensemble of classifiers in the domain of intrusion detection in mobile ad-hoc networks. There they use ensemble of classifiers for predicting intrusion attempts. In [66] ensemble based classification methods were applied to spam filtering. [59] Describes a cost sensitive learning approach which resembles ensemble methods, for recurrence prediction of breast cancer.

All the ensemble algorithms discussed in this section are based on ‘‘passive’’ combining, in that the classification decisions are made based on
static classifier outputs, assuming all the data is available at a centralized location at the same time. Distributed classifier ensembles using “active” or agent-based methods can overcome this difficulty by allowing agents to decide on a final ensemble classification based on multiple factors like classification and confidence [1]. Another application is in face recognition. Chawla and Bowyer [14] addressed the standard problem of face recognition but under different lighting conditions and with different facial expressions. The authors of [41] decided to theoretically examine which combiners do the best job of optimizing the Equal Error Rate (EER), which is the error rate of a classifier when the false acceptance rate is equal to the false rejection rate. [56, 62] explain the practical methods of how to use Weka for advanced data mining tasks including how to include java libraries in Weka.

2.3.5 Other data mining models

Even though popular classification models like decision trees, neural networks and Naive Bayes classifiers are described in detail in the above section, it is worth mentioning about data mining models, which are quite popular among researchers owing to its efficiency and range of applications. Some of them are given below:

2.3.5.1 Lazy classifiers

Lazy learners store the training instances and do no real work until classification time. \textit{IB1} is a basic instance-based learner which finds the training instance closest in Euclidean distance to the given test instance and predicts the same class as this training instance. If several instances qualify as the closest, the first one found is used. \textit{IBk} is a \textit{k}-nearest-neighbor classifier that uses the same distance metric. The number of nearest neighbors (default \textit{k} is 1) can be specified explicitly or determined automatically using leave-one-
out cross-validation, subject to an upper limit given by the specified value. Predictions from more than one neighbor can be weighted according to their distance from the test instance, and two different formulas are implemented for converting the distance into a weight. The number of training instances kept by the classifier can be restricted by setting the window size option. As new training instances are added, the oldest ones are removed to maintain the number of training instances at this size. KStar is a nearest neighbor method with a generalized distance function based on transformations.

2.3.5.2 Association rules

They are really not that different from classification rules except that they can predict any attribute, not just the class, and this gives them the freedom to predict combinations of attributes too. Also, association rules are not intended to be used together as a set, as classification rules are. Different association rules express different regularities that underlie the dataset, and they generally predict different things.

Because so many different association rules can be derived from even a tiny dataset, interest is restricted to those that apply to a reasonably large number of instances and have a reasonably high accuracy on the instances to which they apply to. The coverage of an association rule is the number of instances for which it predicts correctly. This is often called its support. Its accuracy is often called confidence. It is the number of instances that it predicts correctly, expressed as a proportion of all instances to which it applies.

For example, for the rule:

If temperature = cool then humidity = normal, coverage is the number of days that are both cool and normal, and the accuracy is the proportion of
cool days that have normal humidity. It is usual to specify minimum coverage and accuracy values and to seek only those rules whose coverage and accuracy are within these specified limits. The concept of association rules are mentioned here owing to its high applicability in sales data bases.

2.3.5.3 Machine learning and statistics

Data mining can be considered as a confluence of statistics and machine learning. In truth, one should not look for a dividing line between machine learning and statistics because there is a continuum. Some derive from the skills taught in standard statistics courses, and others are more closely associated with the kind of machine learning that has arisen out of computer science. Historically, the two sides have had rather different traditions. While statistics is more concerned with testing hypotheses, machine learning is more concerned with formulating the process of generalization as a search through possible hypotheses. But this is a gross oversimplification. Statistics is far more than hypothesis testing, and many machine learning techniques do not involve any searching at all. In the past, many similar methods were developed in parallel in machine learning and statistics. One is decision tree induction. Four statisticians (Breiman et al. 1984) published a book on Classification and regression trees in the mid-1980s, and throughout the 1970s and early 1980s a prominent machine learning researcher, J. Ross Quinlan, was developing a system for inferring classification trees from examples. These two independent projects produced quite similar methods for generating trees from examples, and the researchers became aware of one another’s work much later only. Most learning algorithms use statistical tests when constructing rules or trees and for correcting models that are “over fitted” in that they depend too strongly on the details of the particular
examples used to produce them. Statistical tests are used to validate machine learning models and to evaluate machine learning algorithms. From the literature survey it could easily be concluded that statistics and data mining are very much correlated topics and helps each other in future developments of respective fields.

2.3.6 The ethics of data mining

The use of data, particularly data about people, has serious ethical implications in data mining. Practitioners of data mining techniques must act responsibly by making themselves aware of the ethical issues that surround their particular application. When applied to people, data mining is frequently used to discriminate like who gets the loan, who gets the special offer, and so on. Certain kinds of discrimination like racial, sexual, religious, and so on are not only unethical but also illegal. However, the complexity of the situation depends on the application. Using sexual and racial information for medical diagnosis is certainly ethical, but using the same information when mining loan payment behaviour is not. Even when sensitive information is discarded, there is a risk that models will be built that rely on variables that can be shown to substitute for racial or sexual characteristics. For example, people frequently live in areas that are associated with particular ethnic identities. So using an area code in a data mining study runs the risk of building models that are based on race, even racial information is explicitly excluded from the data. It is widely accepted that before people make a decision to provide personal information they need to know how it will be used and what it will be used for, what steps will be taken to protect its confidentiality and integrity, what the consequences of supplying or withholding the information are, and any rights of redress they may have. Whenever such information is collected,
individuals should be told these things, not in legalistic small print but straightforwardly in plain language they can understand. The potential use of data mining techniques may stretch far beyond what was conceived when the data was originally collected. This creates a serious problem. It is necessary to determine the conditions under which the data was collected and for what purposes it may be used. Surprising things emerge from data mining. For example, it was reported that one of the leading consumer groups in France has found that people with red cars are more likely to default on their car loans. What is the status of such a “discovery”? What information is it based on? Under what conditions was that information collected? In what ways is it ethical to use it? Clearly, insurance companies are in the business of discriminating among people based on stereotypes like young males pay heavily for automobile insurance. But such stereotypes are not based solely on statistical correlations; they also involve common-sense knowledge about the world. Whether the preceding finding says something about the kind of person who chooses a red car, or whether it should be discarded as an irrelevancy, is a matter of human judgment based on knowledge of the world rather than on purely statistical criteria. When presented with data, it is needed to ask who is permitted to have access to it, for what purpose it was collected, and what kind of conclusions is legitimate to draw from it. The ethical dimension raises tough questions for those involved in practical data mining. It is necessary to consider the norms of the community that is used to dealing with the kind of data involved, standards that may have evolved over decades or centuries but ones that may not be known to the information specialist. In addition to community standards for the use of data, logical and scientific standards must be adhered to when drawing conclusions from it. If one comes up with
conclusions (such as red car owners being greater credit risks), one need to attach caveats to them and back them up with arguments other than purely statistical ones. The point is that data mining is just a tool in the whole process. It is people who take the results, along with other knowledge, and decide what action to be taken. Of course, anyone who uses advanced technologies should consider the wisdom of what they are doing. So the conclusion that can be drawn is like when data mining is conducted in a particular domain, one should seriously address the question whether the objective of mining is useful for the mankind and is there anything non-ethical hidden behind the rules extracted from the data mining process.

2.4 Chapter summary

This chapter gives an introduction to the various data mining strategies popularly used. Models like decision tree, neural network and Naive Bayes classifier are described in detail. Important research works carried out using these models are reviewed in this chapter. Decision trees are very simple to interpret and work faster. Neural networks manage missing values and categorical values very efficiently. Naive Bayes insist that their numeric data should be normally distributed. From the discussions, it can be concluded that for data mining one cannot find a single classifier that outperforms every other but rather one can find a classifier that performs well for a particular domain.