CHAPTER VI
CLASSIFICATION IN HYPOTHYROIDISM

6.1 INTRODUCTION

The problem of classification plays a significant role in analyzing any medical diagnosis. Medical diagnosis is a problem complicated by many factors and involving all of human abilities including intuition and the subconscious. Thyroid disease specially a major disorder known as hypothyroidism. The data is referred from the journal Khanale and Ambilwade (2011), Aurangabad city of Maharashtra state, India are collected. The variables are symptom score, T4 and TSH. T4 (thyroxine) is a hormone produced by the thyroid gland. A T4 test measures the amount of T4 in your blood. A TSH test measures the amount of thyroid stimulating hormone (TSH) in your blood. TSH is produced by the pituitary gland and tells the thyroid gland to make and release the hormones thyroxine (T4) and triiodothyronine (T3).

It is the most common thyroid disorder today. K-Nearest Neighbor (KNN) is one of the most popular algorithms for pattern recognition. Many researchers have found that the KNN algorithm accomplishes very good performance in their experiments on different
data sets. The support vector machine is a semi parametric technique with origins in the machine-learning literature of engineering and computer science. Support vector machine Carl Gold and Peter Sollich (2003) emerged in recent years as powerful techniques both for regression and classification. Der-Chiang Li and Yao Hwei (2008) presented support vector machines mainly used for classification problems. Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a classification or decision.

6.2 CLASSIFICATION

The following preprocessing steps may be applied to the data which helps to improve the accuracy, efficiency, and scalability of the classification.

DATA CLEANING

This refers to the preprocessing of data in order to remove or reduce noise and the treatment of missing values. Most classification algorithms have some mechanisms for handling noisy or missing data; this step can help to reduce confusion during learning.

RELEVANCE ANALYSIS

Relevance analysis, in the form of correlation analysis and attribute subset selection, can be used to detect attributes that do not
contribute to the classification or prediction task. Relevance analysis removes redundant and irrelevant data from the database.

**DATA TRANSFORMATION AND REDUCTION**

The data may be transformed by normalization, particularly when neural networks or methods involving distance measurements are used in the learning step. Normalization involves scaling all values for a given attribute so that they fall within a small specified range, such as -1.0 to 1.0, or 0.0 1.0. The data can also be transformed by generalizing it to higher – level concepts. Concept hierarchies may be used for this purpose. This is particularly useful for continuous valued attributes.

**6.3 K-NEAREST NEIGHBOUR**

K-nearest neighbour is a supervised learning algorithm developed by Cover and Hart in 1967, where the result of new instance query is classified based on majority of k-nearest neighbour category. The purpose of this algorithm is to classify a new object based on memory. Given a query point, we find k number of objects or (training points) closest to the query point. The classification is using majority vote among the classification of the k-objects. Any ties can be broken at random. K nearest neighbour algorithm used neighborhood classification as the prediction value of the new query instance.

K nearest neighbour algorithm is very simple. It works based on
minimum distance from the query instance to the training samples to determine the k nearest neighbors David et al. (2008).

**ALGORITHM**

The main steps of the KNN algorithm are as follows:

1. Assume there are N training objects where each object has \( t \)-attributes and an object can belong to one of the \( m \)-classes.
2. Let \( O \) be an object to be classified.
3. Compute the distance between the object \( O \) and each of the training objects.
4. Let \( d_1, d_2, \ldots, d_N \) be the resulting distances.
5. Arrange the distances in ascending order and identify first \( k \) objects corresponding to the first \( k \) smallest distances to get the set \( C_k \).
6. Let \( x_r \) denote the number of objects in the set \( C_k \) belong to the class \( r \) (\( r = 1, 2, \ldots, m \)). We assign the object to the class \( \lambda \) if, \( \pi_\lambda = \max_{r=1}^{m} x_r \).

6.4 LINEAR SUPPORT VECTOR MACHINE

On the basis of statistical learning theory, Support Vector Machine (SVM) is introduced by Boser et al. (1992). Assume we have learning set of data, \( L = \{(x_i, y_i) : i = 1, 2, \ldots, n \} \)

where \( x_i \in \mathbb{R}^t \) and \( y_i \in \{-1, +1\} \).

The binary classification is to use \( L \) to construct a function

\( f : \mathbb{R}^t \rightarrow \mathbb{R} \)
so that $c(x) = \text{sign}(f(x))$ is a classifier.

The separating function $f$ then classifies each new point $x$ in a test set $T$ into one of two classes, $\Pi_+$ or $\Pi_-$, depending upon whether

$$C(x) = +1 \quad \text{if} \quad f(x) \geq 0,$$

then

$$C(x) = -1 \quad \text{if} \quad f(x) \leq 0,$$

The goal is to have $f$ assign all positive points $y = +1$ in $T$ to $\Pi_+$ and negative points $y = -1$ to $\Pi_-$, but it may not be possible in practical situations.

**THE LINEARLY SEPARABLE CASE**

Suppose the positive and negative data points from the learning set $L$ can be separated by a hyperplane, $\{x : f(x) = \beta_0 + x^T \beta = 0\}$, where $\beta$ is the weight vector with Euclidean norm $\|\beta\|$, and $\beta_0$ is the bias. Consider any separating hyperplane. Let $d_-$ be the shortest distance from the separating hyperplane to the nearest negative data point, and let $d_+$ be the shortest distance from the hyperplane to the nearest positive data point. Then, the margin of the separating hyperplane is defined as $d = d_- + d_+$. If the distance between the hyperplane and its closest observation is maximized, we say that the hyperplane is an optimal separating hyperplane.
If the learning data from the two classes are linearly separable, there exists $\beta_0$ and $\beta$ such that

$$\beta_0 + x_i^T \beta \geq 1, \quad \text{if} \quad y_i = +1 \quad \ldots \quad (6.1)$$

$$\beta_0 + x_i^T \beta \leq 1, \quad \text{if} \quad y_i = -1 \quad \ldots \quad (6.2)$$

If there are data vectors in $L$ such that equality holds in equation (6.1), then these data vectors lie on the hyperplane

$$H_{+1} : (\beta_0 - 1) + x^T \beta = 0$$

If there are data vectors in $L$ such that equality holds in equation (6.2), then these data vectors lie on the hyperplane

$$H_{-1} : (\beta_0 + 1) + x^T \beta = 0.$$ 

If $X_1$ lies on the hyperplane $H_{+1}$ and if $X_1$ lies on the hyperplane $H_{-1}$ then

$$\beta_0 + x_{+1}^T \beta = -1, \quad \text{and} \quad \beta_0 + x_{-1}^T \beta = +1 \quad \ldots \quad (6.3)$$
The difference of these two equations is
\[ x_i^T \beta - x_i' \beta = 2, \]
and their sum is
\[ \beta_0 = -\frac{1}{2} \{ x_i^T \beta + x_i'^T \beta \}. \]

The perpendicular distances of the hyperplane \( \beta_0 + x^T \beta = 0 \) from the points \( x_{-1} \) and \( x_{+1} \) are
\[ d_+ = \frac{\beta_0 + x_{-1}^T \beta}{\| \beta \|} = \frac{1}{\| \beta \|}, \quad d_+ = \frac{\beta_0 + x_{+1}^T \beta}{\| \beta \|} = \frac{1}{\| \beta \|} \]

Hence, the margin of the separating hyperplane is \( d = \frac{2}{\| \beta \|} \). The inequalities (6.1) and (6.2) can be combined into a single set of inequalities, \( y_i (\beta_0 + x_i^T \beta) \) is called the margin of \( (x_i, y_i) \) with respect to the hyperplane \( y_i (\beta_0 + x_i^T \beta) \geq 1 \), \( i = 1, 2, \ldots, n \)

From equation (6.3), we see that \( x_i \) is a support vector with respect to the hyperplane \( \{ x : f(x) = \beta_0 + X^T \beta = 0 \} \)

If its margin equals one
\[ y_i (\beta_0 + x_i^T \beta) = 1 \]

The problem is to find optimal separating hyperplane; namely find the hyperplane that maximizes the margin, \( \frac{2}{\| \beta \|} \)

subject to the conditions
\[ y_i (\beta_0 + X_i^T \beta) \geq 1, \quad i = 1, 2, \ldots, n \]
It is equivalent to finding $\beta_0$ and $\beta$ to

$$\minimizemize \frac{1}{2}\|\beta\|^2$$

subject to

$$y_i(\beta_0 + x_i^T \beta) \geq 1, i = 1, 2, ..., n$$

This is a convex optimization problem.

Since the constraints are $y_i(\beta_0 + X_i \beta) - 1 > 0$

We solve this problem using Lagrangian multipliers.

That is, we multiply the constraints by positive Lagrangian multipliers and subtract each such product from the objective function to from the primal function.

$$F_p(\beta_0, \beta, \alpha) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{n} \alpha_i \left( y_i(\beta_0 + x_i^T \beta) - 1 \right)$$

where $\alpha = (\alpha_1, ..., \alpha_n)^T \geq 0$ is the n-vector of non vector Lagrangian coefficients. We need to minimize the above function with respect to the primal variables $\beta_0$ and $\beta$. The resulting minimum should be maximized with respect to the dual variables $\alpha$.

The Karush-Kuhn Tucker conditions give necessary and sufficient conditions for a solution to a constrained optimization problem. For our primal problem $\beta_0$, $\beta$, and $\alpha$ have to satisfy:

$$\frac{\partial F_p(\beta_0, \beta, \alpha)}{\partial \beta_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \ldots \quad (6.4)$$

$$\frac{\partial F_p(\beta_0, \beta, \alpha)}{\partial \beta} = \beta - \sum_{i=1}^{n} \alpha_i y_i x_{i=0} \quad \ldots \quad (6.5)$$

$$y_i(\beta_0 + x_i^T \beta) - 1 \geq 0 \quad \ldots \quad (6.6)$$
\[ \alpha_i \geq 0 \quad \ldots (6.7) \]

\[ \alpha_i \left[ y_i (\beta_0 + x_i^T \beta) - 1 \right] = 0 \quad \ldots (6.8) \]

For \( i=1, 2, \ldots, n \), the equation (6.8) is known as the Karush-Kuhn-Tucker complementarily condition.

Solving equations (6.4) and (6.5) yields

\[ \sum_{i=1}^{n} \alpha_i y_i = 0 \quad \ldots (6.9) \]

\[ \beta^* = \sum_{i=1}^{n} \alpha_i y_i x_i \quad \ldots (6.10) \]

substituting equation (6.9) and (6.10) into primal objective function yields the minimum value of \( F \).

\( F_p(\beta_0, \beta, \alpha) \), namely,

\[
FD(\alpha) = \frac{1}{2} \left\| \beta^* \right\|^2 - \sum_{i=1}^{n} \alpha_i \left[ y_i (\beta_0^* + x_i^T \beta^*) - 1 \right] \\
= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) + \sum_{i=1}^{n} \alpha_i \\
\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i^T x_j) \quad \ldots (6.11) 
\]

note that the primal variables have been removed from the problem.

We next find the Lagrangian multipliers \( \alpha \) by maximizing the dual functional equation (6.11) subject to the constraints (6.7) and (6.9).

The constrained maximization problem can be written in matrix notation as follows. Find \( \alpha \) to

\[ \text{maximize} \quad F_D(\alpha) = 1^T \alpha - \frac{1}{2} \alpha^T H \alpha \]
subject to $\alpha \geq 0$, $\alpha^T y = 0$

where $y = (y_1, \ldots, y_n)^T$ and $H = (H_{ij})$ is a square $(n \times n)$ matrix with

$$H_{ij} = y_i y_j (x_i^T x_j).$$

If $\alpha$ solves this optimization problem, then

$$\hat{\beta} = \sum_{i=1}^{n} \alpha_i y_i x_i$$

yields the optimal weight vector. If $\alpha > 0$, then, from equation (6.7),

$$y_i (\beta_0^* + x_i^T \beta^*) = 1,$$

And so $x_i$ is a support vector for all observations that are not support vectors, $\alpha_i = 0$.

Let $sv \subseteq \{1, 2, \ldots, n\}$ be the subset of indices that identify the support vectors. Then, the optimal $\beta$ is given by

$$\hat{\beta} = \sum_{i \in sv} \alpha_i y_i x_i,$$

where the sum is taken only over the support vectors,

$$\hat{\beta} = \sum_{i \in sv} \alpha_i y_i x_i.$$

In other words, $\hat{\beta}$ is a linear function only of the support vectors $\{x_i, i \in sv\}$. The primal and dual optimization problem yields the same solution.

Finding the solution involves standard convex quadratic programming methods, and so any local minimum also turns out to be a global minimum. Although the optimal bias $\hat{\beta}_0$ is not determined explicitly by the optimization solution, we can estimate it by solving Kunh Tucker complementary condition for each support vector and then averaging the results. In other words, the estimated bias of the optimal hyperplane is given by
\[
\hat{\beta}_0 = \frac{1}{|sv|} \sum_{i \in sv} \left( 1 - y_i x_i^T \hat{\beta} \right)
\]

where \(|sv|\) is the number of support vectors in \(L\)

It follows that the optimal hyperplane can be written as

\[
\hat{f}(x) = \hat{\beta}_0 + x^T \hat{\beta} = \hat{\beta}_0 + \sum_{i \in sv} \hat{\alpha}_i y_i (x^T x_i) \quad \text{... (6.12)}
\]

clearly, only support vectors are relevant in computing the optimal separating hyperplane; observations that are not support vectors play no role in determining the hyperplane and are, thus, irrelevant to solving the optimization problem. The classification rule is given by

\[
C(x) = \text{sign}\{\hat{f}(x)\}
\]

if \(j \in sv\) then, from equation (6.12),

\[
y_j \hat{f}(x_j) = y_j \hat{\beta}_0 + \sum_{i \in sv} \hat{\alpha}_i y_i (x_j^T x_i) = 1 \quad \text{... (6.13)}
\]

hence, the squared – norm of the weight vector \(\hat{\beta}\) of the optimal hyperplane is

\[
\|\hat{\beta}\|^2 = \sum_{i \in sv} \sum_{j \in sv} \hat{\alpha}_i \hat{\alpha}_j y_i y_j (x_i^T x_j)
\]

\[
\sum_{j \in sv} \hat{\alpha}_j y_j \sum_{i \in sv} \hat{\alpha}_i y_i (x_j^T x_j)
\]

\[
\sum_{j \in sv} \hat{\alpha}_j (1 - y_j \hat{\beta}_0)
\]

by using (6.13)

\[
\sum_{j \in sv} \hat{\alpha}_j \quad \text{... (6.14)}
\]

it follows from (6.14) that the optimal hyperplane has maximum

\[
\text{margin} \quad 2\|\hat{\beta}\|, \quad \text{where} \quad \|\hat{\beta}\| = \left( \sum_{j \in sv} \hat{\alpha}_j \right)^{\frac{1}{2}}
\]
### 6.5 COMPUTATIONAL RESULTS

#### Table 6.1: Classification Using KNN

<table>
<thead>
<tr>
<th>KNN</th>
<th>subclinical</th>
<th>secondary</th>
<th>primary</th>
<th>nohypo</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclinical</td>
<td>14</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>Secondary</td>
<td>0</td>
<td>10</td>
<td>2</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Primary</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>Nohypo</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

#### Table 6.2: Classification Using SVM

<table>
<thead>
<tr>
<th>SVM</th>
<th>subclinical</th>
<th>secondary</th>
<th>primary</th>
<th>nohypo</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subclinical</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>Secondary</td>
<td>1</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>Primary</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Nohypo</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

All forty five observations were classified subclinical, secondary, primary, nohypo using KNN and SVM. The classification is as follows:

- Out of sixteen observations using KNN, fourteen were correctly classified from subclinical to subclinical, and others were misclassified. Similarly using SVM, fifteen were correctly classified from subclinical to subclinical and others misclassified.
Out of thirteen observations using KNN ten were correctly classified others were misclassified. SVM using eleven observations were correctly classified from ‘secondary to secondary and others misclassified.

Out of seven observations, using KNN four observations were correctly classified from ‘primary to primary’ and others were misclassified. Using SVM, five observations were correctly classified others misclassified.

Out of nine observations using KNN, six observations were correctly classified ‘nohypo to nohypo’ and others misclassified. Using SVM, six observations were correctly classified and others misclassified.

Table 6.3: Classification Using KNN and SVM

<table>
<thead>
<tr>
<th>No.</th>
<th>Diagnosis</th>
<th>KNN</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>2</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>3</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>4</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>5</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>6</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>7</td>
<td>nohypo</td>
<td>nohypo</td>
<td>nohypo</td>
</tr>
<tr>
<td>8</td>
<td>nohypo</td>
<td>subclinical</td>
<td>nohypo</td>
</tr>
<tr>
<td>9</td>
<td>primary</td>
<td>subclinical</td>
<td>primary</td>
</tr>
<tr>
<td>10</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>11</td>
<td>subclinical</td>
<td>primary</td>
<td>subclinical</td>
</tr>
<tr>
<td></td>
<td>subclinical</td>
<td>subclinical</td>
<td>nohypo</td>
</tr>
<tr>
<td>----</td>
<td>-------------</td>
<td>-------------</td>
<td>--------</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>15</td>
<td>subclinical</td>
<td>primary</td>
<td>subclinical</td>
</tr>
<tr>
<td>16</td>
<td>primary</td>
<td>primary</td>
<td>primary</td>
</tr>
<tr>
<td>17</td>
<td>nohypo</td>
<td>nohypo</td>
<td>nohypo</td>
</tr>
<tr>
<td>18</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>19</td>
<td>primary</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>20</td>
<td>primary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>21</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>22</td>
<td>nohypo</td>
<td>primary</td>
<td>nohypo</td>
</tr>
<tr>
<td>23</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>24</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>25</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>26</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>27</td>
<td>subclinical</td>
<td>subclinical</td>
<td>secondary</td>
</tr>
<tr>
<td>28</td>
<td>nohypo</td>
<td>nohypo</td>
<td>nohypo</td>
</tr>
<tr>
<td>29</td>
<td>subclinical</td>
<td>subclinical</td>
<td>nohypo</td>
</tr>
<tr>
<td>30</td>
<td>secondary</td>
<td>secondary</td>
<td>nohypo</td>
</tr>
<tr>
<td>31</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
<tr>
<td>32</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>33</td>
<td>primary</td>
<td>subclinical</td>
<td>primary</td>
</tr>
<tr>
<td>34</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>35</td>
<td>subclinical</td>
<td>subclinical</td>
<td>subclinical</td>
</tr>
<tr>
<td>36</td>
<td>nohypo</td>
<td>subclinical</td>
<td>nohypo</td>
</tr>
<tr>
<td>37</td>
<td>secondary</td>
<td>secondary</td>
<td>secondary</td>
</tr>
</tbody>
</table>
### 6.6 CONCLUSION

This work focused the implementation of k-Nearest Neighbor and support vector machines for the hypothyroidism data. From the analysis, it is examined that the formation of classifications will be different for both classification methods. From the histogram, it is seen that the Support Vector Machine accuracy is 82%. KNN gives the
accuracy of 76%. Support vector machines are higher than the accuracy of KNN.

6.7 CLASSIFICATION BY DECISION TREE INDUCTION

Decision tree algorithm is developed by Ross Quinlan in 1975. A decision tree is a flow–chart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distribution. The topmost node in a tree is the root node. Internal nodes are denoted by rectangle, and leaf nodes are denoted by ovals.

In order to classify an unknown sample, the attribute values of the sample are tested against the decision tree. A path is traced from the root to a leaf node which holds the class prediction for that sample.

EXTRACTING CLASSIFICATION RULES FROM DECISION TREES

The knowledge represented in decision trees can be extracted and represented in the form of classification IF – THEN rules. One rule is created for each path from the root to a leaf node. Each attribute–value pair along a given path forms a conjunction in the rule antecedent (“IF” part). The leaf node holds the class prediction, forming the rule consequent (“THEN” part). The IF – THEN rules may be easier for humans to understand, particularly if the given tree is very large.
Tree Pruning: When a decision tree is built, many of the branches will reflect anomalies in the training data due to noise or outliers. Tree pruning methods address this problem of over-fitting the data. Such methods typically use statistical measures to remove the least reliable branches, generally resulting in faster classification and an improvement in the ability of the tree to correctly classify test data.

There are two common approaches to tree pruning.

- Pre-pruning
- Post-pruning

Pre-pruning: In the pre-pruning approach, a tree is “pruned” by halting its construction early (e.g., by deciding not to further split or partition the subset of training samples at a given node. Upon halting, the node becomes a leaf. The leaf may hold the most frequent class among the subset samples, or the probability distribution of those samples.

When constructing a tree, measures such as statistical significance, Chi-Square, information gain etc. can be used to assess the goodness of a split.

Post-pruning: The post-pruning approach removes branches from a “fully grown” tree. A tree node is pruned by removing its branches. In this work, post-pruning technique is applied.
6.8 FUZZY SETS AND MEMBERSHIP FUNCTION

In classical, or crisp, sets the transition for an element in the universe between membership and non-membership in a given set is abrupt and well defined (said to be crisp). For an element in a universe that contains fuzzy sets, this transition can be gradual. This transition among various degrees of membership can be thought of as conformation to the fact that the boundaries of the fuzzy sets are vague and ambiguous. Hence, membership of an element from the universe in this set is measured by a function that attempts to describe vagueness and ambiguity.

A fuzzy set, then, is a set containing elements that have varying degrees of membership in the set. This idea is in contrast with classical, or crisp, sets because members of a crisp set would not be members unless their membership was full, or complete, in that set (i.e., their membership is assigned a value of 1). Elements in a fuzzy set, because their membership need not be complete, can also be members of other fuzzy sets on the same universe.

The idea proposed by Lotfi Zadeh in (1965) suggested that set membership is the key to decision making when faced with uncertainty.

MEMBERSHIP FUNCTION

The membership function Ross (2004) embodies the mathematical representation of membership in a set, and the notation used throughout this text for a fuzzy set is a set symbol with a tilde.
underscore, say $A$, where the functional mapping is given by 
$\mu_{A}(x) \in [0,1]$ and the symbol $\mu_{A}(x)$ is the membership of element $x$ in fuzzy set $A$. Therefore, $\mu_{A}(x)$ is a value on the unit interval that measures the degree to which element $x$ belongs to fuzzy set $A$; equivalently, $\mu_{A}(x) = \text{degree to which } x \in A$.

The simplest membership functions are formed using straight lines. Of these, the simplest is the triangular membership function, named as trimf. It is simply a collection of three points forming a triangle. The trapezoidal membership function, trapmf, has a flat top and is just a truncated triangle curve. These straight line membership functions have the advantage of its simplicity.

![Figure 6.3: Triangular and Trapezoidal Membership Function](image)

6. 9 ATTRIBUTE SELECTION MEASURE

The information gain measure is used to select the test attribute at each node in the tree. Such a measure is referred to as an attribute
selection measure or a measure of the goodness of split. The attribute with the highest information gain is chosen as the test attribute for the current node. This attribute minimizes the information the information needed to classify the samples in the resulting in the resulting partitions and reflects the least randomness or impurity in these partitions. Such an information theoretic approach minimizes the expected number tests needed to classify an object and guarantees that a simple tree is found Gorunescu (2010).

Let \( S \) be a set consisting of \( s \) data samples. Suppose the class label attribute has \( m \) distinct values defining \( m \) distinct classes, \( C_i \) (for \( i=1,...,m \)). Let \( s_i \) be the number of samples of \( S \) in class \( C_i \). The expected information needed to classify a given sample is given by

\[
I(s_1,s_2,...,s_m) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{6.15}
\]

where \( p_i \) is the probability than an arbitrary sample belongs to class \( C_i \) and is estimated by \( s_i/s \). The log function to the base 2 is used since the information is encoded in bits.

Let attribute \( A \) have \( v \) distinct values, \( \{a_1, a_2, ..., a_v\} \). Attribute \( A \) can be used to partition \( S \) into \( v \) subsets, \( \{S_1, S_2, ..., S_v\} \), where \( S_j \) contains those samples in \( S \) that have value \( a_j \) of \( A \). If \( A \) were selected as the test attribute, then these subsets would correspond to the branches grown from the node containing the set \( S \). Let \( s_{ij} \) be the number of samples in \( S \). Let \( s_{ij} \) be the number of samples of class \( C_i \) in a subset \( S_j \). The entropy, or expected information based on the partitioning into subsets by \( A \) is given by
The term \( \sum_{j=1}^{v} \frac{s_{1j} + \cdots + s_{mj}}{s_j} \) acts as the weight of the \( j \)th subset and is the number of samples in the subset divided by the total number of samples in \( S \). The smaller the entropy value is, the greater the purity of the subset partitions. The encoding information that would be gained by branching on \( A \) is

\[
\text{Gain (A)} = I(s_1, s_2, \ldots, s_m) - E(A) \quad \text{...(6.17)}
\]

In other words, \( \text{Gain (A)} \) is the expected reduction in entropy caused by knowing the value of the attribute \( A \). The algorithm computes the information of each attribute. The attribute with the highest information gain is chosen as the test attribute for the given set \( S \). A node is created and labeled with the attribute, branches are created for each value of the attribute, and the samples are partitioned accordingly.

### 6.10 BAYESIAN CLASSIFICATION

Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given sample belongs to a particular class.

Bayesian classification is based on Bayes theorem. A simple Bayesian classifier is known as the naïve Bayesian classifier which is used to compare the performance with decision tree and neural network classifiers. Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.
NAIVE BAYESIAN CLASSIFICATION

Let D be a training set of tuples and their associated class labels. Each tuple is represented by an n-dimensional attribute vector, \( X = (X_1, X_2, \ldots, X_n) \), depicting n measurements made on the tuple from n attributes, respectively, \( A_1, A_2, \ldots, A_n \).

Suppose that there are m classes, \( C_1, C_2, \ldots, C_m \). Given an unknown data sample, \( X \), the classifier will predict that \( X \) belongs to the class having the highest posterior probability, conditioned on \( X \). Thus we maximize \( p(C_i | X) \). The class \( C_i \) for which \( p(C_i | X) \) is maximized is called the maximum posterior probabilities.

\[
p(C_i | X) = \frac{p(X | C_i)p(C_i)}{p(X)}
\]

... (6.18)
equation (6.18) derived from the Bayes theorem. As \( p(X) \) is constant for all classes, only \( p(X | C_i)p(C_i) \) need be maximized.

6.11 COMPUTATIONAL RESULTS

The three input parameters symptom score, T4 and TSH. There are three types of symptom score low, normal and high depending on the patient’s symptoms and its severity and its family history etc. The other variables, T4 range is 0.6 to 2.0 ng dL\(^{-1}\) and TSH 0.5 to 4.6 mlU mL\(^{-1}\).
The shape of the membership function used for this low and high are of trapezoidal type and normal for triangular type.

**Table 6.4: Classification Rule**

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Symptom Score</th>
<th>T4</th>
<th>TSH</th>
<th>Consequence Hypothyroidism</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>high</td>
<td>Low</td>
<td>Low</td>
<td>secondary</td>
</tr>
<tr>
<td>2</td>
<td>high</td>
<td>Low</td>
<td>normal</td>
<td>secondary</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>Low</td>
<td>High</td>
<td>primary</td>
</tr>
<tr>
<td>4</td>
<td>high</td>
<td>Normal</td>
<td>Low</td>
<td>secondary</td>
</tr>
<tr>
<td>5</td>
<td>high</td>
<td>Normal</td>
<td>High</td>
<td>subclinical</td>
</tr>
<tr>
<td>6</td>
<td>low</td>
<td>Normal</td>
<td>normal</td>
<td>nohypo</td>
</tr>
<tr>
<td>7</td>
<td>normal</td>
<td>Low</td>
<td>Low</td>
<td>secondary</td>
</tr>
<tr>
<td>8</td>
<td>normal</td>
<td>Low</td>
<td>normal</td>
<td>secondary</td>
</tr>
<tr>
<td>9</td>
<td>normal</td>
<td>Low</td>
<td>High</td>
<td>primary</td>
</tr>
<tr>
<td>10</td>
<td>normal</td>
<td>Normal</td>
<td>Low</td>
<td>secondary</td>
</tr>
<tr>
<td>11</td>
<td>normal</td>
<td>Normal</td>
<td>High</td>
<td>subclinical</td>
</tr>
<tr>
<td>12</td>
<td>normal</td>
<td>Normal</td>
<td>normal</td>
<td>nohypo</td>
</tr>
<tr>
<td>13</td>
<td>low</td>
<td>Low</td>
<td>Low</td>
<td>secondary</td>
</tr>
<tr>
<td>14</td>
<td>low</td>
<td>Low</td>
<td>Normal</td>
<td>secondary</td>
</tr>
<tr>
<td>15</td>
<td>low</td>
<td>Low</td>
<td>High</td>
<td>primary</td>
</tr>
<tr>
<td>16</td>
<td>low</td>
<td>Normal</td>
<td>Low</td>
<td>secondary</td>
</tr>
<tr>
<td>17</td>
<td>low</td>
<td>Normal</td>
<td>High</td>
<td>subclinical</td>
</tr>
<tr>
<td>18</td>
<td>high</td>
<td>Normal</td>
<td>Normal</td>
<td>subclinical</td>
</tr>
</tbody>
</table>
Figure 6.5: Attribute Selection Measure

The decision trees for the attribute T4 has the highest information and therefore becomes a test attribute at the root node of the decision tree. Branches are grown for each value of T4.
The naive Bayes classification for predict unknown sample X.

\[ X = (\text{symptom} = \text{high}, T4= \text{normal}, TSH= \text{high}) \]

\[ P(X | \text{subclinical}) = 0.6233 \]

\[ P(X | \text{secondary}) = 0 \]

\[ P(X | \text{primary}) = 0 \]

\[ P(X | \text{nohypo}) = 0 \]

\[ P(X | \text{symptom= subclinical})p(\text{symptom= subclinical}) = 0.2632 \]

The naive Bayesian classifier predicts “Diagnosis = subclinical” for sample X.

\[ X = (\text{symptom} = \text{low}, T4= \text{normal}, TSH= \text{low}) \]

\[ P(X | \text{nohypo}) = 0.125 \]

\[ P(X | \text{subclinical}) = 0 \]

\[ P(X | \text{secondary}) = 0 \]

\[ P(X | \text{primary}) = 0 \]

\[ P(X | \text{symptom= nohypo})p(\text{symptom= nohypo}) = 0.0222. \]

The naive Bayesian classifier predicts “Diagnosis = nohypo” for sample X.

6.12 CONCLUSION

This work focused the implementation of attribute selection measure for decision trees and the naive Bayes classification for the hypothyroidism data. From the analysis, it is evident that the classification formed is formation of classifications different for both the methods. In decision trees for attribute selection measure, when compared with other variables T4 has the highest information gain.
In naive Bayes classification the representative probabilities are predicted for an unknown sample. In general, Bayesian classifiers have the minimum error rate in comparison to all other classifiers. Naive Bayes classifier is better than the decision trees.