DETECTION OF CHURNERS IN INTERNET GAMES USING CRM APPROACH: A CASE STUDY ON PISHTI PLUS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

PELIN ERCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2015

Approval of the thesis:

DETECTION OF CHURNERS IN INTERNET GAMES USING CRM APPROACH: A CASE STUDY ON PISHTI PLUS

submitted by **PELIN ERCAN** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Prof. Dr. Ferda Nur Alpaslan _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Mehmet Reşit Tolun
Computer Engineering Department, Aksaray University _____

Prof. Dr. Ferda Nur Alpaslan
Computer Engineering Department, METU _____

Prof. Dr. Ahmet Coşar
Computer Engineering Department, METU _____

Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering Department, METU _____

Assoc. Prof. Dr. Pınar Karagöz
Computer Engineering Department, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: PELIN ERCAN

Signature :

ABSTRACT

DETECTION OF CHURNERS IN INTERNET GAMES USING CRM APPROACH: A CASE STUDY ON PISHTI PLUS

Ercan, Pelin

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Ferda Nur Alpaslan

September 2015, 60 pages

Nowadays, more and more companies start to focus on Customer Relationship Management (CRM) to prevent customer loss. The early detection of future churners is one of the CRM strategies. Since the cost of acquiring new customers is much higher than the cost of retaining the existing ones, it is important to keep existing customers. Churn is an important problem for the game companies as churners impact negatively for potential and existing customers. Data mining can support an individualized and optimized customer management to avoid customer loss.

In this thesis, the problem of player churn in Pishti Plus, which is a multi-player social game, is studied. The purpose is the detection of churners by using the first 24 hours log data of the players. Data used in the prediction model is selected using correlation based filter method. Results of Bayesian Network, Logistic Regression, Sequential Minimal Optimization (SMO), and Simple Classification and Regression Tree (CART) algorithms are compared and an early prediction model is built. Ensemble methods are applied to improve the accuracy of the model. Results indicate that Simple CART algorithm is more successful for predicting churners. The built model predicts the churners with 68.20 % accuracy.

Keywords: CRM, machine learning, data mining, churn prediction, game, Bayesian

Network, Logistic Regression, Simple CART, SMO

ÖZ

MÜŞTERİ İLİŞKİLERİ YÖNETİMİ YAKLAŞIMININ İNTERNET OYUNLARINDA OYUNCU KAYBI PROBLEMİNİN SAPTANMASI İÇİN UYGULANMASI: PİŞTİ PLUS İÇİN BİR ÇALIŞMA

Ercan, Pelin
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Ferda Nur Alpaslan

Eylül 2015 , 60 sayfa

Günümüzde birçok şirket müşteri kaybını azaltmak için Müşteri İlişkileri Yönetimi'ne önem vermeye başlamıştır. Mevcut müşteri kaybının önceden tespit edilmesi Müşteri İlişkieri Yönetimi stratejilerinden biridir. Yeni müşteri çekmek mevcut müşteriyi korumaktan daha maliyetli olduğu için mevcut müşterinin korunması önemlidir. Müşteri kaybı, potansiyel ve mevcut oyuncuları negatif etkilediği için oyun şirketleri için önemli bir sorundur. Veri madenciliği müşteri kaybını önlemek için kişiselleştirilmiş ve optimize edilmiş bir müşteri yönetimi sağlar.

Bu tezde, çok oyunculu sosyal bir oyun olan olan Pişti Plus'ın oyuncu kaybı problemi üzerinde çalışılmıştır. Oyuncuların ilk 24 saatlik verileri kullanılarak oyunu bırakacak oyuncuların tespit edilmesi amaçlanmıştır. Tahmin için kullanılan veriler ilişki tabanlı filtreleme yöntemi kullanılarak seçilmiştir. Bayesian Network, Logistic Regression, SMO ve Simple CART algoritmalarının sonuçları karşılaştırılarak erken tahmin modeli oluşturulmuştur. Elde edilen modelin doğruluğunu arttırmak için ensemble yöntemleri uygulanmıştır. Sonuçlar Simple CART algoritmasının oyunu terk edecek oyuncuları tahmin etmede daha başarılı olduğunu göstermektedir. Geliştirilen model oyunu bırakacak oyuncuları %68.20 doğrulukla tespit etmektedir.

Anahtar Kelimeler: Müşteri İlişkileri Yönetimi, machine learning, veri madenciliği, churn prediction, oyun, Bayesian Network, Logistic Regression, Simple CART, SMO

To My Familiy

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Ferda Nur ALPASLAN for her guidance, encouragement, and support. I also would like to thank Dr. Özgür ALAN, who is the founder and the CEO of SNG ICT, for providing the log data of Pishti Plus, his guidance, encouragement, and support.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AUC	Area Under Curve
CART	Classification And Regression Tree
CFS	Correlation Based Feature Selection
CRM	Customer Relationship Management
GA	Genetic Algorithm
QP	Quadratic Programming
ROC	Receiver Operating Characteristic
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

Today, more and more companies start to focus on Customer Relationship Management, CRM. It is a strategy for building, managing, and establishing loyal and permanent customer relationships. Customers are the most important people for any organization. Every organization wants satisfied customers who remain loyal to them and strengthening their relationships with the organization. Because of this, an organization should have a clear strategy for customers [46].

"Churn is a word derived from change and turn." [30]. Customer churn refers to when a customer end his or her relationship with a company. If it is not controlled carefully, it may bring a company to its knees. Cost of customer churn includes advertisement cost, decrease in income, organizational chaos, planning chaos, budgeting chaos, cost of customer retention, and cost of customer reacquisition. Furthermore, previous studies have shown that gaining new customers is more costly than retaining the existing ones [25]. Therefore, if we have a small enhancement in customer retention, it can make a remarkable increase in profit [21].

One of the CRM strategies is the early detection of future churners. Finding the churners can help companies retain their customers [8]. The companies should be able to predict the behavior of customers correctly and set up links between customer loss and the factors associated with the customer attrition to reduce the customer churn. To identify future churners predictive models can be developed and a numerical measure, which assigns to each client their tendency to churn in terms of probability, can be provided. This information is useful for marketing expedition for the company to retain the customers.

Data mining can provide customer insight, which is essential for set up an successful CRM strategy. It can conduct customized interactions with customers. Therefore, it is possible to increase pleasure of customers and having gainful customer relationships through data analysis. Data mining can support a personalized and optimized customer management all over the phases of the customer life-cycle, from the gaining customers and setting up a healthy relationship with them to the prevention of customer loss and the recovering lost customers [46].

Churn analysis has been studied in different disciplines for decades, especially, in retail banking, insurance, telecommunication, Internet service providers, social networks and game industry. In the game industry, the meaning of churn is stop playing the game. Churn is a big problem for the gaming companies, because churners have a negative impact negative in the word of mouth reports for existing and potential players leading to further erosion of customer base [23]. This can cause a significant decrement in revenues. Since it has a social influence among players, it is more important for Massively Multiplayer Games.

In this thesis, churn prediction in the game industry is studied. Prediction is made using only the first day data of the players and it is the first in this context. Prediction result indicates whether the player will continue to play or not by looking at the first 24 hours log data of the player. Bayesian Network, Logistic Regression, SMO and CART algorithms are used for churn prediction and results are compared. To improve the performance ensemble methods are applied.

The rest of the thesis is organized as follows. Chapter 2 gives information about previous studies. Chapter 3 explains theoretical background and the tool, which is used. In Chapter 4, data and feature set, which are used in this study, are explained in detail. Methods are described in Chapter 5, comparison of algorithms and evaluation are explained in Chapter 6. Finally, Chapter 7 concludes with an overview of the study and discussion of the results.

CHAPTER 2

RELATED WORK

"If you build a game, someone will try it". "They may come but will they stay?" This is a crucial question for game developers [9]. Therefore, there are lots of study in this subject.

Jaya Kawale et al. made a study on EverQuest II to understand the effects of a social network in churn [23]. They used a social influence vector, which has negative and positive influence components, by taking into account personal engagement of players in games and social influence among players. Then, they measured a player personal engagement based on his/her activity patterns and used it in churn prediction.

Their method improved prediction accuracy for their dataset according to prediction using the player engagement factor or the conventional diffusion model. They used the average length of sessions to measure engagement in game play, and found that the churners have a decreasing average session length according to non-churners. In addition, the session lengths of churners were shorter in the later periods of time than in the initial periods.

Players can become bored once they have explored all the game's content, and may stop playing. Hence, game companies need to know how fast players consume content so they can schedule the release of new content accordingly. Once new content goes live, the overall player base may progress through it faster or slower than the developers expected. Knowing the response of players to new content is so important that when the developers realize there is a problem, they often release a "hot x" patch addressing it the same day.

Because of this Thomas Debeauvais et al. measured the effect of game-play, real-life status, and in-game sociality on player loyalty by using the following metrics: stop rate, number of years have been playing the game, and weekly play time. They differentiated players concerning demographic categories such as age, gender, marital status, and region. Then, they analyzed the differences between long-term and short-term players [9].

In another study, Borbora, Z.H. et al. studied the user churn problem in a social gaming environment [7]. They studied the time period just before the player stops using the social gaming product and compared the activity of churners with the regular players. Based on this discriminative analysis the authors identified several features related to signals like engagement, persistence, and enthusiasm. The discriminative features were used to create a distance metric, which they call wClusterDist that gives the distance between two sets of users (ones who are likely to leave versus normal users).

Bauckhage, C. et al. used random process theory to draw inferences about player engagement by using behavioral telemetry data, which belongs to over 250,000 players [5]. They used the information of how long separate players have played five different action-adventure and shooter games, and applied lifetime analysis techniques to detect common patterns. They found that the Weibull distribution has a satisfactory performance of the statistics of total playing times in these games. This means the interest of an ordinary player, who plays one of the games, changes based on a non-homogeneous Poisson process. Therefore, it is possible to forecast when the players stop playing by using data, which belongs to the earliest playtime behavior of them of a game.

Fabian Hadiji et al. addressed, the challenge of predicting player churn in freemium games, and a machine learning approach presented, which can be applied across games, under real-life conditions, i.e. in the wild [15]. Furthermore, they defined features, which are universal for games such as session length, playtime, and session intervals. They examined the importance of these features in predicting player churn, and developed a churn prediction model using them. They tested their approach using data from five commercial games across mobile and web based social-online plat-

forms and predicted churn from the game's social network.

Julian Runge et al. focused on predicting churn for high value players and tried to evaluate the business effect, which could be obtained from a predictive churn model [40]. They compared the prediction performance of four classification algorithms using two games, Diamond Dash and Monster World Flash. Then, they implemented a hidden Markov model to clearly address temporal dynamics. They found that a neural network has the best prediction performance based on the area under curve (AUC).

They designed and implemented an A/B test on one of the games, using free in-game currency as a stimulation to keep players to evaluate the business impact of churn prediction. The results show that connecting players quickly before the predicted churn event reasonably improves the efficiency of communication with players. They also showed that distributing free in-game currency does not affect the churn substantially. They found that changing game-play experience of players is only way to kept them and that cross-linking is an important measurement to cope with churn.

We can conclude that the values such as playtime, session length, average length of sessions, and session intervals are the vulnerable values for churn prediction. In addition, these values are common for most of the games. On the other hand, the earliest playtime behavior of the players can give us clue about future of the player and contacting players quickly before the predicted churn event can improve the efficiency of communication with players. These are taken into consideration in this study.

CHAPTER 3

BACKGROUND

"Data mining is the process of discovering interesting knowledge from large amounts of data." [20]. It includes a combination of techniques from different disciplines such as statistics, high performance computing, database technology, machine learning, neural networks, pattern recognition, information retrieval, data visualization, image and signal processing, and spatial data analysis. In general, data mining consist of the following steps:

1. **Data Integration:** First, all data is collected and added together from the different sources.

2. **Data Selection:** Then, the data will be analyzed is taken from the database.

3. **Data Cleaning:** The collected data may contain missing values, errors, inconsistent or noisy data. In this step, such anomalies are removed.

4. **Data Transformation:** Even after cleaning, the data is not ready for mining. It can be needed to transform the data into a suitable form by using some techniques such as smoothing, aggregation, and normalization.

5. **Data Mining:** In this step, data is ready to apply data mining techniques such as association analysis, clustering, and classification to discover the interesting patterns.

6. **Pattern Evaluation:** This step contains identifying the interesting patterns using some interestingness measurements.

7. **Knowledge Representation:** In this step, visualization and knowledge representation techniques are used to help user to make better decisions by using the extracted information.

By using data mining, regularities, high-level information or interesting knowledge can be brought out and observed from different angles. This information can be used in information management, query processing, decision making, process control etc. In this thesis, data mining is used for future prediction. In the following sections Data Preprocessing, Model Building, and Model Evaluation steps are described.

3.1 Data Preprocessing

Data Preprocessing includes data integration, data selection, data cleaning, and data transformation. Preprocessing starts with collecting the data and retrieving relevant data to the analysis. Then anomalies, such as errors, missing values, noisy or inconsistent data are removed. Finally, data is transformed into appropriate form for mining. In preprocessing, the most important step is data selection, which is defined as feature selection in the following section.

3.1.1 Feature Selection

The purpose of feature selection is to find a subset of inputs by eliminating features, which have no predictive information or are irrelevant. It has proven in both practice and theory, feature selection increases predictive accuracy and reduces complexity of learned results to be effective in enhancing learning efficiency [39].

In supervised learning, purpose of the feature selection is to determine a feature subset, which gives higher classification accuracy. The feature selection methods are typically presented in three classes, which are described in the following section, according to how they combine the selection algorithm and build the model (Figure 3.1) [34]).

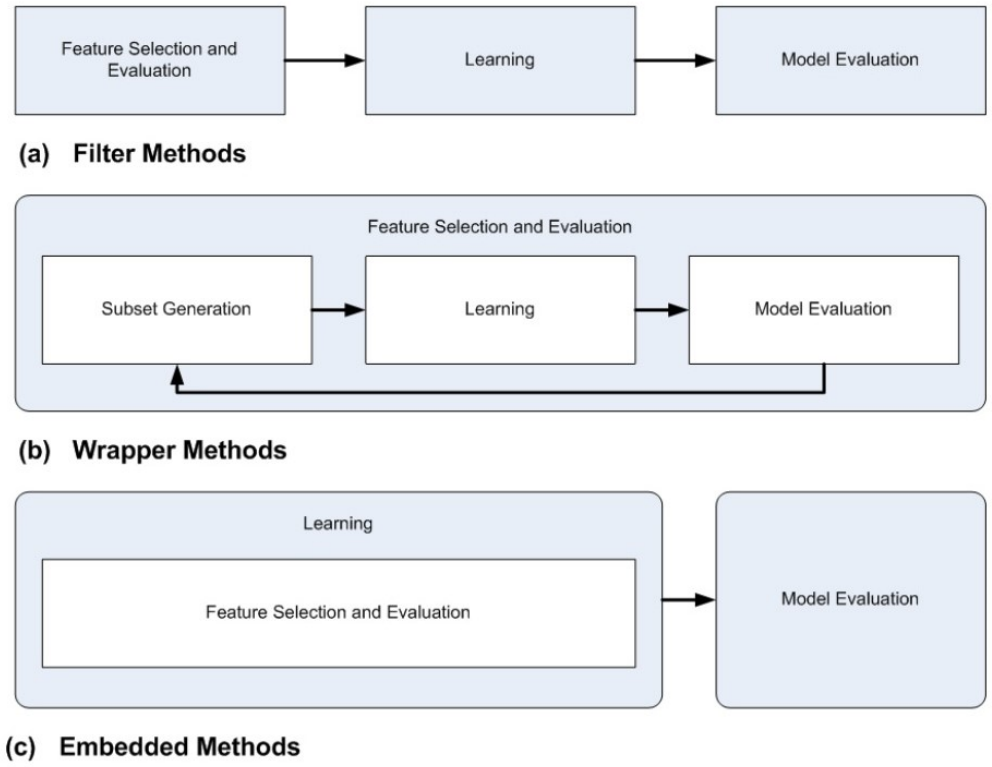

Figure 3.1: Filter, wrapper and embedded feature selection scheme

3.1.1.1 Filter Methods

Filter methods evaluates the relevance of features by just using the actual properties of the data. Generally, a feature relevance value is calculated, and the features have the less relevance are removed. Then, selected features are used as input to the classification algorithm [41].

A widespread disadvantage of filter methods is that they consider each feature separately, and disregard the interaction with the classifier. Thereby, ignoring feature dependencies, can cause worse classification performance than other types of feature selection techniques. Because filter methods do not consider the relationships between variables, they tend to select redundant variables.

One of the advantages of filter methods is that they are independent of the classification algorithm. Other advantages are that they are computationally simple and fast, and it is easy to scale very high-dimensional datasets. As a result, once the feature se-

lection is performed, it can be used by different classifiers. Therefore, filter methods are mainly used as a preprocessing method. In this study Correlation-based Feature Selection Algorithm (CFS), which is a filter method algorithm, is used.

Correlation-based Feature Selection

An attribute subset is good for this method if the attributes it contains are highly correlated with the class attribute and not strongly correlated with one another. CFS Algorithm considers the predictive value of attributes with the degree of inter-redundancy. Since irrelevant features have weak correlation with the class, they can be ignored. As redundant features are highly correlated with the other feature or features, they should be screened out [16].

In CFS algorithm, features are accepted depending on their level of extent in its predicting class. CFS's feature subset evaluation function is given below:

$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \tag{3.1}$$

where M_S is the heuristic merit of feature subset, S contains k number of features. $\bar{r}_{cf}$ is the average correlation between feature and class. $\bar{r}_{ff}$ is the average inter correlation between two features.

Equation 3.1 is the heart of CFS and arranges feature subsets in the search space of all potential feature subsets. In fact, Equation 3.1 is Pearson's correlation coefficient, where all variables have been standardized. It shows the relation between a class and a feature.

CFS is generally used with search strategies such as forward selection, backward elimination, best-first search and genetic search. Forward selection starts with no feature and greedily adds one feature until there is no addition results in a higher evaluation. Conversely, backward elimination starts with whole feature set and greedily removes one feature until the evaluation does not degrade. Best first search starts with forward selection. Then, it uses backward elimination. To prevent investigating

the whole search space, a termination criterion such as ceiling on the total number of combining features with no improvement over the existing best subset can be used.

A genetic algorithm (GA) has three main operators: reproduction, cross-over, and mutation. Reproduction selects good features; crossover combines good features to generate better feature sets; mutation changes a feature locally to create a better feature set [22] [13]. The result is evaluated and tested for stopping criteria of the algorithm in each generation. These three steps are repeated and then re-evaluated, if the stopping criteria is not satisfied. GA is a stochastic general search method, and it can search large spaces effectively, which is generally needed in case of attribute selection. In addition, GA performs a global search, in contrast to many search algorithms, which perform a local, greedy search.

3.1.1.2 Wrapper Methods

In the wrapper approach, the feature subset selection is done using the induction algorithm as a black box. There is no need for the knowledge of the induction algorithm, just the interface of the algorithm is used. The feature subset selection algorithm searches for a subset using the induction algorithm as part of the evaluation function [26].

The wrapper approach makes a search in the space of potential parameters. The search needs an initial state, a stopping state, a state space, and a search engine such as best-first search, hill-climbing, genetic search etc. There are two main disadvantages of these methods:

- The insufficient number of observations increases over-fitting risk, and
- The large number of variables increases computation time.

3.1.1.3 Embedded Method

Embedded methods are different from other feature selection methods in the way feature selection and learning interact. Filter methods do not involve learning. Wrapper

methods involve a learning machine to evaluate the feature subsets, but do not have information about the particular structure of the regression or classification function. Therefore, wrapper methods can be merged with any machine learning algorithm.

In embedded methods, in contrast to wrapper and filter approaches, the feature selection part and the learning part cannot be isolated. The structure of the observed class of functions has an important role for embedded methods [29]. Summary of feature selection techniques are given in Table 3.1 [41].

Table 3.1: Summary of feature selection techniques

		Advantages	Disadvantages	Examples
Filter	Univariate	Fast Scalable Independent of the classifier	Ignores feature dependencies Ignores interaction with the classifier	Chi-square Euclidean distance t-test Information Gain, Gain ratio
	Multivariate	Models feature dependencies Independent of the classifier Better computational complexity than wrapper methods	Slower than univariate techniques Less scalable than univariate techniques Ignores interaction with the classifier	Correlation based feature selection (CFS) Markov blanket filter (MBF) Fast correlation based feature selection (FCBF)
Wrapper	Deterministic	Simple Interacts with the classifier Models feature dependencies Less computationally intensive than randomized methods	Risk of over-fitting More prone than randomized algorithms to getting stuck in a local optimum (greedy search) Classifier dependent selection	Sequential forward selection(SFS) Sequential backward elimination (SBE) Plus q take-away r Beam search
	Randomized	Less prone to local optima Interacts with the classifier Models feature dependencies	Computationally intensive Classifier dependent selection Higher risk of over-fitting than deterministic algorithms	Simulated annealing Randomized hill climbing Genetic algorithms Estimation of distribution algorithms
Embedded		Interacts with the classifier Better computational complexity than wrapper methods Models feature dependencies	Classifier dependent selection	Decision trees Weighted naive Bayes Feature selection using the weight vector of SVM

3.2 Model Building

When building a model there are two options: unsupervised and supervised learning. In unsupervised learning, the correct results are not provided to the model during the training. It is used to cluster the input data in classes according to their statistical properties. In supervised learning, both the desired results and the input is involved by training data.

3.2.1 Classification and Prediction

Classification is the method of obtaining a set of functions, which represent and distinguish data classes. In classification, the derived model is obtained by analyzing a set of training data whose class label is known. Then the training data is used to predict the class of instances whose class label is unknown [17]. The obtained model can be illustrated in different forms, such as decision trees, classification (IF-THEN) rules, neural networks or mathematical formula.

Churn prediction is a binary classification work, which differentiates churners and non-churners. Churn prediction can be made by using various data mining and statistical classification methods. In the literature, there are many prediction algorithms, which have been used to forecasting the customer churn. This section provides a general information about Logistic Regression, SMO, Bayesian Network and Simple CART classification algorithms.

3.2.1.1 Bayesian Network

"Bayesian Network is a specific type of graphical model, which is a directed acyclic graph. That is, all the edges in the graph are directed and there are no cycles." [43]. A Bayesian model defines dependencies among all variables. It can handle missing data. Since it can be used to understand about a problem domain, to learn causal relationships, and to predict the results of intervention. Hence, the model has both probabilistic and causal semantics. It is perfect to represent combining data and prior knowledge. In addition, Bayesian statistical methods in composition with Bayesian networks offer a well-organized approach for keeping away from the over-fitting of data [18].

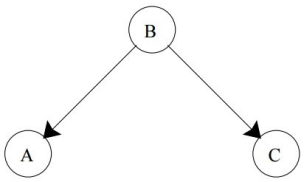

Figure 3.2: A Bayesian network (Probabilities are omitted).

Figure 3.2 illustrates a Bayesian Network. Its set of edges is $E = (B, A), (B, C)$. Since A and C conditionally independent of each other, it can be said that: $P(A|B, C) = P(A|B)$. That means, the probability of A is conditionally depends on B and independent from the value of C. It also can be said that $P(C|A, B) = P(C|B)$. The edges in the Bayesian Network correspond to the joint probability distribution of the connected variables. In this example, the joint distribution of all the variables is

$$P(A, B, C) = P(A|B) \cdot P(B) \cdot P(C|B). \tag{3.2}$$

Normally, given nodes $X = X_1, ...X_n$, the joint probability function for any Bayesian Network is

$$P(X) = \prod_{i=1}^{n} P(X_i|parents(X_i)). \tag{3.3}$$

The joint probability of all the variables is calculated from the product of individual probabilities of the nodes, which is given by using the value of their parents. This indicates that the directed edges represent direct dependence among the variables.

3.2.1.2 Logistic Regression

Regression models aim to determine a functional connection between one variable, which is called the dependent variable (response), and independent (explanatory) variables. Logistic regression can be used, when dealing with qualitative response variable. A qualitative response problem can be divided into binary response problems. Logistic regression converts binary classification problems into linear regression problems using a proper transformation. It can be used in different contexts [44].

3.2.1.3 Sequential Minimal Optimization

Sequential Minimal Optimization (SMO) is an algorithm, which can easily solve the Support Vector Machine (SVM) quadratic programming (QP) problem. SMO divides

the QP problem into QP sub-problems, using Osuna's theorem. It does not use numerical QP optimization steps or any extra matrix storage [36].

It chooses the smallest possible optimization problem in each step to solve. Since the Lagrange multipliers must fit in a linear equality constraint, the smallest possible optimization problem requires two Lagrange multipliers (Figure 3.3) for the standard SVM QP problem. At every step, SMO selects two Lagrange multipliers to jointly optimize, finds the optimal values for them, and updates the SVM to indicate these new optimal values. There are two components: a heuristic for selecting which multipliers to optimize, and an analytic method to solve for the two Lagrange multipliers.

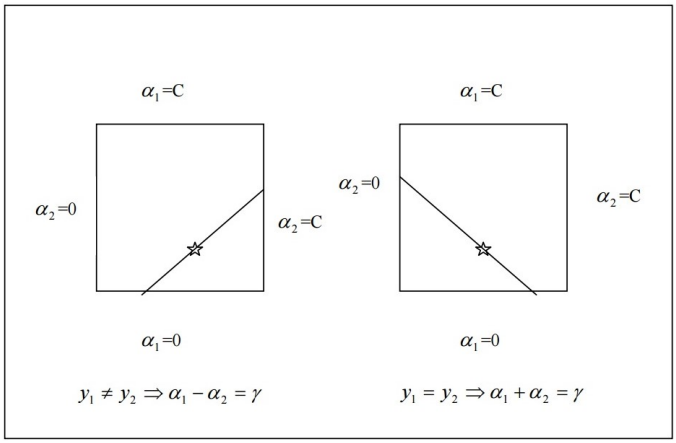

Figure 3.3: The two Lagrange multipliers

The two Lagrange multipliers must conform all the constraints of the full problem. While the linear equality constraint causes the Lagrange multipliers to lie on a diagonal line, the inequality constraints cause them to lie in the box. Thus, one step of SMO guarantees to find an optimum of the objective function on a diagonal line.

3.2.1.4 CART

Until a specified criteria has been fulfilled, a Classification And Regression Tree (CART) is built by recursive divisions of an instance into subgroups. The tree keeps growing until the impurity falls below a user defined threshold. Nodes in a decision tree represent test conditions and branching depends on the value of the attribute, which is tested [30].

The tree represents a group of rules. It classifies the instances by traversing the tree up to a leaf node is reached. The label of this leaf node is assigned as a label of the instance. Figure 3.4 indicates a simplified churn prediction decision tree for the telecommunication sector. The label of the leaves are assigned to the customers as churner or non-churner.

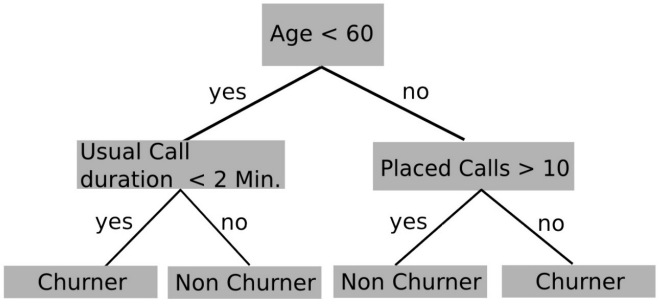

Figure 3.4: A simplified churn prediction decision tree

3.2.2 Ensemble Methods

Model Evaluation involves considering different types of models and choosing the best one according to their predictive performance. There are lots of methods, which are generated to accomplish this. These techniques apply different models to the same data set and then make a comparison their performance to select the best one. These methods are regarded as the heart of predictive data mining, include: Boosting, Bagging (Voting, Averaging), and Stacking (Stacked Generalizations), which are described in this section [47] [44].

An ensemble classifier combines or uses multiple classifiers to improve robustness and accomplishes an improved classification performance from any of the single classifiers. Because of this, an ensemble classifier has better accuracy than single classification techniques and has the advantage that it can be adapted to any changes in the monitored data more accurately than single model. This method uses a divide and conquer approach where a complex problem is divided into multiple sub problems, which are easier to understand and solve. Moreover, this method is more resilient to noise than the use of a single classifier.

The success of the ensemble approach is based on the diversity in the individual classifiers with regard to incorrectly classified instances [31]. There are four ways to do this, the first one is to use distinct training data to train single classifiers, the second one is to use distinct training parameters, the third one is to use distinct features to train the classifiers and the final one is to combine different types of classifier [37].

3.2.2.1 Bagging

The meaning of combining the decisions of various models is merging the different outputs into a single prediction [50]. Outputs of the classifier are often normalized between 0 and 1. These values are depicted as the support given by the classifier to each class. This allow us to ensemble models through algebraic combination rules (majority voting, or combinations of probabilities such as maximum, minimum, sum of, and product of). The easiest way to do this for the classification is to take an average vote in the case of numeric prediction. Boosting and bagging both can use this approach, but they obtain the particular models separately. In bagging, the models have equal weights. On the other hand, in boosting, to give more influence to the more successful model, weighting is used.

In bagging, a number of training sets have the same size are randomly selected from the problem domain and a machine learning method is used to create a decision tree for each dataset. A different feature being selected at a specific node, changes the structure of the subtree under this node. Since that decision tree induction is an unsteady process, slight changes in the training set can cause a remarkable change in branching. This indicates that there are test instances for which some decision trees can accurately predict and others cannot.

Considering the individual decision trees as the experts. The trees can be combined by having them vote on each test instance. The class has the highest vote is taken as the correct one. Usually, predictions made by voting are more trustworthy, since more votes are taken into consideration but improvement is not guaranteed. The combined classifier may rarely be less accurate than a decision tree, which is built from only one of the datasets.

Voting

Voting is the simplest method to combine predictions of different classifiers. Majority voting is the simplest form of voting. In majority voting, each classification model uses just one vote according to its own classification. The common prediction is determined by the majority of the votes, and the class has the highest votes is accepted as the final prediction [38].

In weighted voting, the classifiers have different degrees of effect on the common prediction according to their predictive accuracy. A specific weight is determined for each model by its accuracy. All votes are summed and the final prediction is decided by choosing the class with the highest vote.

It gives a 0 vote for class y_1 and a 1 vote for class y_2, for a binary classification problem, where C_i denotes the classifier, and w_i denotes the weight, the aggregate is given by:

$$S(x) = \frac{\Sigma_{i=1}^{K} w_i C_i(x)}{\Sigma_{i=1}^{K} w_i}. \tag{3.4}$$

For instance, the threshold differentiating classes y_1 and y_2 is chosen 0.5. If $S(x) < 0.5$, the weighted voting method classifies instances x as y_1. If $S(x) > 0.5$, it classifies as y_2. It decides randomly, if $S(x) = 0.5$.

For a non-binary classification problem, this method can be used by linking each class j with a different $S_j(x), j \in 1, 2, ...m$, and by mapping the m-class problem into m binary classification problems. Each $S_j(x)$ produces a confidence value showing the feature x being classified as j versus being classified as non-j to classify an instance x. The final class responds to the $S_j(x), j \in 1, 2, ...m$ with the highest confidence value.

3.2.2.2 Boosting

Boosting is a sequential learning method of which evaluation depends on the previous predictor. The first one learns from the entire data set. Then the following learns from training sets according to the performance of the previous one. To have a higher probability of existing in the training set of the next predictor, the incorrectly classified examples are detected and their weights are increased. It causes different machines, which are specialized in predicting different areas of the dataset [14].

Boosting starts with a base classifier. Then, a second classifier is created to focus on the instances in the training data which is misclassified by the first classifier. The process continues to add classifiers until all base classifiers is used or a limit is reached in accuracy [45]. Boosting is provided in Weka in the AdaBoostM1 (adaptive boosting) algorithm.

In this thesis, AdaBoost algorithm is one of the most commonly used boosting techniques for building a strong classifier as a linear combination of base classifiers, is selected.

AdaBoost

"The AdaBoost algorithm proposed by Yoav Freund and Robert Schapire." [27]. Since it makes an accurate prediction, is simple, and has successful applications, it is one of the most important ensemble methods.

Let X denotes the instance space and Y the set of class labels. Assume $Y = -1, +1$. Given a training set $D = \{(x_1, y_1), (x_2, y_2)$,..., $(x_m, y_m)\}$, where $x_i \in X$ and $y_i \in Y$ $(i = 1, ..., m)$, and a base learning algorithm, the AdaBoost algorithm works as follows.

First, AdaBoost constructs an initial distribution of weights D_t over the training data. It gives equal weights to the all training examples $(x_i, y_i)(i \in 1, ..., m)$. It develops a base learner $h_t : X \rightarrow Y$ by calling the base learning algorithm from D_t and the training set.

Algorithm 1 The AdaBoost algorithm

Input : Data set $D = (x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$;

Base learning algorithm L;

Number of learning rounds T.

Process :

$D_1(i) = 1/m$ ▷ Initialize the weight distribution

for $t = 1, ..., T$:

$h_t = L(D, D_t)$; ▷ Train a weak learner h_t from D using distribution D_t

$\epsilon_t = Pr_{i \sim D_i}[h_t(x_i \neq y_i)]$; ▷ Measure the error of h_t

$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$; ▷ Determine the weight of h_t

$$D_{t+1}(i) = \frac{D_t(i)}{Z_i} \times \begin{cases} \exp(-\alpha_i) & \text{if } h_t(x_i) = y_i \\ \exp(\alpha_i) & \text{if } h_t(x_i) \neq y_i \end{cases}$$

$$= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$ ▷ Update the distribution, where Z_t is a normalization factor which enables D_{t+1} be a distribution

end

Output : $H(x) = sign \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$

It uses the training set to test h_t, and increases the weights of the misclassified examples. Consequently, an updated weight distribution D_{t+1} is derived. Adaboost generates another base learner by recalling the base learning algorithm from the training set and D_{t+1}. This process is repeated for T times. The last model is obtained by using T base learners. The weights of the learners are decided in the training process and majority voting of the learners is used in the final model.

In application, the base learning algorithm can be a learning algorithm, which directly uses weighted training set. If not, the weights could be used by sampling the training instances.

3.2.2.3 Stacking

Stacking is a technique of combining multiple classifiers [44]. Different from bagging and boosting, stacking is usually used to merge different classifiers, e.g. decision tree, neural network, rule induction, naïve bayes, logistic regression, etc.

Stacking consists of two levels, which are base learner as level-0 and stacking model learner as level-1. Base learner (level-0) uses different models to learn from a dataset. The outputs of each of the models are used to generate a new dataset. In this new dataset, each instance is relevant to the real value that it is supposed to predict. Then, that dataset is used by stacking model learner (level-1) to provide the final output.

For example, the predicted classifications from the three base classifiers, decision tree, naïve bayes, and rule induction can be used as input variables for the nearest neighbor classifier. In this situation, the nearest neighbor classifier is a stacking model learner, which will learn from the data how to combine the predictions from the separate classifiers to achieve the best classification accuracy.

3.2.3 Diversity in Classifier Ensemblers

The success of the ensemble methods is based on the diversity in the individual classifiers with respect to misclassified instances. Nevertheless, measuring diversity is not a simple task since there is no commonly accepted formal definition. In this study diversity is measured using Q statistic, correlation, double fault, disagreement, and inter-rater agreement [4] [28].

Let $Z = z_1, ..., z_N$ be a labeled data set, $z_j \in R^n$ coming from the classification problem in question. The output of a classifier D_i is represented as an N-dimensional binary vector $y_i = [y_{1,i}, ..., y_{N,i}]^T$, such that $y_{j,i} = 1$, if D_i recognizes correctly z_j, and 0, otherwise, $i = 1, ..., L$.

Table 3.2: A 2x2 table of the relationship between classifiers.

	D_k correct (1)	D_k wrong (0)
D_i correct (1)	N^{11}	N^{10}
D_i wrong (0)	N^{01}	N^{00}

Total, $N = N^{00} + N^{01} + N^{10} + N^{11}$

3.2.3.1 The Q statistics

This measurement is based on Yule's Q statistic. It is used to detect the similarity of two classifiers' outputs:

$$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \tag{3.5}$$

where N^{ab} is the number of elements z_j of Z for which $y_{i,j} = a$ and $y_{i,k} = b$ (Table 3.2). Q varies between –1 and 1. The expected value of Q is 0 for statistically independent classifiers. Classifiers tend to label the same instance correctly will have positive values, and the classifiers which have errors on different instances will have negative values.

3.2.3.2 The correlation coefficient ρ

The correlation between two binary classifier outputs (correct/incorrect), y_i and y_k, is

$$\rho_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11} + N^{10})(N^{01} + N^{00})(N^{11} + N^{01})(N^{10} + N^{00})}}. \tag{3.6}$$

For any two classifiers, Q and ρ have the same sign, and it can be said that $|\rho| \leq |Q|$.

3.2.3.3 The disagreement measure

This measure was used to describe the diversity between two classifiers by Skalak [42]. Then, it is used for measuring diversity in decision forests by Ho [19]. It is the fraction of the number of statements in which one classifier is incorrect and the other is correct over the total number of observations. Its formula is,

$$Dis_{i,k} = \frac{N^{01}N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}. \tag{3.7}$$

3.2.3.4 The double-fault measure

This measure was used to make a pairwise diversity matrix for a classifier pool and then to choose classifiers that are most irrelevant by Giacinto and Roli [12]. It is the ratio of the cases that have been incorrectly classified by both classifiers.

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}}. \tag{3.8}$$

3.2.3.5 Measurement of interrater agreement κ

It is a statistic produced as a measure of inter-rater reliability, κ. Measurement of interrater agreement can be used when separate classifiers identify instances (z_j) to measure the level of agreement while correcting for chance (Fleiss) [11]. It has connection to the intra-class correlation coefficient and the significance test of Looney [33] [30].

Fleiss (1981) defines the pairwise κ_p as,

$$\kappa_p = \frac{2(N^{11}N^{00} - N^{01}N^{10})}{(N^{11} + N^{10})(N^{01} + N^{00}) + (N^{11} + N^{01})(N^{10} + N^{00})}. \tag{3.9}$$

3.3 Model Evaluation

Evaluating the performance of data mining technique is vital for machine learning [24]. Evaluation method helps us to survey the performance and efficiency of any model. It is important to understand the quality of the model, for selecting the most acceptable model or from a given set of models, and for purify parameters in the iterative process of learning. In this section, evaluation measures used in this study are described.

3.3.1 Confusion Matrix

Given a binary classification problem with two classes: positive and negative, the confusion matrix is a contingency table of 2x2. Columns represent to the predicted values by the classification model, and the observed values are represented at the rows [8], as in Table 3.3. The following measures can be obtained from the confusion matrix.

Table 3.3: The confusion matrix

		Predicted Class	
		0	1
Actual Class	0	True Negatives	False Positives
	1	False Negatives	True Positives

True Positive (TP): Number of positive instances correctly predicted.

True Negative (TN): Number of negative instances correctly predicted.

False Positive (FP): Number of negative instances incorrectly predicted as positive.

False Negative (FN): Number of positive instances incorrectly predicted as negative.

Accuracy: Fraction of correctly classified instances over whole instances.

Precision: Fraction of positive instances correctly predicted over the instance which has declared as positive. If the number of false positive is low, the value of precision becomes higher.

Specificity: Fraction of negative instances correctly predicted by the classifier.

Recall or sensitivity: Fraction of positive instances correctly predicted by the classifier. If the number of false negative is low, the value of recall becomes higher.

Table 3.4: The confusion matrix for a binary classification problem is given in Table 3.3

Measure	Formula
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Error rate = 1 - Accuracy	$\frac{FP+FN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Specificity	$\frac{TN}{TN+FP}$
Recall (Sensitivity)	$\frac{TP}{TP+FN}$

3.3.2 ROC Curve

A Receiver Operating Characteristic (ROC) is a two dimensional graph. The ratio of false positives (1- specificity) is plotted on the Y axis and the ratio of true positives (precision) is plotted on the X axis [8]. Points on the ROC curve represent sensitivity/specificity pairs corresponding to a specific decision threshold. The optimal balance point between specificity and sensitivity can be established via this graph. On the other hand, the ROC analysis lets us to understand the predictive ability of a classifier free of any threshold. The area under the ROC curve (called AUC) is a common measurement for comparing the accuracy of various classifiers. It evaluates ability of the method to correctly classify. If AUC of the classifier is closer to 1, accuracy of the classifier is higher. The classifier has the greatest AUC is considered the best.

3.3.3 F-Measure

This measure a harmonic mean of Precision and Recall is null whenever one of the two values is null. The value of F increases proportionally to the increase of precision

and recall. A high value of F-Measure indicates that the model performs better on the positive class [6].

$$F - Measure = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = 2 \cdot \frac{TP}{(2TP + FP + FN)}. \tag{3.10}$$

3.3.4 Kappa Statistic

Cohen's kappa measures the agreement between two raters each of whom classify N items into C mutually exclusive categories [49]. The equation for kappa is:

$$\kappa = \frac{Pr(a) - Pr(e)}{1 - Pr(e)}, \tag{3.11}$$

where $Pr(a)$ is the relative observed agreement among raters. $Pr(e)$ is the probability of chance agreement which is decided calculating the probabilities of each observer random decision.

Table 3.5: Defination of Kappa values

Kappa Statistic	**Strength of Agreement**
<0	Poor
0.00-0.20	Slight
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Substantial
0.81-1	Almost perfect

κ equals to 1, if the raters totally agree. If the agreement is not more than what would be expected by chance (as defined by $Pr(e)$), then κ equals to 0. The interpretation of kappa is given in Table 3.5.

3.3.5 Loss Functions

Let D^T be (X_1, y_1), (X_2, y_2), ... , (X_d, y_d) where X_i is the set of n-dimensional test tuples with d many tuples, and linked real values, y_i , for a response variable, y [32]. It is not easy to say precisely whether the predicted value, $y_i^{'}$, for X_i is accurate, because predictors return a continuous value instead of a label. Therefore, it is looked at how far the predicted value is from the real value, rather than focusing on whether there is an exact match between $y_i^{'}$ and y_i. Loss functions measure the error between the predicted value, $y_i^{'}$, and y_i. The most common loss functions are:

$$\textbf{Absolute error: } |y_i - y_i^{'}|, \tag{3.12}$$

$$\textbf{Squared error: } (y_i - y_i^{'})^2. \tag{3.13}$$

It can be said that error rate is the average loss over the test set. Therefore, it results in the following error rates.

$$\textbf{Mean absolute error: } \frac{\sum_{i=1}^{d} |y_i - y_i^{'}|}{d}, \tag{3.14}$$

$$\textbf{Mean squared error: } \frac{\sum_{i=1}^{d} (y_i - y_i^{'})^2}{d}. \tag{3.15}$$

Unlike the mean absolute, the **mean squared error** exaggerates the existence of outliers. Square root of the mean squared error called the **root mean squared error** has the same magnitude as the quantity predicted.

If just predicted $\bar{y}$, the mean value for y from the training data D exists, the error to be relative to what it would have been may be wanted. In this situation, the total loss can be normalized as follows:

$$\textbf{Relative absolute error: } \frac{\sum_{i=1}^{d}|y_i - y_i^{'}|}{\sum_{i=1}^{d}|y_i - \bar{y}|}, \tag{3.16}$$

$$\textbf{Relative squared error: } \frac{\sum_{i=1}^{d}(y_i - y_i^{'})^2}{\sum_{i=1}^{d}(y_i - \bar{y})^2}, \tag{3.17}$$

where $\bar{y}$ is the mean value of the y_i's of the training data, that is

$$\bar{y} = \frac{\sum_{i=1}^{t} y_i}{d}. \tag{3.18}$$

To find the **root relative squared error**, the root of the relative squared error can be taken. It is useful in that it has the same units as the quantity being estimated.

3.3.6 K-Fold Cross Validation

Cross validation is a method for evaluating how the results of a statistical analysis will generalize to an independent data set. In k-fold cross validation, the initial data is randomly divided into k equal sized folds, $D_1, D_2, ..., D_k$ [17]. In iteration i, fold D_i is used to test and the others are used to train the model. In the first iteration, subsets $D_2, ..., D_k$ are used for training and results are tested on D_1; the second iteration subsets $D_1, D_3, ..., D_k$ are used for training and D_2 for testing; and so on.

Training and testing are performed k times. Different from the holdout and random sub sampling methods, each sample is used once for testing, and the same number of times for training. The accuracy is calculated by dividing number of correct classifications from the k iterations by the total number of tuples in the initial data. Stratified 10-fold cross validation is generally used for calculating accuracy because of its relatively low bias and variance even if it is possible to use more folds.

3.4 Weka

WEKA is a data mining tool, which is developed by the University of Waikato in New Zealand [3]. It is a group of machine learning algorithms, which are written in JAVA language, for data mining tasks. It implements algorithms, which are applied directly to a dataset, for data preprocessing, regression, association rules, clustering, and classification. WEKA also involves visualization tools.

WEKA is an open source software issued under General Public License. It normally uses the data file in ARFF file format, which includes special tags to indicate different things in the data file. In thesis, Weka is used.

CHAPTER 4

DATA AND FEATURE SET

4.1 Data

In this study the log data of Pishti Plus is used. Pişti (pronounced "pishti") is a popular Turkish card game. It is generally played using a standard 52 card pack by four people in partnerships. The direction of play is anticlockwise, and partners sitting opposite. Cards are played to a central pile. If there is a matching between the previous card and the played card, or playing a jack, the central pile can be captured. Points are scored for certain captured cards. "The word 'pişti' describes a capture of a pile containing only one card, for which extra points are scored" [2].

Pishti Plus, which is developed by SNG ICT, is a multi-player social betting game. Pishti Plus different from "pişti" in POT. Each turn, a player puts chips to the pot, aside from regular bet. Whoever breaks the POT wins all the money in it. 3 regular pishties at 2 players table or 2 regular pishties at 4 players table are required to break the POT. Or, whoever makes a pishti with Jack breaks the POT directly.

Figure 4.1 shows the graph of churners according the number of days after they download the game. The statistics of Pishti Plus indicates that almost half of the 30.672 players stop playing in the first day. Hence, churn prediction is applied for the first day churners.

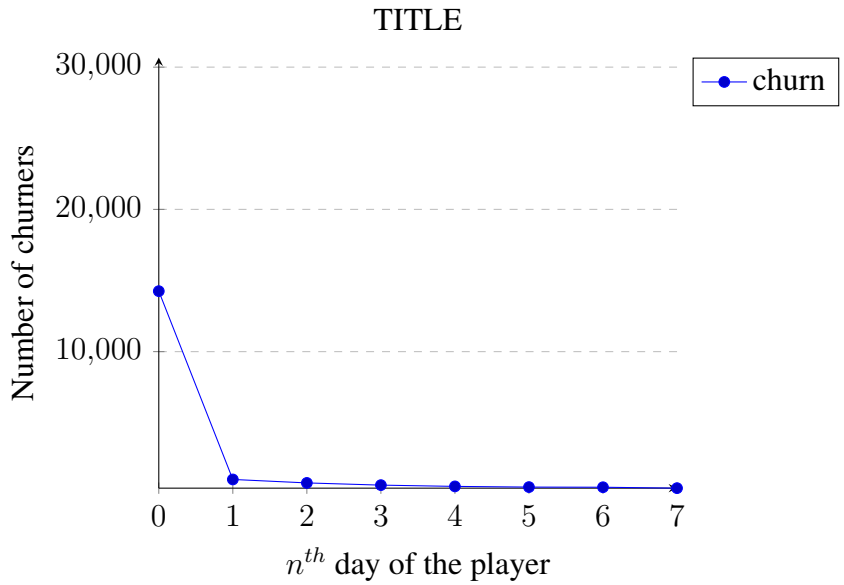

Figure 4.1: Churn graph of Pishti Plus

4.2 Data Preprocessing

Forecasting the future churn, a very strong model should be acquired and this model can only be constructed if we have a vigorous dataset [35]. In this step data elimination and feature extraction are applied. The extracted feature set is given in Appendix A. This preprocessing step took almost 70 percent of total time.

In this study 25.825.864 rows of data belong to 30.672 players were used. This data set contains the data, which is collected between the times "2014-08-09 14:36:17" and "2014-12-08 10:29:30". In the final data set, each row represents the feature vector of the player.

There are 148 players who start the game at the last day, which this data is taken. They start playing after time "2014-08-08 14:36:17". Therefore, their data doesn't consist of feature set of 24 hours. First of all, these players were eliminated. In addition to this, 4 players had inconsistent data were also eliminated.

The remaining data indicates that 5.414 players have never won or lost in the first 24 hours, and 3.639 of them are churners. Since they didn't play the game, their features indicating the total number of plays, maximum amount of chips played in the game

Table 4.1: Player data

Data	**Number of Players**
# of total player	30.672
# of players who haven't completed the game	5.414
# of players who start the game in the last 24 hours	148
# of inconsistent data	4
# of players which is used in this study	25.106
# of non-churners which is used in this study	14.641
# of churners which is used in this study	10.465

etc. are all null. These players were analyzed separately.

Finally, the data belongs to 25.106 players was used in this study. The players play Pishti just for one day were labeled as churners. After this process, 10.465 players were labeled as churners and 14.641 of them as non-churners. Summary of the data set is given in Table 4.1.

CHAPTER 5

THE METHODOLOGY

5.1 Feature Selection

In principle, wrapper method can find the most useful features, but it is prone to over-fitting. In contrast to filter and wrapper approaches, the feature selection part and the learning part cannot be separated in embedded methods. On the other hand, embedded methods and wrapper methods are scheme-dependent attribute subset evaluators, which evaluate the subset using the machine learning algorithm, is used for learning. They are simple and direct, but slow.

Filter methods are scheme-independent attribute subset evaluators. Although filter methods may fail to choose the most useful features, they are relatively robust against over-fitting [48]. Because of these reasons, filter method is used for feature selection in preprocessing step.

Correlation-based Feature Selection (CFS) algorithm is selected as a filter method algorithm. It is nearly as good as wrapper, and much faster. Hence, CFS is used to decide the best subset of features. Since genetic algorithm performs a global search, CfsSubsetEval Algorithm is combined with genetic algorithm.

In this study, 31 attributes are used to determine the best feature set. 10 fold cross-validation is used with CfsSubsetEval Algorithm. Full feature set and the result of feature selection are given in Appendix A. Features appear at least 40% of folds are selected (Table 5.1). Since both avg_half_game and $total\_half\_game$ features are selected, their affects are compared by eliminating one of them. Because of eliminat-

ing avg_half_game increased the accuracy, only $total\_half\_game$ is used.

Table 5.1: Feature set

General Features	
fbid	Id of player
total_session	Total number of sessions in the first day of player
total_play	Total number of
total_half_game	Total number of games which are unfinished
session_with_no_play	Total number of session with no play
sequential_win	Total number of games which are won sequentially in twice
Game Specific Features	
total_chip	Total amount of chip which player has
total_pishti	Total number of pishti
total_vpishti	Total number of pishti with jack of player
avg_pot	Arithmetic average of number of pots per game
total_level	Level of user
sequential_pishti	Total number of pishti which is done sequentially in twice
Game Specific Character Based Features	
max_chip_played	Maximum amount of chip which player played
total_create	Total number of table create of player
total_sit	Total number of table sit of player
total_join	Total number of join to games
total_play_now	Total number of click play now event
want_buy	Total number of actions to buy
Class Feature	
is_churn	Is player active after the first day or not

The selected features are grouped under three categories: general, game specific, and game specific character based. General features are the features, which exist in every game. In addition to total play, total session numbers, and total number of games, which are sequentially won, total half game number is included. Total half game means a player start the game but does not have any result like win or lose. It can be because of technical problems or just because of the player does not want to continue. Session with no play indicates that player start the game to do something like buying some chips, inviting friends etc. but not for playing the game.

In this study, interestingly, fbid of players are selected as a feature. The reason is that since April the 30^{th} 2014, Facebook uses version 2.0 and the user ID became app-scoped since then. App-scoped means that the ID for the same user will be different between applications [1]. Players who use version 2.0 or higher have 10 digits id and the others have 6 digits.

Game specific features are total amount of chips the player has, total number of pishti and pishti with jack, average number of pot per game, level of the user, and total number of pishties, which are done consecutively. Although these features are specific to Pishti Plus, level of the user can be used in every level based games.

Game specific character based features actually give some characteristic information about player. If maximum number of chip played is higher, it is possible to say that the player likes to take some risk and is self-confident. High number of total create means that the player is dominant and high number of total sit means that the player is not dominant. Total join may be interpreted as the player looking for someone to play. Total play now indicates that player wants play again. In addition to these, want buy is calculated by the events in game as: click on buy chips button, vip membership button, offer button or opened store.

Finally, is_churn indicates whether the player will load the Pishti Plus again or not. The players who load the game in the following days are non-churners and the others are churners. If is_churn is 0 the player is a non-churner, otherwise (s)he is a churner.

Take into account that the importance of the features can change according to the selected algorithm. Because of this reason, after the best classification algorithm is selected, feature selection is applied a second time to find the most important features. In the second feature selection, the whole attributes are used and wrapper method is applied.

After the preprocessing step, data divided into two groups: players who plays the game at least once, and the players who have never completed the game but have some other activities. The same methods are applied both of the two groups with different feature sets.

5.2 Classification and Prediction

Churn prediction is a binary classification work, which distinguishes churners and non-churners. There are many prediction algorithms to forecast the customer churn. In this study, Logistic Regression, SMO, Bayesian Network and Simple CART clas-

sification algorithms are used and the results of the algorithms are compared. Then, ensemble methods (boosting, bagging, and stacking) are applied to improve the accuracy by combining different types of classifiers.

To understand the quality of the model for selecting the most acceptable one some evaluation measures are used. Overall accuracy, sensitivity, and specificity measures are used to compare the success of the algorithms. In addition to this, AUC values are compared. F-measure is used to find which algorithm is better on the positive class. Kappa statistics are used to understand how much better the results of the algorithms are than the results expected by chance. On the other hand, loss functions are used to see the error rates of the algorithms.

5.3 Ensemble Methods

Dietterich [10] reported that there are three major reasons why an ensemble classifier is usually significantly better than a single classifier. First, the training data does not always give enough information for selecting a single accurate hypothesis. Second, the learning progress of the weak classifier may not be perfect. And third, the hypothesis space being searched may not contain the true target function while an ensemble classifier can give a good approximation. Because of these reasons, ensemble methods are used to combine multiple classifiers to improve robustness and achieve an improved classification performance from any of the constituent classifiers. Boosting, bagging, and stacking are applied and the results are compared.

First, AdaBoost algorithm one of the most extensively used boosting methods for building a strong classifier as a linear combination of base classifiers is applied to each of the four classifiers. Secondly, stacking is used to combine different techniques of classifiers. The most successful algorithm is used as meta classifier and combined with the other algorithms. Thirdly, voting is applied as bagging method. The most successful two algorithms are merged using some combination rules: majority voting, maximum probability, minimum probability, sum of probabilities, and product of probabilities. Finally, the ensemble method which outputs the highest overall accuracy is selected.

5.4 Diversity of Classifiers

Disagreement measure, Q statistic, correlation coefficient, double-fault measure, and measurement of interrater agreement κ are calculated for each of the four classifiers. Diversity of classification algorithms are compared and the relationship between diversity and accuracy is examined to find the most suitable diversity measurement for this study.

In this study, the first three algorithm pairs, which have the highest diversity, in each diversity measures are selected and the common pairs are detected. Then, the ensemble results of the pairs are investigated to see the affects this approach.

CHAPTER 6

EVALUATION AND EXPERIMENTAL RESULTS

In this section, prediction performances of Logistic Regression, SMO, Bayesian Network and Simple CART algorithms used very common in churn prediction are compared using the feature set, which is produced by filter method. Accuracy of these algorithms are presented in Table 6.1.

Table 6.1: Accuracy of the algorithms

Algorithm	Sensitivity (%)	Specificity (%)	Overall accuracy (%)
Bayesian Network	76.1	57.5	65.24
Logistic Regression	69.9	64.7	66.87
Simple CART	61.5	**71.9**	**67.56**
SMO	**79.2**	54.3	64.69

Table 6.1 indicates that Simple CART is the most successful algorithm to predict the churners and has the highest overall accuracy. It has 71.9% specificity and correctly classified the 67.56% of the players. On the other hand, SMO has the highest sensitivity, which means that it predicts the churners with 79.2% of accuracy.

In Weka the kappa statistics defines the agreement of prediction with the true class. Kappa statistics is about 0.32 for each algorithm that means there is a fair agreement, which is more than by chance (Table 3.5). F-Measure indicates that Simple CART algorithm is more successful to predict non-churners. On the other hand, root relative squared error, root mean squared error, and ROC area show that accuracy of SMO is less than the accuracy of the other algorithms. Since the output of SMO is 1 or 0, it is more prone to error (Table 6.2).

Table 6.2: Results of the algorithms

	Bayesian Network	Logistic Regression	SMO	CART
Kappa statistic	0.3196	**0.3367**	0.3158	0.3335
Mean absolute error	**0.3517**	0.4164	0.3531	0.4107
Root mean squared error	0.5488	0.4548	0.5942	**0.4547**
Relative absolute error (%)	**72.3389**	85.65	72.6299	84.477
Root relative squared error (%)	111.3076	92.2452	120.5239	**92.2318**
Weighted Avg. of F-Measure	0.653	0.671	0.646	**0.676**
ROC Area	0.725	**0.727**	0.668	0.721

Because of Simple CART has the highest overall accuracy, feature selection is applied a second time to find the most important features (Table 6.3). In the second feature selection the whole features are used and wrapper subset evaluator algorithm is applied with the best first search to data.

Table 6.3: The results of Simple CART algorithm

Sensitivity (%)	74.4
Specificity (%)	58.8
Overall accuracy (%)	67.90
Kappa statistic	0.3347
Mean absolute error	0.413
Root mean squared error	0.4558
Relative absolute error (%)	84.955
Root relative squared error (%)	92.4404
Weighted Avg. of F-Measure	0.678
ROC Area	0.719

Important features are selected using 10 fold cross validation. The values of importance indicate the percentage of number of times the features selected. Features appear in the final feature set more than 50% are selected. Selected features and their importance are given in Figure 6.1.

The results indicate that general features: total session and sequential win are important for churn prediction in Pishti Plus. In addition, character based features: max chip played, total create and total join are also important features. Total level is selected from game specific features category. Total level also exists in the games which have levels. Therefore, this method can be applied to other games. On the other hand, this study support the Jaya Kawale's hypothesis: churners have less number of ses-

sions compared to the non-churners. Finally, we can conclude that the features related to player character can be more informative for churn prediction.

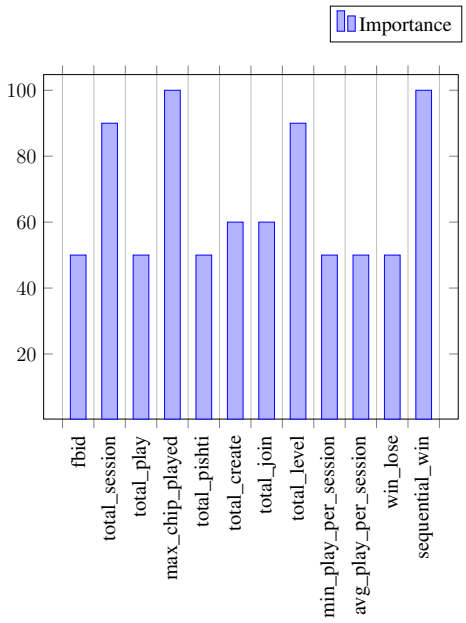

Figure 6.1: Feature-Importance graph

In order to improve the accuracy, ensemble methods are applied. First, AdaBoost algorithm is applied to each of the four algorithms. Boosting increased the accuracy of Bayesian algorithm, but affected negatively the other algorithms substantially (Table 6.4).

Table 6.4: Results of boosting

Algorithm	Sensitivity (%)	Specificity (%)	Overall accuracy (%)
Bayesian Network	69.1	65.7	**67.15**
Logistic Regression	70.5	64.1	66.81
Simple CART	52.3	**68.4**	61.71
SMO	**79.4**	54.1	64.66

Then, stacking is applied. Since it is the most successful algorithm, Simple CART algorithm is used as the meta classifier and is combined with the other algorithms. Stacking Simple CART algorithm with Bayesian Network and Logistic Regression increased the accuracy compared to using just Bayesian Network or Logistic Regres-

sion. Stacking Simple CART with SMO decreased the accuracy (Table 6.5).

Table 6.5: Results of stacking

Algorithm	Sensitivity (%)	Specificity (%)	Overall accuracy (%)
Bayesian Network	**68.9**	65.3	66.81
Logistic Regression	61.0	**71.9**	**67.36**
SMO	66.9	67.3	61.11

Finally, voting is applied as the bagging method. Because Logistic Regression and Simple CART algorithms are the best algorithms, average/product of/minimum/maximum probability rules are applied to these two algorithms. They all gave the same result, which is given in Table 6.6. Bagging method increased the accuracy insignificantly compared to using just Simple CART or Logistic Regression alone.

Table 6.6: Results of voting

Sensitivity (%)	Specificity (%)	Overall accuracy (%)
62.7	71.6	**67.88**

The reason why we couldn't improve the accuracy is investigated and diversity of algorithms are examined. It can be seen that Simple CART and Logistic Regression, which is the only pair accuracy is incremented, is one of the first three algorithm pairs, which has higher diversity. On the other hand, it can be said that the diversity of these algorithms are very low, and because of this ensemble could not increase the accuracy very much (Table 6.7).

Table 6.7: Diversity of algorithms

Algorithms	q statistics	disagreement	double fault	kappa statistics	correlation coefficient
Bayesian-CART	**0.91**	**0.17**	**0.24**	**0.62**	**0.0069641**
Bayesian-Logistic	0.96	**0.12**	0.28	0.74	0.0081170
Bayesian-SMO	0.98	0.09	0.31	0.81	0.0085801
CART-Logistic	**0.95**	**0.12**	**0.26**	**0.72**	**0.0078544**
CART-SMO	**0.88**	**0.19**	**0.24**	**0.57**	**0.0060164**
Logistic-SMO	0.97	0.11	0.29	0.77	0.0081559

The methods applied for the first group of players are also applied to 5.414 players,

who have never won or lost any game in the first 24 hours, using different features. CfsSubsetEval algorithm is applied with genetic algorithm to select features. 10 fold cross-validation is used for CfsSubsetEval Algorithm. Full feature set and the results of feature selection are given in Appendix B. Features appear at least 70% of folds are selected (Table 6.8).

Table 6.8: Feature set

General Features	
app_loaded	number of application load
Game Specific Features	
vpishti	number of pishti with jack
Game Specific Character Based Features	
sit_to_table	number of sit table
buy_chips_button	number of click buy chips button
join_room	number of join room
send_chips	number of send chips
Class Feature	
is_churn	Is player active after the first day or not

The prediction performances of Logistic Regression, SMO, Bayesian Network and Simple CART algorithms are compared using the feature set, which is produced by filter method. Accuracy of these algorithms are given in Table 6.9. Results indicate that Simple CART Algorithm and Bayesian Network are the most successful algorithms to predict the churners and have the highest overall accuracy. By looking at Table 6.9 and Table 6.10 it can be seen that there is no important improvement in churn prediction for the 5.414 players.

Table 6.9: Accuracy of the algorithms

Algorithm	Sensitivity (%)	Specificity (%)	Overall accuracy (%)
Bayesian Network	92.7	20.9	**69.19**
Logistic Regression	98.4	4.5	67.60
Simple CART	92.7	21.7	**69.45**
SMO	100.0	0.0	67.22

To improve the accuracy ensemble methods are used (Table 6.11). First AdaBoost algorithm is applied to Simple CART and Bayesian Network algorithms. Then bagging and stacking methods are applied. Again bagging is the most successful method to

Table 6.10: Results of the algorithms

	Bayesian Network	Logistic Regression	SMO	CART
Kappa statistic	0.1633	0.0378	0	0.1726
Mean absolute error	0.4212	0.4285	0.3279	0.4159
Root mean squared error	0.4596	0.4627	0.5726	0.4584
Relative absolute error (%)	95.5628	97.2146	74.3844	94.3563
Root relative squared error (%)	97.8962	98.5616	121.9742	97.655
Weighted Avg. of F-Measure	0.64	0.567	0.54	0.644
ROC Area	0.604	0.608	0.5	0.592

improve churn prediction in this dataset.

Table 6.11: Results of ensemble methods

Algorithm	Method	Sensitivity (%)	Specificity (%)	Overall accuracy (%)
Bayesian Network	Boosting	92.7	20.9	69.19
Simple CART	Boosting	91.3	24.0	69.23
Bayesian & CART	Bagging	94.0	19.6	**69.63**
Bayesian & CART	Stacking	92.9	21.1	69.36

The method used in this thesis for churn prediction divides data into two groups of players depending on whether they complete the game at least once or don't. If the user complete the game in the first day at least once we used the average weighting in voting as bagging method and used Simple CART algorithm with Logistic Regression. Otherwise, we applied the average weighting in voting as bagging method and used Simple CART algorithm with Bayesian Network. Final results are given in Table 6.12 and Figure 6.2.

Table 6.12: Final Results

Sensitivity (%)	Specificity (%)	Overall accuracy (%)
65.98	70.77	68.20

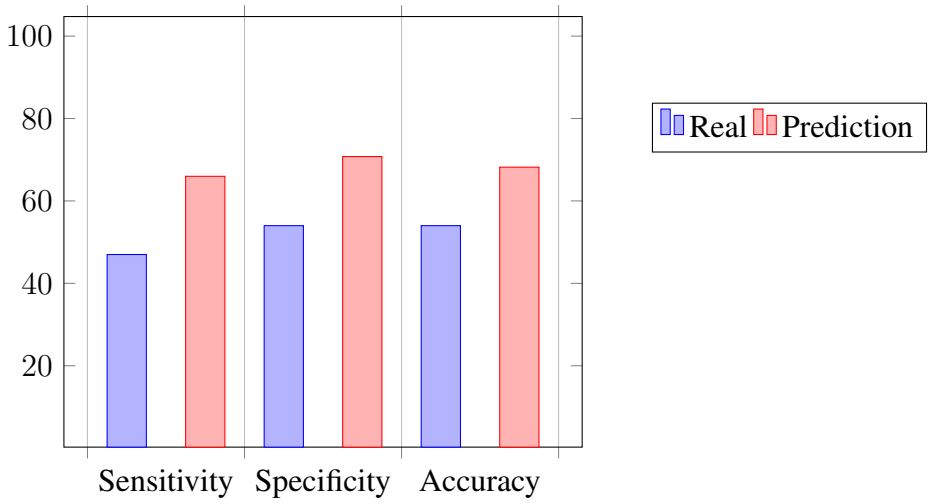

Figure 6.2: Success of prediction

CHAPTER 7

DISCUSSION AND CONCLUSION

In this thesis, CRM approaches were used to predict churners for internet games. Multiplayer social game Pishti Plus was chosen as the case study. Churn prediction is done using the first day actions of players. Weka is used as a data mining tool. In preprocessing, players, who haven't complete the first 24 hours or have inconsistent data, are eliminated.

The players are divided into two groups: players who completed the game at least once, and players who have started to play but haven't completed or never played but had some activities such as sending chips inviting friends. This two groups are analyzed separately. The players who load Pishti just for one day are labeled as churner and others labeled as non-churner.

For each of the two groups, CFS algorithm is used as a feature selection algorithm since it is scheme-independent, and is relatively robust against over-fitting. In addition, it performs nearly as good as wrapper, and much faster. Then, Bayesian Network, Logistic Regression, SMO and Simple CART algorithms are used for churn prediction and results are compared. Finally, ensemble methods: boosting, bagging and stacking are applied to improve the accuracy.

This study indicates that bagging is better than other ensemble methods. For the first group, bagging Simple CART algorithm with Logistic Regression gives 67.88% accuracy. For the second group, bagging Simple CART algorithm with Bayesian Network gives 69.63% accuracy. In conclusion, whether Pishti Plus players will churn or not can be predicted with 68.20 % accuracy using the first day data of players.

The results indicate that total session and sequential win are the most important features for churn prediction and they exist in every kind of games. On the other hand, the other important feature total level exists in level based games. Because of this, the same study can be applied to other games and common features can be found for churn prediction in game industry as a feature work. In addition, including friendship table may also increase the accuracy, since previous works indicate that churners affect negatively existing players.

The results also show that features, which may include some information about the character of the player can be more informative for churn prediction. In the future, a different feature set has personalized information such as age, gender, and job can be added to database and the effects of these features can be analyzed.

In this study, interestingly, fbid (Facebook ID) is selected as an important feature for churn prediction. When the reason is investigated it is founded that Facebook has changed the id format of users. Players have the newest version of Facebook are more prone to churn. These players who have the newest version can be newer players or the players who follow the innovations. As a feature work this can be investigated.

REFERENCES

[1] Facebook - Upgrade Guide of Facebook. `https://developers.facebook.com/docs/apps/upgrading#appscope`. last visited on August 2008.

[2] SNG ICT - Pishti Plus. `https://www.sngict.com/games/pishtiplus/pishtiplus.html`. last visited on August 2008.

[3] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. `http://www.cs.waikato.ac.nz/ml/weka/`. last visited on August 2008.

[4] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Ensemble diversity measures and their application to thinning. *Information Fusion*, 6(1):49–62, 2005.

[5] C. Bauckhage, K. Kersting, R. Sifa, C. Thurau, A. Drachen, and A. Canossa. How players lose interest in playing a game: An empirical study based on distributions of total playing times. *2012 IEEE Conference on Computational Intelligence and Games, CIG 2012*, pages 139–146, 2012.

[6] M. Bekkar, H. K. Djemaa, and T. A. Alitouche. Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, 3(10):27–38, 2013.

[7] Z. H. Borbora and J. Srivastava. User Behavior Modelling Approach for Churn Prediction in Online Games. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 51–60. IEEE, Sept. 2012.

[8] M. Clemente, V. Giner-Bosch, and S. San Matías. Assessing classification methods for churn prediction by composite indicators. *Manuscript, Dept. of Applied Statistics, OR & Quality, Universitat Politècnica de València, Camino de Vera s/n*, 46022:1–31, 2010.

[9] T. Debeuvais, D. J. Shapiro, B. Nardi, N. Ducheneaut, and N. Yee. If You Build It They Might Stay: Retention Mechanisms in World of Warcraft. *6th International Conference on the Foundations of Digital Games (FDG 2011)*, pages 1–8, 2011.

[10] T. G. Dietterich. Machine-learning research: Four current directions. *AI magazine*, 18(4):97–136, 1997.

[11] J. L. Fleiss. *Statistical Methods for Rates and Proportions*. Wiley Series in Probability and Statistics. Wiley, 1981.

[12] G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9-10):699–707, 2001.

[13] A. Gowda Karegowda, M. Jayaram, and a.S .Manjunath. Feature Subset Selection using Cascaded GA and CFS: A Filter Approach in Supervised Learning. *International Journal of Computer Applications*, 23(2):1–10, 2011.

[14] M. Graczyk, T. Lasota, B. Trawiński, and K. Trawiński. Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5991 LNAI(PART 2):340–350, 2010.

[15] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting, and C. Bauckhage. Predicting Player Churn in the Wild. *IEEE Computational Intelligence in Games*, pages 131–139, 2014.

[16] M. a. Hall. Correlation-based Feature Selection for Machine Learning. *Methodology*, 21i195-i20(April):1–5, 1999.

[17] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. 3 edition, 2012.

[18] D. Heckerman. A Tutorial on Learning With Bayesian Networks. *Innovations in Bayesian Networks*, 1995(November):33–82, 1996.

[19] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[20] H. Jiawei and M. Kamber. Data mining: concepts and techniques. *San Francisco, CA, itd: Morgan Kaufmann*, pages 377–385, 2001.

[21] N. Kamalraj and a. Malathi. A Survey on Churn Prediction Techniques in Communication Sector. *International Journal of Computer Applications*, 64(5):39–42, 2013.

[22] A. G. Karegowda, a. S. Manjunath, and M. a. Jayaram. Comparative study of attribute selection using gain ration and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, 2(2):271–277, 2010.

[23] J. Kawale, A. Pal, and J. Srivastava. Churn prediction in MMORPGs: A social influence based approach. *Proceedings - 12th IEEE International Conference on Computational Science and Engineering, CSE 2009*, 4:423–428, 2009.

[24] B. Kiranmai and A. Damodaram. A Review on Evaluation Measures for Data Mining Tasks. 3(7):7217–7220, 2014.

[25] C. Kirui, L. Hong, W. Cheruiyot, H. Kirui, C. Engineering, and I. Technology. Predicting Customer Churn in Mobile Telephony Industry Using Probabilistic Classifiers in Data Mining. *IJCSI International Journal of Computer Science Issues*, 10(2):165–172, 2013.

[26] R. Kohavi and R. Kohavi. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[27] R. Kumar and D. Verma. Classification Algorithms for Data Mining: A Survey. *International Journal of Innovations in Engineering*, 1(2):7–14, 2012.

[28] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[29] T. N. Lal, O. Chapelle, J. Weston, and a. Elisseeff. Embedded Methods. 0, 2004.

[30] V. Lazarov and M. Capota. Churn Prediction. *TUM computer science*, 2007.

[31] K. C. Lee and H. Cho. Performance of Ensemble Classifier for Location Prediction Task: Emphasis on Markov Blanket Perspective. *International Journal of U- & E-Service, Science & Technology*, 3(3):1–12, 2010.

[32] K. M. Leung. Estimating and Reducing the Error of a Classifier or Predictor, 2007.

[33] S. W. Looney. A Statistical Technique for Comparing the Accuracies of Several Classifiers. *Pattern Recogn. Lett.*, 8(1):5–9, 1988.

[34] G. Naqvi. International Master ' s Thesis A Hybrid Filter-Wrapper Approach for Feature Selection Ghayur Naqvi. 2012.

[35] V. P. Narkhede. Data Mining Tools for Predicting Churn Behaviour of Bank Customers. 1632:56–62.

[36] J. C. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Advances in Kernel MethodsSupport Vector Learning*, 208:1–21, 1998.

[37] R. Polikar. Robi Polikar ©. *Ieee Circuits And Systems Magazine*, pages 21–45, 2006.

[38] A. L. Prodromidis and S. J. Stolfo. A Comparative Evaluation of Meta-Learning Strategies over Large and Distributed Data Sets. *16 Intl. Conf. Machine Learning, Workshop on Meta-learning*, pages 18–27, 1999.

[39] M. Ramaswami and R. Bhaskaran. A Study on Feature Selection Techniques in Educational Data Mining. *Journal of Computing*, 1(1):7–11, 2009.

[40] J. Runge and P. Gao. Churn Prediction for High-Value Players in Casual Social Games. 2014.

[41] Y. Saeys, I. n. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics, 2007.

[42] D. Skalak. The sources of increased accuracy for two proposed boosting algorithms. *Proc. American Association for Artificial Intelligence, . . .* , pages 120—-125, 1996.

[43] T. Stephenson. An introduction to Bayesian network theory and usage. *Idiap-Rr 00-03*, 2000.

[44] I. Syarif, E. Zaluska, A. Prugel-Bennett, and G. Wills. Application of bagging, boosting and stacking to intrusion detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7376 LNAI:593–602, 2012.

[45] A. Tiwari and A. Prakash. Improving classification of J48 algorithm using bagging , boosting and blending ensemble methods on SONAR dataset using WEKA. (9):207–209, 2014.

[46] K. Tsiptsis and A. Chorianopoulos. *Data Mining Techniques in CRM*. 2010.

[47] A. Tsymbal, M. Pechenizkiy, and P. Cunningham. Diversity in search strategies for ensemble feature selection. *Information Fusion*, 6(1):83–98, 2005.

[48] M. Usman. *Improving Knowledge Discovery through the Integration of Data Mining Techniques*. IGI Global, 2015.

[49] A. J. Viera and J. M. Garrett. Understanding interobserver agreement: The kappa statistic. *Family Medicine*, 37(5):360–363, 2005.

[50] I. H. Witten, E. Frank, and M. a. Hall. *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*. 2011.

APPENDIX A

FEATURE SET 1

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval
Search:weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1
Relation:     pishti
Instances:    25106
Attributes:   32
              fbid
              total_session
              total_play
              total_half_game
              avg_half_game
              total_chip
              max_chip_played
              total_pishti
              avg_pishti
              total_vpishti
              total_pot
              avg_pot
              total_create
              total_sit
              total_join
              total_level
              total_play_now
              friendly
              buy_offer
              want_buy
              max_play_per_session
              min_play_per_session
```

```
                session_with_no_play
                avg_play_per_session
                first_event
                last_event
                win_lose
                pos_neg
                create_sit
                sequential_win
                sequential_pishti
                is_churn
Evaluation mode:10-fold cross-validation

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

number of folds (%)  attribute
           4( 40 %)     1 fbid
          10(100 %)     2 total_session
           9( 90 %)     3 total_play
           4( 40 %)     4 total_half_game
           4( 40 %)     5 avg_half_game
           8( 80 %)     6 total_chip
           9( 90 %)     7 max_chip_played
           4( 40 %)     8 total_pishti
           0(  0 %)     9 avg_pishti
           8( 80 %)    10 total_vpishti
           3( 30 %)    11 total_pot
           9( 90 %)    12 avg_pot
          10(100 %)    13 total_create
           6( 60 %)    14 total_sit
          10(100 %)    15 total_join
           5( 50 %)    16 total_level
           7( 70 %)    17 total_play_now
           1( 10 %)    18 friendly
           0(  0 %)    19 buy_offer
           5( 50 %)    20 want_buy
           0(  0 %)    21 max_play_per_session
           0(  0 %)    22 min_play_per_session
           5( 50 %)    23 session_with_no_play
           0(  0 %)    24 avg_play_per_session
           0(  0 %)    25 first_event
```

```
0(  0 %)    26 last_event
3( 30 %)    27 win_lose
0(  0 %)    28 pos_neg
0(  0 %)    29 create_sit
9( 90 %)    30 sequential_win
8( 80 %)    31 sequential_pishti
```

APPENDIX B

FEATURE SET 2

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval
Search:weka.attributeSelection.GeneticSearch -Z 20 -G 20 -C 0.6 -M 0.033 -R 20 -S 1
Relation:     pishti
Instances:    5414
Attributes:   17
              fbid
              app_loaded
              create_table
              sit_to_table
              pishti
              vpishti
              pot
              buy_chips_button
              buy_vip_membership_button
              playnow_clicked
              round_started
              join_room
              store_opened
              send_chips
              invite_friend
              buy_offer_button
              is_churn
Evaluation mode:10-fold cross-validation

=== Attribute selection 10 fold cross-validation (stratified), seed: 1 ===

number of folds (%)  attribute
```

```
 0(  0 %)    1 fbid
10(100 %)    2 app_loaded
 0(  0 %)    3 create_table
 7( 70 %)    4 sit_to_table
 0(  0 %)    5 pishti
 8( 80 %)    6 vpishti
 2( 20 %)    7 pot
 7( 70 %)    8 buy_chips_button
 0(  0 %)    9 buy_vip_membership_button
 0(  0 %)   10 playnow_clicked
 0(  0 %)   11 round_started
 9( 90 %)   12 join_room
 2( 20 %)   13 store_opened
 9( 90 %)   14 send_chips
 0(  0 %)   15 invite_friend
 2( 20 %)   16 buy_offer_button
```