

**YAŞAR UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

MASTER THESIS

**A COMPARISON OF THE PERFORMANCE OF
ENSEMBLE CLASSIFICATION METHODS IN
TELECOM CUSTOMER CHURN ANALYSIS**

Gökçe KALABALIK

Thesis Advisor: Prof. Dr. Mehmet Cudi OKUR

Department of Computer Engineering

Presentation Date: 03.03.2016

**Bornova-İZMİR
2016**

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a dissertation for the degree of master of science.



Prof. Dr. Mehmet Cadi OKUR (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a dissertation for the degree of master of science.

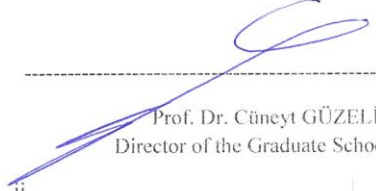


Asst. Prof. Dr. Mete EMİNAĞAOĞLU

I certify that I have read this thesis and that in my opinion it is fully adequate,
in scope and in quality, as a dissertation for the degree of master of science.



Asst. Prof. Dr. İbrahim ZİNCİR



Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

A COMPARISON OF THE PERFORMANCE OF ENSEMBLE CLASSIFICATION METHODS IN TELECOM CUSTOMER CHURN ANALYSIS

KALABALIK, Gökçe

MSc in Computer Engineering

Supervisor: Prof. Dr. Mehmet Cudi OKUR

March 2016, xx pages

Data mining is used to analyze mass databases in order to discover hidden information. Churn analysis based on classification is one of the most common applications of data mining. It is used to predict the behavior of customers who are most likely to change the provided telecom service. In this way, specific campaigns can be created for them. Customer churn is one of the most significant problems that affect business nowadays. The main purpose of churn prediction is to classify the customers into two types. These two types are customers who leave the company and customers who continue doing their business with the company. In order to identify future churners, predictive models based on past data can be developed. However, it has become more difficult to assess the proper classification methods for churn prediction applications since the number of classification models have also increased. In the area of telecom churn prediction, conventional statistical prediction methods are used mostly. This thesis examines combining multiple machine learning algorithms using ensemble methods to increase the accuracy measures of the existing prediction methods. The major aim is to evaluate classification results in telecom customer churn management using bagging, boosting, and random forest ensemble classification methods. Weka software tool has been used to evaluate the performance of common bagging, boosting, and random forest techniques. The results indicate moderate improvements in classification accuracies and other measures. Based on the results, it can be said that ensemble methods with a good base learner are efficient in churn classification. This thesis comprises of eight sections which include these subjects, their applications, and the results.

Keywords: Data Mining, Churn Analysis, Telecom Churn, Classification, Ensemble Methods, Bagging, Boosting, Random Forest

ÖZET

TELEKOMÜNİKASYON SEKTÖRÜ MÜŞTERİ AYRILMA ANALİZİNDE BİRLEŞTİRMELİ SINIFLANDIRMA YÖNTEMLERİ PERFORMANSLARININ KARŞILAŞTIRILMASI

Gökçe KALABALIK

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Mehmet Cudi OKUR

Mart 2016, xx sayfa

Veri madenciliği, saklı bilgiyi ortaya çıkarmak için büyük veri kümelerini analiz etme sürecidir. Sınıflandırmaya dayanarak yapılan müşteri ayrılma analizi veri madenciliğinin en yaygın uygulama alanlarından biridir. Bu analiz, telekomünikasyon servis sağlayıcılarını değiştirme eğilimi gösteren müşterilerin tutumunu tahmin etmekte kullanılır. Böylelikle, bu müşteriler için özel kampanyalar oluşturulabilir. Günümüzde, ayrılacak müşteriler iş hayatını etkileyen en önemli problemlerden biridir. Müşteri ayrılma analizinin esas amacı müşterileri iki tipte sınıflandırmaktır. Bu iki tip müşteri; şirketten ayrılanlar ve şirketle işlerini yürütmeye devam edenlerdir. Gelecekte şirketten ayrılma eğilimi olan müşterileri saptamak için geçmiş verilere dayalı tahmin edici modeller geliştirilebilir. Bununla birlikte, sınıflandırma yöntemlerinin sayısı arttığından dolayı müşteri ayrılma analizi tahmini uygulamaları için uygun sınıflandırma yöntemlerini belirlemek daha da zor bir hal aldı. Telekomünikasyon sektöründe müşteri ayrılma analizi tahmininde, geleneksel istatistiksel tahmin yöntemleri çoğunlukla kullanılmaktadır. Bu tez, çoklu makine öğrenmesi algoritmalarının, birleştirmeli sınıflandırma yöntemlerini mevcut tahmin etme metotlarının ölçü doğruluğunu artırmak için kullanarak birleştirilmesini inceler. Başlıca amaç, bagging, boosting ve random forest birleştirmeli sınıflandırma yöntemlerini kullanarak telekomünikasyon sektöründe müşteri ayrılma yönetimi sınıflandırma sonuçlarının değerlendirmeye alınmasıdır. Yaygın bagging, boosting ve random forest tekniklerinin performansını değerlendirmek için Weka yazılım aracı kullanılmıştır. Sonuçlar sınıflandırma doğrulukları ve diğer ölçülerde makul iyileşmelere işaret etmektedir. Sonuçlara dayanarak, iyi bir sınıflandırma tabanı ile kullanılan birleştirmeli sınıflandırma yöntemlerinin müşteri ayrılma analizi tespitinde etkili olduğunu söylemek mümkündür. Bu tez; bu konuları, uygulamalarını ve sonuçlarını içeren sekiz bölümden oluşmaktadır.

Anahtar sözcükler: Veri Madenciliđi, Müşteri Ayrılma Analizi, Telekomünikasyon Sektöründe Müşteri Ayrılma Analizi, Sınıflandırma, Birleştirmeli Sınıflandırma Yöntemleri, Bagging, Boosting, Random Forest

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Prof. Dr. Mehmet Cudi OKUR for his guidance and support throughout the research and writing phases of my thesis.

Furthermore, I would like to thank to the academic staff of Computer Engineering and Software Engineering Departments of Yaşar University, for their interest in my work and in my seminar presentation which gave me an opportunity to put my theoretical knowledge into practice.

Finally, I would like to thank to my parents for their constant support.

Gökçe KALABALIK
İzmir, 2016

TEXT OF OATH

I declare and honestly confirm that my study, titled “A Comparison of the Performance of Ensemble Classification Methods in Telecom Customer Churn Analysis” and presented as a Master’s Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions, that all sources from which I have benefited are listed in the bibliography, and that I have benefited from these sources by means of making references.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
ÖZET	iv
ACKNOWLEDGEMENTS	vi
TEXT OF OATH	vii
TABLE OF CONTENTS	viii
INDEX OF FIGURES	xi
INDEX OF TABLES	xiv
INDEX OF SYMBOLS AND ABBREVIATIONS	xv
1 INTRODUCTION	1
2 LITERATURE REVIEW	3
3 CLASSIFICATION MODELS	7
3.1 Decision Trees	7
3.2 Rule Based Classifiers	11
3.3 Other Classification Methods	11
3.3.1 Neural Networks Based Classification	11
3.3.2 Support Vector Machines	13

3.4	Performance Evaluation Methods	14
3.4.1	Confusion Matrix Based Performance Measures	15
3.4.2	Receiver Operating Characteristic Curve	17
3.4.3	Error Rates	18
4	Ensemble Methods	19
4.1	Bagging	22
4.2	Boosting	23
4.3	Random Forest	25
5	IMPLEMENTATION	27
5.1	Dataset Description	27
5.2	Dataset Variables Selection	28
6	Experimental Results	30
6.1	Base Classifiers: Decision Stump and J48 Implementation	30
6.1.1	Decision Stump for Full Data Set	30
6.1.2	J48 Decision Tree for Full Dataset	32
6.2	Implementation Results of Bagging Ensemble Classification	34
6.2.1	Bagging-Decision Stump Base-Full Dataset	34
6.2.2	Bagging-J48 Decision Tree Base-Full Dataset	36

6.2.3	Reduced Dataset Bagging-J48 Decision Tree Base	37
6.3	Implementation Results of Boosting Ensemble Classification	38
6.3.1	Boosting-Decision Stump Base-Full Dataset	38
6.3.2	Boosting-J48 Decision Tree Base-Full Dataset	40
6.3.3	Reduced Dataset Boosting-J48 Decision Tree Base	41
6.4	Implementation Results of Random Forest Ensemble Classification	42
6.4.1	Random Forest Classification for Full Dataset	42
6.4.2	Random Forest Classification for Reduced Dataset	43
7	Evaluation of the Ensemble Classification Methods Based on Weighted Performance Values	45
7.1	Accuracy Measures	45
7.1.1	Decision Stump Base Implementation Accuracy Results Comparisons	46
7.1.2	J48 Base Implementation Accuracy Results Comparisons	48
7.1.3	Reduced Dataset Implementation Accuracy Results Comparisons	50
7.2	Error Rate Comparisons	52
8	CONCLUSION AND FUTURE WORK	58
	REFERENCES	59
	CURRICULUM VITEA	62

INDEX OF FIGURES

Figure 3.1 A simplified churn prediction decision tree (Almana et al., 2014)	8
Figure 3.2 Decision Tree Learning Algorithm (Almana et al., 2014)	9
Figure 3.3 An example of a decision stump on Irish flower dataset	10
Figure 3.4 A decision tree for the concept buys computer (Han and Kamber, 2006)	10
Figure 3.5 Real Neuron and Artificial Neuron Model (Larose, 2005)	12
Figure 3.6 Maximum Margin Hyperplane	14
Figure 3.7 Different Outcomes of a Two-Class Prediction	15
Figure 3.8 ROC Curve	17
Figure 4.1 General Idea of Ensemble Methods	19
Figure 4.2 An ensemble of linear classifiers (Oza, 2009)	21
Figure 4.3 An ensemble of linear classifiers (Oza, 2009)	23
Figure 4.4 AdaBoost Algorithm	24
Figure 4.5 Random Forest	26
Figure 5.1 Statistical report of churn dataset (Vis et al., 2009)	27
Figure 5.2 Final optimal subset of features (Kozielski et al., 2015)	29
Figure 6.1 Weka Output of Decision Stump Algorithm on the Full Dataset	31
Figure 6.2 ROC Curve of Decision Stump Algorithm on the Full Dataset	31
Figure 6.3 Weka Output of J48 Algorithm on the Full Dataset	32

Figure 6.4 ROC Curve of J48 Algorithm on the Full Dataset	32
Figure 6.5 Weka Output of Bagging Decision Stump Base on the Full Dataset	34
Figure 6.6 ROC Curve of Bagging Decision Stump Base on the Full Dataset	35
Figure 6.7 Weka Output of Bagging J48 Base on the Full Dataset	36
Figure 6.8 ROC Curve of Bagging J48 Base on the Full Dataset	36
Figure 6.9 Weka Output of Bagging J48 Base on the Reduced Dataset	37
Figure 6.10 ROC Curve of Bagging J48 Base on the Reduced Dataset	37
Figure 6.11 Weka Output of Boosting Decision Stump Base on the Full Dataset	38
Figure 6.12 ROC Curve of Boosting Decision Stump Base on the Full Dataset	39
Figure 6.13 Weka Output of Boosting J48 Base on the Full Dataset	40
Figure 6.14 ROC Curve of Boosting J48 Base on the Full Dataset	40
Figure 6.15 Weka Output of Boosting J48 Base on the Reduced Dataset	41
Figure 6.16 ROC Curve of Boosting J48 Base on the Reduced Dataset	41
Figure 6.17 Weka Output of Random Forest on the Full Dataset	42
Figure 6.18 ROC Curve of Random Forest on the Full Dataset	43
Figure 6.19 Weka Output of Random Forest on the Reduced Dataset	43
Figure 6.20 ROC Curve of Random Forest on the Reduced Dataset	44
Figure 7.1 Accuracy Measures for Full Dataset Decision Sump Base Implementation	
Results	47

Figure 7.2 Accuracy Measures for Full Dataset J48 Base Implementation Results	49
Figure 7.3 Accuracy Measures for Reduced Dataset Implementation Results	51
Figure 7.4 Error Rates for Full Dataset Decision Stump Base Implementation Results	53
Figure 7.5 Error Rates for Full Dataset J48 Base Implementation Results	55
Figure 7.6 Error Rates for Reduced Dataset Implementation Results	57

INDEX OF TABLES

Table 7.1 Accuracy Results Comparisons for Decision Stump Base Implementation	46
Table 7.2 Accuracy Results Comparisons for J48 Base Implementation	48
Table 7.3 Accuracy Results Comparisons for Reduced Dataset	50
Table 7.4 Decision Stump Base Error Rate Results Comparisons	52
Table 7.5 J48 Base Error Rate Results Comparisons	54
Table 7.6 Error Rate Results Comparisons for Reduced Dataset	56

INDEX OF SYMBOLS AND ABBREVIATIONS

Abbreviations

PCT	Percentage
RF	Random Forest
AUC	Area Under Curve

1 INTRODUCTION

Data mining explores large, high-dimensional, and multi-type data sets that have meaningful structure or patterns with the help of statistical and computational methodologies. The fundamental purpose of data mining is to support the discovery of patterns in data to transform information into knowledge. Another purpose is to support decision making process or to explain and justify it. Availability of qualified data on business activities, integration of data repositories into data warehouses, the exponential increase in data processing and storage capabilities, and decrease in cost have led to the rapid development of data mining applications. In today's competitive business world, data mining applications have become so widely used due to the more intense competition at the global scale and the need of making accurate and timely decisions. Data mining focuses on finding interesting and meaningful patterns from large datasets. For this reason, there have also been numerous scientific, health and security related applications.

Nowadays, huge amounts of data are being collected and warehoused. The amount of available data has increased and it has provided the opportunity to automatically find and uncover valuable information and to transform it into valuable knowledge. As computers have become cheaper and more powerful, competitive pressure has been stronger. With the widespread use of low-cost massive data storage technologies and the Internet, large amounts of data have been available for analysis. The organizations that are capable of transforming data into information and knowledge can use them in order to make quicker and more effective decisions and thus to achieve a competitive advantage (Vercellis, 2009).

In today's competitive business world, information and knowledge has become the absolute power for both launching and managing companies. In terms of strategic decision making, more reliable decision support systems and mechanisms with the aid of IT and automated business intelligence models are needed. In recent years, predicting customer churn with the purpose of retaining customers has received an increasing attention due to the competitive business environments.

For many companies, finding reasons of losing customers, measuring customer loyalty and regaining customer have become very important concepts (Gürsoy, 2010).

Companies usually create special marketing tools in order to avoid losing their customers since it is more challenging to obtain new ones.

The subject of this thesis is the evaluation of classification results in telecom churn analysis using bagging, boosting, and random forest ensemble methods. Throughout the thesis; the classification models, decision trees, rule based classifiers, and other classifiers are reviewed in order to identify common approaches within the context of data mining. Afterwards, bagging, boosting, and random forest ensemble methods are explained. All of the algorithms are implemented in Weka 3.7 software tool which is comprised of a collection of machine learning algorithms developed at the University of Waikato in New Zealand. In the implementation phase, initially the dataset is introduced. Telco churn dataset has 3332 customer records with 21 attributes. It is a complete dataset which has no missing values for each attribute throughout all the records. After introducing the features, Decision Stump and J48 algorithms under the trees section within Weka classifiers are implemented separately on the full dataset. After that, Bagging and AdaBoostM1 algorithms under the meta section and Random Forest algorithm under the trees section within Weka classifiers are implemented separately. For the algorithms Bagging and AdaBoostM1 each of DecisionStump and J48 algorithms are used as the base algorithms. Afterwards, the same algorithms are used with a reduced dataset which includes most effective attributes for classification tasks. According to feature selection models, the optimal reduced number of the attributes is decreased to 11. (Kozielski et al., 2015). After completing all the implementation phases within Weka, the results are evaluated using common performance evaluation methods. The evaluation criteria include the following measures and statistics: The percentage of correctly classified instances, the percentage of incorrectly classified instances, true positive rate, and precision, F Measure, ROC Area and Kappa Statistic. The computed values are compared for each base algorithm, bagging, boosting, and random forest results. In terms of error rates, MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), RAE (Relative Absolute Error) and RRSE (Root Relative Squared Error) values are compared for each base algorithm, bagging, boosting, and random forest results. Based on these metrics, comparison results and evaluations of the methods are presented.

2 LITERATURE REVIEW

The fundamental aim of customer churn prediction is identifying customers with a high tendency to leave a company. Customer churn is a common concern of most companies in business environments. Churn occurs when a customer leaves a company. It is a significant issue for most businesses since keeping an existing customer is cheaper than finding a new one. The company can focus on likely churners and try to keep them in case churn can be predicted. From the customer churn perspective, customers can be classified into two types churners and non-churners. Customers who leave the company are called as churners, whereas customers who continue their business with the company are called as non-churners. Improvement of churn prediction can increase profit of the company. The telecommunication industry is dynamic with a large base of customers. Among all industries which suffer from this issue, telecommunications industry can be considered in the top of the list with approximate annual churn rate of 30% (Jahromi, 2009).

The telecommunication industry is dynamic with a large base of customers. Churn prediction and management have become a significant issue especially for the mobile operators. As Gürsoy (2010) indicates the telecommunications sector acquires huge amount of data because of the rapidly changing technologies, the increase in the number of subscribers and many value added services. Due to the uncontrolled and rapid spreading of this field, losses have also increased. Therefore, it has become vital for the operators to acquire the amount invested and to gain at least a minimum profit within a very short period of time (Umayaparvathi et al., 2012). With the help of identification methods, customers who have a tendency to leave a telecom service provider preventive measures can be taken beforehand.

Mobile operators aim to keep their customers and satisfy their needs. To achieve this, they need to predict the customers who have a tendency to churn and then make use of the limited resources to retain those customers. In the telecommunication industry, classification and other data mining methods are used to reveal their profitable and stable customers. Classification methods mainly focus on predicting the customers who have a tendency to leave a certain company based on the user characteristics, user behaviors and quality of services. Developing effective strategies to win more customers and to retain the existing ones contribute to the

survival and good profitability of telecom companies. With the help of these strategies, a telecom company can grow and manage a large customer base to increase profits via telecommunication services like voice data transmission and broadband in a mass scale. However, in order to develop such kind of strategies, the reasons should be known why an existing customer chooses to discontinue his/her telecommunication company. It is very critical to identify churning timely for a company to keep pace with competitive and up-to-date telecommunication industry, in today's dynamic business world, with rapid advances of related technologies, products, and services.

Companies in telecommunication industry have detailed call records within their databases. In their study Rygielski, Wang, Yen et al. (2002) presented that these companies can segment their customers by using call records for developing price and promotion strategies. By making use of data mining techniques, the customers who have a tendency not to make any payments can be detected beforehand. In this way, financial loss of telecom companies can also be reduced. Deviation determination method is one of the methods that is used for these types of analysis. Customers are divided into clusters according to their usage patterns. Customers with inconsistent features are detected and preventive measures are initiated for them.

Within their study; Ren, Zeng, and Wu (2009) presented a clustering method based on genetic algorithm for telecommunication customer subdivision. Initially, the features of telecommunication customers like calling and consuming behavior are extracted. Afterwards, the similarities between the multidimensional feature vectors of telecommunication customers are computed and mapped as the distance between samples on a two-dimensional plane. Eventually, the distances are adjusted to approximate the similarities gradually by genetic algorithm.

Analysis results from a big Taiwan telecom provider pointed out that the proposed approach has pretty good prediction accuracy by using customer demography, billing information, call detail records, and service changed log to build churn prediction mode by making use of Artificial Neural Networks (Chang, 2009).

In their study, Abbasimehr et al. (2014) indicate that as the results show the application of ensemble learning has brought a significant improvement for individual base learners in terms of three performance indicators i.e., AUC,

sensitivity, and specificity. Boosting gave them best results among all other methods. These results indicate that ensemble methods can be a best candidate for churn prediction tasks (Abbasimehr et al., 2014).

As Almana et al. (2014) pointed out within their study, decision tree based techniques, neural network based techniques and regression techniques are generally applied in customer churn.

Hung et al. (2006) indicated that both decision tree and neural network techniques can deliver accurate churn prediction models by using customer demographics, billing information, contract/service status, call detail records, and service change log.

Ensemble learning algorithms have received an increasing attention over last several years. Since these algorithms generate multiple base models using traditional machine learning algorithms and combine them into an ensemble model, their performance is usually better than single models. Amongst the ensemble learning algorithms, bagging and boosting are two of the most popular algorithms due to their good empirical results and theoretical support. An obvious approach to making decisions more reliable is to combine the output of several different models and several machine learning techniques. By learning an ensemble of models and using them in combination; they can often increase predictive performance over a single model. These are general techniques that are able to be applied to most classification tasks and numeric prediction problems (Witten et al., 2011).

Throughout this thesis, an ensemble method which originates from statistical machine learning called bagging (Breiman, 1996) is used. It consists of sequentially computing a base classifier from resampled versions of the training sample in order to obtain a committee of classifiers (Lemmens and Croux, 2006). The final classifier is then obtained by taking the average over all committee members. Applying bagging algorithm on a database is simple and easy even it requires a bit more computation time; it does not need any extra information when compared to the one training sample needs. As Lemmens and Croux (2006) reflect, there is a growing literature showing that committees usually perform better than the base classifiers. Breiman (1996) suggests classification tree as the base classifier. As Lemmens and Croux

(2006) point out more sophisticated versions of bagging with the use of weighted sampling schemes exist under the name of boosting.

The boosting ensemble method which is used throughout this thesis is Real Adaboost (Freund and Schapire, 1996). The main principle of boosting comprises of sequentially applying the base learner to adaptively reweighted versions of the initial dataset. It has been proposed that misclassified observations are assigned an increased weight in the next iteration and the weights given to previously correctly classified observations are reduced consequently (Lemmens and Croux, 2006). The main idea is based on forcing the classifier to focus on the instances which are difficult to classify. As Lemmens and Croux (2006) point out, boosting procedure requires software that allows assigning weights to the observations of the training sample when computing the base classifier. Lemmens and Croux (2006) also reflect that a key difference between bagging and boosting is the initial classification rule which is preferably a weak learner, for instance; a classifier that has a slightly lower error rate than random guessing. Lemmens and Croux (2006) indicate that using decision stumps for example binary trees with only two terminal nodes for Real Adaboost is suggested since such a weak base classifier would have a low variance but a high bias. As Lemmens and Croux (2006) point out after iterations of the boosting algorithm, the bias should be reduced, while the variance would remain moderate. In principle, boosting should therefore outperform bagging since it not only reduces the variance, but also the bias (Lemmens and Croux, 2006).

Random Forest is another ensemble method which is used throughout this thesis. Although it is under the trees section within Weka classifiers, it stands for a class of ensemble methods that is particularly designed for decision tree classifiers. Random Forest algorithm combines predictions made by plenty of decision trees. A popular algorithm for learning random forests builds a randomized decision tree in each iteration of the bagging algorithm, and often produces excellent predictors (Witten et al., 2011).

3 CLASSIFICATION MODELS

3.1 Decision Trees

Decision Tree is as a tree-shaped structure that depicts sets of decisions and produces rules for the classification of a dataset. It can also be described as a structure that is used to divide a large collection of records into sequentially smaller sets of records by applying a sequence of simple decision rules. Decision trees are based on divide-and-conquer concept. A decision tree consists of three types of nodes:

- Root Node
- Internal Node
- Leaf or Terminal Node

As Clemente et al. (2010) state the fundamental logic behind Decision Tree is producing a classification of observations into groups and then obtaining a score for each group. CART (Classification and Regression Trees) algorithm is the most widely used tree algorithm amongst the statistical algorithms. CART analysis is based on predicting or classifying cases according to a response variable.

In terms of customer churn prediction, decision trees are the most common methods amongst the classification models. In order to evaluate a dataset using decision trees, classification is done by altering the tree until a leaf node is reached. When classifying customer records, class labels of churner or non-churner are assigned to the leaf node. Fig 3.1 illustrates a simplified decision tree for customer churn prediction in telecom sector.

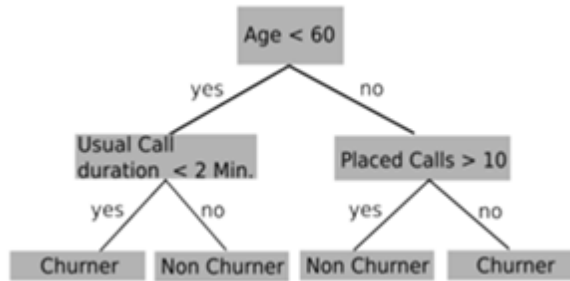


Figure 3.1 A simplified churn prediction decision tree (Almana et al., 2014)

Decision trees has a top-down structure. A decision tree learning algorithm (ID3) is illustrated in Figure 3.2. If all the examples belong to the same class, the algorithm just returns a leaf node of that class. If there are no attributes left with which to produce a nonterminal node, then the algorithm has to return a leaf node. It returns a leaf node of the class which is most frequently seen in the training set. If none of them is true, then the algorithm finds the one attribute value test that comes closest to splitting the whole training set into parts such that each part only comprises of examples of one class. When such an attribute is chosen, the training set is split based on that attribute. It means that for each value of the attribute, a training set is produced such that all the examples in the set have value for the selected attribute. The learning algorithm is called recursively for each of the training sets.

```

Decision_Tree_Learning( $T, A, g$ )
  if  $T_{i,g}$  is the same for all  $i \in \{1, 2, \dots, |T|\}$ ,
    return a leaf node labeled  $T_{1,g}$ .
  else if  $|A| = 0$ ,
    return a leaf node labeled  $\operatorname{argmax}_c \sum_{i=1}^{|T|} I(T_{i,g} = c)$ .
  else
     $b = \text{Choose\_Best\_Attribute}(T, A)$ 
    Set tree to be a nonterminal node with test  $b$ .
    for each value  $v$  of attribute  $b$ ,
       $t_v = \emptyset$ 
    for each example  $T_i \in T$ ,
       $v = T_{i,b}$ 
      Add example  $T_i$  to set  $t_v$ .
    for each value  $v$  of attribute  $b$ ,
       $\text{subtree} = \text{Decision\_Tree\_Learning}(t_v, A - b)$ 
      Add a branch to tree labeled  $v$  with subtree subtree.
  return tree.

```

Figure 3.2 Decision Tree Learning Algorithm (Almana et al., 2014)

Decision Stump and J48 are two of the widely used decision tree algorithms that are also used throughout this thesis. Decision Stump and J48 algorithms are under the trees section of Weka classifiers. A decision stump class comprises of a one-level decision tree with one root node that is instantly connected to the terminal nodes. A decision stump makes a prediction based on single criteria. In Figure 3.3, it can be observed that decision stump algorithm discriminates between two of three classes of Irish flower dataset. The value of petal width is measured in centimeters. Viola-Jones face detection algorithm employs AdaBoost with decision stumps as weak learners (Viola and Jones, 2004).

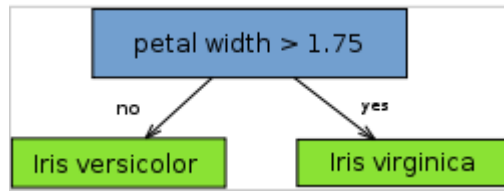


Figure 3.3 An example of a decision stump on Irish flower dataset

J48 is a class for generating a pruned or unpruned C4.5 decision tree. J48 is Weka’s implementation of C4.5 decision tree learner. J48 actually implements a later and slightly improved version called C4.5 revised version 8, which was the last public version of this family of algorithms before the commercial implementation C5.0 was released (Witten et al., 2011). ID3, C4.5, and CART adopt a greedy approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner, in addition to this most algorithms for decision tree induction also follow such a top-down approach, which starts with a training set of tuples and their associated class labels, then the training set is recursively partitioned into smaller subsets as the tree is being built (Han and Kamber, 2006). Figure 3.4 illustrates whether a customer at AllElectronics is likely to purchase a computer or not. Each internal node depicts a test on an attribute and each leaf node depicts a class having the value of either yes or no.

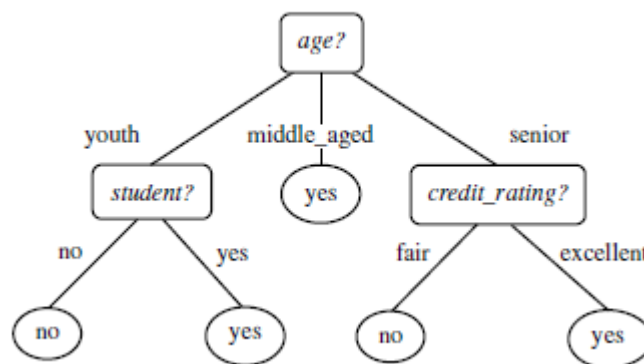


Figure 3.4 A decision tree for the concept buys computer (Han and Kamber, 2006)

3.2 Rule Based Classifiers

Rules are an effective way of representing information or bits of knowledge. A rule-based classifier uses a set of IF-THEN rules for classification. An IF-THEN rule is an expression of the form:

IF condition THEN conclusion.

The “IF”-part (or left-hand side) of a rule is described as the rule antecedent or precondition. The “THEN”-part (or right-hand side) is the rule consequent. In the rule antecedent, the condition comprises of one or more *attribute tests* (such as *age = youth*, and *student = yes*) that are processed with the logical AND operator.

A rule R can be assessed by its coverage and accuracy. Given a tuple, X , from a class labeled data set, D , let n_{covers} be the number of tuples covered by R ; $n_{correct}$ be the number of tuples correctly classified by R . We can define the coverage and accuracy of R as (Han et al., 2006):

$$coverage(R) = \frac{n_{covers}}{|D|}$$

$$accuracy(R) = \frac{n_{correct}}{n_{covers}}$$

Within Weka tool, there are different rule-based classifiers under the rules section. DecisionTable, DTNB, JRip, OneR, ZeroR, and PART algorithms are the most widely used ones.

3.3 Other Classification Methods

3.3.1 Neural Networks Based Classification

Artificial neural network (ANN) is another common classification method. ANNs are a product of early artificial intelligence work aimed at modeling the inner workings of the human brain as a way of creating intelligent systems (Lyle, 2007).

Although many artificial intelligence researchers have focused on different directions, ANNs are still useful in many domains which contain noise.

Figure 3.5 shows that a real neuron uses dendrites to gather inputs from other neurons and combines the input information, generating a nonlinear response when some threshold is reached (“firing”), which it sends to other neurons using the axon (Larose, 2005). This figure also illustrates an artificial neuron model that is used in most neural networks. The inputs (x_i) are collected from upstream neurons (or the data set) and combined through a combination function such as summation (Σ), which is then input into a (usually nonlinear) activation function to produce an output response (y), which is then channeled downstream to other neurons (Larose, 2005). As Alman et al. (2014) pointed out, neural network-based approaches in the prediction of customer churn in line with cellular wireless services is used (Almana et al., 2014).

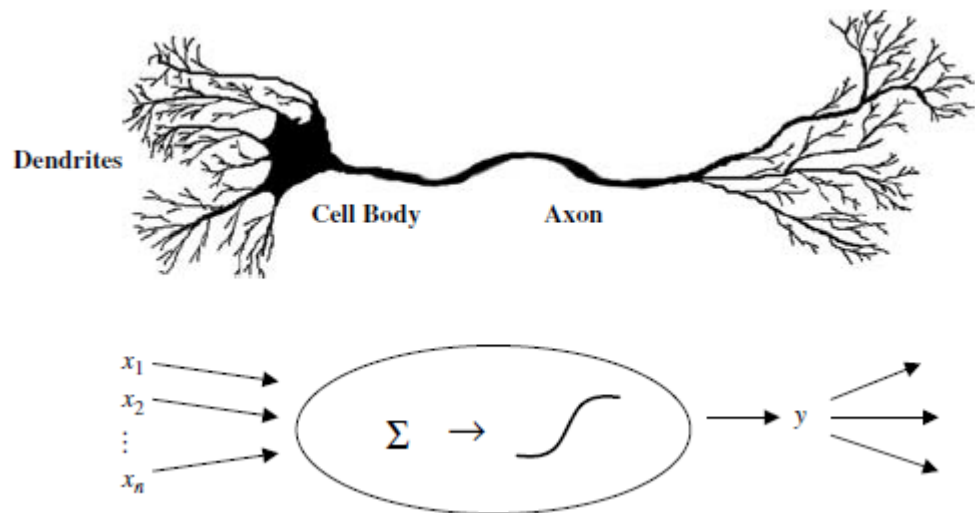


Figure 3.5 Real Neuron and Artificial Neuron Model (Larose, 2005)

3.3.2 Support Vector Machines

Support vector machines are a recent machine learning method for discrete classification and continuous prediction. Support vector machines are based on the idea that concepts that are linearly separable are easy to learn. Support vector machines operate on the idea that by expanding the feature space of the domain to be learned the concepts involved may become linearly separable (Lyle, 2007).

As Lyle (2007) indicates in his study, the maximum margin hyperplane is a hyperplane in the new space that provides the greatest separation between the classes involved. The maximum margin hyperplane, which is illustrated in Figure 3.6, can be detected with the help of finding the convex hulls of the classes involved. If the classes are linearly separable, the convex hulls will not overlap. The maximum margin hyperplane can be described as the orthogonal to the shortest line between the convex hulls and intersects it at its midpoint. In their paper, Brandusoiu and Todorean (2013) built four predictive models for subscribers' churn in mobile telecommunications companies, using SVM algorithm with different kernel functions. By evaluating the results, from the technical point of view, we observe that for predicting both churners and non-churners, the model that uses the polynomial kernel function performs best, having an overall accuracy of 88.56% (Brandusoiu and Todorean, 2013).

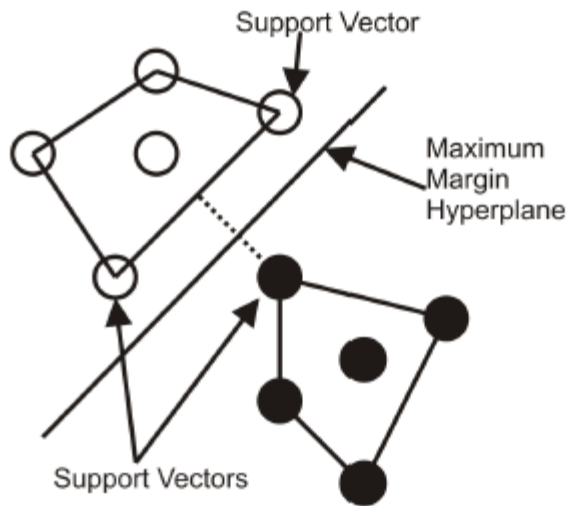


Figure 3.6 Maximum Margin Hyperplane

3.4 Performance Evaluation Methods

Predictive models produce a numerical measure that assigns to each customer their tendency to churn with the help of probability. Clemente et al. (2010) state that this probabilistic classifier can be turned into a binary one using a certain threshold to determine a limit between classes. The accuracy of a model is an indicator of its capability to predict the target class for future observations. The proportion of observations of the test set correctly classified by the model can be described as the most basic indicator. The error rate can be calculated using the ratio between the number of errors and the number of cases examined.

In terms of classification instead of focusing on the number of cases correctly or incorrectly classified, it is more critical to analyse the type of error made. From the churn prediction perspective, as Clemente et al. (2010) state it is normal that the churn rate is much lower than the retention rate in the company which causes a class imbalance problem. For these kinds of problems, it is more appropriate to make use of decision matrices.

3.4.1 Confusion Matrix Based Performance Measures

Confusion matrix for two classes is a binary classification problem with two possible values; positive (+) and negative (-). In this case, confusion matrix can be described as a contingency table of 2x2 which has rows containing observed values and columns containing predicted values as it can be seen in Figure 3.7.

		Predicted Class	
		Yes	No
Actual Class	Yes	TP	FN
	No	FP	TN

Figure 3.7 Different Outcomes of a Two-Class Prediction

In order to assess the classification results, in the two-class case with classes *yes* and *no* a single prediction has the four different possible outcomes shown in Figure 3.7. The *true positives* (TP) and *true negatives* (TN) are correct classifications. A *false positive* (FP) is when the outcome is incorrectly predicted as *yes* (or positive) when it is actually *no* (negative). A *false negative* (FN) is when the outcome is incorrectly predicted as negative when it is actually positive. The *true positive rate* is TP divided by the total number of positives, which is TP + FN; the *false positive rate* is FP divided by the total number of negatives, which is FP + TN (Witten et al., 2011):

- Overall accuracy measures the percentage of correct classified is calculated via the following formula:

$$PCC = \frac{TP + TN}{TP + TN + FP + FN}$$

- Sensitivity, in other words true positive rate, measures the proportion of positive examples which are predicted to be positive. In this study, sensitivity refers to the percentage of correctly classified in class “Churn”.

$$sensitivity = \frac{TP}{TP + FN}$$

- Specificity, in other words true negative rate, measures the proportion of negative examples which are predicted to be negative. In this study, specificity refers to the percentage of correctly classified in class “Non-Churn”.

$$specificity = \frac{TN}{FP + TN}$$

- Recall is defined as the true positive rate or sensitivity, and precision is defined as positive predictive value (PPV); True negative rate is also called as specificity.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

- F Measure combines precision and recall is the harmonic mean of precision and recall;

$$F = 2. \frac{Precision.Recall}{Precision + Recall}$$

- Kappa Statistic makes a comparison between the accuracy of the system and the accuracy of a random system.

$$kappa = \frac{totalAccuracy - randomAccuracy}{1 - randomAccuracy}$$

- MCC (Matthews’s correlation coefficient) is a measure of the quality of two-class classification. MCC is a correlation coefficient between the observed and predicted binary classification having a value between -1 and +1. Having a MCC value of +1 indicates a perfect prediction, 0 means not better than random guessing, and -1 indicates a controversy between predicted and observed classification results.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3.4.2 Receiver Operating Characteristic Curve

Receiver Operating Characteristic (ROC) chart is a two-dimensional plot. ROC curves depict the performance of a classifier without regard to class distribution or error costs; they plot the true positive rate (sensitivity) on the vertical axis against the false positive rate (specificity) on the horizontal axis (Witten et al., 2011). Figure 3.8 illustrates how a ROC curve looks like. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. By using this graph, the optimal balance point between sensitivity and specificity can be detected. ROC analysis also provides the chance of assessing the predictive ability of a classifier independent of any threshold. The area under the ROC curve which is called AUC is a common measure for comparing the accuracy of various classifiers. ROC evaluates the ability of a method to correctly classify the instances. According to this approach, the classifier with the greatest AUC will be accepted better. If the AUC of a classifier is closer to 1, it means that its accuracy is higher. As Clemente et al., (2010) states the AUC can be interpreted intuitively as the probability that at a couple of clients, one loyal and one that churns, the method correctly classify both of them.

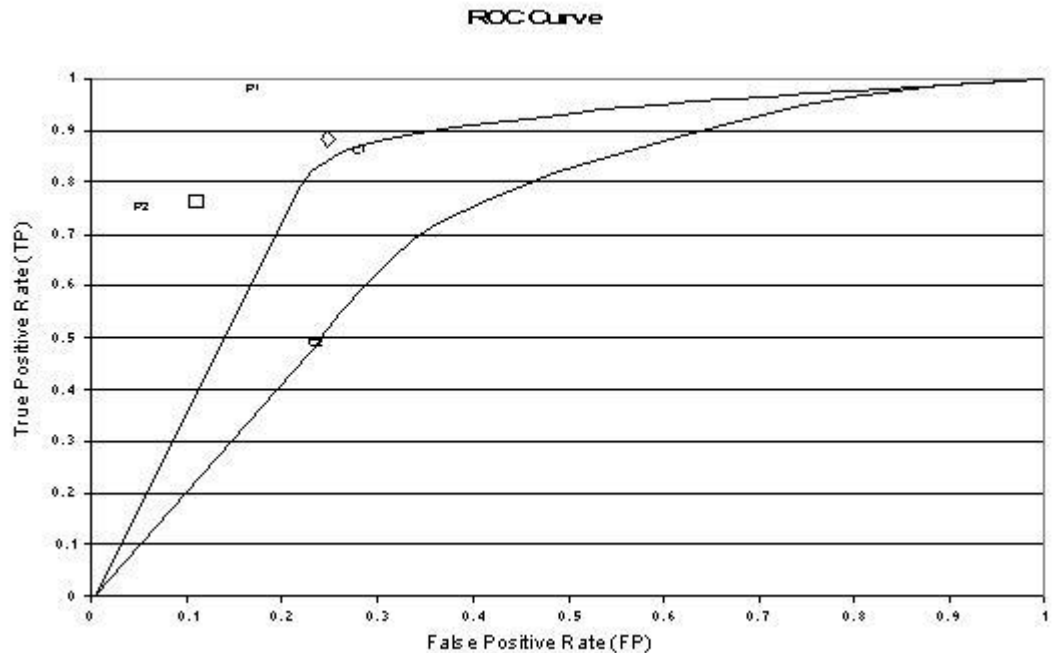


Figure 3.8 ROC Curve

3.4.3 Error Rates

The mean absolute error (MAE) is defined as the quantity used to measure how close predictions or forecasts are to the eventual outcomes; the root mean square error (RMSE) is defined as frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed; Relative Absolute Error (RAE) is a measure of the uncertainty of measurement compared to the size of the measurement; The root relative squared error (RRSE) is defined as a relative to what it would have been if a simple predictor had been used (Vijayarani et al., 2013). The predicted values on the test instances are p_1, p_2, \dots, p_n ; the actual values are a_1, a_2, \dots, a_n . Notice that p_i means the numerical value of the prediction for the i^{th} test instance (Witten et al., 2011).

$$\text{Mean - absolute error} = \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{n}$$

$$\text{Mean - squared error} = \frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{n}$$

$$\text{Relative - absolute error} = \frac{|p_1 - a_1| + \dots + |p_n - a_n|}{|a_1 - \bar{a}| + \dots + |a_n - \bar{a}|}$$

$$\text{Relative - squared error} = \frac{(p_1 - a_1)^2 + \dots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \dots + (a_n - \bar{a})^2}$$

4 Ensemble Methods

Ensemble learning is a machine learning model which is based on training multiple learners in order to solve the same problem. As illustrated in Figure 4.1, apart from ordinary machine learning approaches which try to learn one hypothesis from training data, ensemble methods try to construct a set of hypotheses and combine them to be used. They can all, more often than not, increase predictive performance over a single model. And they are general techniques that are able to be applied to classification tasks and numeric prediction problems (Witten et al., 2011).

An ensemble method comprises of a number of learners which are called as base learners. The ability to generalize an ensemble is usually much stronger than base learners. Ensemble learning strategy is very powerful in terms of enhancing weak learners which perform better than random guessing to be strong learners which can make accurate decisions. It is noteworthy, however, that although most theoretical analysis work on weak learners, base learners used in practice are not necessarily weak since using not-so-weak base learners often results in better performance (Zhou, 2009).

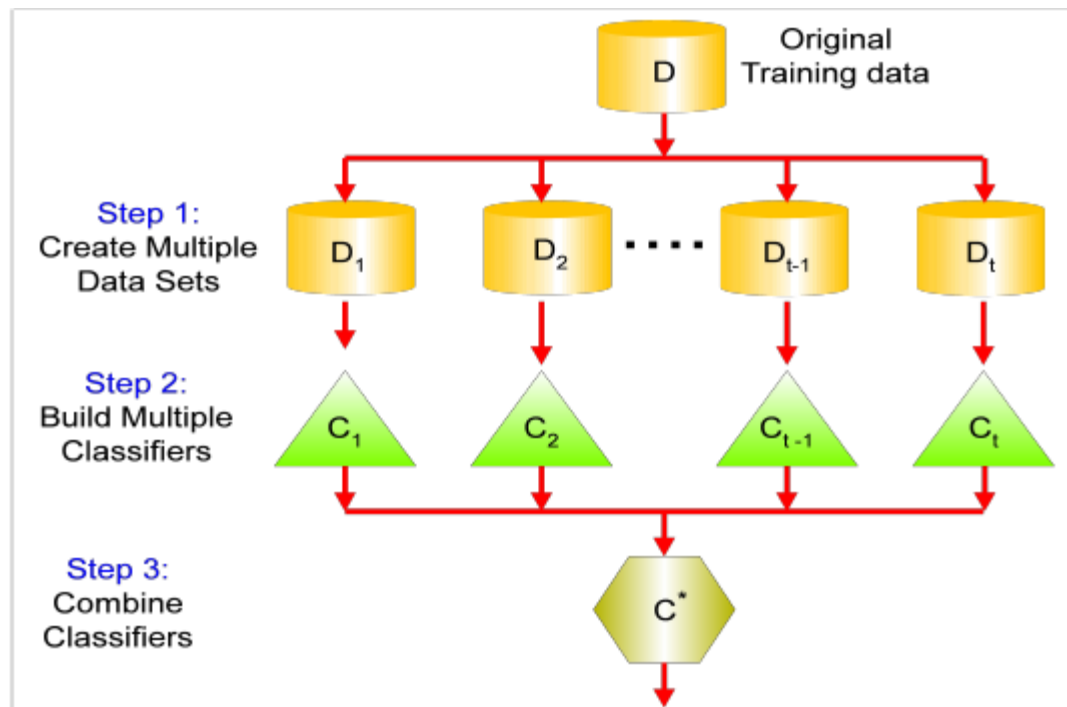


Figure 4.1 General Idea of Ensemble Methods

There are 2 necessary conditions for an ensemble classifier to perform better than a single classifier:

- 1-) the base classifiers should be independent of each other
- 2-) the base classifiers should do better than a classifier that performs random guessing

Ensemble classifiers combine multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, and correct decisions are reinforced.

Multiple training sets can be created by resampling the data according to some sampling distribution. Sampling distribution determines how likely it is that an example will be selected for training; it may vary from one trial to another. Classifier is built from each of the training set using a particular learning algorithm.

Reduced error rates by Bagging & Boosting

Suppose there are 25 base classifiers

Each classifier has error rate, $\epsilon = 0.35$ Assume errors made by classifiers are uncorrelated. Probability that the ensemble classifier makes a wrong prediction:

$$P(X \geq 13) = \sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1-\epsilon)^{25-i} = 0.06$$

This is considerably lower than the error rate $\epsilon = 0.35$.

In the following figure, an ensemble of linear classifiers is illustrated. Each line A, B, and C correspond to a linear classifier. The boldface line illustrates the ensemble that classifies new examples with the help of the majority vote of A, B, and C.

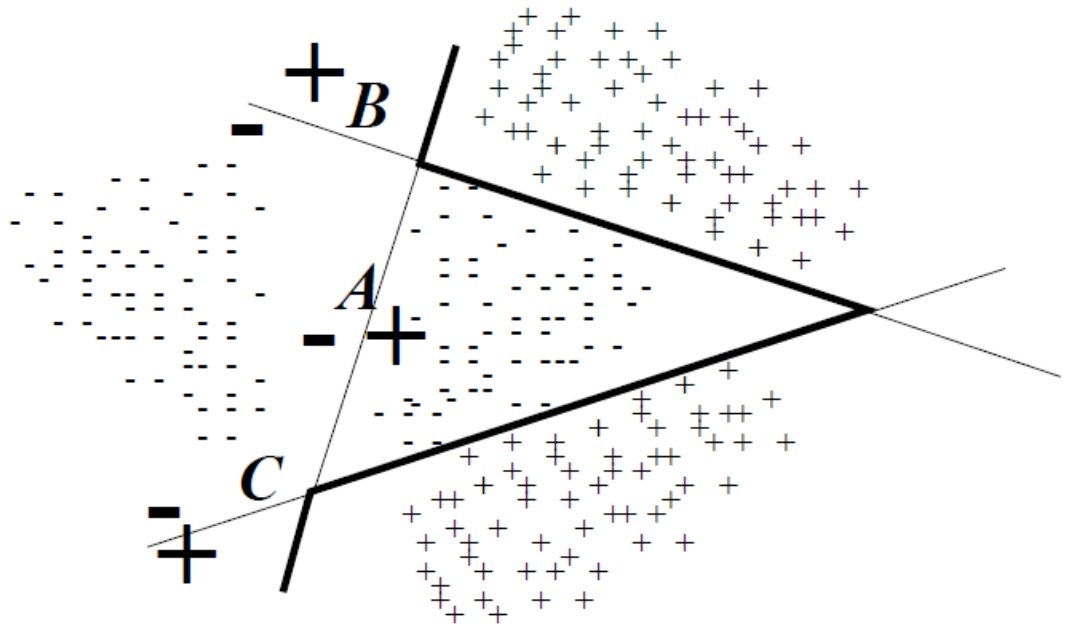


Figure 4.2 An ensemble of linear classifiers (Oza, 2009)

In order to depict a more accurate picture of a situation, consulting a group of experts is better than consulting just one expert in terms of solving everyday problems. For instance, a patient has a set of symptoms and instead of consulting just one doctor; he/she decides to consult a few doctors in order to make sure about his/her illness. Consulting many doctors, and then based on their diagnosis, he/she can get a fairly accurate idea of the diagnosis. In this example, doctors can be considered as classifiers by analogy. Combining conjectures and judgments of a group of experts lead to more accurate decision when compared to consulting just one expert. As Lyle (2007) points out, if multiple base classifiers are combined, it has often been found that the group is more accurate than the individual, though improvement is not guaranteed.

For this thesis, general ensemble learning methods namely bagging and boosting and RF are used expecting to produce more accurate predictions when compared to the predictions produced by the base classifiers.

4.1 Bagging

Bagging (Breiman 1996) is an ensemble learning method whose name is derived from bootstrap aggregation. Bagging is a kind of meta-algorithm and it is a special case of model averaging. It is originally designed for classification and usually applied to decision tree models (Sewell, 2008). Bagging makes use of multiple versions of a training set by using the bootstrap namely sampling with replacement. In order to train a different model, each of these data sets is used. The outputs of the models are combined by averaging (in case of regression) or voting (in case of classification) to create a single output. Bagging is easy to implement amongst the ensemble learning methods. It is the simplest method used to improve the performance of a classifier. This method is developed by Leo Breiman (1996) and it is based on aggregating classifiers in order to increase predictive accuracy. Basic idea of bagging is producing different versions of the same classifier using the same training data set. In order to do this, examples are chosen randomly with replacement. It may cause that some examples may be repeated or left out of a training set. As a result of this phase, classifiers are created using the same training sample. The next phase is established by combining all the predictions of each individual classifier to create a final prediction. The final prediction is usually obtained by voting method. The average of all the predictions is performed in this way.

The algorithms for bagging and sampling with replacement are given in figure 4.3. In these algorithms, T is the original training set of N examples, M is the number of base models to be learned, L_b is the base model learning algorithm, the h_i 's are the base models, $\text{random_integer}(a, b)$ is a function that returns each of the integers from a to b with equal probability, and $I(A)$ is the indicator function that returns 1 if A is true and 0 otherwise (Oza, 2009). In order to create a bootstrap training set from an original training set of size N , we perform N Multinomial trials, where in each trial; we draw one of the N examples. For each trial, each example has probability $1/N$ of being drawn. The part of the algorithm shown in figure 4.1.1 does exactly this; for N times, the algorithm chooses a number r from 1 to N and adds the r th training example to the bootstrap training set S . Clearly, some of the original training examples will not be selected for inclusion in the bootstrap training set and others will be chosen one time or more (Oza, 2009).

```

Bagging( $T, M$ )
For each  $m = 1, 2, \dots, M$ 
     $T_m = \text{Sample\_With\_Replacement}(T, |T|)$ 
     $h_m = L_b(T_m)$ 
Return  $h_{fin}(x) = \arg \max_{y \in Y} \sum_{m=1}^M I(h_m(x) = y)$ .

Sample_With_Replacement( $T, N$ )
 $S = \{\}$ 
For  $i = 1, 2, \dots, N$ 
     $r = \text{random\_integer}(1, N)$ 
    Add  $T[r]$  to  $S$ .
Return  $S$ .

```

Figure 4.3 An ensemble of linear classifiers (Oza, 2009)

4.2 Boosting

Boosting (Schapire 1990) is a kind of meta-algorithm which can be received as a model averaging method. It is the most widely used ensemble method amongst the other ensemble methods. Initially a weak classifier is created that it suffices that its accuracy on the training set is only slightly better than a random guessing (Sewell, 2008). A succession of models is built in an iterative fashion. Each of them is trained on a data set in which points misclassified by the previous model are assigned more weight. Eventually, all of the successive models are weighted based on their success. The outputs are combined by making use of voting for classification. This method was developed by Freund and Schapire (1996) as a way of iteratively creating models which complement those that have been created previously (Lyle, 2007). Boosting has common features with bagging that it uses only one type of base classifier, on the other hand, instead of relying on a uniform randomly selected training sets of classifiers, the training sets are based on the strengths and weaknesses of the previously created classifiers. The fundamental difference between bagging and boosting is the addition of a weight to each of the examples in the training set. At the

beginning all the weights are set to one, in this way each training example is given equal importance to begin with. At this point it is time to generate the first classifier.

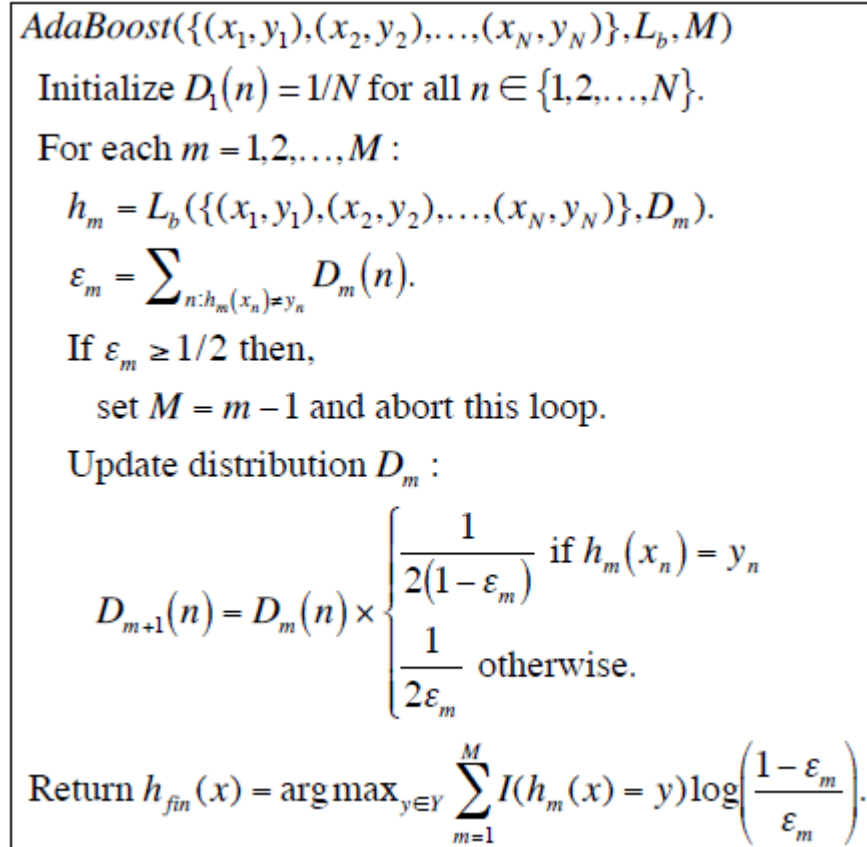


Figure 4.4 AdaBoost Algorithm

The AdaBoost algorithm is illustrated in Figure 4.4. The inputs of the algorithm are a set of N training examples, a base model learning algorithm L_b , and the number M of base models that we want to combine. AdaBoost is originally designed for two-class classification problems, although it is generally used with a large number of classes. This description is based on the assumption that there are two classes. The first phase in AdaBoost is to construct an initial distribution of weights D_1 over the training set. This distribution assigns equal weight to all N training examples. Now the loop of the algorithm starts. In order to produce the first base model, we call L_b with distribution D_1 over the training set i . After getting back a model h_1 , we calculate its error ϵ_1 on the training set itself, which is just the sum of the weights of the training examples that h_1 classifies incorrectly (Oza, 2009). We

require that $\epsilon_1 < 1/2$ which is the weak learning assumption that the error should be less than what we would achieve through random guessing. In case this condition is not satisfied, then we stop and return the ensemble consisting of the previously-created base models. If this condition is satisfied, then we calculate a new distribution D_2 over the training examples as follows. Examples that were correctly classified by h_1 have their weights multiplied by $1/(2(1-\epsilon_1))$. Examples that were misclassified by h_1 have their weights multiplied by $1/(2\epsilon_1)$. Because of our condition $\epsilon_1 < 1/2$, correctly classified examples have their weights reduced and misclassified examples have their weights increased. Specifically, examples that h_1 misclassified have their total weight increased to $1/2$ under D_2 and examples that h_1 correctly classified have their total weight decreased to $1/2$ under D_2 . After that we go into the next iteration of the loop to construct base model h_2 using the training set and the new distribution D_2 . The key is that the next base model will be created by a weak learner; therefore, at least some of the examples misclassified by the previous base model will have to be correctly classified by the current base model. In this way, boosting forces subsequent base models to correct the mistakes that are made by the previous models. M base models are produced. The ensemble returned by AdaBoost is a function that takes a new example as input and returns the class that gets the maximum weighted vote over the M base models, where each base model's weight is $\log((1-\epsilon_m)/\epsilon_m)$, which is proportional to the base model's accuracy on the weighted training set presented to it (Oza, 2009).

4.3 Random Forest

Random Forest is a class of ensemble methods especially designed for decision tree classifiers. The logic behind its structure is that it combines predictions made by many decision trees. In a random forest algorithm, each tree is produced based on a bootstrap sample and the values of a distinct set of random vectors. The random vectors are produced based on a fixed probability distribution. The structure of generating a random forest is based on sampling a dataset with replacement, then selecting m variables from p variables randomly and creating a tree in this way, after creating more trees by repeating the same procedures, the results are combined eventually. Fig 4.3.1 shows the structure of random forests.

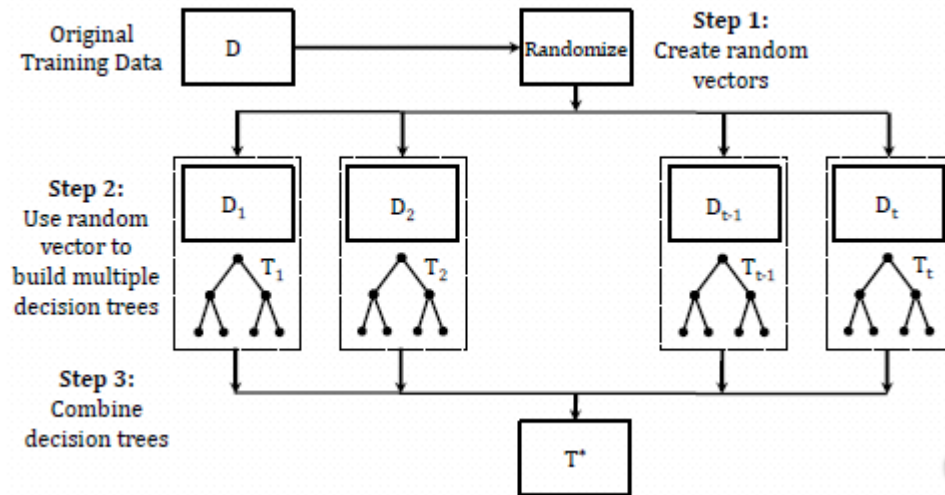


Figure 4.5 Random Forest

5 IMPLEMENTATION

5.1 Dataset Description

This study is performed on the telco dataset. It is a customer database from University of California, Department of Information and Computer Science, Irvine, CA. The dataset contains historical records of customer churn. There are 3332 instances within dataset with 21 attributes for each customer. Churn is the output variable having the value of either true or false. For each customer record, we can find out information about their corresponding inbound/outbound calls count, inbound/outbound SMS count, and voice mail. In the following table, statistical report of the churn dataset is illustrated in detail. By investigating this dataset, it can be observed that this dataset is a complete dataset with no missing values.

Table 1: Statistical report of churn dataset

Item	Type	Distinct	Missing	Unique	Min	Max	Mean	StdDev
State	Nominal	51	0	NaN	NaN	NaN	NaN	NaN
Account Length	Numeric	212	0	16	1	1	101	40
Area Code	Numeric	3	0	0	408	510	437	42
Phone	Nominal	3333	0	3333	NaN	NaN	NaN	NaN
Int'l Plan	Nominal	2	0	NaN	NaN	NaN	NaN	NaN
VMail Plan	Nominal	2	0	NaN	NaN	NaN	NaN	NaN
VMail Msg	Numeric	46	0	4	0	51	8	14
Day Mins	Numeric	1667	0	770	0	351	180	54
Days Calls	Numeric	119	0	10	0	165	100	20
Days Charge	Numeric	1667	0	770	0	60	30	9
Eve Mins	Numeric	1611	0	709	0	364	201	51
Eve Calls	Numeric	123	0	17	0	170	100	20
Eve Charge	Numeric	1440	0	585	0	31	17	4
Night Mins	Numeric	1591	0	586	23	395	201	51
Night Calls	Numeric	120	0	11	33	175	100	19
Night Charge	Numeric	933	0	236	1	18	9	2
Intl Mins	Numeric	162	0	16	0	20	10	3
Intl Calls	Numeric	21	0	3	0	20	4.5	2.5
Intl Charge	Numeric	162	0	16	0	5.4	2.8	0.8
CusServ Calls	Numeric	10	0	0	0	9	1.5	1.3
Churn	Nominal	2	0	0	NaN	NaN	NaN	NaN

Figure 5.1 Statistical report of churn dataset (Vis et al., 2009)

5.2 Dataset Variables Selection

Full dataset includes 21 variables, but the variables including State, Area Code, and Phone do not contain relevant information that can be used for prediction. In that case we have reduced the number of predictors from 21 to 18. The target variable is Churn which has two values, one of them for each customer: yes or no, telling if a customer is a cherner or not. It is only useful for identification purposes.

According to Rulex (Rulex, Inc., 2014) model, the number of variables reduces to 11 variables out of 21. Another study also suggests reducing the number of variables to 11 variables out of 21 as shown in Figure 5.2. The only difference with the Rulex model is that Rulex includes Account Length variable, whereas this study includes State variable. When our algorithms are applied on both of these reduced datasets, the Rulex model generated more efficient results, as a result the Rulex model is chosen as the reduced dataset.

In this thesis, bagging, boosting, and random forest ensemble methods are applied to the datasets that are described above. These algorithms are considered for telecom customer churn classification. In order to perform this study, bagging, boosting, and random forest ensemble methods and decision trees algorithms as base classifiers are used and their results are compared with each other in terms of major performance criteria. Throughout this thesis, decision stump is referred as a weak learner as the base of bagging and boosting ensemble methods. Since it is a weak learner and reducing the number of attributes does not affect its accuracy results, it is only implemented on the full dataset. The classification task comprises of predicting churn based on customer behaviors. Since for many companies, revealing reasons of losing customers, evaluating customer loyalty, taking preventions not to lose customers and developing strategies to regain the customers who churned have become very critical issues.

Attribute Name	Distinct Count	Means	StdDev	Categorical Values
State	51	25.269	14.737	-
Intl_Plan	2	0.097	0.296	Y=323, N=3010
VMail_Message	46	8.099	13.688	-
Day_Mins	1667	179.77	54.467	-
Day_Calls	119	100.43	20.069	-
Eve_Mins	1611	200.98	50.741	-
Night_Mins	1591	200.87	50.574	-
Intl_Mins	162	10.237	10.792	-
Intl_Calls	21	4.479	2.461	-
CutServ_Calls	10	1.563	1.315	-
Churn?	2	-	-	Class Label

Figure 5.2 Final optimal subset of features (Kozielski et al., 2015)

Throughout this thesis, within WEKA software tool decision stump and J48 base classifiers are implemented purely and they are used as base classifiers for bagging and boosting ensemble methods. Each method is applied on the full dataset with 18 variables and on the reduced dataset with 11 selected variables Rulex (Rulex, Inc., 2014) model.

6 Experimental Results

In this section, Decision Stump and J48 Decision Tree classification results are presented first and then the results of the ensemble methods including bagging, boosting and Random Forest are presented. Numerical outputs involving accuracy measures as well as graphical ROC curve outputs are analysed by considering the full and reduced feature sets.

A common method of error rate prediction of a learning algorithm is using stratified tenfold cross-validation. According to this test option in Weka, the dataset is divided into 10 parts initially. These parts comprise of samples which represent approximately the same proportions of the original dataset (Witten et al., 2011). Each part is used in turn for testing while the other parts are used for training. Eventually, the average of the error rates for 10 runs is estimated. For all of our experiments within this study, tenfold cross-validation strategy is applied.

6.1 Base Classifiers: Decision Stump and J48 Implementation

6.1.1 Decision Stump for Full Data Set

Weka output of the Decision Stump algorithm results with statistical values are presented in Figure 6.1 and Figure 6.2. The figures include prediction errors, accuracy metrics, confusion matrix and areas under the ROC curves for both classes. Weighted averages of accuracy values are also available from the output of this algorithm.

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      2873           86.2245 %
Kappa statistic                    0.2397
Mean absolute error                 0.2218
Root mean squared error             0.3345
Relative absolute error             89.3948 %
Root relative squared error         95.015 %
Coverage of cases (0.95 level)     100 %
Mean rel. region size (0.95 level) 100 %
Total Number of Instances          3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,201   0,026   0,571     0,201   0,297     0,280   0,610   0,273   1
                0,974   0,799   0,878     0,974   0,924     0,280   0,610   0,885   0
Weighted Avg.   0,862   0,687   0,833     0,862   0,833     0,280   0,610   0,796

=== Confusion Matrix ===

  a    b  <-- classified as
 97 386 |   a = 1
 73 2776 |  b = 0

```

Figure 6.1 Weka Output of Decision Stump Algorithm on the Full Dataset

As explained before, Decision Stump is a weak learner for this kind of problems. We use it here for comparison with the other more powerful classification methods. The results in Figure 6.1 are consistent with expectations from a weak learner in that they represent very low TP rate (0,201) in churn group and very high FP rate (0,799) in non-churn group. The MCC value (0.28) and ROC area value (0,61) are also low. For more succesful classifiers both of these values should be closer to their maximal values of 100%.

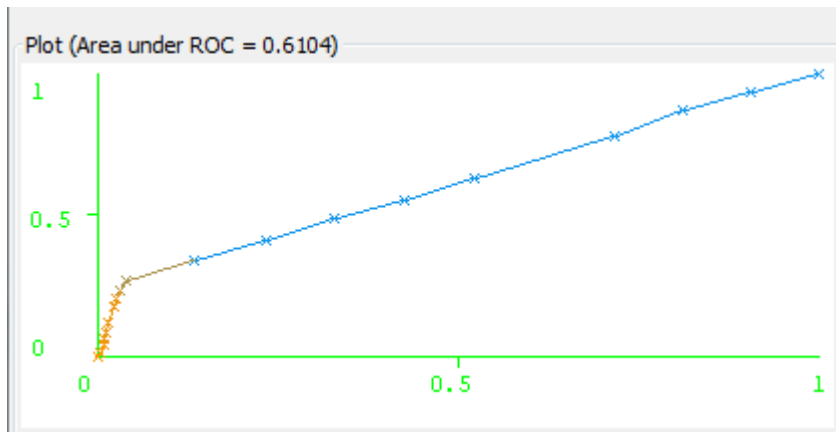


Figure 6.2 ROC Curve of Decision Stump Algorithm on the Full Dataset

6.1.2 J48 Decision Tree for Full Dataset

J48 Decision Trees which have good classification properties for most data types. It is also expected to give better results for telecom churn data. The Weka output for J48 implementation and ROC curve are presented in Figure 6.3 and Figure 6.4.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      3142           94.2977 %
Kappa statistic                    0.7539
Mean absolute error                 0.0806
Root mean squared error            0.2299
Relative absolute error             32.4957 %
Root relative squared error        65.3031 %
Coverage of cases (0.95 level)     95.9784 %
Mean rel. region size (0.95 level)  54.6669 %
Total Number of Instances          3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,725   0,020   0,860     0,725   0,787     0,758   0,855    0,748    1
                0,980   0,275   0,955     0,980   0,967     0,758   0,855    0,942    0
Weighted Avg.   0,943   0,238   0,941     0,943   0,941     0,758   0,855    0,914

=== Confusion Matrix ===

  a  b  <-- classified as
350 133 |  a = 1
 57 2792 |  b = 0
```

Figure 6.3 Weka Output of J48 Algorithm on the Full Dataset

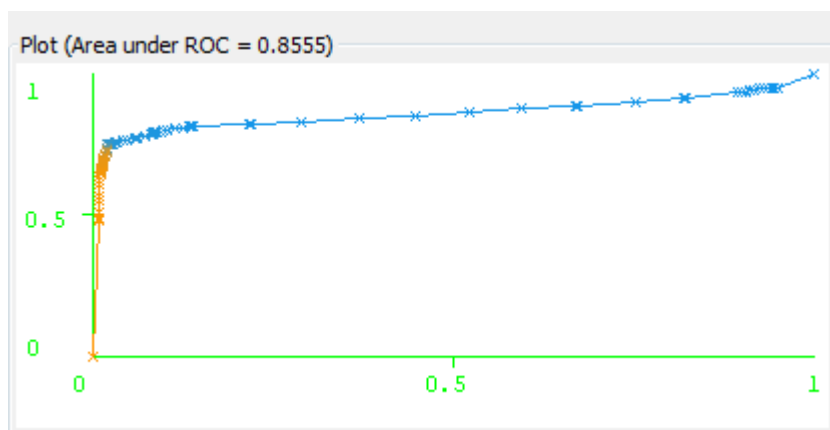


Figure 6.4 ROC Curve of J48 Algorithm on the Full Dataset

As can be seen from Figure 6.4 TP rate, for the churn group has increased to 0.73 and FP rate has decreased to 0.02. These rates and overall correct classification rate have also improved considerably for the non-churn group. The correct classification rate 0.943 is a high value for this dataset. These results are also reflected in higher MCC (0.758) and ROC area (0.855) values. These and other performance measures display the superiority of J48 based classification over simple Decision Stumps.

Similar results have also been obtained for the reduced data implementation. These results indicate that feature selection successfully reduced the dataset size with almost no loss in accuracy metrics of the J48 Classifier.

6.2 Implementation Results of Bagging Ensemble Classification

As the first example of ensemble classification, implementation results for bagging are presented, using Decision Stump and J48 as base classifiers both on full and reduced datasets.

6.2.1 Bagging-Decision Stump Base-Full Dataset

The Weka output for the full data set and the ROC curve are presented in Figure 6.5 and Figure 6.6. Decision Stump is used as a weak base learner for the bagging ensemble classification.

```
Correctly Classified Instances      2855          85.6843 %
Kappa statistic                    0.1368
Mean absolute error                 0.2215
Root mean squared error            0.3292
Relative absolute error            89.2865 %
Root relative squared error        93.519 %
Coverage of cases (0.95 level)     100 %
Mean rel. region size (0.95 level) 100 %
Total Number of Instances          3332

=== Detailed Accuracy By Class ===

          TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
          0,108   0,016   0,531     0,108   0,179     0,191   0,729    0,377    1
          0,984   0,892   0,867     0,984   0,922     0,191   0,729    0,920    0
Weighted Avg.   0,857   0,765   0,818     0,857   0,814     0,191   0,729    0,842

=== Confusion Matrix ===

  a    b  <-- classified as
52 431 |   a = 1
46 2803 |  b = 0
```

Figure 6.5 Weka Output of Bagging Decision Stump Base on the Full Dataset

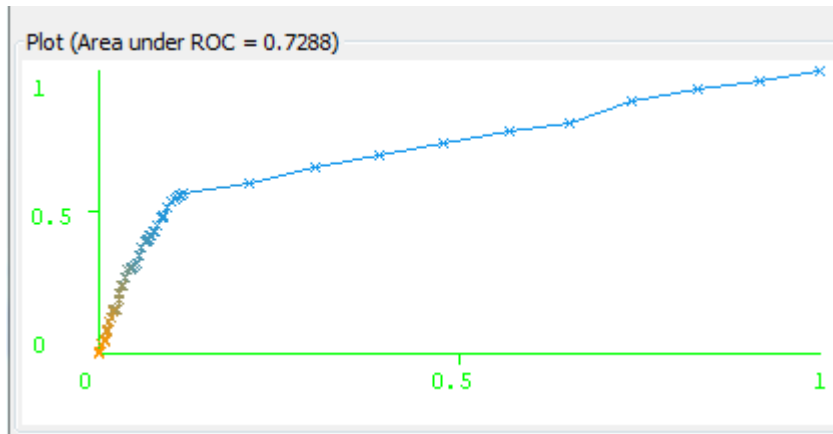


Figure 6.6 ROC Curve of Bagging Decision Stump Base on the Full Dataset

The accuracy values in Figure 6.5 do not show any improvements in comparison with simple Decision Stump classification. Some of the results including FP and TP ratios for both classes are even worse than simple Decision Stump implementation. As the Figure 6.6 indicates, the only improvement is in the area under ROC curve which has increased from 0.61 to 0.73. The results indicate that Decision Stump is not a good choice as the base learner for this kind of bagging ensemble classifier formation.

6.2.2 Bagging-J48 Decision Tree Base-Full Dataset

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      3165           94.988 %
Kappa statistic                    0.7819
Mean absolute error                 0.0823
Root mean squared error            0.2043
Relative absolute error             33.1764 %
Root relative squared error        58.0348 %
Coverage of cases (0.95 level)    97.8091 %
Mean rel. region size (0.95 level) 59.7089 %
Total Number of Instances         3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,739   0,014   0,897     0,739   0,810     0,787   0,913    0,850     1
                0,986   0,261   0,957     0,986   0,971     0,787   0,913    0,970     0
Weighted Avg.   0,950   0,225   0,948     0,950   0,948     0,787   0,913    0,952

=== Confusion Matrix ===

  a    b  <-- classified as
357 126 |  a = 1
 41 2808 |  b = 0

```

Figure 6.7 Weka Output of Bagging J48 Base on the Full Dataset

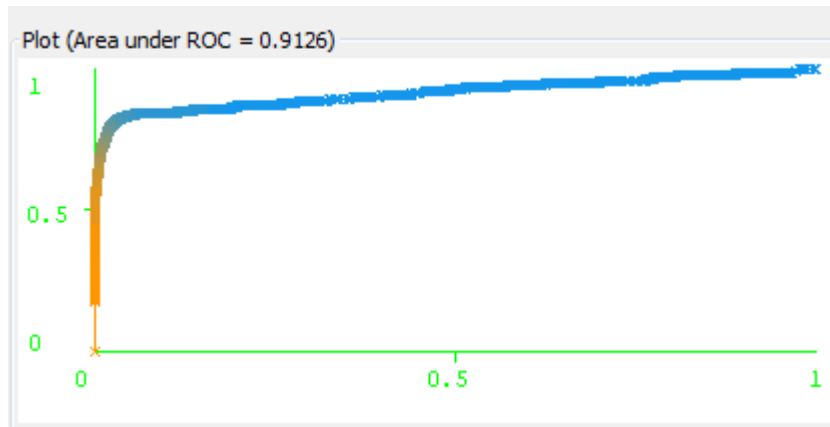


Figure 6.8 ROC Curve of Bagging J48 Base on the Full Dataset

The results in Figure 6.7 and Figure 6.8 show some improvements in performance values in comparison with simple J48 implementation. The TP and FP rates are better as well as MCC and ROC area values in bagging ensemble implementation. The ROC area for example increased from 0.855 to 0.913. These results indicate that, even though moderate, improvements are possible in

classification performance values when bagging is implemented with a reasonably good learner like J48.

6.2.3 Reduced Dataset Bagging-J48 Decision Tree Base

```

Correctly Classified Instances      3167          95.048 %
Kappa statistic                    0.7845
Mean absolute error                0.082
Root mean squared error           0.2032
Relative absolute error            33.0544 %
Root relative squared error        57.7267 %
Coverage of cases (0.95 level)    97.7491 %
Mean rel. region size (0.95 level) 58.8085 %
Total Number of Instances         3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,741   0,014   0,899     0,741   0,813     0,789   0,919   0,854    1
                0,986   0,259   0,957     0,986   0,971     0,789   0,919   0,973    0
Weighted Avg.   0,950   0,223   0,949     0,950   0,948     0,789   0,919   0,955

=== Confusion Matrix ===

  a    b  <-- classified as
358 125 |   a = 1
 40 2809 |   b = 0

```

Figure 6.9 Weka Output of Bagging J48 Base on the Reduced Dataset

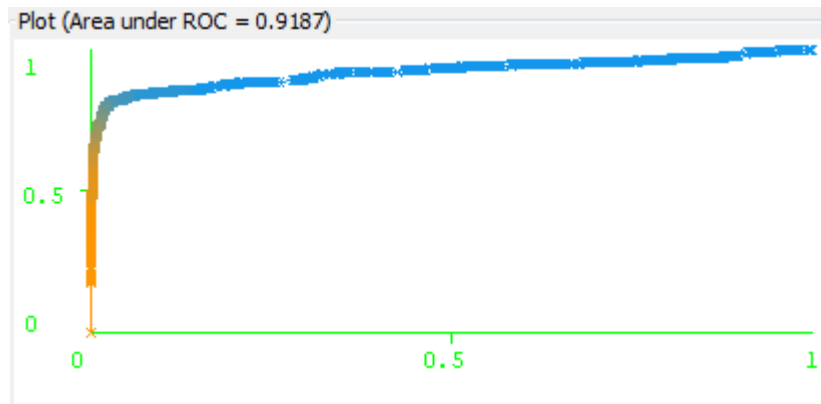


Figure 6.10 ROC Curve of Bagging J48 Base on the Reduced Dataset

The results in Figure 6.9 And Figure 6.10 show that the performance value results are essentially similar to the full data set results in that, there have been slight reductions in TP rates and increases in FP rates. The overall performance criteria like F measure, MCC and ROC area also show somewhat negligible reductions.

6.3 Implementation Results of Boosting Ensemble Classification

In this section, the results for Boosting Ensemble classification are presented for Decision Stump and J48 as base learners. The implementations are performed using Weka Adaboost algorithm.

6.3.1 Boosting-Decision Stump Base-Full Dataset

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      2894           86.8547 %
Kappa statistic                    0.2968
Mean absolute error                 0.1768
Root mean squared error            0.3099
Relative absolute error             71.2552 %
Root relative squared error        88.0136 %
Coverage of cases (0.95 level)    98.7395 %
Mean rel. region size (0.95 level) 82.9832 %
Total Number of Instances         3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,251   0,027   0,614     0,251   0,356     0,334   0,840    0,511    1
                0,973   0,749   0,885     0,973   0,927     0,334   0,840    0,958    0
Weighted Avg.   0,869   0,645   0,845     0,869   0,844     0,334   0,840    0,893

=== Confusion Matrix ===

  a    b  <-- classified as
121  362 |   a = 1
 76 2773 |   b = 0
```

Figure 6.11 Weka Output of Boosting Decision Stump Base on the Full Dataset

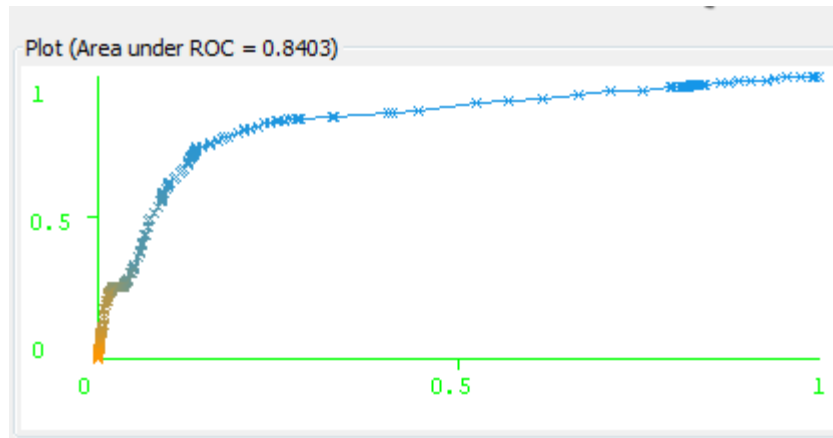


Figure 6.12 ROC Curve of Boosting Decision Stump Base on the Full Dataset

It can be seen from the results in Figure 6.11 and Figure 6.12 that, Decision Stump is not a good choice as a base learner for boosting ensemble classification. Because performance values, although better than simple Decision Stump implementation, are worse than simple J48 and bagging results. Especially, very low MCC values (0.334) and relatively low ROC area value (0.84) and higher accuracy related errors support this claim.

6.3.2 Boosting-J48 Decision Tree Base-Full Dataset

```

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      3164           94.958 %
Kappa statistic                    0.7871
Mean absolute error                 0.0502
Root mean squared error            0.2172
Relative absolute error            20.2426 %
Root relative squared error        61.7005 %
Coverage of cases (0.95 level)    95.8884 %
Mean rel. region size (0.95 level) 50.8703 %
Total Number of Instances         3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,772   0,020   0,865     0,772   0,816     0,789   0,906    0,852    1
                0,980   0,228   0,962     0,980   0,971     0,789   0,908    0,970    0
Weighted Avg.   0,950   0,198   0,948     0,950   0,948     0,789   0,908    0,953

=== Confusion Matrix ===

 a  b  <-- classified as
373 110 |  a = 1
 58 2791 | b = 0

```

Figure 6.13 Weka Output of Boosting J48 Base on the Full Dataset

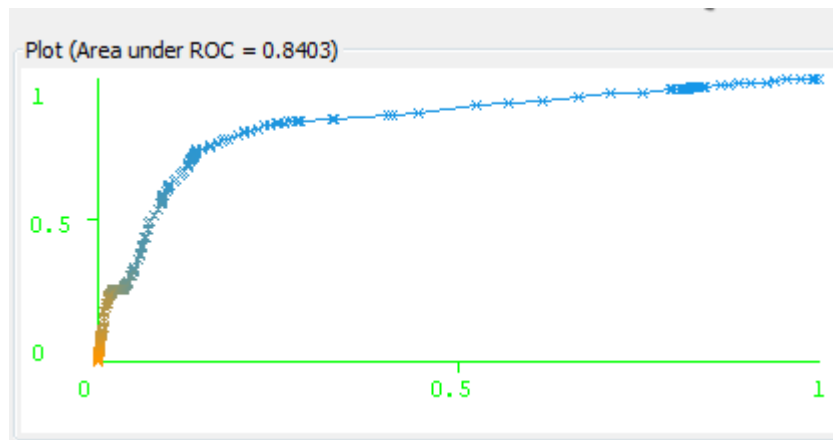


Figure 6.14 ROC Curve of Boosting J48 Base on the Full Dataset

The results in Figure 6.13 and Figure 6.14 show considerable improvements in all performance measures in comparison with simple and Decision Stump base implementation. It is also possible to conclude that, in general, bagging and boosting classification results are comparable for J48 base implementation. The accuracy values, ROC area and MCC are very close in both cases.

6.3.3 Reduced Dataset Boosting-J48 Decision Tree Base

```

Correctly Classified Instances      3167           95.048 %
Kappa statistic                    0.7876
Mean absolute error                0.0492
Root mean squared error           0.2145
Relative absolute error            19.8199 %
Root relative squared error        60.9414 %
Coverage of cases (0.95 level)    95.9184 %
Mean rel. region size (0.95 level) 50.8703 %
Total Number of Instances         3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,758   0,017   0,884     0,758   0,816     0,791   0,917   0,862    1
                0,983   0,242   0,960     0,983   0,971     0,791   0,919   0,974    0
Weighted Avg.   0,950   0,210   0,949     0,950   0,949     0,791   0,919   0,958

=== Confusion Matrix ===

  a    b  <-- classified as
366 117 |    a = 1
 48 2801 |    b = 0

```

Figure 6.15 Weka Output of Boosting J48 Base on the Reduced Dataset

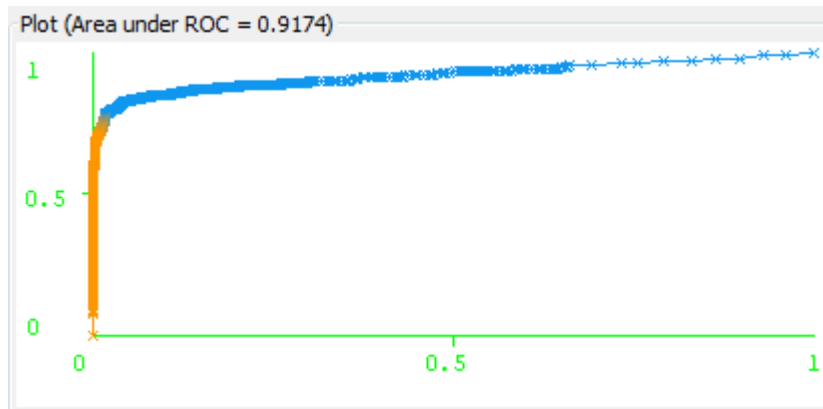


Figure 6.16 ROC Curve of Boosting J48 Base on the Reduced Dataset

The results in Figure 6.15 and Figure 6.16 show that the accuracy values and other statistics are very close to the full dataset results. Consistent with the previous results, we can conclude that feature selection is an effective data reduction method and should be attempted in ensemble churn analysis.

6.4 Implementation Results of Random Forest Ensemble Classification

Random Forest is a similar ensemble classification method to bagging and boosting. Therefore, its performance results are expected to be close to the other methods we have considered.

6.4.1 Random Forest Classification for Full Dataset

```
Correctly Classified Instances      3164           94.958 %
Kappa statistic                    0.7824
Mean absolute error                 0.1042
Root mean squared error            0.2169
Relative absolute error            42.019 %
Root relative squared error        61.6039 %
Coverage of cases (0.95 level)     98.4694 %
Mean rel. region size (0.95 level) 70.9484 %
Total Number of Instances          3332

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,747   0,016   0,887     0,747   0,811     0,786   0,913    0,833    1
                0,984   0,253   0,958     0,984   0,971     0,786   0,913    0,972    0
Weighted Avg.   0,950   0,218   0,948     0,950   0,948     0,786   0,913    0,952

=== Confusion Matrix ===

  a    b  <-- classified as
361 122 |   a = 1
 46 2803 |  b = 0
```

Figure 6.17 Weka Output of Random Forest on the Full Dataset

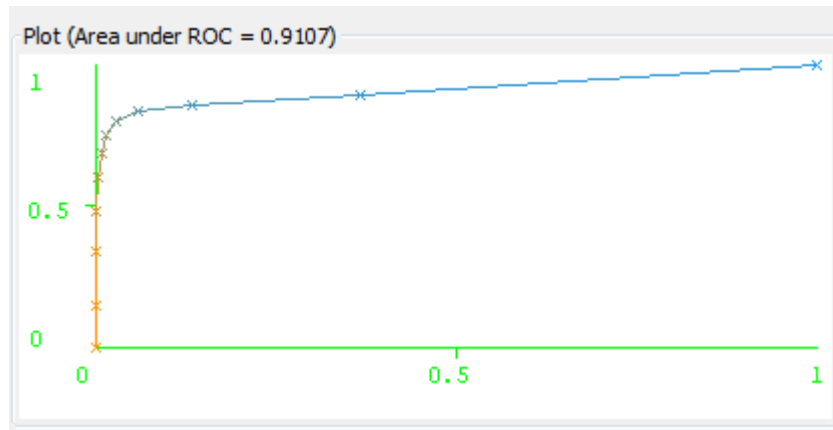


Figure 6.18 ROC Curve of Random Forest on the Full Dataset

Figure 6.17 and Figure 6.18 display similar results of performance measures to bagging and boosting with J48 as base learner. With all these ensemble classification methods we obtain MCC values close to 80% and ROC areas close to 91%. The results obtained by such methods are accepted to be fairly good for the classification problem in this study.

6.4.2 Random Forest Classification for Reduced Dataset

```

Correctly Classified Instances      3166           95.018 %
Kappa statistic                    0.7854
Mean absolute error                 0.1057
Root mean squared error             0.2182
Relative absolute error             42.5997 %
Root relative squared error         61.984 %
Coverage of cases (0.95 level)     98.4394 %
Mean rel. region size (0.95 level) 71.1435 %
Total Number of Instances          3332

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
      0,752   0,016   0,888     0,752   0,814     0,789   0,911    0,830    1
      0,984   0,248   0,959     0,984   0,971     0,789   0,911    0,971    0
Weighted Avg.   0,950   0,215   0,949     0,950   0,948     0,789   0,911    0,951

=== Confusion Matrix ===

  a  b  <-- classified as
363 120 |  a = 1
 46 2803 |  b = 0

```

Figure 6.19 Weka Output of Random Forest on the Reduced Dataset

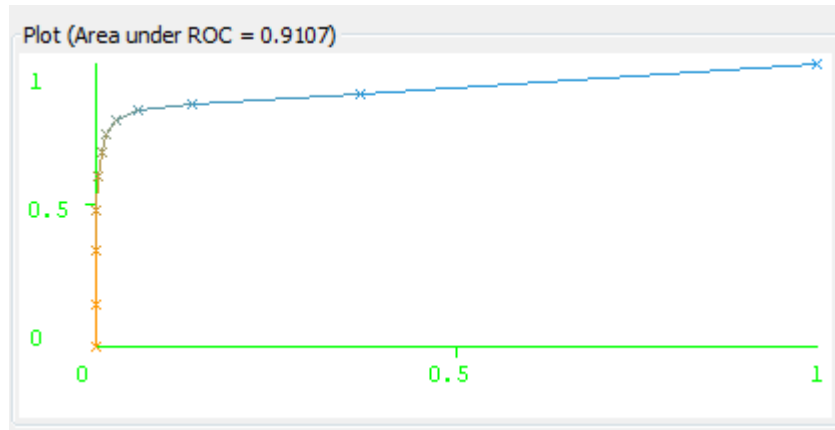


Figure 6.20 ROC Curve of Random Forest on the Reduced Dataset

Similar to the previous results, performance measure values in the reduced dataset are close to those obtained for the full dataset in case of random forest classification. This finding demonstrates again the importance of dataset size reduction by feature selection.

7 Evaluation of the Ensemble Classification Methods Based on Weighted Performance Values

7.1 Accuracy Measures

The performance comparisons in this section are all based on the weighted average values of the accuracy measures. The accuracy measures that are summarized in the tables are as follows: Percentage of correctly classified instances, percentage of incorrectly classified instances, true positive rate, F Measure, precision, ROC Area MCC and Kappa Statistic. Precision can be described as the proportion of churn cases in the results amongst all cases. F Measure is a way of combining recall and precision values into a single performance measure. ROC Area is a way of plotting same information in a normalized form. Kappa Statistic is a measure of agreement between observed and predicted classes.

7.1.1 Decision Stump Base Implementation Accuracy Results Comparisons

Table 7.1 Accuracy Results Comparisons for Decision Stump Base Implementation

	DecisionStump (%values)	Bagging (% values)	Boosting (% values)
Correctly Classified	86,22	85,68	86,85
Incorrectly Classified	13,78	14,32	13,15
TP Rate	86,2	85,7	86,9
Precision	83,3	81,8	84,5
F Measure	83,3	81,4	84,4
ROC Area	61,0	72,9	84,0
Kappa Statistics	23,97	13,68	29,68
MCC	28,0	19,1	33,4

The weighted performance values in Table 7.1 display the shadowing effect of using weighted values instead of individual classes of Churn and Non Churn. All accuracy values, including correct and incorrect classification percentages, precision and F measure values turn out to be similar because of weighting the individual class values. However, the last three statistics namely; ROC area, Kappa Statistic and MCC show reasonable differences for the three methods. The lower weighted values of the statistics show that Decision Stump is not a good choice as a base classifier for two class churn problems. The following graph also displays this fact where the lower values of MCC and Kappa Statistic are noteworthy.

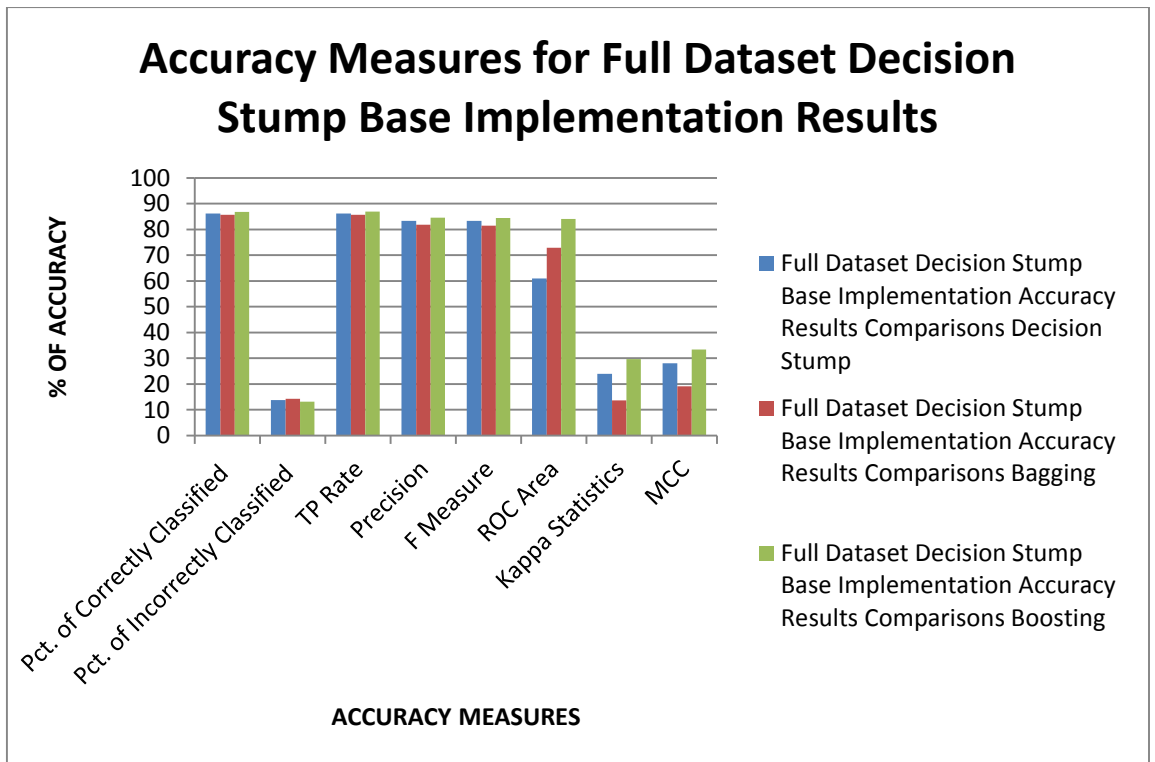


Figure 7.1 Accuracy Measures for Full Dataset Decision Stump Base Implementation Results

From the graph, it can be observed that when bagging and boosting algorithms are used the accuracy of the algorithm increases and performs better when compared to the results of the Decision Stump algorithm itself. The area under the ROC curve

increases relatively when the algorithm is used as the base classifier of the bagging and boosting algorithms.

7.1.2 J48 Base Implementation Accuracy Results Comparisons

Table 7.2 Accuracy Results Comparisons for J48 Base Implementation

	J48 values)	(% Bagging (% values)	Boosting (% values)	Random Forest (% values)
Correctly Classified	94,30	94,99	94,96	94,96
Incorrectly Classified	5,70	5,01	5,04	5,04
TP Rate	94,3	95,0	95,0	95,0
Precision	94,1	94,8	94,8	94,8
F Measure	94,1	94,8	94,8	94,8
ROC Area	85,5	91,3	90,8	91,3
Kappa Statistics	75,39	78,19	78,71	78,24
MCC	75,8	78,7	78,9	78,6

The performance values in Table 7.2 show considerable improvements in performance values when the base classifier is J48. Weighted correct classification rates as well as true classification rates and precision values are about 95% for all classifiers including Random Forest. If the class difference is not clear and both classes are equally important, these results are very good. But, for churn problems, this may not be true as correct results for churn identification is more important. Therefore, using weighted performance values is not a good option in churn analysis.

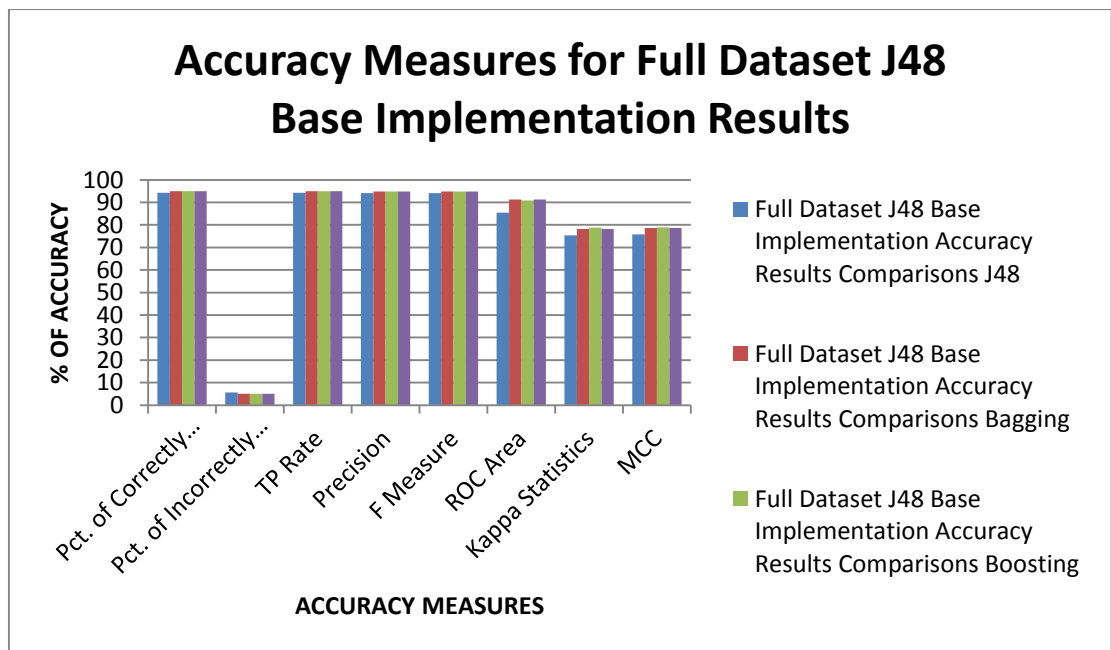


Figure 7.2 Accuracy Measures for Full Dataset J48 Base Implementation Results

From the graph, it can also be observed that when the weighted bagging and boosting algorithms are used, the accuracy of the algorithm increases and performs better when compared to the results of J48 algorithm itself. The area under the ROC curve, Kappa Statistic, and MCC values all increase relatively when the algorithm is used as the base classifier.

7.1.3 Reduced Dataset Implementation Accuracy Results Comparisons

Table 7.3 Accuracy Results Comparisons for Reduced Dataset

	J48 values)	(% Bagging (% values)	Boosting (% values)	Random Forest (% values)
Correctly Classified	94,45	95,05	95,05	95,02
Incorrectly Classified	5,55	4,95	4,95	4,98
TP Rate	94,4	95,0	95,0	95,0
Precision	94,2	94,9	94,9	94,9
F Measure	94,2	94,8	94,9	94,8
ROC Area	86,6	91,9	91,9	91,1
Kappa Statistics	75,79	78,45	78,76	78,54
MCC	76,3	78,9	79,1	78,9

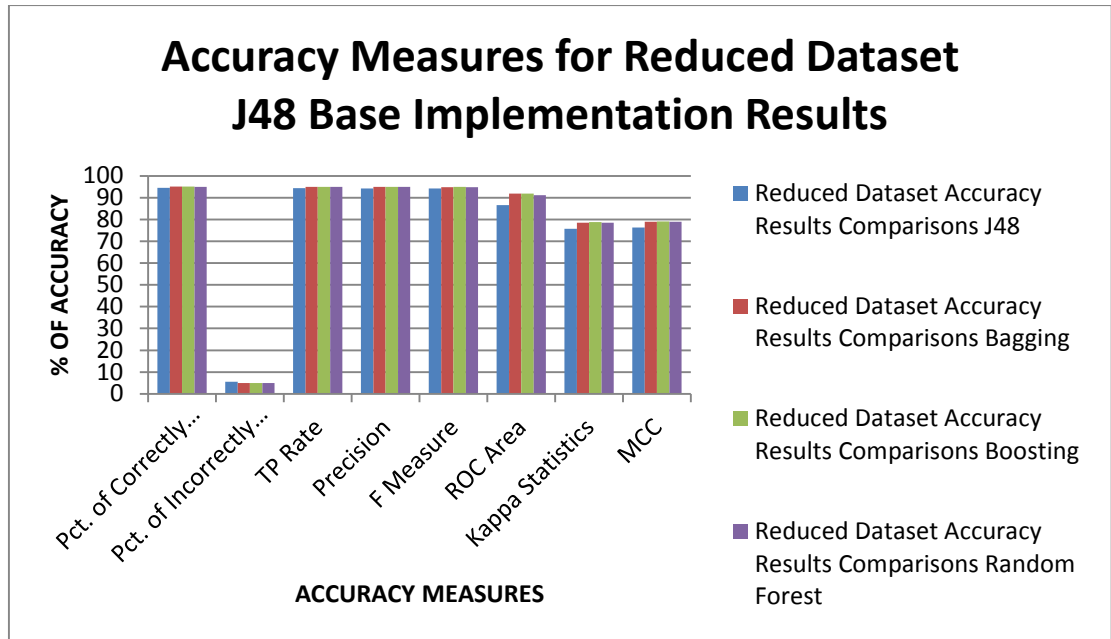


Figure 7.3 Accuracy Measures for Reduced Dataset Implementation Results

From the Table 7.3 and Figure 7.3, similar results to the full data set can be observed: When bagging and boosting algorithms are used the accuracy of the algorithm increases and performs better when compared to the results of the algorithm itself. The area under the ROC curve, Kappa Statistic and MCC values increase relatively when the algorithm is used as the base classifier of the bagging and boosting algorithms.

7.2 Error Rate Comparisons

The error statistics compare true values to their estimates, but they do it in a slightly different way. They tell "how far away" the class estimated values from the true classes are. Sometimes mean square roots are used and sometimes absolute values - this is because when using squared differences, the extreme values have more influence on the result. Generally, accuracy measures; percentage of correctly classified instances, percentage of incorrectly classified instances, true positive rate, F Measure, precision, ROC Area MCC and Kappa Statistic are used in binary classification, but the error statistics are also given in this study as an additional information. The major error rate statistics output by Weka are compared in this section for the classification methods by considering full and reduced datasets. The following table displays the Mean Absolute Error (M.A.E), Root Mean Square Error (R.M.S.E), Relative Absolute Error (R.A.E) and Root Relative Squared Error (R.R.S.R) for the Decision Stump based implementations.

Table 7.4 Decision Stump Base Error Rate Results Comparisons

	MAE	RMSE	RAE	RRAE
Decision Stump (% values)	22,18	33,45	89,39	95,02
Bagging (% values)	22,15	32,92	89,29	93,52
Boosting (% values)	17,68	30,99	71,26	88,01

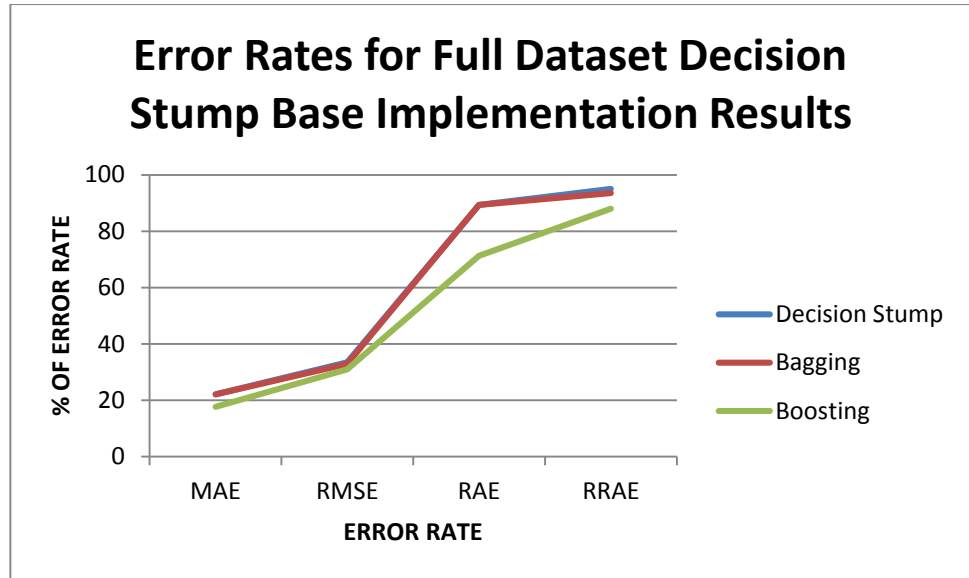


Figure 7.4 Error Rates for Full Dataset Decision Stump Base Implementation Results

From the table and graph, it can be observed that when bagging and boosting algorithms are used the error rate of the algorithm decreases and performs better when compared to the results of the Decision Stump algorithm itself. These results are consistent with the results obtained for the accuracy measures.

Table 7.5 J48 Base Error Rate Results Comparisons

	MAE	RMSE	RAE	RRAE
J48 (% values)	8,06	22,99	32,50	65,30
Bagging (% values)	8,23	20,43	33,18	58,03
Boosting (% values)	5,02	21,72	20,24	61,70
Random Forest (% values)	10,42	21,69	42,02	61,60

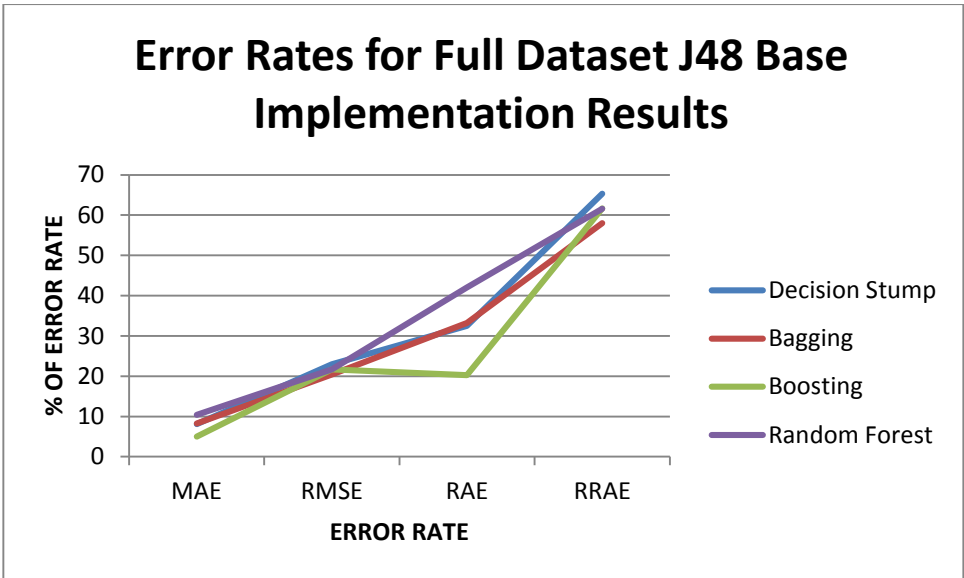


Figure 7.5 Error Rates for Full Dataset J48 Base Implementation Results

From the Table 7.5 and Figure 7.5, it can be observed that when bagging and boosting algorithms are used the error rate of the algorithm decreases and performs better when compared to the results of the algorithm itself. Another point is that the error values for Random Forest are generally higher in both Decision Stump and J48 based implementations.

The error estimates for reduced dataset are given below in Table 7.5 and Figure 7.5. It can be observed that when bagging and boosting algorithms are used the error rate of the algorithm decreases and performs better when compared to the results of the algorithm itself. These results are similar to those obtained for the full dataset.

Table 7.6 Error Rate Results Comparisons for Reduced Dataset

	MAE	RMSE	RAE	RRAE
J48 (% values)	8,03	22,67	32,37	64,40
Bagging (% values)	8,2	20,32	33,05	57,73
Boosting (% values)	4,92	21,45	19,82	60,94
Random Forest (% values)	10,57	21,82	42,60	61,98

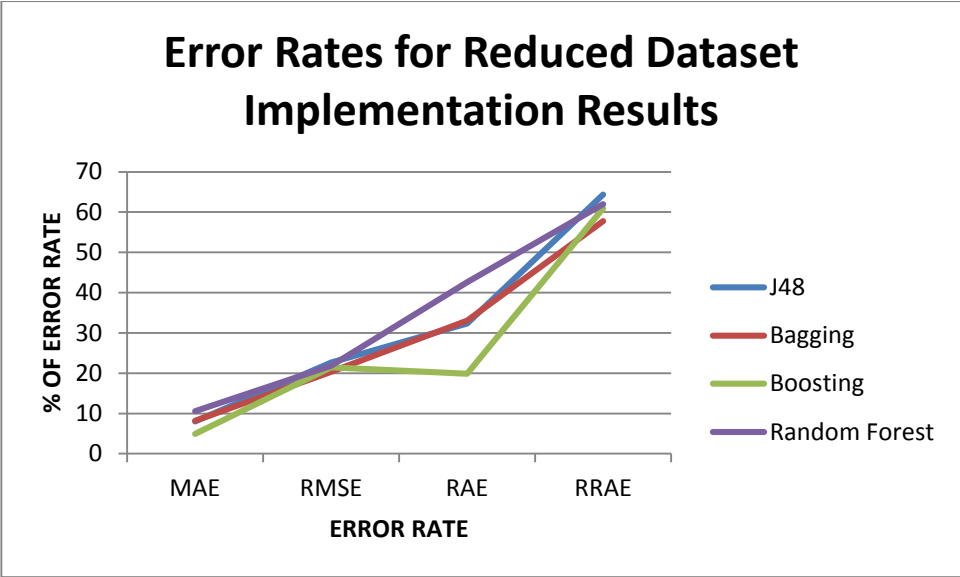


Figure 7.6 Error Rates for Reduced Dataset Implementation Results

8 CONCLUSION AND FUTURE WORK

In this thesis, the performance of the three ensemble methods including bagging, boosting, and random forest are compared. In comparisons, using Decision Stump and J48 as base learners, the performance measures for various combinations of the algorithm options are compared systematically. Decision Stump is chosen as a weak base learner as suggested in the literature. On the other hand, J48 is used as an alternative and powerful base learner for bagging and boosting ensemble classification methods. The results indicate that application of ensemble learning methods provide significant improvements over individual base learners from the perspectives of AUC, sensitivity, and specificity performance indicators. Using Decision Stump as base learner does not improve ensemble classification results. However, J48 as base learner turns out to be a very good option as the implementation results demonstrate improvements in all accuracy measures and error rate statistics. Even though, random forest is applied without using any method as the base learner, it also gives good classification results.

Among all of the three ensemble methods that we applied, boosting with J48 base provided the best results. Based on these results, it can be said that ensemble methods with a good base learner are efficient in churn classification. Another finding of this study is that attribute elimination by a good feature selection method is quite effective. Because, in every implementation option that we considered, reduced datasets gave very close results to the full dataset options. The results also indicate that, using weighted averages of classification statistics for churn and non-churn groups is not a good approach. The weighted values as output by Weka generally display a shadowing effect on the results for individual groups.

In order to move this study one step further, different ensemble methods such as LogitBoost, Stacking, Dagging, BatchPredictorVote, EnsembleSelection methods other than Bagging, AdaBoostM1, and RandomForest with different base learners may be applied on bigger datasets and the accuracy measures of different ensemble methods can be examined. By using more than one base classifier, accuracy measures may be improved. The predictive powers of individual features can be further evaluated and specific recommendations based on these evaluations can be formed for telecom companies.

REFERENCES

- Abbasimehr, H., Setak, M., Tarokh, M.,** 2014, A Comparative Assessment of the Performance of Ensemble Learning in Customer Churn Prediction, The International Arab Journal of Information Technology, 11(6).
- Almana, A.M., Aksoy, M.S., Alzahrani, R.,** 2014, A Survey On Data Mining Techniques In Customer Churn Aanalysis For Telecom Industry, Int. Journal of Engineering Research and Applications, 4(5), 165-171pp.
- Brandusoiu, I., Todorean, G.,** 2013, Churn Prediction In the Telecommunications Sector Using Support Vector Machines, Annals of the Oradea University, Fascicle of Management and Technological Engineering, 1.
- Breiman, L.,** 1996, Machine Learning, 26:123-140pp.
- Chang, Y.T.,** 2009, Applying Data Mining to Telecom Churn Management, International Journal of Reviews in Computing, 69-77pp.
- Clemente, M., Giner-Bosch V., San Matias, S.,** 2010, Assessing classification methods for churn prediction by composite indicators, Universitat Politecnica de Valencia, Spain, Department of Applied Statistics, Operations Research and Quality.
- Freund, Y., Schapire, R.E.,** 1996, Experiment With a New Boosting Algorithm, 148-156pp.
- Gürsoy, U. T.,** 2010, Customer churn analysis in telecommunication sector, İstanbul University Journal of the School of Business Administration, 39(1), 35-49pp.
- Han, J., & Kamber, M.,** 2006, Data Mining: Concepts and Techniques (2nd ed.), USA: Elsevier Inc.
- Hung, S.-Y., Yen, D. C., Wang, H.-Y.,** 2006, Applying data mining to telecom churn management, Expert Systems with Applications, 31, 515-524pp.

Jahromi, A. T., 2009, Predicting Customer Churn in Telecommunications Service Providers, Master Thesis, Lulea University of Technology, Department of Business Administration and Social Sciences, Division of Industrial marketing and e-commerce.

Kozielski, S., Mrozek, D., Kasproski, P., Malysiak-Mrozek, B., Kostrzewa, D., 2015, Beyond Databases, Architectures, and Structures (11th ed.), Ustron, Poland, Springer.

Larose, D. T., 2005, Discovering Knowledge In Data: An Introduction To Data Mining, New Jersey: John Wiley & Sons Inc.

Lemmens, A., Croux, C., 2006, Bagging And Boosting Classification Trees To Predict Churn, Katholieke Universiteit Leuven, Department of Applied Economics.

Lyle, A., 2007, Baseball Prediction Using Ensemble Learning, Master Thesis, University of Georgia, Graduate School.

Oza, N. C., 2009, Ensemble Data Mining Methods, USA, NASA Ames Research Center.

Ren, H., Zheng, Y., Wu, Y., 2009, Clustering Analysis of Telecommunication Customers, The Journal of China Universities of Post and Telecommunications, 16(2), 114-116pp.

Ridi, A., 2014, Churn Analytics for a Telecom, URL: <http://www.rulexinc.com/site/churn-analysis-2/> (Date of Access: February 2, 2016)

Rygielski, C., Wang, J., & Yen, D., 2002, Data mining techniques for customer relationship management, Technology in Society, 24(4), 483-502pp.

Sewell, M., 2008, Ensemble Methods, University College London, Department of Computer Science.

Umayaparvathi, V., Iyakutti, K., 2012, Applications of Data Mining Techniques in Telecom Churn Prediction, International Journal of Computer Applications, 42(20).

Vercellis, C., 2009, Business Intelligence: Data Mining and Optimization for Decision Making, UK: John Wiley & Sons Ltd.

Vijayarani, S., Muthulakshmi, M., 2013, Evaluating The Efficiency Of Rule Techniques For File Classification, International Journal of Research in Engineering and Technology, 2(10).

Viola, P., Jones, M. J., 2004, Robust Real-Time Face Detection, International Journal of Computer Vision, 57(2), 137-154pp.

Vis, J., Zwet, R., 2009, Churn in Telecom dataset (unpublished)

Witten, I. H., Frank, E., & Hall, M. A., 2011, Data Mining Practical Machine Learning Tools and Techniques (3th ed.), USA: Elsevier Inc.

Zhou, Z.-H., 2009, Ensemble Learning, In: S. Z. Li ed. Encyclopedia of Biometrics, Berlin: Springer, 270-273pp.

CURRICULUM VITEA

Gökçe Kalabalık was born in 1987, İzmir – Turkey. She graduated from İhsan Doğramacı Bilkent University, Computer and Instructional Technology Teacher Education Department, and graduated from İhsan Doğramacı Bilkent University, Master of Arts in Computer and Instructional Technology Teacher Education with a CGPA of 3.85 in 2010. Her professional career started in 2010 as an ICT (Information and Communication Technologies) teacher in a private collage in İzmir.

After 3 years of work experience in a private college as an ICT teacher, she decided to start her Master of Science (M.Sc.) education in Yaşar University in order to utilize her academic development in a professional academic environment through the most efficient support of related courses and practical projects.