BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

UNDERGRADUATE THESIS

Knowledge Discovery From Academic Data Using Data Mining Technique

Author:
Md.Mostafizur RAHMAN
Sabid Bin Habib

Supervisor:
Dr. ASM Latiful Hoque

Department of Computer Science And Engineering

February 22, 2016

Declaration of Authorship

We, Md.Mostafizur Rahman and Sabid Bin Habib, declare that this thesis titled, "Knowledge Discovery From Academic Data Using Data Mining Technique" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ____

Date: ____

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Abstract

Department of Computer Science And Engineering

Undergraduate Thesis

Knowledge Discovery From Academic Data Using Data Mining Technique

by Md.Mostafizur RAHMAN Sabid Bin Habib

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Acknowledgements

The acknowledgments and the people to thank go here, don't forget to include your project advisor…

Contents

Declaration of Authorship **iii**

Abstract **v**

Acknowledgements **vii**

1 Introduction **1**

2 Literature Study **3**
2.1 Knowledge Discovery Steps 3
2.2 Data Mining Concepts 3
2.3 Preprocessing 3
2.4 Classification 3
2.4.1 Basic Concept 3
2.4.2 General Approach to Classification 3
2.4.3 Decision Tree Induction 4
2.4.4 Decision Tree Induction Algorithm 5
2.4.5 Attributes Selection Measures 6
Information Gain 7
Gain Ratio 8
Gini Index 8
Other Attribute Selection Measures 9

3 Analysis of BIIS Data **11**
3.1 Scope 11
3.2 Database Structure 11
3.3 Problems in Existing Structure 11

4 Preprocessing **13**
4.1 Technique And Design 13
4.2 Algorithms 13
4.2.1 Cleaning 13
4.2.2 Reduction 13
4.2.3 Integration 13
4.2.4 Normalization 13
4.3 Results 13

5 Classification **15**
5.1 Decision Tree 15
5.2 Algorithm 17
5.3 Result of Classification 19
5.4 Statistical Analysis 19
5.5 Result of Statistical Analysis 20
5.5.1 Department wise Performance 20

EEE Department 20
CSE Department 20
IPE Department 22
ME Department 22
CE Department 23
MME Department 23
CHE Department 25
NAME Department 25
URP Department 26
ARCH Department 26
WRE Department 28

6 Conclusion **29**
6.1 Summary of Thesis 29
6.2 General Findings 29
6.3 Future Works 29

A Appendix Title Here **31**

List of Figures

2.1 Classification 4
2.2 A Decision Tree 5

5.1 A Decision Tree From Training Data 16
5.2 Performance of EEE Department 20
5.3 Performance of CSE Department 21
5.4 Performance of IPE Department 22
5.5 Performance of ME Department 23
5.6 Performance of CE Department 24
5.7 Performance of MME Department 24
5.8 Performance of CHE Department 25
5.9 Performance of NAME Department 26
5.10 Performance of URP Department 27
5.11 Performance of ARCH Department 27
5.12 Performance of WRE Department 28

List of Tables

5.1 Training Data Sample 16
5.2 Training Data Sample 19
5.3 Final Output Sample 19
5.4 Performance of EEE department 20
5.5 Performance of CSE Department 21
5.6 Performance of IPE Department 22
5.7 Performance of ME Department 22
5.8 Performance of CE Department 23
5.9 Performance of MME Department 23
5.10 Performance of CHE Department 25
5.11 Performance of NAME Department 25
5.12 Performance of URP Department 26
5.13 Performance of ARCH Department 26
5.14 Performance of WRE Department 28

For/Dedicated to/To my…

Chapter 1

Introduction

Chapter 2

Literature Study

2.1 Knowledge Discovery Steps

2.2 Data Mining Concepts

2.3 Preprocessing

2.4 Classification

2.4.1 Basic Concept

Classification is a form of data analysis that extracts models describing important data classes. Such models,called classifiers, predict categorical (discrete, unordered) class labels. For example, we can build a classification model to categorize bank loan applications as either safe or risky. Such analysis can help provide us with a better understanding of the data at large. Many classification methods have been proposed by researchers in machine learning, pattern recognition, and statistics. Most algorithms are memory resident, typically assuming a small data size. Recent data mining research has built on such work, developing scalable classification and prediction techniques capable of handling large amounts of disk-resident data. Classification has numerous applications, including fraud detection, target marketing, performance prediction, manufacturing, and medical diagnosis.

The data analysis task is classification, where a model or classifier is constructed to predict class (categorical) labels.This model is a predictor.Regression analysis is a statistical methodology that is most often used for numeric prediction; hence the two terms tend to be used synonymously, although other methods for numeric prediction exist. Classification and numeric prediction are the two major types of prediction problems.

2.4.2 General Approach to Classification

Data classification is a two-step process, consisting of a learning step (where a classification model is constructed) and a classification step (where the model is used to predict class labels for given data).

In the first step, a classifier is built describing a predetermined set of data classes or concepts. This is the learning step (or training phase), where a classification algorithm builds the classifier by analyzing or "learning from"

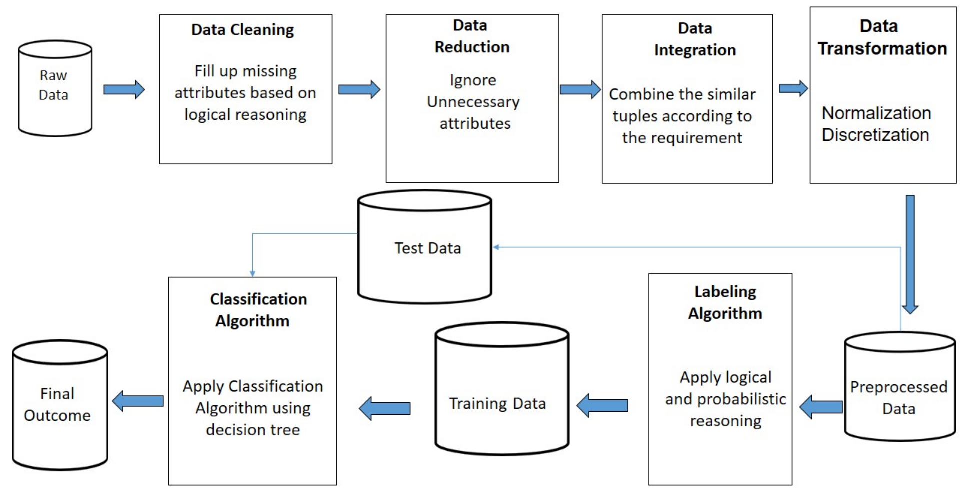

FIGURE 2.1: General Process of Classification

a training set made up of database tuples and their associated class labels. A tuple, X, is represented by an n-dimensional attribute vector, $X = (x_1, x_2, ..., x_n)$ depicting n measurements made on the tuple from n database attributes, respectively,$A_1, A_2, ...A_n$.[1] Each tuple, X, is assumed to belong to a predefined class as determined by another database attribute called the class label attribute. The class label attribute is discrete-valued and unordered. It is categorical (or nominal) in that each value serves as a category or class. The individual tuples making up the training set are referred to as training tuples and are randomly sampled from the database under analysis. In the context of classification, data tuples can be referred to as samples, examples, instances, data points, or objects.[2]

Figure 2.1 shows the general procedure of classification.

2.4.3 Decision Tree Induction

Decision tree induction is the learning of decision trees from class-labeled training tuples. A decision tree is a flowchart-like tree structure, where each internal node (nonleaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node. The decision tree in Figure 2.2 is a tree for the concept buy_computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class. The benefits of having a decision tree are

- It does not require any domain knowledge.
- is easy to comprehend.

[1] Each attribute represents a "feature" of X. Hence, the pattern recognition literature uses the term feature vector rather than attribute vector.

[2] In the machine learning literature, training tuples are commonly referred to as training samples. Throughout this text, we prefer to use the term tuples instead of samples.

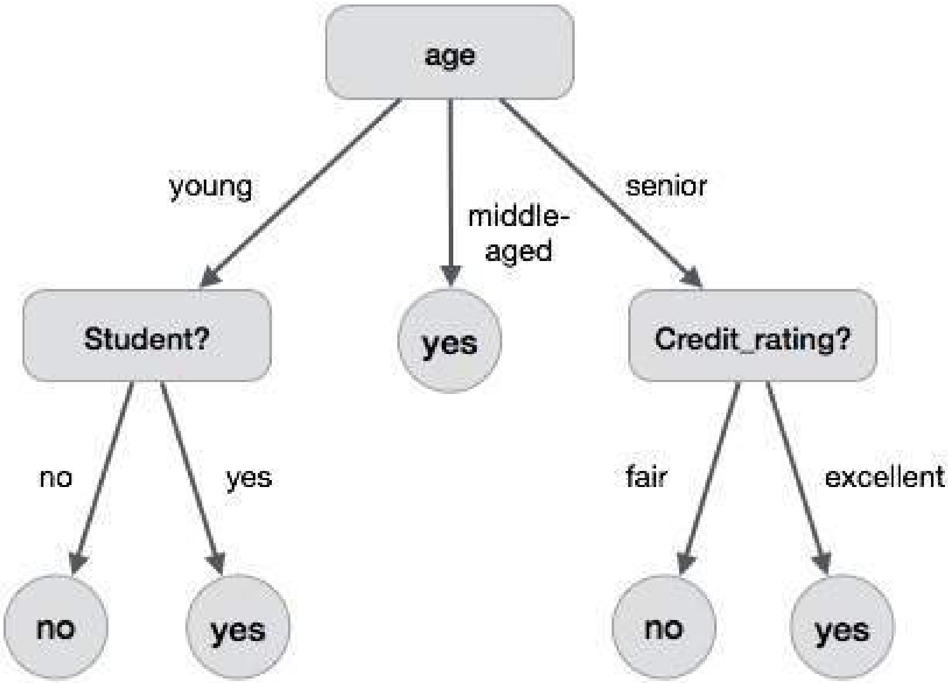

FIGURE 2.2: A Decision Tree

- The learning and classification steps of a decision tree are simple and fast.

2.4.4 Decision Tree Induction Algorithm

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algorithm, there is no backtracking; the trees are constructed in a top-down recursive divide-and-conquer manner.

Input :

- Data partition, D, which is a set of training tuples and their associated class labels;
- $attribute\_list$, the set of candidate attributes;
- $Attribute\_selection\_method$.

Output: A decision tree.

1: **procedure**
2: create a node N;
if *tuples in* D *are all of the same class,* C, **then**
return N as leaf node labeled with the class C ;
end
if $attribute\_list$ *is empty* **then**
return N as leaf node labeled with the majority class in D ;
end
3: apply $Attribute\_selecction\_method(D, attribute\_list)$ to find best splittingcriterion ;
4: label node N with $splitting_criterion$;
if $splitting\_criterion$ *is discrete valued and multiway splits allowed* **then**
$attribute\_list \leftarrow attribute\_list - splitting\_ttribute$;
end
for *each outcome* j *of* $splitting\_criterion$ **do**
5: let D_j be the set of data tuples in D satisfying outcome j ;
if $D_j isempty$ **then**
attach a leaf labeled with the majority class in D to node N ;
end
else
attach a leaf labeled with the majority class in D to node N ;
end
end
6: return N ;
7: **end procedure**

Algorithm 1: Generate_decision_tree

Algorithm 1 is the general approach to build a decision tree.

2.4.5 Attributes Selection Measures

An attribute selection measure is a heuristic for selecting the splitting criterion that "best" separates a given data partition, D, of class-labeled training tuples into individual classes. If we were to split D into smaller partitions according to the outcomes of the splitting criterion, ideally each partition would be pure (i.e., all the tuples that fall into a given partition would belong to the same class). Conceptually, the "best" splitting criterion is the one that most closely results in such a scenario. Attribute selection measures are also known as splitting rules because they determine how the

tuples at a given node are to be split.
The attribute selection measure provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure[3] is chosen as the splitting attribute for the given tuples. If the splitting attribute is continuous-valued or if we are restricted to binary trees, then, respectively, either a split point or a splitting subset must also be determined as part of the splitting criterion. The tree node created for partition D is labeled with the splitting criterion, branches are grown for each outcome of the criterion, and the tuples are partitioned accordingly.Three popular attribute selection measures are

- Information gain
- Gain ration
- Gini index

The notation used herein is as follows. Let D, the data partition, be a training set of class labeled tuples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for $i = 1, 2, ...$). Let be the set of tuples of class C_i in D. Let $|D|$ and $|C_{i,D}|$ denote the number of tuples in D and $C_{i,D}$ respectively.

Information Gain

ID3 uses information gain as its attribute selection measure. This measure is based on pioneering work by Claude Shannon on information theory, which studied the value or "information content" of messages. Let node N represent or hold the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify the tuples in the resulting partitions and reflects the least randomness or "impurity" in these partitions. Such an approach minimizes the expected number of tests needed to classify a given tuple and guarantees that a simple (but not necessarily the simplest) tree is found.

The expected information needed to classify a tuple in D is given by

$$Info(D) = -\sum(p_i * \log(p_i))$$

where p_i is the nonzero probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. A log function to the base 2 is used, because the information is encoded in bits. $Info(D)$ is just the average amount of information needed to identify the class label of a tuple in D. Note that, at this point, the information we have is based solely on the proportions of tuples of each class. $Info(D)$ is also known as the entropy of D. How much more information would we still need (after the partitioning) to arrive at an exact classification? This amount is measured by

$$Info_A(D) = \sum \frac{|D_j|}{|D|} * Info(D_j)$$

[3]Depending on the measure, either the highest or lowest score is chosen as the best (i.e., some measures strive to maximize while others strive to minimize).

nformation gain is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A). That is,

$$Gain(A) = Info(D) - Info_A(D)$$

Gain Ratio

The information gain measure is biased toward tests with many outcomes. That is, it prefers to select attributes having a large number of values. For example, consider an attribute that acts as a unique identifier such as $product\_ID$. A split on $product\_ID$ would result in a large number of partitions (as many as there are values), each one containing just one tuple. Because each partition is pure, the information required to classify data set D based on this partitioning would be $Info_product\_ID(D) = 0$. Therefore, the information gained by partitioning on this attribute is maximal. Clearly, such a partitioning is useless for classification.

C4.5, a successor of ID3, uses an extension to information gain known as gain ratio, which attempts to overcome this bias. It applies a kind of normalization to information gain using a "split information" value defined analogously with $Info(D)$ as

$$SplitInfo_A(D) = -\sum(\frac{|D_j|}{|D|} * \log_2(\frac{|D_j|}{|D|}))$$

This value represents the potential information generated by splitting the training data set, D, into v partitions, corresponding to the v outcomes of a test on attribute A. Note that, for each outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D. It differs from information gain, which measures the information with respect to classification that is acquired based on the same partitioning. The gain ratio is defined as

$$GainRatio = \frac{Gain(A)}{SplitInfo_A(D)}$$

Gini Index

The Gini index is used in CART. Using the notation previously described, the Gini index measures the impurity of D, a data partition or set of training tuples, as

$$Gini(D) = 1 - \sum(p_i^2)$$

where pi is the probability that a tuple in D belongs to class C_i and is estimated by $|C_{i,D}|/|D|$. The sum is computed over m classes.

When considering a binary split, we compute a weighted sum of the impurity of each resulting partition. For example, if a binary split on A partitions D into D_1 and D_2, the Gini index of D given that partitioning is

$$Gini_A(D) = \frac{|D_1|}{|D|}Gini(D_1) + \frac{|D_2|}{|D|}Gini(D_2).$$

The reduction in impurity that would be incurred by a binary split on a discrete- or continuous-valued attribute A is

$$\delta Gini(A) = Gini(D) - Gini_A(D).$$

Other Attribute Selection Measures

Many other attribute selection measures have been proposed. CHAID, a decision tree algorithm that is popular in marketing, uses an attribute selection measure that is based on the statistical χ^2 test for independence. Other measures include C-SEP (which performs better than information gain and the Gini index in certain cases) and G-statistic (an information theoretic measure that is a close approximation to χ^2 distribution).
Attribute selection measures based on the Minimum Description Length (MDL) principle have the least bias toward multivalued attributes. MDL-based measures use encoding techniques to define the "best”" decision tree as the one that requires the fewest number of bits to both (1) encode the tree and (2) encode the exceptions to the tree(i.e., cases that are not correctly classified by the tree). Its main idea is that the simplest of solutions is preferred.
Other attribute selection measures consider multivariate splits (i.e., where the partitioning of tuples is based on a combination of attributes, rather than on a single attribute). The CART system, for example, can find multivariate splits based on a linear combination of attributes. Multivariate splits are a form of attribute (or feature) construction, where new attributes are created based on the existing ones. (Attribute construction was also discussed in

Chapter 3

Analysis of BIIS Data

3.1 Scope

3.2 Database Structure

3.3 Problems in Existing Structure

Chapter 4

Preprocessing

4.1 Technique And Design

4.2 Algorithms

4.2.1 Cleaning

4.2.2 Reduction

4.2.3 Integration

4.2.4 Normalization

4.3 Results

Chapter 5

Classification

There are five class labels in our classification model.They are

- Excellent
- Good
- Moderate
- Poor
- Very Poor

Applying ID3 classification algorithm a decision tree was created using training data and this knowledge in decision tree was used to find the class label of test data set.

5.1 Decision Tree

The training data was prepared after data preprocessing as described in Chaper 4.The final attributes in the training data are

- Student Id
- Department
- Hall Status
- Gender
- Attendance marks
- Class test marks
- Earned CGPA
- Completed Credit
- Final status according to our reasoning as Status

A sample of the training data set looks like Table 5.1.

A hypothetical decision tree can be derived from the training data as depicted in Figure 5.1.

TABLE 5.1: Training Data Sample

SID	Department	Hall	Gender	Attendance	ClassTest	Cgpa	Credit	Status
4650	2	0	1	1	0.83	3.72	1	excellent
4755	9	1	1	0.82	0.71	3.39	1	poor
4769	1	1	1	0.97	0.83	3.76	0.98	moderate
4975	3	1	0	0.99	0.76	3.50	1	good
5064	10	0	1	0.33	0.33	2.61	0.75	very poor

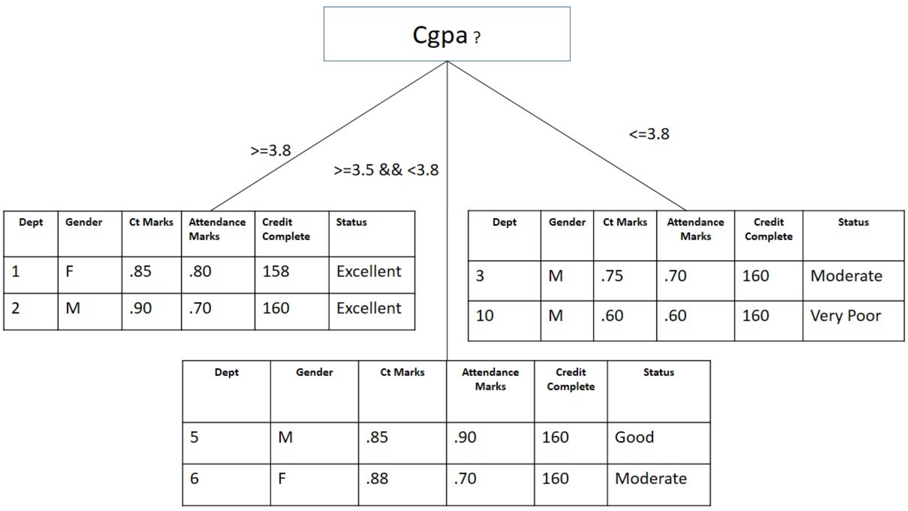

FIGURE 5.1: A Decision Tree From Training Data

5.2 Algorithm

The algorithm used is ID3 Algorithm. So, Information gain as described in Section 2.4.5 is used as splitting criterion.
We used the structure of the algorithm described in Algorithm 1 in our implementation. We used Java programming language for implementation.

The pseudocode for the final implementation of the algorithm is shown

Input :

- Data partition, D, which is a set of training tuples and their associated class labels;
- $attribute\_list(SID, Department, Hall, Attendance, ClassTest, Cgpa, CreditComplete)$, the set of candidate attributes;
- $Attribute\_selection\_method : Informationgainwithmajorityvoting.$

Output: A decision tree.

procedure
create a node N;
if *tuples in* D *are all of the same class,* C, **then**
return N as leaf node labeled with the class C ;
end
if $attribute\_list$ *is empty* **then**
return N as leaf node labeled with the majority class in D ;
end
apply $Attribute\_selecction\_method(D, attribute\_list)$ to find best splittingcriterion ;
label node N with $splitting_criterion$;
if $splitting\_criterion$ *is Dept or Hall_Status or Gender* **then**
split according to the discrete values of the attribure;
$attribute\_list \leftarrow attribute\_list - splitting\_attribute$;
end
else
▷ $splitting\_criterion$ is Cgpa or Attendance or Classtest or CreditCompleted.
split 3 ways according to the values of the attribute for example for Cgpa divide at 3.8 and 3.5 into 3 parts;
$attribute\_list \leftarrow attribute\_list - splitting\_attribute$;
end
for *each outcome* j *of* $splitting\_criterion$ **do**
let D_j be the set of data tuples in D satisfying outcome j ;
if $D_j isempty$ **then**
attach a leaf labeled with the majority class in D to node N ;
end
else
attach a leaf labeled with the majority class in D to node N ;
end
end
return N ;
end procedure

Algorithm 2: Generate_decision_tree_For_BIIS_Data

TABLE 5.2: Training Data Sample

SID	Department	Hall	Gender	Attendance	ClassTest	Cgpa	Credit
4487	9	1	1	0.98	0.77	3.64	1
4488	9	0	0	0.72	0.68	3.18	1
4489	9	0	0	0.93	0.69	3.49	1
4490	9	1	1	0.61	0.59	3.02	0.96
4492	11	1	1	0.99	0.72	3.06	0.99
4493	11	1	1	0.71	0.68	3.07	0.91
4488	9	0	0	0.72	0.68	3.18	1
4489	9	0	0	0.93	0.69	3.49	1
4490	9	1	1	0.61	0.59	3.02	0.96
4492	11	1	1	0.99	0.72	3.06	0.99
4493	11	1	1	0.71	0.68	3.07	0.91
4494	11	1	0	0.89	0.78	3.25	0.99
4496	11	0	1	0.98	0.73	3.22	0.99

TABLE 5.3: Final Output Sample

SID	Department	Hall	Gender	Attendance	ClassTest	Cgpa	Credit	Status
4487	9	1	1	0.98	0.7 7	3.64	1	good
4488	9	0	0	0.72	0.68	3.18	1	moderate
4489	9	0	0	0.93	0.69	3.49	1	good
4490	9	1	1	0.61	0.59	3.02	0.96	poor
4492	11	1	1	0.99	0.72	3.06	0.99	moderate
4493	11	1	1	0.71	0.68	3.07	0.91	poor
4488	9	0	0	0.72	0.68	3.18	1	moderate
4489	9	0	0	0.93	0.69	3.49	1	good
4490	9	1	1	0.61	0.59	3.02	0.96	very poor
4492	11	1	1	0.99	0.72	3.06	0.99	poor
4493	11	1	1	0.71	0.68	3.07	0.91	very poor
4494	11	1	0	0.89	0.78	3.25	0.99	moderate
4496	11	0	1	0.98	0.73	3.22	0.99	moderate

5.3 Result of Classification

Test data set before applying algorithm looks like Table 5.2.

After applying algortihm as desribed in Algorithm 2 the results as shown in Table 5.3 are found.

5.4 Statistical Analysis

We analyzed the final results and found some relevant statistical results. The categories are

- Department wise performance
- Impact of gender on performance
- Impact of hall status on performance

TABLE 5.4: Performance of EEE department

Class Label	Percent
Excellent	18%
Good	42%
Moderate	32%
Poor	5%
Very Poor	3%

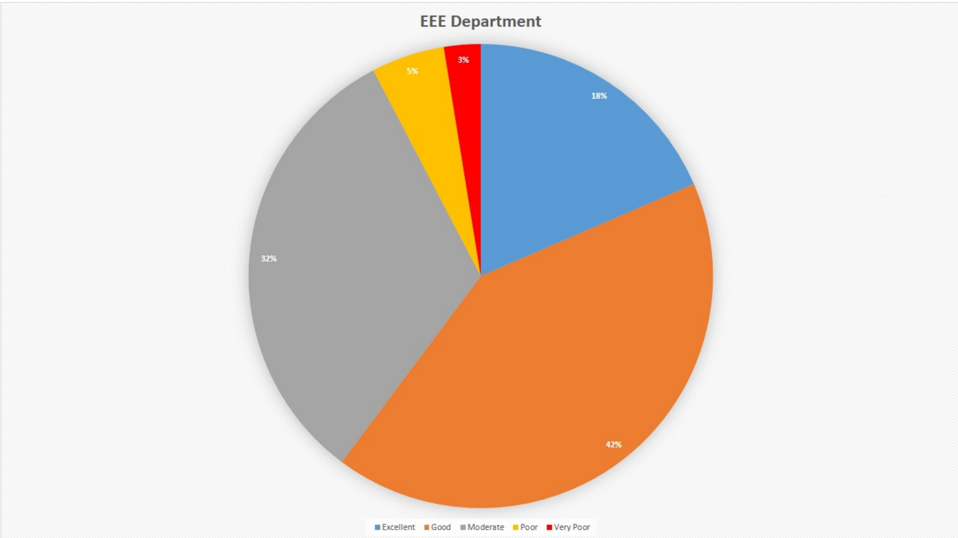

FIGURE 5.2: Performance of EEE Department

- Impact of classtest marks on performance
- Impact of attendance marks on performance
- Impact of cgpa on performance
- Impact of credit completion on performance

The details of the findings are discussed in Section 5.5

5.5 Result of Statistical Analysis

5.5.1 Department wise Performance

EEE Department

The overall performance of EEE department is shown in Figure 5.2. According to our classifier the percentage of each class label of EEE department is shown in Table 5.4

CSE Department

The overall performance of CSE department is shown in Figure 5.3. According to our classifier the percentage of each class label of CSE department is shown in Table 5.5

TABLE 5.5: Performance of CSE Department

Class Label	Percent
Excellent	21%
Good	32%
Moderate	12%
Poor	30%
Very Poor	5%

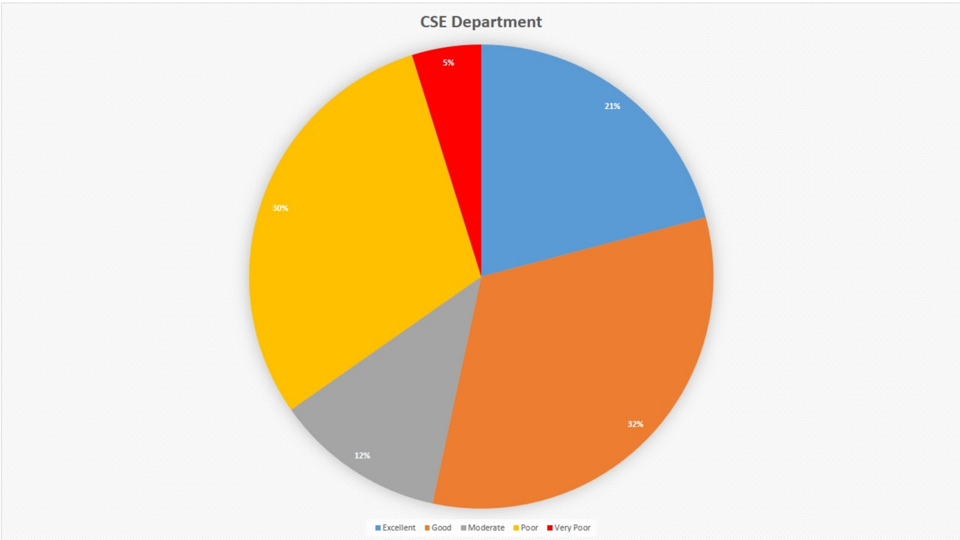

FIGURE 5.3: Performance of CSE Department

TABLE 5.6: Performance of IPE Department

Class Label	Percent
Excellent	13%
Good	33%
Moderate	17%
Poor	23%
Very Poor	14%

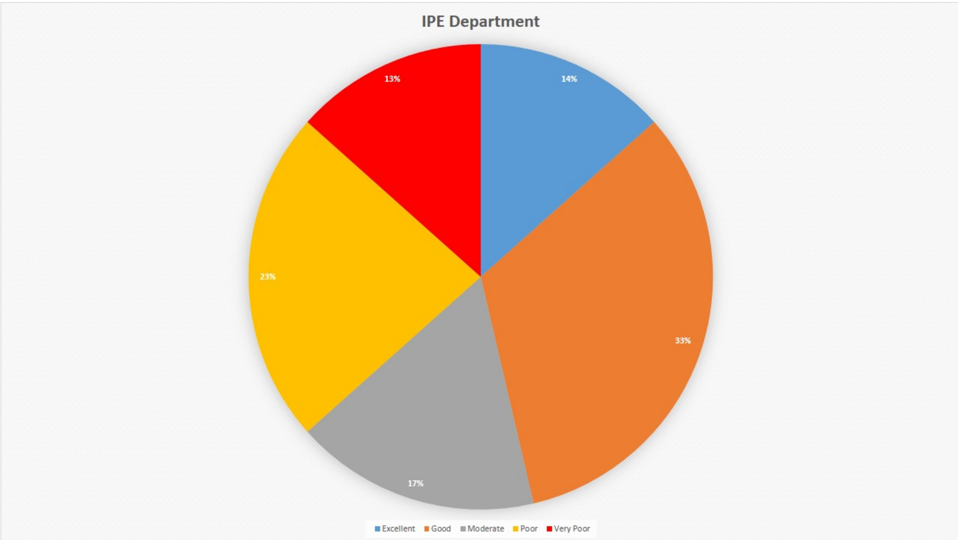

FIGURE 5.4: Performance of IPE Department

IPE Department

The overall performance of IPE department is shown in Figure 5.4. According to our classifier the percentage of each class label of IPE department is shown in Table 5.6

ME Department

The overall performance of ME department is shown in Figure 5.5. According to our classifier the percentage of each class label of ME department is shown in Table 5.7

TABLE 5.7: Performance of ME Department

Class Label	Percent
Excellent	5%
Good	42%
Moderate	37%
Poor	6%
Very Poor	10%

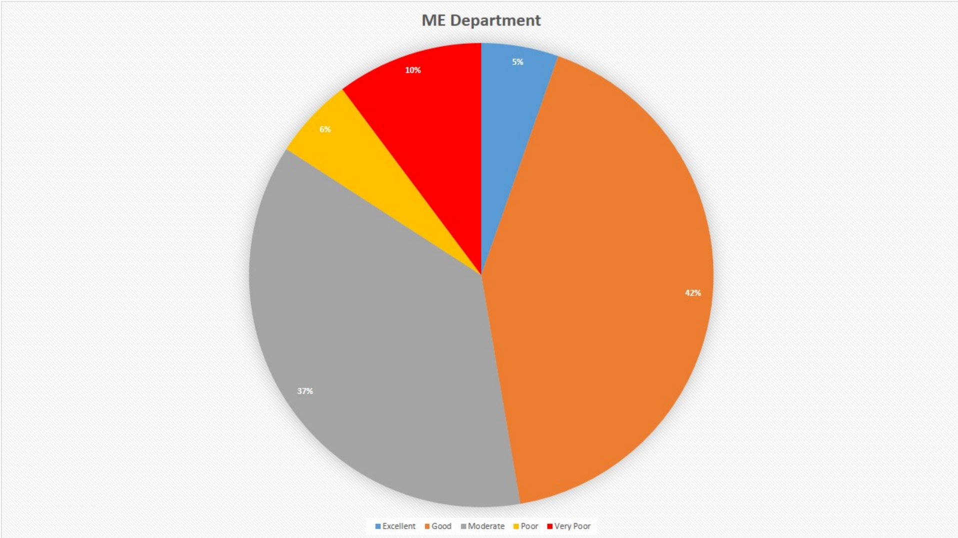

FIGURE 5.5: Performance of ME Department

TABLE 5.8: Performance of CE Department

Class Label	Percent
Excellent	5%
Good	27%
Moderate	47%
Poor	15%
Very Poor	6%

CE Department

The overall performance of CE department is shown in Figure 5.6. According to our classifier the percentage of each class label of CE department is shown in Table 5.8

MME Department

The overall performance of MME department is shown in Figure 5.7. According to our classifier the percentage of each class label of MME department is shown in Table 5.9

TABLE 5.9: Performance of MME Department

Class Label	Percent
Excellent	16%
Good	37%
Moderate	22%
Poor	14%
Very Poor	11%

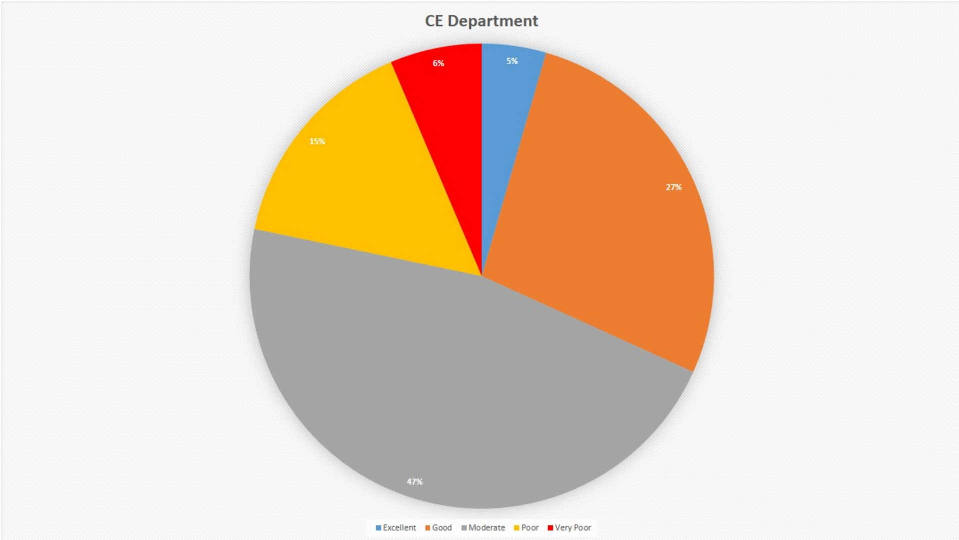

FIGURE 5.6: Performance of CE Department

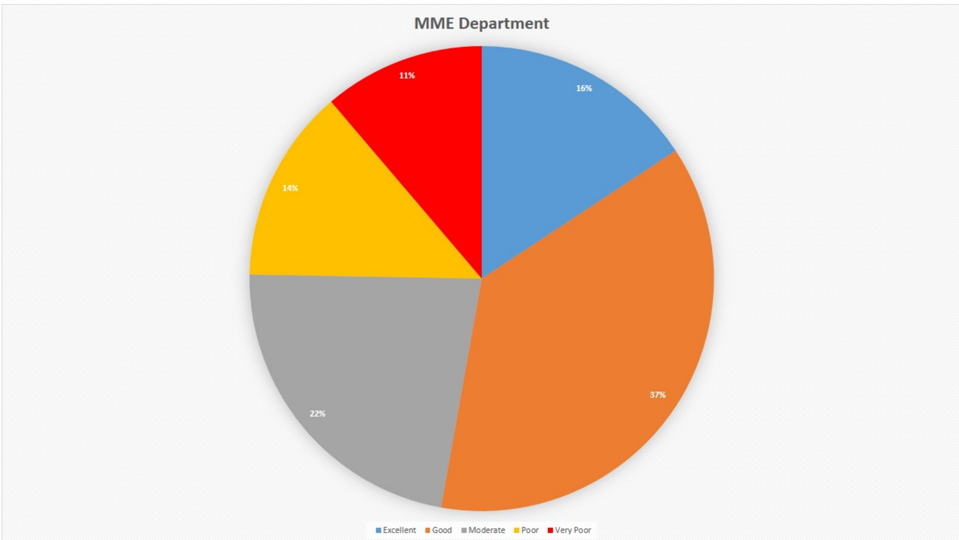

FIGURE 5.7: Performance of MME Department

TABLE 5.10: Performance of CHE Department

Class Label	Percent
Excellent	5%
Good	62%
Moderate	13%
Poor	8%
Very Poor	12%

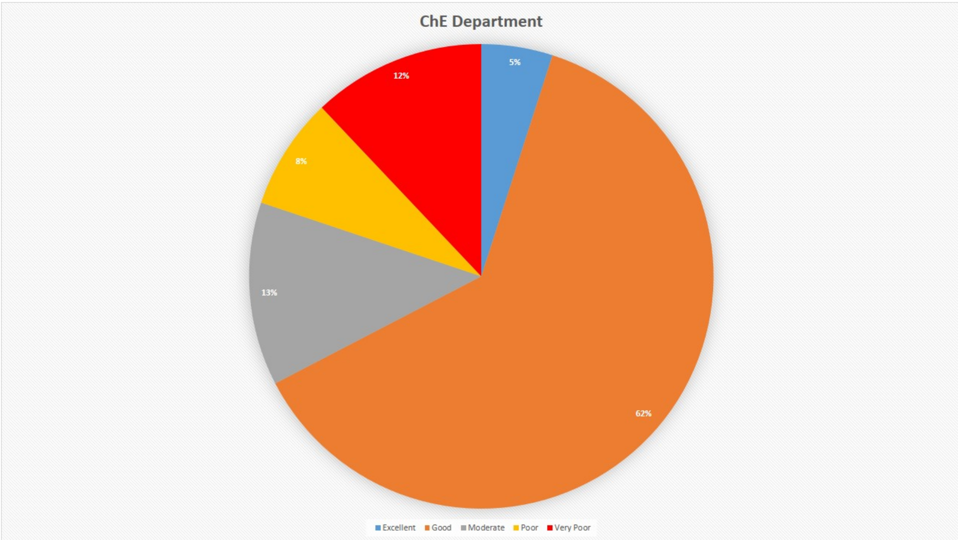

FIGURE 5.8: Performance of CHE Department

CHE Department

The overall performance of CHE department is shown in Figure 5.8. According to our classifier the percentage of each class label of CHE department is shown in Table 5.10

NAME Department

The overall performance of NAME department is shown in Figure 5.9. According to our classifier the percentage of each class label of NAME department is shown in Table 5.11

TABLE 5.11: Performance of NAME Department

Class Label	Percent
Excellent	8%
Good	48%
Moderate	27%
Poor	10%
Very Poor	7%

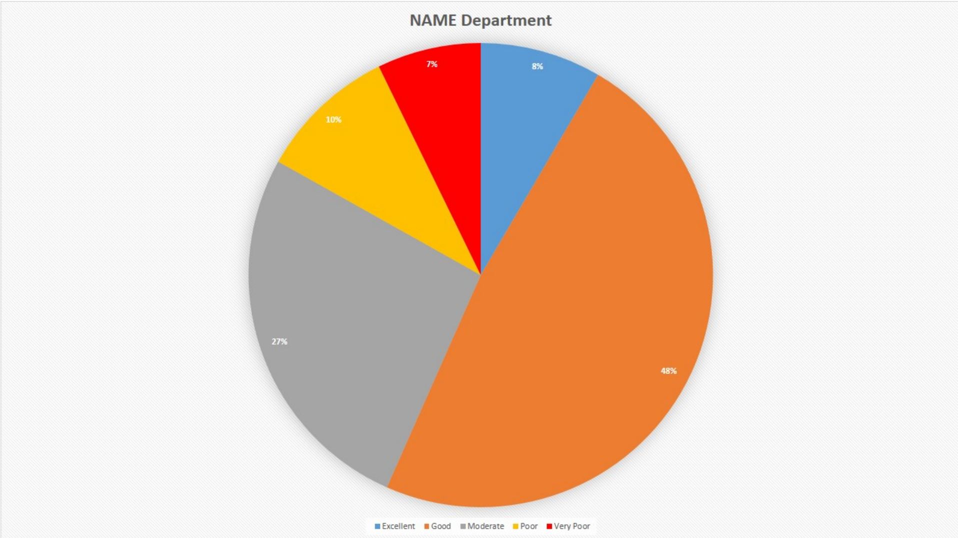

FIGURE 5.9: Performance of NAME Department

TABLE 5.12: Performance of URP Department

Class Label	Percent
Excellent	9%
Good	40%
Moderate	26%
Poor	11%
Very Poor	14%

URP Department

The overall performance of URP department is shown in Figure 5.10. According to our classifier the percentage of each class label of URP department is shown in Table 5.12

ARCH Department

The overall performance of ARCH department is shown in Figure 5.11. According to our classifier the percentage of each class label of ARCH department is shown in Table 5.13

TABLE 5.13: Performance of ARCH Department

Class Label	Percent
Excellent	2%
Good	8%
Moderate	32%
Poor	16%
Very Poor	39%

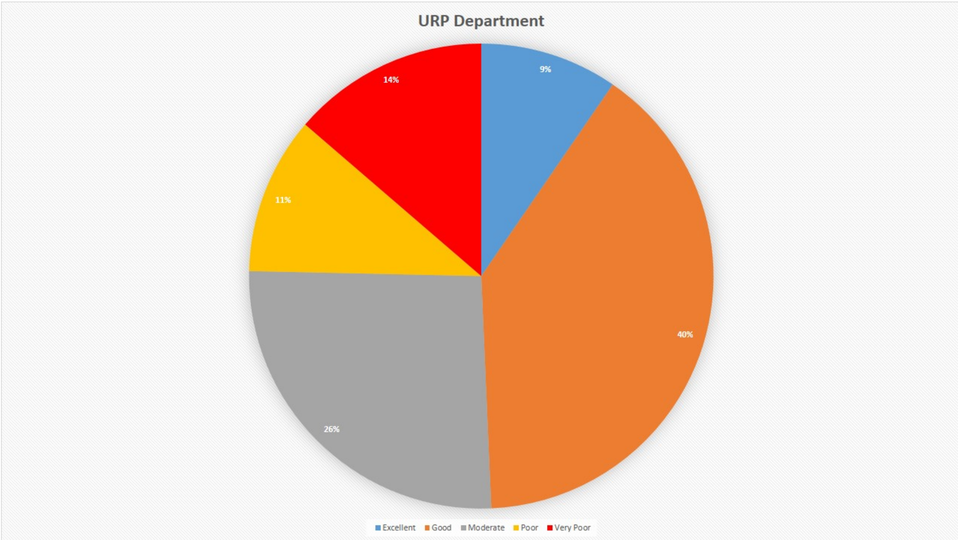

FIGURE 5.10: Performance of URP Department

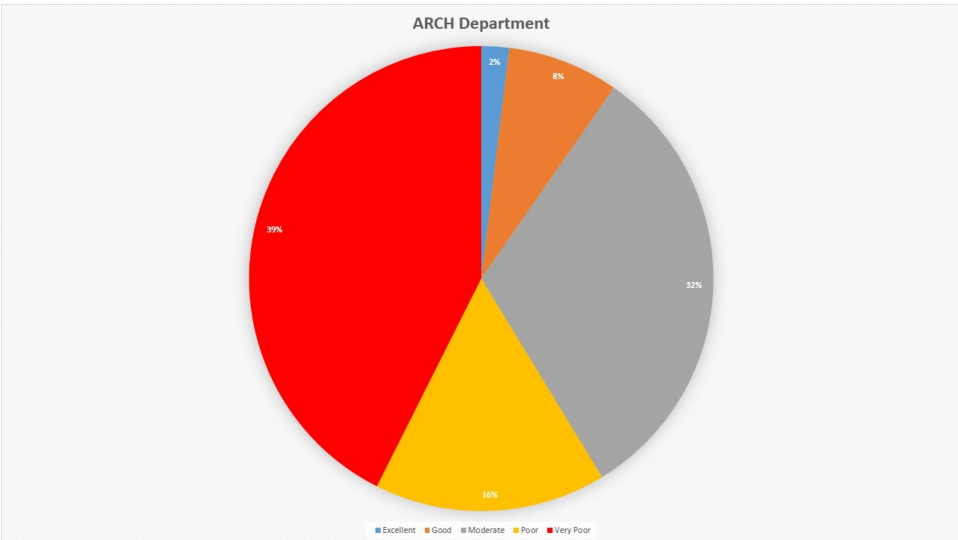

FIGURE 5.11: Performance of ARCH Department

TABLE 5.14: Performance of WRE Department

Class Label	Percent
Excellent	8%
Good	30%
Moderate	35%
Poor	12%
Very Poor	15%

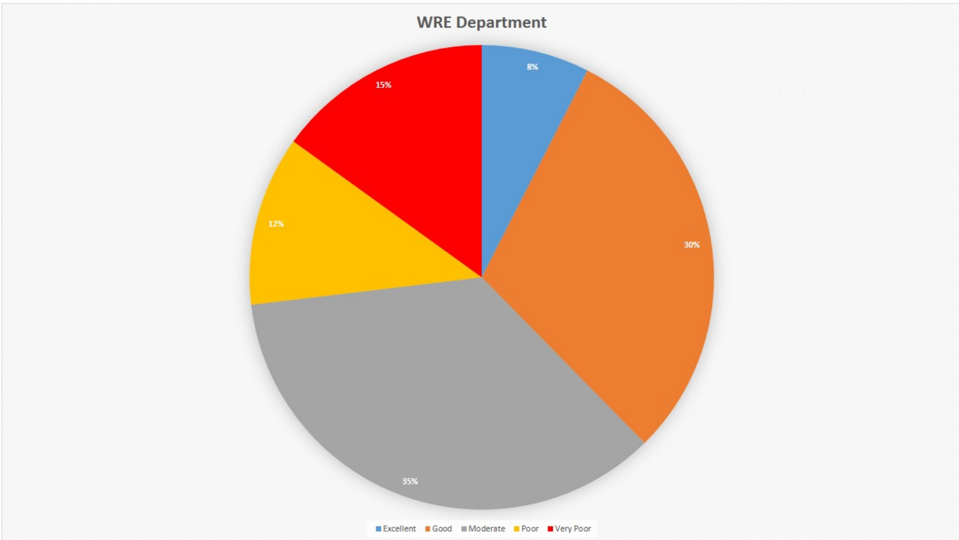

FIGURE 5.12: Performance of WRE Department

WRE Department

The overall performance of WRE department is shown in Figure 5.12. According to our classifier the percentage of each class label of WRE department is shown in Table 5.14

Chapter 6

Conclusion

6.1 Summary of Thesis

6.2 General Findings

6.3 Future Works

Appendix A

Appendix Title Here

Write your Appendix content here.