RECOMMENDER SYSTEM FOR IMPROVING CHURN RATE

by

Yuehua Duan

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2022

Approved by:

Dr. Zbigniew W. Ras

Dr. Aidong Lu

Dr. Angelina Tzacheva

Dr. Moutaz Khouja

ABSTRACT

YUEHUA DUAN. Recommender system for improving churn rate. (Under the direction of DR. ZBIGNIEW W. RAS)

Customer churn leads to higher customer acquisition cost, lower volume of service consumption and reduced product purchase. Reducing the outflow of the customers by 5% can double the profit of a typical company. Therefore, it is of significant value for companies to reduce customer outflow. In this dissertation research, we mainly focus on identifying the customers with high chance of attrition and providing valid and trustworthy recommendations to reduce customer churn.

We designed and developed a customer attrition management system that can predict customer churn and yield actionable and measurable recommendations for the decision makers to reduce customer churn. Moreover, reviews from leaving customers reflect their unfulfilled needs, while reviews of active customers show their satisfactory experience. In order to extract the action knowledge from the unstructured customer review data, we thoroughly applied aspect-based sentiment analysis to transform the unstructured review text data into a structured table. Then, we utilized rough set theory, action rule mining and meta-action triggering mechanism on the structured table to generate effective recommendations for reducing customer churn. Lastly, in practical applications, an action rule is regarded as interesting only if its support and confidence exceed the predefined threshold values. If an action rule has a large support and high confidence, it indicates that this action can be applied to a large portion of customers with a high chance. However, there is little research focused on improving the confidence and coverage of action rules. Therefore, we proposed a guided semantic-aided agglomerative clustering algorithm to improve the discovered action rules.

ACKNOWLEDGEMENTS

There are many who helped me along the way on this journey. I would not get this far without their help.

First and foremost, I would like to express my deep gratitude to my advisor, Prof. Zbigniew W. Ras for his constructive and persistent guidance throughout my research development. He is so inspiring to help me by sharpening my thinking and bringing my research work to a higher level. I admire his great expertise and personality, and I will continue learning from him.

I would like to thank my Dissertation Committee members, Prof. Aidong Lu, Prof. Angelina A. Tzacheva and Prof. Moutaz Khouja, for their valuable feedback and guidance.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Customer churn refers to the loss of existing customers who cease the business relationship with a company in a period of time [1]. Recent studies [2, 3, 4] indicate that reducing customer churn can be significantly beneficial. According to [5, 6], reducing customer churn by 5% can lead to double profit for a typical company. Mediating and preventing customer churn become increasingly important to companies [7]. In order to minimize customer churn and maintain a long-term customer relationship, it is critical for companies to detect customer churn proactively and take effective strategies to reduce it.

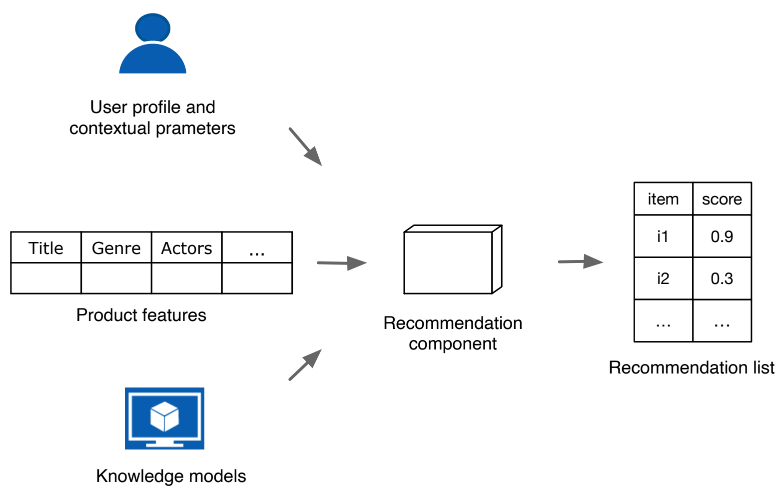
Churn detection aims to recognize early churn signals and distinguish the likely leaving customers from the others who will not leave. A Churn predicting model, which makes inferences for future situations by extracting patterns from the available data, is widely used in recognizing the early signals of customer churn [8]. Garcia et al. [9] classified churn prediction models into three groups: churn prediction models that are built based on statistics; churn prediction models that are utilizing machine learning methods; churn prediction models that are applying other methods.

In the group of statistics-based churn prediction models, Logistic Regression is well suited for describing and testing hypotheses about relationships between a categorical outcome variable and one or more categorical (or continuous) predictor variables [10]. It is a popular multivariate statistics method in churn prediction, owning to its conceptual simplicity as well as its quick and robust results. Decision Tree is an important knowledge extraction tool in data mining that is widely used in churn prediction. It is a tree-shaped structure representing sets of decisions capable of generating classification rules for a specific dataset [11]. Popular choices of Decision Tree in churn data analysis include C 5.0 classification tree, classification and regression tree (CART), and Chisquared Automatic Interaction Detector (CHAID).

Machine learning methods, such as Artificial Neural Networks (ANN), Support Vector Machine (SVM), EXtreme Gradient Boosting (XGBoost), are widely used in churn prediction. ANN is a popular approach to address the churn prediction problem. Haykin [12] described ANN as a massively parallel combination of simple processing unit which can gain knowledge from environment through a learning process and store the knowledge in its connections. SVM is able to optimally separate two classes of objects (e.g. leaving customers and retained customers) through the generation of a multivariate maximally separating hyperplane [9]. It is a popular tool in predicting churn because of its lower number of controlling parameters and better generalization capability [13]. XGBoost is an efficient and scalable implementation of gradient boosting framework, which includes efficient linear model solver and tree learning algorithm. It supports various objective functions, including regression, classification and ranking [14].

Customer Relationship Management (CRM), Decision Support Systems (DSS) and Recommender Systems are widely used to assist the decision-making process in retaining existing customers and reducing customer churn [15]. CRM is a combination of people, processes and technology that seeks to understand a company's customers [16]. As an integrated approach, CRM focuses on customer retention and relationship development, and it helps to lower the customer acquisition cost and raise business revenue [17]. DSS is defined by Sprague et al. [18] as a class of information systems that can support the decision-making activities by utilizing the transaction processing systems and interacting with other parts of the overall information systems. By utilizing computer-based resources, such as data, documents and models, managers are provided with analytical capabilities in the decision-making process. Recommender systems are defined as a subclass of information filtering systems that seek to predict the rating or preference a user would give to an item [19]. Recommender systems are categorized into collaborative filtering recommender systems, content-based recommender systems, demographic recommender systems, knowledge-based recommender systems, community-based recommender systems, and hybrid recommender systems [20].

Figure 1.1: Knowledge-based recommender system



Knowledge-based recommender systems show advantage in such scenarios: when there is not enough available ratings, or when time span plays an important role, or when the customers define their requirements explicitly. As shown in Figure 1.1, the input of a typical knowledge-based recommender system includes user profile and contextual parameters, product features, and knowledge models. This input will be processed by the recommendation component. At the endpoint, a recommendation list as output will be given [19]. Knowledge-based recommender systems can be classified into constraint-based and case-based. In constraint-based recommender systems, the explicitly predefined recommendation rules should be fulfilled. If there is no solution matching the recommendation rules, the constraints will be relaxed until a corresponding solution is found. In case-based recommender systems, different types of similarity measures are given, and items similar to the specified requirements are retrieved as recommendations.

It is imperative for companies to recognize the early churn signals and build recommender systems that can provide effective recommendations for reducing customer churn. In this dissertation, we focus on building recommender systems that can effectively improve customer churn rate. It mainly contains three parts.

- For the customer churn problem in the heavy equipment repair and service sector, we design and develop a Customer Attrition Management System that can identify customers with a high chance of attrition and yield actionable and measurable recommendations for the decision makers to reduce customer churn. The Customer Attrition Management System comprises two stages.
  - Stage I is to identify potential leaving customers by utilizing the early churn signal.
  - Stage II is to build a recommender system that can provide effective and measurable strategies for decision makers to reduce customer churn.
- Reviews from leaving customers reflect their unfulfilled needs, while reviews of active customers reflect their satisfactory experience. We use aspect-based sentiment analysis to transform the unstructured review text data into a structured table, then extract action rules from the structured table to generate recommendations for reducing customer churn.
- Action rules with high quality can affect many more customers. However, there is little research focused on improving the quality of the action rules. We propose a guided semantic-aided agglomerative clustering algorithm to improve the quality of discovered action rules.

Next, we present the structure of this dissertation. In Section 2, we describe the related work on customer churn study. In Section 3, we briefly recall the important background knowledge for better understanding the methodology proposed in this dissertation. In Section 4, we give the description of the dataset used in this research. In Section 5, we present the design and implementation of the Customer Attrition Management System. In Section 6, we present the design and implementation of the recommender system, which is developed by utilizing the unstructured review data. In Section 7, we give the design and implementation of the semantic-aided agglomerative clustering algorithm. In Section 8 we give the conclusion of this dissertation.

CHAPTER 2: RELATED WORK

When it comes to customer churn, related research has three main directions. First, research focuses on pursuing higher precision in churn prediction. Second, research concentrates on churn determinants analysis. Third, study on the cost and effect of different strategies in the decision-making process for improving customer churn rate.

There are many different kinds of churn models aimed to give insight on how to predict customer churn. Datta et al. [21] proposed a cellular subscriber churn modeling system, in which Decision Tree and genetic algorithms are used for feature selection, and cascade neural network is used for predicting churn scores. Kim et al. [22] applied binomial logit model to analyze subscriber churn in the telecommunication industry. Wang et al. [23] built a churn prediction model on a credit card holders' behaviors dataset from a commercial bank in China. In their study, 12 classification algorithms are compared based on predictive performance. Huang et al. [24] studied a set of features for land-line customer churn prediction, and they also conducted churn prediction comparison based on 7 classification algorithms. Li et al. [25] obtained imbalanced customer data from a major Chinese airline for a study on predicting potential churn customers. F-Measure and G-Mean were used to evaluate the performance of the new ensemble model they proposed. De Caigny et al. [26] investigated the value by adding textual data into customer churn prediction (CCP) models, which use real life data from a European financial services provider on convolutional neural networks (CNNs) to validate the framework.

The above research works focus on predicting the occurrence of customer churn, which cannot be directly applied to the decision-making process due to the lack of recognizing mechanisms for churn determinants. Thus, it is important for researchers to conduct studies to analyze the determinants of customer churn. Gerpott et al. [27] conducted their research on a sample of 684 residential customers of digital cellular network operators in Germany. Based on their analysis, customer satisfaction shows significant impact on customer loyalty. Larivière and Van [28] conducted their research on the savings and investment customers from a large financial service provider in Belgium. Kaplan-Meier is used for discovering the time of a churn event. Varshney and Gupta [29] used the association rule to determine factors affecting churn in telecommunication sectors in India. Their conclusion is that service satisfaction plays an important role in improving customer churn rate. Troncoso [30] collected 23,195 VOC (voice of the customer) interactions (made by 14,531 customers) facilitated from a Chilean bank. They proposed a text mining process to automate the extraction of customer churn determinants from the voice of the customer interactions. Coşer et al. [31] applied an exploratory data analysis to examine the main characteristics of customers that influence the propensity to churn. They used machine learning algorithms and the optimization technique (Grid Search) to obtain the optimal predictive model, and evaluate the model by using Area Under the Curve (AUC).

The above studies focus on recognizing churn determinants, which lack the important information on the cost and effect of different strategies. Therefore, another important topic in churn study is to build the decision support systems (recommender systems) to handle customer churn. Daskalaki et al. [32] built a decision support system to handle customer insolvency for a large telecommunication company. The strategy in this study is to predict churner customers by using the Decision Tree and Neural Network. Kim et al. [33] conducted their study on a dataset from Korean mobile telecommunication service, and investigated the strategies for reducing customer churn. Their conclusion is that customer-oriented services and long-term relationship with customers show great impact in increasing customer loyalty. Burez and Van [34] compared churn prediction performances among Markov chains, a random forest model, and a basic logistic model with data from a European pay-TV company. Their empirical results show that three alternative courses of action are equally effective to reduce customer attrition: (1) giving free incentives (enhancing the service), (2) organizing special events to pamper customers and (3) obtaining feedback on customer satisfaction through questionnaires. Wang et al. [35] used the Decision Tree algorithm to analyze more than 4000 members over three months. The conditional rules produced by the Decision Tree algorithm show the characteristics of customer behavior that can lead to customer loss. They used such rules as strategy recommendations to prevent future customer loss. Lemmens and Gupta [36] defined a profit-based loss function, which weighs more heavily with those customers who have the greatest (positive or negative) impact on the retention campaign profit. They empirically demonstrate that this weighted estimation can boost the profitability of the retention campaign. Some limitations of this approach include: it does not take the long-run impact of retention interventions into account. Moreover, it does not model the relationship between a customer's sensitivity to an intervention and expectations of retention offers in the future.

All the above studies fail to propose an adaptable system that can provide actionable recommendations. Moreover, much of the research only deals with a survey data. It leads to the problem that lacking actual customer transaction data may not fully reflect the customers' future re-patronage decisions. Additionally, most of the churn analysis is frequently adapted in telecommunication, banking and so forth. Churn analysis is rarely carried out in the repair and service sector owing to the data unavailability.

CHAPTER 3: BACKGROUND KNOWLEDGE

This chapter presents the theoretical background of the proposed approaches in this dissertation, and we will cover Action Rules, Meta-actions, Reduct, and Text Mining.

3.1 Action Rules

As the source of action rules discovery, decision system [37] is defined as a pair $S = (U, A \cup \{d\})$, where U is a nonempty, finite set of objects called the universe. $A \cup \{d\}$ is a nonempty, finite set of attributes, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called the domain of a. Attribute d is called the decision. We assume that $A = A_s \cup A_f$. Attributes in A_s are called stable attributes for the set U if their values assigned to the objects from U cannot be changed in time. For instance, "Date of Birth" is a stable attribute. Attributes in $A_f \cup \{d\}$ are called flexible attributes and their values can be changed in time. For instance, "interest rate" on a customer bank account is an example of a flexible attribute [38].

The concept of an action rule was proposed by Ras and Wieczorkowska [39]. An action rule is defined as a rule extracted from a decision system that describes a transition that may occur within its objects by reclassifying them from one state to another, with respect to the decision attribute. More formally, action rule is defined as a term: $[(\omega) \wedge (\alpha \to \beta) \Rightarrow (\Phi \to \Psi)]$, where ω denotes a conjunction of fixed stable attributes, $(\alpha \to \beta)$ are proposed changes in values of flexible attributes, and $(\Phi \to \Psi)$ is a desired change of decision attribute values.

An atomic action set is a singleton set defined as an expression $(a, a_1 \to a_2)$ in S. If $a_1 = a_2$, then a is stable on a_1. Action sets are the smallest collection of sets such that:

- If t is an atomic action set, then t is an action set.
- If t_1, t_2 are action sets, then $t_1 \cup t_2$ is a candidate action set.
- If t is a candidate action set and for any two atomic actions $(a, a_1 \rightarrow a_2)$, $(b, b_1 \rightarrow b_2)$ contained in t, we have $a \neq b$, then t is an action set.

Table 3.1: Decision System S

	a	**b**	**c**	**d**
x_1	a_1	b_1	c_1	d_1
x_2	a_2	b_1	c_1	d_1
x_3	a_2	b_2	c_2	d_2
x_4	a_2	b_1	c_2	d_2
x_5	a_1	b_2	c_1	d_2

For example, Table 3.1 shows a decision system S, where $X = \{x_1, x_2, x_3, x_4, x_5\}$, $A = \{a, b, c, d\}$ and $V = \{a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2\}$. Additionally, let us assume that attributes a, c are stable, and b, d are flexible where d is the decision attribute. One of the action sets mined from this information system is represented by $\{(a, a_1), (c, c_1), (b, b_1 \rightarrow b_2), (d, d_1 \rightarrow d_2)\}$. The meaning of the corresponding action rule $[(a, a_1) \wedge (c, c_1) \wedge (b, b_1 \rightarrow b_2)] \Rightarrow (d, d_1 \rightarrow d_2)$ is that when the stable attribute a is a_1, c is c_1, and if the flexible attribute changes from b_1 to b_2, then it is expected that d_1 will change to d_2.

We use support, confidence, and coverage to measure the quality of extracted action rules. The higher the support, confidence, and coverage of action rules, the more valuable they are to the users [40]. An action rule can be represented as the pair of classification rules [39]. For instance, $r = \{(a, a_1) \wedge (c, c_1) \wedge (b, b_1 \rightarrow b_2) \Rightarrow (d, d_1 \rightarrow d_2)\}$ is represented by $r_1 = \{(a, a_1) \wedge (c, c_1) \wedge (b, b_1) \Rightarrow (d, d_1)\}$ and $r_2 = \{(a, a_1) \wedge$

$(c, c_1) \wedge (b, b_2) \Rightarrow (d, d_2)\}$. Support of r is defined by Equation (3.1).

$$Support(r) = Support(r_1) \tag{3.1}$$

Confidence of r is defined by Equation (3.2).

$$Confidence(r) = Confidence(r_1) * Confidence(r_2) \tag{3.2}$$

Coverage of an action rule r is defined as the number of objects transferred from $Support(d_1)$ to $Support(d_2)$ divided by the number of objects in $Support(d_2)$.

There are two approaches for action rule mining: rule-based approach and object-based approach. In the rule-based approach, a prior extraction of classification rules is needed. Systems DEAR (Discovering Extended Action-Rules) [41] and DEAR2 [42] belong to this category. In the object-based approach, action rules are extracted directly from the database. For instance, ARED (Action Rule Extraction from a Decision Table) [43] is an example of such a system.

3.2 Meta-actions

The action rule approach can be further enhanced by meta-actions, which act as a triggering mechanism for action rules. In other words, meta-actions are the actions that need to be executed in order to trigger the corresponding atomic actions. Meta-actions were initially proposed by Wang et al. [44] to address the actionability problem in data mining. Meta-actions are defined as actions which trigger changes of flexible attributes, either directly or indirectly, because of the correlations among certain attributes in the system [45]. Such correlations can be reflected by an influence matrix.

Table 3.2 presents an example of an influence matrix, which describes the triggering type of correlations between meta-actions and atomic actions. In this table, b is a

Table 3.2: Influence matrix for meta-actions and atomic actions

	a	**b**	**c**
M_1		b_1	$c_2 \rightarrow c_1$
M_2	$a_2 \rightarrow a_1$	b_2	
M_3	$a_1 \rightarrow a_2$		$c_2 \rightarrow c_1$
M_4		b_1	$c_1 \rightarrow c_2$
M_5	$a_1 \rightarrow a_2$	b_1	$c_1 \rightarrow c_2$

stable attribute, a, c are flexible attributes. In the first row, the meta-action M_1 triggers the change in the flexible attribute $(c, c_2 \rightarrow c_1)$ assuming that the value of b is b_1; while in the second row, the meta-action M_2 triggers the change in the flexible attribute $(a, a_2 \rightarrow a_1)$ assuming that the value of b is b_2. Only classification features are involved in an expert knowledge concerning meta-actions [46]. When some classification features are correlated with the decision feature, then change of their values will cascade to the decision value through such correlations. For instance, M_4 can trigger atomic actions (b, b_1) and $(c, c_1 \rightarrow c_2)$, such change will cascade to the decision value d. Therefore, an action rule $r = [(b, b_1) \wedge (c, c_1 \rightarrow c_2) \Rightarrow (d, d_1 \rightarrow d_2)]$ can be triggered by M_4, because no constraint is placed on the attribute a.

The effect of meta-actions is defined by Kuang and Ras [47]. Assume that a set of meta-actions $M = \{M_1, M_2, ..., M_n : n > 0\}$ can trigger a set of action rules, $\{r_1, r_2, ..., r_m : m > 0\}$ which domains have no overlapping objects. The support of M is defined as the summation of all the support of covered action rules. The confidence of M is calculated by averaging the confidence of all the covered action rules.

$$Support(M) = \sum_{i=1}^{m} support(r_i) \tag{3.3}$$

$$Confidence(M) = \frac{\sum_{i=1}^{m} support(r_i) \cdot confidence(r_i)}{\sum_{i=1}^{m} support(r_i)} \tag{3.4}$$

The effect of applying M is defined as the product of support and confidence of M,

$$\eta(M) = support(M) \cdot confidence(M) \tag{3.5}$$

3.3 Reduct

Rough set theory was introduced by Polish mathematician Pawlak [48]. It stems from research on logical properties of decision systems. In a decision system $S = (U, A \cup \{d\})$, we are interested in attributes that are sufficient for the decision-making process. Such a goal leads us to the definition of reducts, the smallest subsets of attributes in a decision system which are sufficient to provide semantically the same knowledge as the knowledge described using all attributes.

Before we recall the definition of a reduct in S, it is necessary to introduce the concept of a discernibility relation. Let objects $x, y \in U$ and $B \subset A$. We say that x, y are discernible by B if there exists $a \in B$ such that $a(x) \neq a(y)$. Objects x, y are indiscernible by B when they are identical on B, that is, $a(x) = a(y)$ for each $a \in B$.

A set of attributes B included in A is called a reduct in S if:

- B keeps the discernibility of A, i.e., for each $x, y \in U$, if x, y are discernible by A, then they are also discernible by B.
- B is irreducible, i.e., none of its proper subsets keeps discernibility properties of A.

There are many heuristic strategies developed to find reducts. The most popular one is based on the discernibility matrix and the discernibility function, which needs conversion to DNF to find reducts [49]. The discernibility matrix [50] of a decision system $S = (U, C \cup \{d\})$ is a symmetric $|U| \times |U|$ matrix with entries defined by Equation (3.6).

$$c_{ij} = \{a \in C | a(x_i) \neq a(x_j)\} \quad i, j = 1, ..., |U| \tag{3.6}$$

So, each c_{ij} contains only attributes that differ between objects x_i and x_j.

Discernibility function f_S is a Boolean function, and it is defined as the conjunction of all logical disjunctions of discernibility matrix elements [51], as shown in Equation (3.7):

$$f_S(a_1, ..., a_m) = \wedge\{\vee c_{ij}|\ 1 \leq j \leq i \leq |U|\} \tag{3.7}$$

There are many other approaches for finding reducts, such as Dynamic Reducts [52], Johnson Reducer [53] and so on. Reduct in rough set theory provides a powerful method for feature engineering. Unlike other methods, attributes reduction requires neither additional information nor external parameters. Instead, it only uses information within the given data. Moreover, it is a minimal subset of attributes that keeps the characteristics of the full dataset.

3.4 Text Mining

We use technologies in text mining to discover knowledge from customer comments. This section contains two parts. The first part is a brief introduction to text preprocessing. The second part is about existing approaches to sentiment analysis, especially aspect-based sentiment analysis.

3.4.1 Text Preprocessing

Tokenization, normalization and noise removal are the three important components in text preprocessing [54]. Tokenization is to split longer strings of text into smaller pieces or tokens. For instance, "It is great that we won." The result of tokenization is {it, is, great, that, we, won}. In this way, we can break down long sentences into smaller units to process.

Normalization refers to a series of tasks that can put all the text into the same level so as to allow text processing to proceed uniformly. Some steps are listed below:

- Convert all text to the same case.

- Remove punctuation.
- Remove numbers.
- Strip white space.
- Remove stop words.
- Stemming (eliminate affixes and keep the word stem).
  - For instance, *leaving* → *leave*.
- Lemmatization (capture canonical forms based on a word's lemma).
  - For instance, *better* → *good*.

Noise removal is a task-oriented procedure. It can occur before or after previously mentioned steps. For instance, if the text that to be prepossessed comes from the internet, we will need to remove the text header and footer, the HTML and XML tags, and so on.

3.4.2 Sentiment Analysis

We use sentiment analysis to identify and extract information of the polarity or emotion from customer comments. This section gives an introduction to sentiment analysis, with focus on aspect-based sentiment analysis. There are mainly four different levels of sentiment analysis, document level, sentence level, aspect level, and concept level. The document-level sentiment analysis classifies the entire document opinion into a positive, negative or neutral sentiment [55]. Sentence-level sentiment analysis aims to determine whether a sentence expresses a positive, negative or neutral opinion, and it works well in the settings when there is only one aspect (and its polarity) in a given text. However, in reality, there are many aspects and polarities in a given text. A more general and complicated task is to predict the aspects mentioned in a sentence and the sentiments associated with each one of them. This generalized task is called aspect-based sentiment analysis [56]. In aspect-based sentiment analysis, two fundamental tasks are aspect extraction and sentiment classification.

3.4.2.1 Aspect Extraction

Aspect extraction aims to extract the opinion targets from texts. For instance, in the statement "This cellphone has a high-resolution screen," the word "screen" is the aspect, and it is a feature of the product. The approach of aspect extraction can be classified to three groups: unsupervised approach, semi-supervised approach and supervised approach [57].

Unsupervised Approach

To extract the aspects based on the frequency or statistical information of the noun and noun phrases is intuitive, because nouns that frequently appear are the important aspects [58]. Bootstrapping is another unsupervised approach, it can be viewed as an iterative clustering technique which in each iteration valuable candidate is added to adjust current seed set [59]. The iteration continues until meeting a pre-defined threshold or satisfying a stopping criterion. Zhu et al. [60] used multi-aspect bootstrapping (MAB) for aspect extraction. Popescu and Etzioni [61] modified the technique proposed by Hu et al. [58]. They introduced point-wise mutual information (PMI) to increase the precision of aspect extraction.

Semi-supervised

Dependency parser gives the relationship between all the words in a sentence. The concept of phrase dependency parsing was introduced by Wu et al. [62]. It aims to extract relations between product aspects and opinion words. Yu et al. [63] used phrase dependency parser to extract the aspects from reviews. They rank different aspects by taking the aspect frequency and user's opinion into account.

Lexicon-based method extracts aspects based on a list of lexicon. Wei et al. [64] proposed a semantic-based product aspect extraction technique. With all the aspects extracted, the authors used a list of positive and negative adjectives to identify the subjectivity of the opinion words. Based on this list, the potential aspects that are not aspects are pruned out.

Supervised

As aspect extraction belongs to information extraction problem, supervised learning method that is used in information extraction can also be applied in aspect extraction. One of the most dominant methods is based on sequential learning. Hidden Markov Models (HMM) and Conditional Random Fields (CRF) are two popular sequential learning methods. Dictionary-based methods utilize a dictionary list for aspect extraction. Cruz et al. [65] proposed to generate a domain oriented taxonomy for aspect extraction. Jiang et al. [66] proposed a tree-based approach for aspect extraction. They used a generalized aspect sentiment tree to extract aspects, and they defined four tree kernel spaces to identify aspects from the reviews. Li et al. [67] formulated the opinion target extraction problem as a shallow semantic parsing problem. The sentence is represented by a parse tree. A predicate is used to identify the corresponding semantic arguments in the sentence. They used several heuristic rules to map the opinion targets into several constituents and to prune out those arguments which do not satisfy the rules.

3.4.2.2 Sentiment Classification

Another important step in sentiment analysis is to apply an appropriate technique to classify the sentiments. There are many different classification methods proposed in literature, which fall into two groups: machine learning approach and lexicon-based approach [68].

Machine Learning Approach

Machine learning based sentiment classification method can be divided into three groups: Supervised, Semi-supervised, and Unsupervised learning methods. Supervised learning is a function that maps an input to an output based on example input-output pairs. In supervised learning sentiment classification, training text samples with categorized labels are used to classify the sentiment of the testing data. Some popular and effective methods are Naive Bayes, Maximum Entropy, Support Vector Machine(SVM), Decision Tree, Artificial Neural Network, and K-nearest Neighbor. Pang et al. [69] compared the performance of sentiment classification among Naive Bayes, Maximum Entropy and SVM. Alfaro et al. [70] compared the performance of SVM and the K-nearest Neighbor algorithm for text classification and sentiment analysis by using weblog data as dataset.

Semi-supervised learning method aims to use unlabeled data together with a small size of labeled data to build the classifiers. Multi view learning method is a semi-supervised learning method. In multi view learning, k models are produced based on k views, which can effectively improve the overall performance of the classification. It can be used in cross-lingual sentiment analysis, which utilizes the labeled data of the source language for compensating the lack of labeled data in the target language [71].

In unsupervised learning, the training data is not labeled yet. The main idea is putting the unlabeled data into different groups, and the members of each group are similar to each other from a particular point of view. In this way, data of same clusters have the maximum similarity, while data in different clusters share the minimum similarity. K-means is a typical clustering algorithm for machine learning, and it is popular in sentiment classification. K-means clustering assigns all data into k groups by minimizing the objective function [72]. It is defined as Equation 3.8.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} ||\rho - \mu_i||^2 \tag{3.8}$$

where k is the number of clusters, and μ_i is the centroid of a cluster C_i, and ρ is the data point in each cluster. Claypo and Jaiyen [72] proposed an opinion mining method on Thai restaurants' review by using K-means clustering. The agglomerative method is another popular unsupervised learning sentiment classification method. Starting with a set of n objects to be clustered, it groups these objects into successively fewer than n sets, arriving eventually at a single set containing all the n objects [73]. De and Kopparapu [74] used agglomerative algorithm as the clustering technique for collecting opinions.

Lexicon-based Approach

Lexicon is defined as the vocabulary of a person, language, or branch of knowledge. Lexicon-based sentiment classification uses adjectives as indicators of the semantic orientation of the texts [58]. First, a list of adjectives and the corresponding semantic orientation values are compiled into a dictionary. Then, for any given text, all adjectives are extracted and annotated with their semantic orientation value, using the dictionary scores. The semantic orientation scores are in turn aggregated into a single score for the text [75]. Lexicon-based method can be divided into two categories: dictionary-based approach and corpus-based approach.

In dictionary-based method, a sentiment dictionary is generated. Firstly, a few sentiment words are selected manually. Then, based on bootstrapping technique, more words are added by searching in online dictionaries. This process continues until the collection of words is stable, no new words are detected [76]. SentiWordNet is a popular sentiment dictionary built on top of WordNet [77], according to the notions of positivity, negativity, and neutrality. Each synset is associated with three numerical scores: Pos(s), Neg(s), and Obj(s) which indicate how positive, negative, and objective (i.e., neutral) the terms contained in the synset are [78]. SentiSense classifies WordNet synsets [77] with emotional meanings. It consists of 5,496 words and 2,190 synsets labeled with an emotion from a set of 14 emotional categories which are related by an antonym relationship. SentiSense has been developed semi-automatically using several semantic relations between synsets in WordNet [79]. AFINN is a strength-oriented lexicon [80]. There are 564 positive words scored from 1 to 5, and 964 negative words with score ranging from -1 to -5. Liu and Hu's Opinion Lexicon [58] is a list of English positive and negative opinion words or sentiment words (around 6800 words). VADER (Valence Aware Dictionary and sEntiment Reasoner) was created from a generalized, valence-based, human-curated gold standard sentiment lexicon. It is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, for instance, Twitter.

CHAPTER 4: DATASET

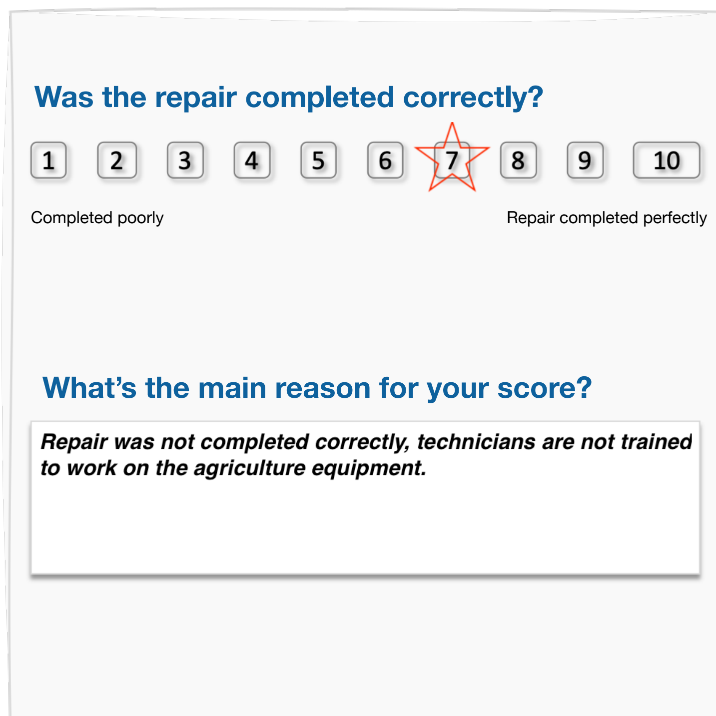
4.1 Introduction

The dataset used in this dissertation was collected by a consulting company based in Charlotte, North Carolina. The company made telephone surveys to query customers' satisfaction on repair services done by 34 heavy equipment repair companies. Among all the customers who have been served by these 34 companies, more than 25,000 customers were randomly chosen to take the survey. There are more than 400,000 records ranging from 2011 to 2017. The dataset consists of features related to:

- Clients details: ID, Group, Name, Division, etc.
- Survey details: ID, Name, Type, Date Interviewed, Completed by, etc. Based on which customer satisfaction area we focus on, the type of the survey is classified as field, shop, parts, rentals and sales. Field means sending technician to customers' for repairing.
- Customer details: ID, Name, Contact Phone, City, State, Zip, etc.
- Order details: ID, Invoice Number, Invoice Amount, etc.
- Benchmark questions (survey questions) and attached notes.
  - The answer to the benchmark question is recorded as a rating score, ranging from 1 to 10.
  - Answer to the open-ended question ("What's the main reason for your score?") is recorded as a textual data saved as attached notes (we rename it as "review" to reflect its function). Customers can give their thoughts, feelings, complaints, and expectations freely in their answers.

An example is shown in Figure 4.1, one of the benchmark questions is "What's your overall satisfaction?" and the customer being surveyed rates 7 to this benchmark question. For the open-ended question "What's the main reason for your score?" the customer's answer is *"Repair was not completed correctly, technicians are not trained to work on the agriculture equipment."* From his answer, we get the clue that the customer rates 7 instead of a higher score (like 9 or 10) because he is not content with the repair.

Figure 4.1: Survey example

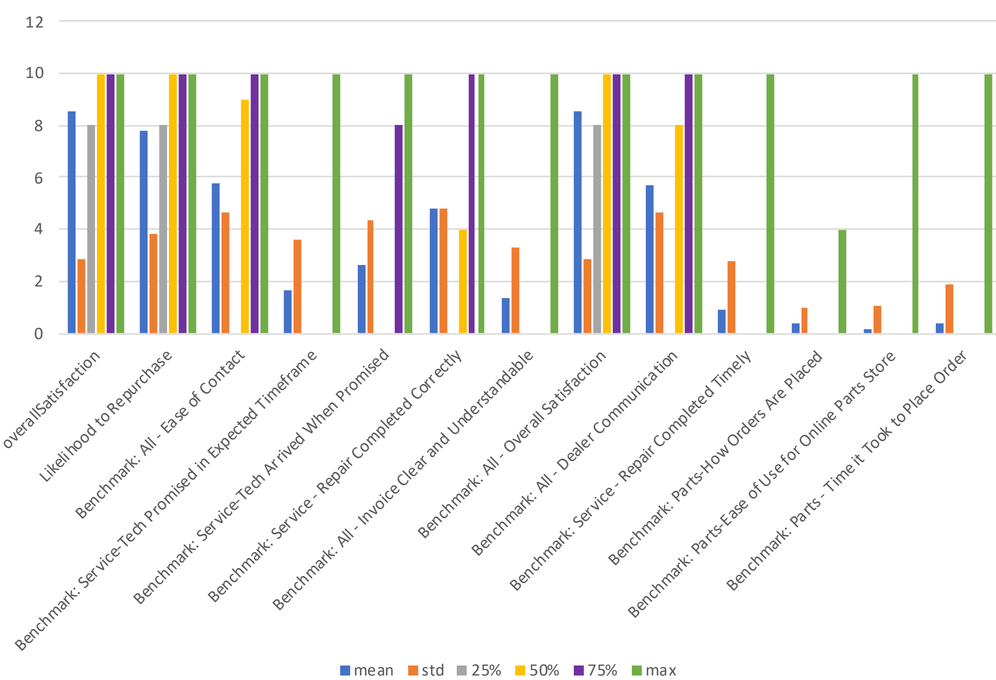


Based on different topics, the survey is divided into five sections: service, sales, rental, parts, and all-field (which covers all the topics: service, sales, rental, parts). In each category, there are different benchmark questions, which served as the questions that were raised to customers during the survey. We have more than 200 benchmark questions in total.

4.2 Data Prepossessing

In reality, when taking time constraint into consideration, it is impossible to ask all the questions when conducting the survey on a certain customer. Only some small subsets of questions are asked. Therefore, the dataset we are dealing with is highly incomplete and multidimensional. Figure 4.2 shows the descriptive statistics of the customer ratings to a few example benchmark questions. In data preprocessing,

Figure 4.2: Descriptive statics



we focus on dealing with the missing values. Considering two scenarios: when the benchmark question was not asked, or when most customers did not give their ratings to certain benchmark questions. In these scenarios, the benchmark columns contain many null values, which bear little information. We set 95% as the threshold for deletion based on our discussion with the domain experts. The final rule is that when there is more than 95% null values in a benchmark column, such benchmark column is removed.

After data cleansing, there are 48 benchmark questions left, some example benchmark questions are shown in Table 4.1.

Table 4.1: Example benchmark questions

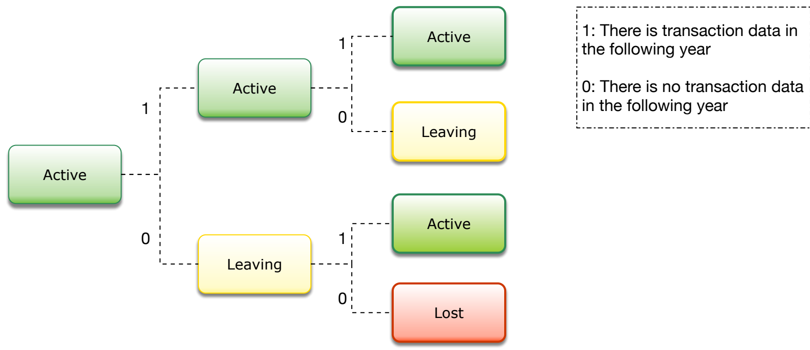
Topic	**Features**
All-Field	Ease of Contact Likelihood to be Repeat Customer Has Issue Been Resolved
Service	Tech Promised in Expected Time frame Tech Arrived When Promised Final Invoice Matched Expectation
Parts	How Orders Are Placed Ease of Use for Online Parts Store Explanation of Delivery Options and Cost
Rental	Equipment Available Equipment Delivered When Expected Equipment Clean and Working
Sales	Product Specs Provided by Salesperson Complete Pricing Information Provided by Sale Responsiveness of Salesperson to Information

4.3 Decision Attribute Generation

In the original survey dataset, there is no attribute that can summarize the information about customer churn status (active, leaving or lost). Therefore, we consulted with domain experts for the definition of active, leaving, and lost customers. First, there are customers who prefer not to be called, and they ask to be added into the "Do Not Call" list. Such customers might have transaction data in the following years, but lack of succeeding survey data. Moreover, not all the customers were given a consistent telephone survey. Customers might be surveyed in a certain year. While for the following year, they were not surveyed, but they have follow-up transactions. Taking the above scenarios into consideration, we combined the survey data with the transaction data together to label customer status to reflect the true re-patronize behavior of customers.

According to the definition proposed by the domain expert, we label the customer status into three categories: active, leaving, and lost. Active customers are those

Figure 4.3: Customer labelling



who made transactions minimum once a year. Leaving customers are those who did not make transactions in the subsequent year. Lost customers are those who stopped doing business for the following two or more years.

In Figure 4.3, we give an example of the customer status labelling based on the transaction data and survey data in three consecutive years. Data in 2012 and 2013 are used to mark customer status of 2011. For an active customer in 2011, if there is no transaction data in 2012, he is labeled as leaving. Additionally, when there is no transaction record in 2013, we updated his status as lost.

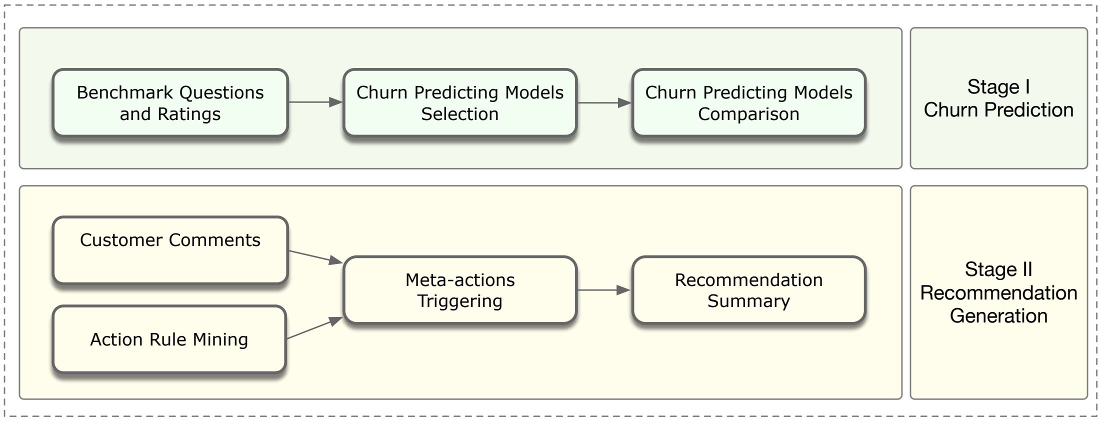
CHAPTER 5: OVERVIEW OF CUSTOMER ATTRITION MANAGEMENT SYSTEM

In this chapter, we present the two-stage Customer Attrition Management System, which is embracing both the power of predicting customer churn and the ability to provide measurable recommendations for reducing customer churn. In order to achieve a higher prediction precision in detecting the leaving customers, we compared several widely used churn prediction models, and pick the one which gives an overall best predicting performance. Once the leaving customers are identified, it is of great necessity to improve customer retention. Therefore, we then built a knowledge-based recommender system to provide actionable recommendations for decision makers to reduce customer churn.

5.1 Methodology

Figure 5.1 gives the design of the Customer Attrition Management System, and it contains two stages. Stage I is Churn Prediction, and Stage II is Recommendation Generation. Stage I focuses on identifying the leaving customers by utilizing customer churn models. After we identify the churn cases, the next step is to generate recommendations on improving churn rate. Thus, in Stage II, we run action rule mining to discover actionable patterns that can reclassify customers from leaving to retained. Then we extract meta-actions from the associated customer comments to generate the appropriate recommendations. Finally, we summarize the recommendations and calculate their effectiveness scores.

Figure 5.1: Design of the Customer Attrition Management System



5.1.1 Churn Prediction

The most commonly used churn prediction modelling techniques are Decision Tree, regression analysis, ANN, and SVM [9]. As stated in Neslin, et al. [81], Logistic Regression, which is of conceptual simplicity, gives quick and robust results compared to other regression analysis models. Therefore, we choose Logistic Regression as the regression analysis model. XGBoost is highly flexible and versatile, and it is widely used in churn prediction [82], and we add XGBoost into the churn predication performance comparison. Therefore, without loss of generality, we select Decision Tree, Logistic Regression, ANN, SVM and XGBoost as the candidate customer churn prediction models.

We compare the prediction performance of these churn prediction models based on accuracy score, AUC (Area Under Curve), precision, recall score, and F1 score. TP (True Positive: actually positive, and classified as positive), FP (False Positive), FN (False Negative), and TN (True Negative) are used in the definition of these measurements. Accuracy is the ratio of the correctly predicted observation to the total observations.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.1}$$

Precision is defined as the ratio of correctly predicted positive observations to the total predicted positive observations. It stands for the accuracy of positive predictions.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

Recall is also called true positive rate. It describes how the model's prediction covers the positive class.

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

F1 score is the weighted average of Precision and Recall.

$$F1\_Score = 2 * (\frac{Precision * Recall}{Precision + Recall}) \tag{5.4}$$

AUC (Area Under Curve): The Receiver Operating Characteristic (ROC) curve plots the true positive rate vs the false positive rate, and AUC is computing the area under the curve.

5.1.2 Recommendations Generation

Once we have identified the potential churn scenario, it is necessary to generate valid and effective recommendations to improve churn rate. We apply action rule mining and the meta-action triggering mechanism to build a knowledge-based recommender system to generate appropriate and effective recommendations.

5.1.2.1 Action Rule Mining

As introduced in 3.1, action rule is defined as an expression: $[(\omega) \wedge (\alpha \rightarrow \beta) \rightarrow (\Phi \rightarrow \Psi)]$, where ω denotes a conjunction of fixed stable attributes, $(\alpha \rightarrow \beta)$ are proposed changes in values of flexible attributes, and $(\Phi \rightarrow \Psi)$ is a desired change of decision attribute value. We define survey type as the stable attribute because it reflects the type of survey (service, sales, rental, parts, and all-field) which is stable by its nature. Customer ratings to the benchmark questions provide hints about what changes can be made for a given group of customers to reclassify them from an undesired status to a desired one. Therefore, the benchmark questions are defined as the flexible attributes. Customer status gives the churn status (active, leaving or lost) of a customer, and we define it as the decision attribute.

5.1.2.2 Meta-actions Extraction

Meta-actions are the triggers that can be used for activating action rules. Meta-actions are extracted from relevant comments left by customers in that domain, respectively [83]. For instance, $r = [(a, a_1) \wedge (b, b_1 \rightarrow b_2)] \Rightarrow (d, d_1 \rightarrow d_2)$ is an action rule where a is a stable attribute and b is a flexible attribute. The decision attribute is d, and the desired changes for d is from d_1 to d_2. The clues for generating meta-actions are in the comments of records matching $(a, a_1) \wedge (b, b_1) \wedge (d, d_1)$ and $(a, a_1) \wedge (b, b_2) \wedge (d, d_2)$. For instance, in action rule (5.5):

$$[(Benchmark\_1, 5 \rightarrow 8) \wedge (Benchmark\_2, 6 \rightarrow 10)] \\ \Rightarrow (Customer\_Status, Leaving \rightarrow Active) \quad (5.5)$$

when the rating for Benchmark_1 increases from 5 to 8 and the rating for Benchmark_2 changes from 6 to 10, then customers' status is expected to change from leaving to active. The rating and associated comments are shown in Table 5.1. To trigger this action rule, we need to find the meta-actions from the related comments.

Customer 2061000 expresses his negative feeling regarding technician and billing: technician not knowledgeable ("*technicians are not very knowledgeable*") and billing issue (*"issues with billing in the past"*) while customer 3078023 conveys her positive attitude towards technician in the two aspects of knowledge and communication: knowledgeable technicians (*"he is very knowledgeable, very reliable"*) and proactive communication (*"communicated effectively by providing excellent feedback"*). Therefore, based on the knowledge mining from text comments, the meta-actions can be summarized as: knowledgeable technicians, reasonable prices, and proactive commu-

Table 5.1: Example of benchmark ratings and associated comments

Customer ID	**Status**	**BM*1**	**BM2**	**Comments**
2061000	Leaving	5	6	He stated the technicians are not very knowledgeable. He said he has also had some issues with billing in the past.
3078023	Active	8	10	She said, "I am very happy with Greg, the technician, for he is very knowledgeable, very reliable, communicated effectively by providing excellent feedback, and he is wonderful and amazing."

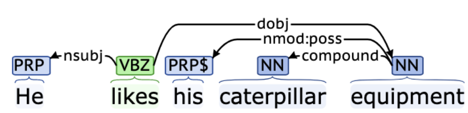
∗ BM: Benchmark

nication. These meta-actions can work as recommendations to trigger the action rule (5.5). Mining meta-actions from text can be summarized into five steps:

Step 1: Apply sentiment dictionaries to identify the opinion sentences. We use VADER (Valence Aware Dictionary and sEntiment Reasoner) [84] as our main reference for detecting the opinion words and assigning sentiment orientations. VADER not only gives the polarity (positivity and negativity) score, but also tells about how positive or negative a sentiment is. In VADER, "-1" stands for extreme negative; while "+1" means most extreme positive. Moreover, in order to catch the domain-oriented sentiment words, we also built a domain-oriented supplement sentiment dictionary by consulting domain experts.

Step 2: Apply structured heuristic rules to filter out the aspect-opinion pairs (aspects and the associated opinions) based on dependency parser tree [85] of each review sentence.

Figure 5.2: Sentence dependency parser



A dependency parser focuses on analyzing the grammatical structure of a sentence and establishing the relationships between the "head" words and words that modify those heads [86]. Take review "He likes his caterpillar equipment." as an example. In this sentence, "(caterpillar) equipment" is an aspect, and "likes" is a sentiment word. Figure 5.2 shows a dependency parse tree of this short sentence. The arrow from the word "likes" to the word equipment indicates that "equipment" modifies "likes", and the label "dobj" (direct object) assigned to the arrow describes the exact nature of the dependency.

Step 3: Cluster the aspect-opinion pairs into predefined classes. Based on suggestions from domain experts, we cluster the aspect-opinion pairs into predefined classes: {service: correctly, timely}, {communication: proactive, timely}, {technician: friendly, knowledgeable} and {price/invoice: reasonable, competitive}.

Step 4: Generate meta-actions. We use the predefined classes to generate appropriate meta-actions, which are essentially recommendations that we target.

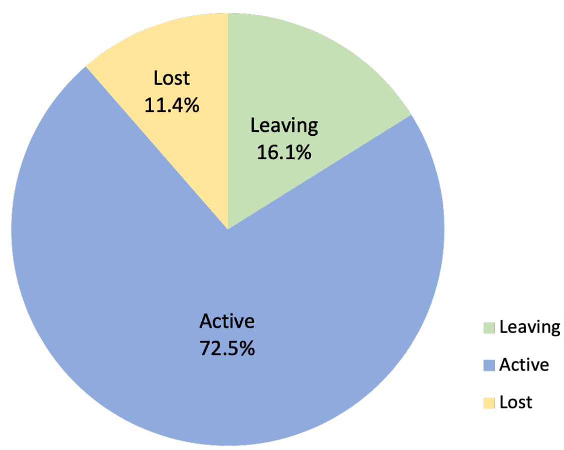
5.1.2.3 Recommendations Summary

We summarize the recommendations based on meta-actions. In order to find the smallest and most efficient set of triggers for a discovered set of action rules, we adopt the strategy presented by Tarnowska and Ras [87]. They introduced the concept of meta-nodes. It is forming a tree structure where each meta-node is uniquely associated with a set of action rules and their triggers. The lower the nodes in the tree, the more action rules are activated (which usually means more triggers are needed). Users are provided with several optional meta-nodes to choose from, and the groups of meta-actions (triggers) linked with these meta-nodes are provided, where each one is sufficient to activate all associated action rules. We also calculate the effectiveness score of different meta-action sets. The score can provide decision-makers with measurable indicators before adopting these recommendations.

5.2 Evaluation

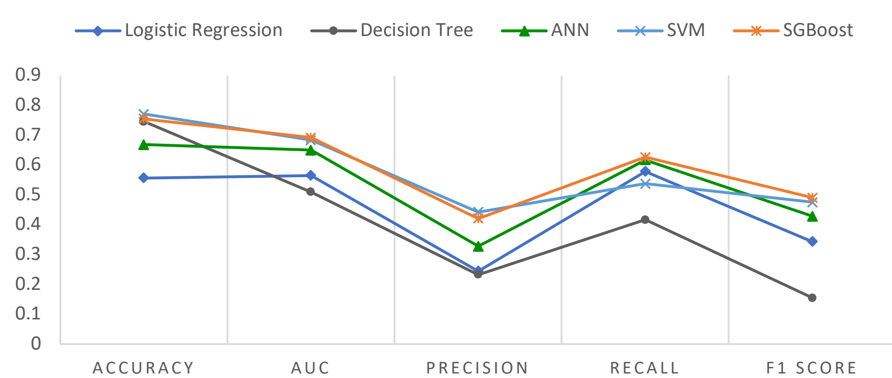
As introduced in Chapter 4, we have collected the surveys to query customers' satisfaction on repair services done by 34 heavy equipment repair companies. We conduct the customer churn study on the dataset from one of these 34 companies, which is deeply suffering the customer churn issue. In this dataset, there are 22,558 records ranging from 2011 to 2017. Among these customer records, 11.4% are lost, 16.1% are leaving and 72.5% are active, as shown in Figure 5.3.

Figure 5.3: Customer status distribution



We use Scikit-learn [88] to compare the prediction performance among ANN, Logistic Regression, Decision Tree, SVM and XGBoost. Figure 5.4 shows the performance

Figure 5.4: Churn prediction model performance comparison



comparison among these prediction models based on accuracy score, AUC, precision, recall score and F1 score. From the experiment result, we can see that XGBoost gives an overall best performance among these five prediction models. Thus, in the first stage of the Customer Attrition Management System, we pick XGBoost as the churn prediction model to predict and recognize the leaving customers.

The second stage is recommendation generation. We use LISp-Miner [89] to discover the action rules, and we extract action rules with a minimum support of 10 and minimum confidence of 80%. In our experiment, survey type is defined as the stable attribute. Benchmark questions after data cleansing are selected as the flexible attributes. Customer status is the decision attribute, and the desired change for the decision attribute is from leaving to active. Table 5.2 presents some example action rules we have discovered. Let's comment on the first action rule AR1. The stable

Table 5.2: Example of discovered action rules

Action rules	Support	Confidence
AR1 $SurveyType(Field)$: $(All\_Does\_Customer\_have\_Future\_Needs, 0 \rightarrow 1)$ $\wedge$ $(Service\_Final\_Invoice\_Matched\_Expectations,$ $8 \rightarrow 10) \Rightarrow (CustomerStatus, Leaving \rightarrow Active)$	21	97.4%
AR2 $SurveyType(Field)$: $(All\_Ease\_of\_Contact, 8 \rightarrow 10) \wedge$ $(Service\_Final\_Invoice\_Matched\_Expectation, 6 \rightarrow 9)$ $\Rightarrow (CustomerStatus, Leaving \rightarrow Active)$	56	96.8%
AR3 $SurveyType(Shop)$: $(Service\_Repair\_Completed\_Correctly, 7 \rightarrow 10)$ $\Rightarrow (CustomerStatus, Leaving \rightarrow Active)$	34	96.1%
AR4 $SurveyType(Shop)$: $(All\_Dealer\_Communication, 8 \rightarrow 10)$ $\Rightarrow (CustomerStatus, Leaving \rightarrow Active)$	175	95.9%
AR5 $SurveyType(Shop)$: $(Service\_Repair\_Completed\_When\_Promised, 9 \rightarrow 10)$ $\Rightarrow (CustomerStatus, Leaving \rightarrow Active)$	89	95.0%
AR6 $SurveyType(Field)$: $(All\_Ease\_of\_Contact, 3 \rightarrow 7)$ $\wedge$ $(Service\_Tech\_Arrived\_When\_Promised, 7 \rightarrow 10)$ $\Rightarrow (CustomerStatus, Leaving \rightarrow Active)$	103	94.2%

part is Survey Type (Field) and the flexible parts are "All-Does Customer have Future Needs" and "Service-Final Invoice Matched Expectations." The decision part is "Customer Status." When rating of "All-Does Customer have Future Needs" changes from 0 to 1 and rating of "Service-Final Invoice Matched Expectations" changes from 8 to 10, then "Customer Status" is expected to change from leaving to active.

In Table 5.3, we give experiment results of meta-actions that can trigger the action sets listed in Table 5.2. We can see that the relationship between atomic action sets and meta-actions is "many-to-many relationships." Each of the atomic actions can be

Table 5.3: Action sets and meta-actions

No.	**Atomic action sets**	**Meta-actions**
AR1	(All_Does_Customer_have_Future_Needs, $0 \rightarrow 1$) (Service_Final_Invoice_Matched_Expectations, $8 \rightarrow 10$)	Service Correctly Communication Proactive Invoice Reasonable Price Competitive
AR2	(All_Ease_of_Contact,$8 \rightarrow 10$) (Service_Final_Invoice_Matched_Expectation, $6 \rightarrow 9$)	Communication Proactive Invoice Reasonable
AR3	(Service_Repair_Completed_Correctly, $7 \rightarrow 10$)	Service Correctly Technician Knowledgeable
AR4	(All_Dealer_Communication, $8 \rightarrow 10$)	Technician Friendly, Communication Proactive
AR5	(Service_Repair_Completed_When_Promised, $9 \rightarrow 10$)	Service Correctly, Service Timely
AR6	(All_Ease_of_Contact, $3 \rightarrow 7$) (Service_Tech_Arrived_When_Promised, $7 \rightarrow 10$)	Service Timely Service Correctly Communication Proactive

triggered by one or more meta-actions. At the same time, each of the meta-actions can invoke one or more atomic actions. For instance, the atomic action sets in action rule AR1 are (All: Does Customer have Future Needs, $0 \rightarrow 1$) and (Service: Final Invoice Matched Expectations, $8 \rightarrow 10$). The meta-actions that can trigger such changes are: to ensure service completed correctly, to ensure proactive communication, to ensure reasonable invoice, and to ensure competitive price.

Table 5.4 shows the experimental results of each meta-node and its effective score.

Table 5.4: Meta-node and its effect

Meta-node	**Effect**	**Meta-actions**
...	...	...
30	52.59	Communication proactive Price competitiveness Service completed correctly Price Reasonable Technician friendly Technician Knowledgeable
31	49.34	Communication proactive Service completed correctly Communication timely Technician friendly Technician Knowledgeable
32	40.83	Price reasonable Price Competitiveness Service completed correctly Technician Knowledgeable
33	34.89	Communication proactive Price competitiveness Communication timely Technician Knowledgeable
...	...	...

The effect of each meta-node is calculated by Equation (3.5). For instance, meta-node 30 contains meta-actions communication proactive, price competitiveness, service completed correctly, technician friendly, technician knowledgeable. The effective score of these meta-actions is 52.59. Based on the meta-actions (recommendations) and the information on the effectiveness score, the decision makers can find the trade-off between the effect and the cost when adopting these recommendations.

CHAPTER 6: RECOMMENDER SYSTEM BASED ON ACTION RULES AND SENTIMENT MINING

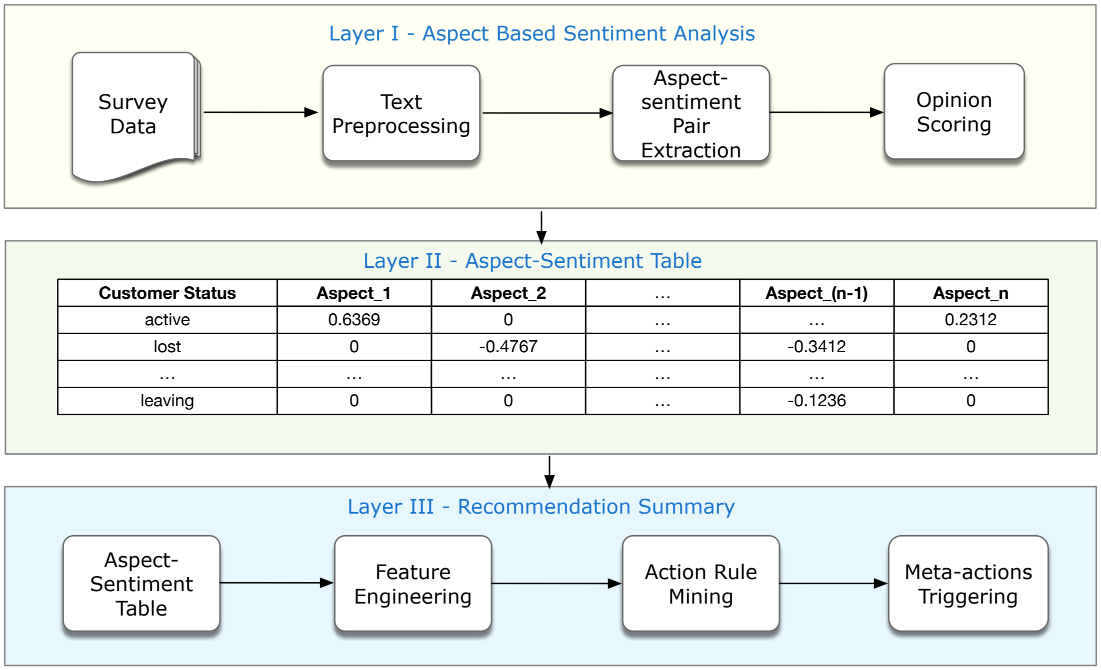
Reviews from leaving customers reflect their unfulfilled needs, while reviews of active customers reflect their satisfactory experience. To further enhance the efficiency of the Customer Attrition Management System, we propose to extract action knowledge from the unstructured customer review data. First, we use aspect-based sentiment analysis to transform the unstructured review text data into a structured table. In the newly created sentiment table, aspects extracted from the text are attributes, and the associated opinion scores are the attribute values. Then, we extract action rules from this newly built sentiment table, as the sentiment table is built on top of text mining. Therefore, we can get recommendations directly from the result of action rule mining, which will save us from the effort of extracting the meta-actions.

6.1 Methodology

Figure 6.1 shows the design of the system, which includes three layers. The first layer is Aspect-based Sentiment Analysis. The second layer is "Aspect-Sentiment Table," and the third layer is Recommendation Summary.

In the first layer, we extract the aspects and the associated sentiments as aspect-sentiment pairs by applying aspect-based sentiment analysis. In the second layer, we build an "Aspect-Sentiment Table" based on the information retrieved from the first layer. In the third layer, we conduct feature engineering, action rule mining and meta-action extraction on the "Aspect-Sentiment Table" built in the second layer. The meta-actions extracted are used to generate recommendations on improving customer churn rate.

Figure 6.1: Recommender system for improving customer churn rate



6.1.1 Aspect-based Sentiment Analysis

The aspect-based sentiment analysis contains three steps: text preprocessing, aspect-sentiment pair extraction and opinion scoring. In the process of aspect-based sentiment analysis, we first get customer ID and customer comment (review). Second, in 'Text preprocessing', we apply text preprocessing on the review. Third, in 'Aspect-sentiment pair extraction', we pass the processed sentence to a series of structured rules to extract aspects and associated sentiments. Forth, in 'Sentiment scoring', we assign a sentiment score to the opinion word. Finally, we accumulate all aspects and merge the results.

6.1.1.1 Text Preprocessing

Text preprocessing is an important step in text mining. Here, the goal of text preprocessing is to prepare the text data for aspect-sentiment pair extraction. We need to put all text into the same level to make the processing proceed uniformly (for example, by converting all text to the same case). Moreover, we need to remove the noisy words as well, for instance, special characters. In this section, we conduct the following steps for text preprocessing: converting to lowercase, checking spellings, removing special characters (for instance #,*), removing single characters, lemmatization and stemming.

6.1.1.2 Aspect-sentiment Pair Extraction

We use structured heuristic rules to filter out the aspect-sentiment pairs based on the dependency parser tree of each review. We use Python package spaCy [90] to recognize the grammatical dependency relationships so as to extract the aspect-opinion pairs from customer reviews. We consider five main dependency paths [91], and the steps for extracting the aspect-sentiment pairs are given below:

1. Get the dependency tree for every sentence with the help of spaCy.

2. Given a dependency tree, extract the aspect-opinion pair by matching dependency rules below:

- Adjectival Modifier Rule: Describe the relation between the nominal (aspect) and the adjective or adjectival phrase (opinion modifier) which modifies the meaning of the nominal.

- Copular Complement Rule: Define the relation among the syntactic subject (aspect) of a clause, copular verb and its complement (opinion modifier).

- Adjectival Complement Rule: Describe the relation among the nominal subject (aspect), the verb and the adjectival complement (opinion modifier)

- Direct Object Rule: Define the relation between the verb (opinion modifier) and the object of the verb (aspect).

- Adverbial Modifier (to a passive verb) Rule: Describe the relation among the syntactic subject (aspect) of a passive clause, the verb and the adverb modifier (opinion modifier).

3. Given a matched aspect-opinion pair, if negations are captured, then mark the aspect-opinion pair as "neg-". We consider the following negation scenarios.

- Negation word no : *'Gary said there was no communication'*
- Negation with not or n't: *'The parent company wasn't responsive'*

6.1.1.3 Sentiment Scoring

We use lexicon-based sentiment classification to assign sentiment scores to the opinion words. As a widely used lexicon-based sentiment dictionary, VADER (Valence Aware Dictionary for sEntiment Reasoning) [92] is used to assign the proper sentiment scores. It outperforms eleven typical state-of-practice benchmarks including LIWC, ANEW, the General Inquirer, SentiWordNet, and machine learning oriented techniques relying on Naive Bayes, Maximum Entropy, and Support Vector Machine (SVM) algorithms [84]. Moreover, it not only gives the polarity (positivity and negativity) score, but also tells about how positive or negative a sentiment is. "-1" stands for extreme negative, while "+1" means most extreme positive. That is, for positive sentiment, the compound score ≥ 0.05. The value of compound score of neutral sentiment is $-0.05 <$ compound score < 0.05. Negative sentiment has a compound score ≤ -0.05.

However, only referencing VADER to detect the opinion words is not sufficient due to its generality. We also added domain-oriented opinion words into the library based on our own learning.

6.1.2 Aspect-Sentiment Table

We combine customer churn status (active, leaving, lost) and aspect-sentiment pairs together to build the Aspect-Sentiment Table. We assign "0" to the aspects that customers did not mention, based on the assumption that they are neutral to such aspects. A sample table is given by Table 6.1. User 1061000 is active, and the score for technician and repair is positive, which means that in his comments, he expressed satisfaction about technician and repair. For user 1138023, the sentiment score for repair is negative, which is a signal that he is not content with the repair he got.

Table 6.1: Sample aspect-sentiment table

C_ID	*C_Status*	*Survey_ Type*	*Technician*	*Knowledge*	...	*Repair*
1061000	Active	Field	0.3412	0	...	0.6369
1138023	Leaving	Field	0	0	...	-0.1835
1571030	Lost	Field	0	-0.1348	...	0
2818061	Leaving	shop	0.2584	0	...	0
...						
3209865	Active	rental	0	0	...	0

6.1.3 Recommendation Summary

We apply feature engineering, action rule mining and the meta-action triggering mechanism to get the recommendations. In order to find the optimal and essential subset of features, we remove the unnecessary features regarding the decision attribute. At the same time, characteristics of the dataset should be kept after feature engineering. Therefore, in feature engineering, we firstly rank the aspects based on its frequency in customer comments, and keep the important ones which rank at the top level [58]. Then we find reducts of these important aspects.

Action rule mining is applied on the newly created table. As we have introduced in 3.1, action rule is defined as $[(\omega) \wedge (\alpha \rightarrow \beta) \Rightarrow (\Phi \rightarrow \Psi)]$ where ω denotes conjunction of fixed stable attributes, $(\alpha \rightarrow \beta)$ are proposed changes in values of flexible attributes, and $(\Phi \rightarrow \Psi)$ is a desired change of the decision attribute. We define customer status, which indicates customer churn information, as the decision attribute. Survey type tells us about the type of service, for instance "field trips" or "in-shop repair," which work as stable attributes naturally. Attributes generated from reviews are the flexible attributes. We aim to recognize the actionable patterns that can improve the customer churn rate.

Action rules provide us recommendations about changes needed to reclassify customers from an undesired to a desired state. However, in the decision process, knowledge on how to invoke these changes is still needed in order to provide useful recommendations. We use meta-actions as a triggering mechanism to activate action rules. Meta-action is defined as a high-level action that can activate changes of flexible attributes, either directly or indirectly, because of correlations among them. On one hand, the meta-actions are extracted from the text data by identifying certain keywords, grouping them using a semantic similarity measure, and then assigning tags to these groups. On the other hand, the column names of the "Aspect-Sentiment Table" are extracted from the review text data, and then grouped based on semantic similarity. Therefore, these aspects, which are listed in the "Aspect-Sentiment Table" as new attribute names, are essentially the meta-actions.

6.2 Evaluation

In our experiment, we use Hadoop research cluster to conduct the experiments. The Hadoop cluster is the parallel mode environment in our experiment. It has 16 nodes/192 computing cores and 87 TBs Hadoop Distributed File System (HDFS). Each computer node has dual Intel 2-93GHz 6-core processors and 64GB RAM.

We conduct the customer churn study on the dataset from one of these 34 companies, which is deeply suffering from the customer churn issue. In this dataset, there are 22,558 records ranging from 2011 to 2017. Based on this dataset, we build the "Aspect-Sentiment Table". Table 6.2 gives the properties of the "Aspect-Sentiment Table" we have built. By ranking the occurrence frequency of all the candidate aspects, we keep aspects ranking at the top level as the whole aspect set. We use reduct to remove the unnecessary features regarding the decision attribute so as to get the

"Aspect-reducts", which are the optimal and essential subsets of the whole aspect set.

Table 6.2: Properties of aspect-sentiment table

Property	**Aspect-sentiment table**
Aspect-reducts	survey, parts, communication, price, skill technician, experience, repair, service, timeliness
Decision attributes distribution	Active - 72.5% Leaving - 16.1% Lost - 11.4%

6.2.1 Evaluation Procedure

We extract action rules with the minimum support of 10 and minimum confidence of 80%, and we set the customer status as the decision attribute, the survey type as the stable attribute. Aspect-reducts (as shown in Table 6.2) are selected as the flexible attributes. In order to achieve higher confidence and support in action rule mining while not losing the important information, we preprocessed the Aspect-Sentiment Table by keeping the sentiment score with one digit after the decimal point.

In our evaluation, we conduct two experiments. The first experiment is mining actionable patterns that can reclassify customers from leaving to active. We conduct the second experiment to extract the meta-actions and evaluate the derived recommendations.

6.2.2 Experiment I: Mining Action Rules with Customer Status from Leaving to Active

For customers who are lost, the probability of getting them back is much lower than getting the leaving customers back. Therefore, our target is to find actionable patterns that can get leaving customers back. In the Experiment I, we focus on mining action rules that can reclassify customers from leaving to active. We set aspect-reducts as the flexible attributes. Examples of discovered action rules for customer status changing from leaving to active are shown in Table 6.3.

Table 6.3: Example of action rules that reclassify customers from leaving to active

Action rules	**Confidence**	**Support**
AR1 $Survey(Field)$: $[(Communication, 0 \rightarrow 0.4)$ $\wedge(Technician, 0 \rightarrow 0.5)]$ $\Rightarrow (Status, Leaving \rightarrow Active)$	100%	52
AR2 $Survey(Field)$: $[(Price, -0.3 \rightarrow 0)$ $\wedge(Technician, 0 \rightarrow 0.4)]$ $\Rightarrow (Status, Leaving \rightarrow Active)$	100%	47
AR3 $Survey(Field)$: $[(Parts, 0 \rightarrow 0.4)$ $\wedge(Repair, 0.1 \rightarrow 0.3)]$ $\Rightarrow (Status, Leaving \rightarrow Active)$	92%	134
AR4 $Survey(Field)$: $(Technician, 0 \rightarrow 0.2)$ $\wedge(Parts, 0 \rightarrow 0.4)]$ $\Rightarrow (Status, Leaving \rightarrow Active)$	91%	66
AR5 $Survey(Shop)$: $(Price, -0.4 \rightarrow 0.2)$ $\Rightarrow (Status, Leaving \rightarrow Active)$	88%	128
AR6 $Survey(Rental)$: $[(Communication, 0 \rightarrow 0.6)$ $\wedge(Parts, 0.2 \rightarrow 0.4)]$ $\Rightarrow (Status, Leaving \rightarrow Active)$	88%	79
AR7 $Survey(Parts)$: $(Repair, -0.2 \rightarrow 0)$ $\Rightarrow (Status, Leaving \rightarrow Active)$	87%	1938

Let us comment on rule AR1. The survey type is field, which tells the service is given on field (technician was sent to the customers' for heavy equipment repairing). The action rule says that when customers' opinions towards communication change from 0 to 0.4 and towards technician change from 0 to 0.5, then the customer status is expected to change from leaving to active. For AR2, the survey type is field. This action rule says that when customers' opinions towards price change from -0.3 to 0 and towards technician change from 0 to 0.4, then the customer status is expected to change from leaving to active.

6.2.3 Experiment II: Meta-actions and Recommendation Evaluation

As the "Aspect-Sentiment Table" is built on top of text mining, the aspects which are listed as the column names are essentially meta-actions. Therefore, we can get recommendations directly from the result of action rule mining. From the experiment results, we can see that price, repair, parts, communication and technician are the main concerns of customers in heavy equipment repair and service sector. In table 6.4, we listed the detailed strategies and the associated comments on price, repair, parts, communication and technician. To evaluate the derived recommendations, we use effectiveness defined in Equation (6.1).

$$\eta = \frac{N_{recommend}}{N_{leaving}} \tag{6.1}$$

where, $N_{recommend}$ is the number of leaving customers whose major issues are covered by the derived recommendations. $N_{leaving}$ is the number of all the leaving customers. In our experiment, the η we got is 72%.

Table 6.4: Meta-actions and recommendations

	Detailed Meta-actions	**Customer Comments**
Price	Competitive and reasonable price	*"The prices are too high." "We are not satisfied with the invoice which contained charges for parts that we didn't order." "I feel the bill was very high for what he did."*
Repair	Repair completed timely and correctly	*"The repair was not done correctly and as a result the equipment was destroyed and had to be replaced." "Repair took longer than expected."*
Parts	Good parts availability	*"The parts were not in stock." "Poor parts service because the parts he needed were not available."*
Communication	Proactive and effective communication	*"I was not provided with much communication." "They have poor communication and need to work on it."*
Technician	Knowledgeable and experienced technician	*"Since the technician was unable to diagnose the underlying issue." "Another, or underlying, problem exists and the technician did nothing about it."*
	Technician working in a timely manner.	*"The technician didn't arrive as scheduled." "The tech was late 2 out of 5 days."*

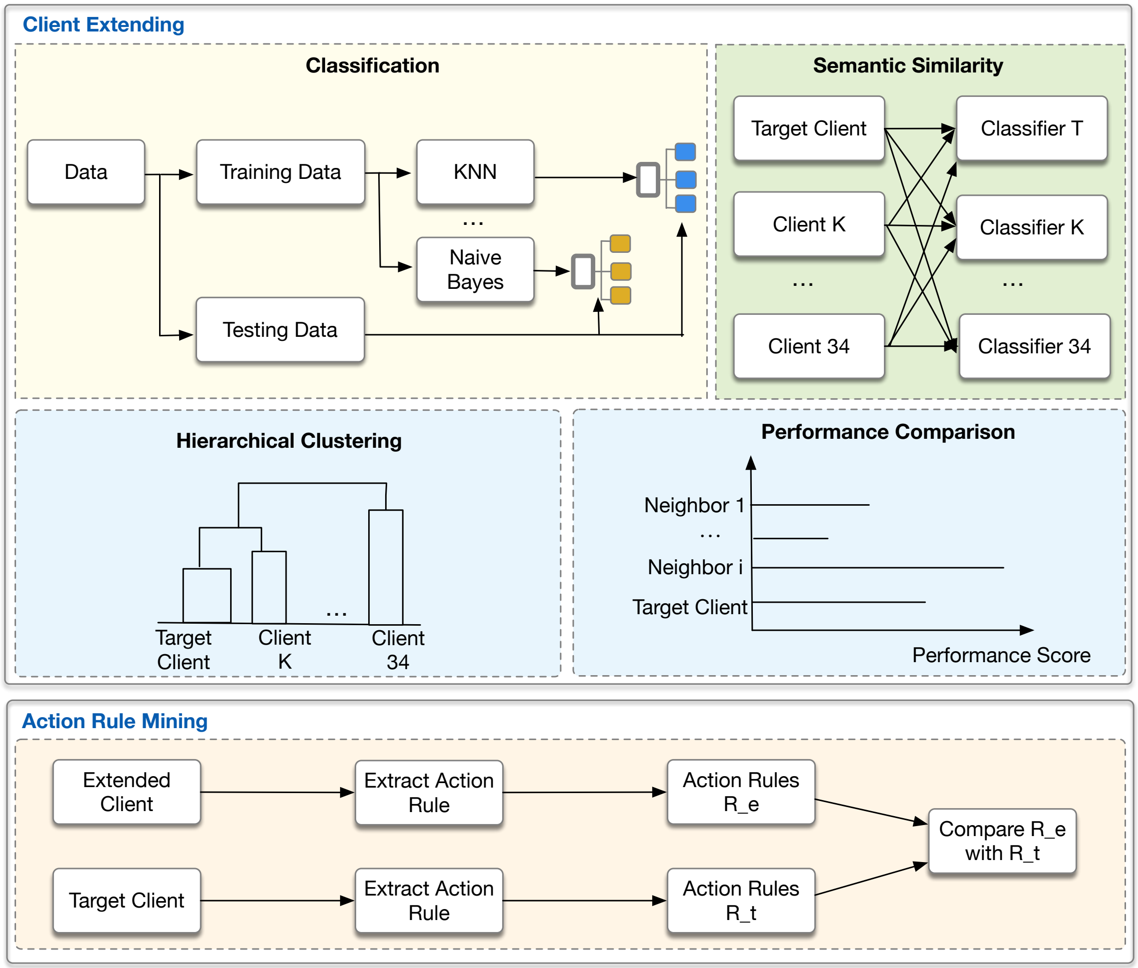
CHAPTER 7: DISCOVERING ACTION RULES FOR MAKING RECOMMENDATIONS TO RETAIN CUSTOMERS

Action rule mining plays an important role in building the Customer Attrition Management System. In this chapter, we propose an algorithm to improve the quality of the target client's discovered action rules by utilizing the other semantic similar but better performing clients. Support, confidence, and coverage are used to measure the quality of the discovered action rules. In practical applications, the action rules are regarded as interesting only if their support and confidence exceed the predefined threshold values. If an action rule has a large support and high confidence, it indicates that this action can be applied to a large portion of customers with a high chance [41]. Little research was done on improving the quality of the discovered action rules. In this chapter, we propose to apply a guided semantic-aided agglomerative clustering algorithm to improve the quality of discovered action rules. The idea is to pick up only such clients which are doing better in business than the given client. By doing that, the given client can follow business recommendations from better performing clients.

7.1 Methodology

The guided semantic-aided agglomerative clustering algorithm comprises two modules - the client extending module and the action rule extraction module. The client extending module is made up of four synergistic components : classification, semantic similarity, hierarchical clustering, and performance comparison. The system design is shown in 7.1.

Figure 7.1: Design of the semantic-aided agglomerative clustering algorithm



7.1.1 Client Extending

A client is a target client if it needs to reduce its customer churn. In the client extending task, the appropriate clients are selected to extend the target client dataset to an extended dataset. In the context of customer churn study, two clients are seen as semantically similar if they agree (to a certain level) on the knowledge concerning active, leaving and lost customers. Therefore, the accuracy of the classifier (that used to classify customers to active, leaving and lost) is utilized to calculate the semantic similarity score between two clients. Moreover, based on the fact that higher accuracy of a classifier leads to higher quality of action rules, some widely used classification algorithms are compared in order to find the best one.

7.1.1.1 Classification

Without loss of generality, Random Forest, K-Nearest Neighbor (KNN), Naive Bayes, Support Vector Machines (SVM) and Logistic Regression are picked for classification performance comparison [9], and the classifier which gives the best performance is picked.

We extract 10 random samplings of the same size from the whole dataset, covering all customers of the 34 clients. There are 1,000 instances in each sampling. Then, these 10 samplings are split into training and testing datasets and processed using selected classification algorithms. Table 7.1 gives the accuracy, precision and F1 among these five algorithms. Among these classification algorithms, Random Forest shows an overall the best performance. Therefore, the chosen classifier is Random Forest, since it is the most accurate.

Table 7.1: The accuracy, precision and F1 of the five algorithms

	Accuracy	**Precision**	**F1**
Logistic Regression	0.883	0.872	0.875
SVM	0.838	0.773	0.793
Naive Bayes	0.787	0.788	0.784
KNN	0.872	0.864	0.861
Ransom Forest	0.917	0.916	0.915

7.1.1.2 Semantic Similarity

Once the best classifier is recognized, the accuracy of the best classifier is utilized to calculate the semantic similarity between two clients. Assume that a and b are the best classifiers for client i and client j respectively, then the semantic similarity score $S(i,j)$ is:

$$S(i,j) = \frac{\left(\left|Accuracy_{a\_i} - Accuracy_{a\_j}\right| + \left|Accuracy_{b\_j} - Accuracy_{b\_i}\right|\right)}{2} \qquad (7.1)$$

where $Accuracy_{a\_i}$ is the accuracy of applying the classifier a to the client i and $Accuracy_{a\_j}$ is the accuracy of applying the classifier a to the client j. Similarly, $Accuracy_{b\_j}$ is the accuracy of applying the classifier b to the client j and $Accuracy_{b\_i}$ is the accuracy of applying the classifier b to the client i.

For all clients, each one of them has their semantic similarity score computed with other clients. Based on the semantic similarity score between each pair of the clients, a semantic similarity distance-based matrix is built: Let $R \in \mathbb{R}^{N \times N}$ be a client-client matrix, N is the number of clients. R_ij represents the semantic similarity score $S(i, j)$ between client i and client j. In Table 7.2, we give the semantic similarity distance-based matrix of the first five clients. For instance, we can see that the semantic similarity distance between client 1 and client 2 is 0.036458956.

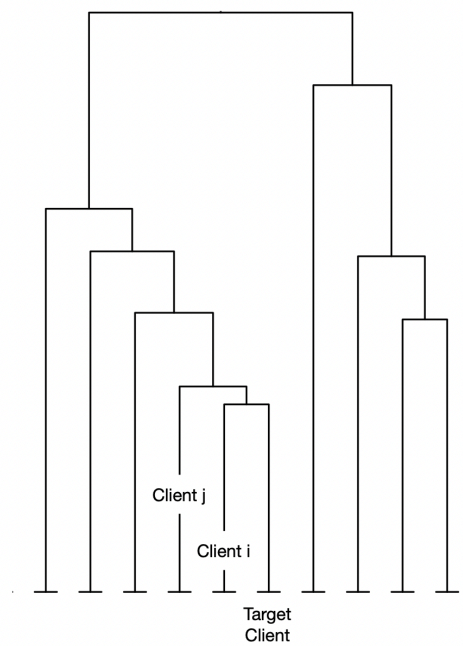
Table 7.2: Semantic similarity distance-based matrix (Client 1-5)

	Client1	Client2	Client3	Client4	Client5
Client1	0				
Client2	0.036458956	0			
Client3	0.064336716	0.053786974	0		
Client4	0.036769177	0.019516244	0.050204144	0	
Client5	0.039030461	0.036457999	0.053123487	0.031540564	0

7.1.1.3 Hierarchical Clustering

Based on the semantic similarity distance-based matrix, we run hierarchical clustering to build a dendrogram, which is used to reflect the semantic similarity relationship among all the clients. According to the semantic similarity relationship in the dendrogram, we locate the nearest neighbor clients of the target client. Figure 7.2 gives an example dendrogram, in which each leaf represents one of the clients. When the difference between two leaves is smaller, then the corresponding clients share more semantic similarity.

Figure 7.2: An example dendrogram



7.1.1.4 Performance Comparison

Once the nearest neighbors of the target client are identified, then their performances are evaluated. The performance score $PeSc$ of a client is defined as:

$$PeSc = \frac{N[active] - N[leaving]}{N[all]} \tag{7.2}$$

where $N[active]$ is the number of active customers, and $N[leaving]$ is the number of leaving customers and $N[all]$ is all customers, no matter active, leaving or lost. If the performance score of the nearest neighbor client is higher than the target client, then this neighbor client is used for extending the target client. Otherwise, the next level client in the dendrogram is checked. For instance, in the dendrogram example (as shown in Figure 7.2), the performance score of client i is first evaluated, which is the nearest neighbor for the target client in the dendrogram. If its performance score is lower than or equal to the target client, then we go upper-level to check the next neighbor, which is client j.

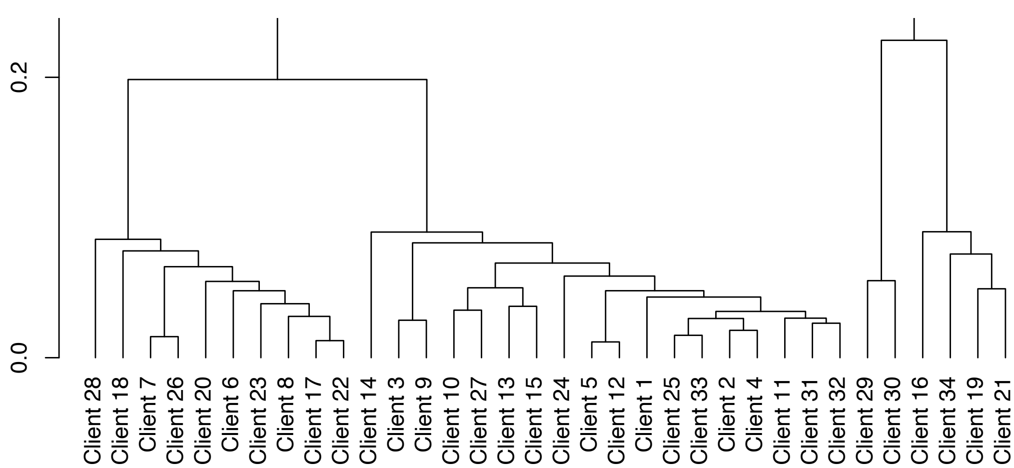
7.1.2 Action Rule Mining

We extend the dataset of a target client by adding datasets of semantically similar but better performing clients. Then we extract action rules from this extended dataset. As introduced in 3.1, action rule is defined as an expression: $[(\omega) \wedge (\alpha \rightarrow \beta) \rightarrow (\Phi \rightarrow \Psi)]$, where $(\Phi \rightarrow \Psi)$ is a desired change of decision attribute value. To reduce customer churn, we define churn status (active, leaving and lost) as the decision attribute value, and the desired change is to reclassify customers from leaving to active.

7.2 Evaluation

We conducted the customer churn study based on the dataset from all the 34 companies, and we focus on Client 22 (the target client) that is deeply suffering from the customer churn issue. In our experiment, we compare the confidence of action rules extracted from the extended dataset with the confidence of action rules discovered from the target dataset. Following the client extending method, we build the dendrogram among these 34 clients, as shown in Figure 7.3.

Figure 7.3: Clients Dendrogram



Client 22 is the target client, and we can see that the nearest neighbors of client

22 are client 17 and client 8, which means client 22 shares more semantic similarity with client 17 and client 8 when compared with other clients. We then check the performance score of client 17 and client 8 based on Equation 7.2, these two clients have a higher performance score than the target client. Thus, we select client 17 and client 8 for extending the target client. Therefore, the extended dataset contains dataset from client 22, client 17 and client 8. The example action rules extracted are shown in Table 7.3. $Conf\_Tar$ stands for the confidence of action rules extracted from the target dataset (Client 22). Similarly, $Conf\_Ext$ stands for the confidence of action rules extracted from the extended dataset. Take the first action rule, for example, it has confidence 92.5% in the target dataset; while the same action rule has confidence 95.6% in the extended dataset.

Table 7.3: Example of action rules' confidence comparison

Action rules	**Conf_Tar**	**Conf_Ext**
SurveyType (Field): (All_Dealer_Communication, 9 → 10) ∧ (All_Invoice_Accuracy, 9 → 10) ∧ ⇒ (Customer Status, Leaving → Active)	92.5%	95.6%
(All_Dealer_Communication, 9 → 10) ∧ (All_Overall_Satisfaction, 8 → 10) ∧ (Service_Tech_Equipped_to_do_Job, 9 → 10) ⇒ (Customer Status, Leaving → Active)	92.3%	92.8%
SurveyType (Field): (All_Dealer_Communication, 9 → 10) ∧ (Service_Tech_Arrived_When_Promised, 8 → 10) ⇒ (Customer Status, Leaving → Active)	92.1%	94.6%
SurveyType (Field): (All_Dealer_Communication, 9 → 10) ∧ (Service_Tech_Equipped_to_do_Job, 9 → 10) ∧ ⇒ (Customer Status, Leaving → Active)	92%	93.9%
SurveyType (Field): (All_Invoice_Accuracy, 9 → 10) ∧ (All_Likelihood_to_be_Repeat_Customer, 8 → 9) ∧ (All_Overall_Satisfaction, 9 → 10) ∧ ⇒ (Customer Status, Leaving → Active)	91.8%	93.1%

We use R_target, $R\_extended$ to represent all action rules extracted from the target dataset and the extended dataset, respectively. Then the confidence of the two action rule sets towards the target dataset is compared.

Now, we give definition of the confidence of an action rule set towards a dataset. Let R be the action rule set extracted from a dataset D and there are M action rules in total. Let r_i denote one of the action rules which belongs to action rule set R, while $Conf\_r_i$ denotes the confidence of r_i, and N_i denotes the supporting set of r_i (there is no overlapping between sets N_i and N_j). We define the confidence of action rule set R towards dataset D as

$$Confidence_{R\_D} = \frac{\sum_{i=1}^{M}(Conf\_r_i * card(N_i))}{\sum_{i=1}^{M} card(N_i)} \tag{7.3}$$

In our experiment, the confidence of action rule set R_target towards the target dataset is 0.807. The confidence of action rule set $R\_extended$ towards the target dataset is 0.892. Given the fact that the action rule set $R\_extended$ is generated using our proposed algorithm and has 10.5% improvement for the confidence as compared to the original method, we conclude that the proposed algorithm can effectively improve the quality of the action rules towards the target dataset.

CHAPTER 8: CONCLUSIONS

Customer churn is a major issue to most companies, and recommender systems that are utilizing the action rule mining technology show great value in the application of reducing customer churn. In this dissertation, we first presented a two-stage Customer Attrition Management System, which can not only detect customer attrition proactively by utilizing churn prediction model, but also provide actionable strategies for decision makers to address the customer churn issue. We then propose two methodologies to further enhance the efficiency of the Customer Attrition Management System.

Firstly, reviews from the leaving customers reflect their unfulfilled needs, and reviews of the active customers reflect their satisfactory experience. Thus, customer reviews contain valuable information about the customers' opinions and attitudes. In order to better catch the customers' opinions, we use aspect-based sentiment analysis to transform the unstructured review text data into a structured "Aspect-Sentiment Table". In the newly created "Aspect-Sentiment Table", the aspects extracted from the review text are the attributes, and the associated opinion scores are the values of the attributes. Statistical method and reduct are applied for feature engineering. Then, action rule mining and meta-action triggering mechanism are used to extract the actionable recommendations. The action rules aim to reclassify customers from leaving to active, and the detailed meta-actions are recommendations that can effectively reduce customer churn.

Secondly, in action rule mining, confidence, support and coverage are used to measure the quality of the discovered action rules. In reality, action rules with higher confidence and support are more useful to users. However, there is little research focused on improving the quality of the action rules. We propose a semantic-aided agglomerative clustering algorithm by utilizing the knowledge extracted from semantically similar clients to the given client. The idea is to pick up only such clients which are doing better in business than the given client. By doing that, the given client can follow business recommendations from better-performing clients. To implement the semantic-aided agglomerative clustering algorithm, some widely used customer churn classification modules are compared and the overall best classifier is recognized. Then the semantic similarity-based distance matrix between two clients is calculated by utilizing the precision of the best classifier. Next, a hierarchically structured dendrogram is built based on the semantic similarity-based distance matrix. The nearest neighbors of the target client are chosen to be the candidate clients for extending the dataset of the target client. Then the performance score of the candidate clients and the target client are compared. Those clients whose performance scores are higher than the target client are used for extending it. Next, action rules are extracted from the target dataset as well as the extended dataset. Then the coverage of the two action rule sets is compared based on the confidence of the action rule set toward the target dataset.

REFERENCES

[1] S. Renjith, "B2c e-commerce customer churn management: Churn detection using support vector machine and personalized retention using hybrid recommendations," *International Journal on Future Revolution in Computer Science & Communication Engineering (IJFRCSCE)*, vol. 3, no. 11, pp. 34–39, 2017.

[2] F. F. Reichheld and W. E. Sasser, "Zero defeofions: Quoliiy comes to services," *Harvard business review*, vol. 68, no. 5, pp. 105–111, 1990.

[3] P. E. Pfeifer, "The optimal ratio of acquisition and retention costs," *Journal of Targeting, Measurement and Analysis for Marketing*, vol. 13, no. 2, pp. 179–188, 2005.

[4] J. D. Farquhar and T. Panther, "Acquiring and retaining customers in uk banks: An exploratory study," *Journal of Retailing and Consumer Services*, vol. 15, no. 1, pp. 9–21, 2008.

[5] W. Kowalczyk and F. Slisser, "Modelling customer retention with rough data models," in *European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 4–13, Springer, 1997.

[6] D. Van den Poel and B. Lariviere, "Customer attrition analysis for financial services using proportional hazard models," *European journal of operational research*, vol. 157, no. 1, pp. 196–217, 2004.

[7] J. Griffin, *Customer loyalty: How to earn it, how to keep it.* Jossey-Bass San Francisco, CA, 2002.

[8] A. K. Ahmad, A. Jafar, and K. Aljoumaa, "Customer churn prediction in telecom using machine learning in big data platform," *Journal of Big Data*, vol. 6, no. 1, pp. 1–24, 2019.

[9] D. L. García, À. Nebot, and A. Vellido, "Intelligent data analysis approaches to churn as a business problem: a survey," *Knowledge and Information Systems*, vol. 51, no. 3, pp. 719–774, 2017.

[10] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The journal of educational research*, vol. 96, no. 1, pp. 3–14, 2002.

[11] S. Jun Lee and K. Siau, "A review of data mining techniques," *Industrial Management & Data Systems*, vol. 101, no. 1, pp. 41–46, 2001.

[12] S. Haykin, *Neural networks: a comprehensive foundation.* Prentice Hall PTR, 1994.

[13] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[14] T. Chen, T. He, M. Benesty, *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[15] G. H. Van Bruggen and B. Wierenga, *Marketing decision making and decision support: Challenges and perspectives for successful marketing management support systems.* Now Publishers Inc, 2010.

[16] T. Coltman, T. M. Devinney, and D. F. Midgley, "Customer relationship management and firm performance," *Journal of Information Technology*, vol. 26, no. 3, pp. 205–219, 2011.

[17] I. J. Chen and K. Popovich, "Understanding customer relationship management (crm)," *Business process management journal*, 2003.

[18] R. H. Sprague Jr and E. D. Carlson, *Building effective decision support systems.* Prentice Hall Professional Technical Reference, 1982.

[19] A. Felfernig, S. Polat-Erdeniz, C. Uran, S. Reiterer, M. Atas, T. N. T. Tran, P. Azzoni, C. Kiraly, and K. Dolui, "An overview of recommender systems in the internet of things," *Journal of Intelligent Information Systems*, vol. 52, no. 2, pp. 285–309, 2019.

[20] L. Guo, J. Liang, Y. Zhu, Y. Luo, L. Sun, and X. Zheng, "Collaborative filtering recommendation based on trust and emotion," *Journal of Intelligent Information Systems*, vol. 53, no. 1, pp. 113–135, 2019.

[21] P. Datta, B. Masand, D. Mani, and B. Li, "Automated cellular modeling and prediction on a large scale," *Artificial Intelligence Review*, vol. 14, no. 6, pp. 485–502, 2000.

[22] H.-S. Kim and C.-H. Yoon, "Determinants of subscriber churn and customer loyalty in the korean mobile telephony market," *Telecommunications policy*, vol. 28, no. 9-10, pp. 751–765, 2004.

[23] G. Wang, L. Liu, Y. Peng, G. Nie, G. Kou, and Y. Shi, "Predicting credit card holder churn in banks of china using data mining and mcdm," in *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, vol. 3, pp. 215–218, IEEE, 2010.

[24] B. Huang, M. T. Kechadi, and B. Buckley, "Customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1414–1425, 2012.

[25] Y. Li, J. Wei, K. Kang, and Z. Wu, "An efficient noise-filtered ensemble model for customer churn analysis in aviation industry," *Journal of Intelligent & Fuzzy Systems*, vol. 37, no. 2, pp. 2575–2585, 2019.

[26] A. De Caigny, K. Coussement, K. W. De Bock, and S. Lessmann, "Incorporating textual information in customer churn prediction models based on a convolutional neural network," *International Journal of Forecasting*, vol. 36, no. 4, pp. 1563–1578, 2020.

[27] T. J. Gerpott, W. Rams, and A. Schindler, "Customer retention, loyalty, and satisfaction in the german mobile cellular telecommunications market," *Telecommunications policy*, vol. 25, no. 4, pp. 249–269, 2001.

[28] B. Larivière and D. Van den Poel, "Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services," *Expert Systems with Applications*, vol. 27, no. 2, pp. 277–285, 2004.

[29] N. Varshney and S. Gupta, "Mining churning factors in indian telecommunication sector using social media analytics," in *International Conference on Data Warehousing and Knowledge Discovery*, pp. 405–413, Springer, 2014.

[30] C. A. M. Troncoso, *Predicting Customer Churn using Voice of the Customer. A Text Mining Approach.* PhD thesis, The University of Manchester (United Kingdom), 2019.

[31] A. COŞER, A. Aldea, M. M. Maer-Matei, and L. BEŞIR, "Propensity to churn in banking: What makes customers close the relationship with a bank?," *Economic Computation & Economic Cybernetics Studies & Research*, vol. 54, no. 2, 2020.

[32] S. Daskalaki, I. Kopanas, M. Goudara, and N. Avouris, "Data mining for decision support on customer insolvency in telecommunications business," *European Journal of Operational Research*, vol. 145, no. 2, pp. 239–255, 2003.

[33] M.-K. Kim, M.-C. Park, and D.-H. Jeong, "The effects of customer satisfaction and switching barrier on customer loyalty in korean mobile telecommunication services," *Telecommunications policy*, vol. 28, no. 2, pp. 145–159, 2004.

[34] J. Burez and D. Van den Poel, "Crm at a pay-tv company: Using analytical models to reduce customer attrition by targeted marketing for subscription services," *Expert Systems with Applications*, vol. 32, no. 2, pp. 277–288, 2007.

[35] Y.-F. Wang, D.-A. Chiang, M.-H. Hsu, C.-J. Lin, and I.-L. Lin, "A recommender system to avoid customer churn: A case study," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8071–8075, 2009.

[36] A. Lemmens and S. Gupta, "Managing churn to maximize profits," *Marketing Science, Forthcoming*, 2020.

[37] Z. Pawlak, "Information systems theoretical foundations," *Information systems*, vol. 6, no. 3, pp. 205–218, 1981.

[38] Z. W. Ras, E. Wyrzykowska, and L.-S. Tsay, “Action rules mining,” in *Encyclopedia of Data Warehousing and Mining, Second Edition*, pp. 1–5, IGI Global, 2009.

[39] Z. Ras and A. Wieczorkowska, “Action-rules: How to increase profit of a company,” in *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 587–592, Springer, 2000.

[40] A. A. Tzacheva and Z. W. Ras, “Association action rules and action paths triggered by meta-actions,” in *2010 IEEE International Conference on Granular Computing*, pp. 772–776, IEEE, 2010.

[41] Z. W. Ras and L.-S. Tsay, “Discovering extended action-rules (system dear),” in *Intelligent Information Processing and Web Mining*, pp. 293–300, Springer, 2003.

[42] L.-S. Tsay and Z. Ras, “Action rules discovery: system dear2, method and experiments,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 17, no. 1-2, pp. 119–128, 2005.

[43] S. Im and Z. Ras, “Action rule extraction from a decision table: Ared,” in *Foundations of Intelligent Systems, LNAI, Vol. 4994*, pp. 160–168, Springer, 2008.

[44] K. Wang, Y. Jiang, and A. Tuzhilin, “Mining actionable patterns by role models,” in *null*, p. 16, IEEE, 2006.

[45] H. A. Wasyluk and Z. W. Raś, “Action rules approach to solving diagnostic problems in clinical medicine,” *G. Devlin, Decision Support Systems Advances in*, pp. 99–106, 2010.

[46] H. Touati, Z. W. Raś, and J. Studnicki, “Meta-actions as a tool for action rules evaluation,” in *Feature Selection for Data and Pattern Recognition, Studies in Computational Intelligence, Chapter 9, Vol. 584*, pp. 177–197, Springer, 2015.

[47] J. Kuang and Z. Ras, “In search for best meta-actions to boost businesses revenue,” in *Flexible Query Answering Systems, Advances in Intelligent Systems and Computing, Vol. 400*, pp. 431–443, Springer, 2015.

[48] Z. Pawlak, “Rough sets,” *International journal of computer & information sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[49] D. Chen, L. Zhang, S. Zhao, Q. Hu, and P. Zhu, “A novel algorithm for finding reducts with fuzzy rough sets,” *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 385–389, 2011.

[50] A. Skowron and C. Rauszer, “The discernibility matrices and functions in information systems,” in *Intelligent decision support*, pp. 331–362, Springer, 1992.

[51] Y. Zhao, Y. Yao, and F. Luo, “Data analysis based on discernibility and indiscernibility,” *Information Sciences*, vol. 177, no. 22, pp. 4959–4976, 2007.

[52] J. G. Bazan, A. Skowron, and P. Synak, "Dynamic reducts as a tool for extracting laws from decisions tables," in *International Symposium on Methodologies for Intelligent Systems*, pp. 346–355, Springer, 1994.

[53] J. Komorowski and A. Øhrn, "Modelling prognostic power of cardiac tests using rough sets," *Artificial intelligence in medicine*, vol. 15, no. 2, pp. 167–191, 1999.

[54] A. Clark and I. Tim, "Pre-processing very noisy text," in *Proc. of Workshop on Shallow Processing of Large Corpora*, pp. 12–22, 2003.

[55] P. Patil and P. Yalagi, "Sentiment analysis levels and techniques: A survey," *space*, vol. 1, p. 6.

[56] B. Wang and M. Liu, "Deep learning for aspect-based sentiment analysis," *Stanford University report*, 2015.

[57] T. A. Rana and Y.-N. Cheah, "Aspect extraction in sentiment analysis: comparative analysis and survey," *Artificial Intelligence Review*, vol. 46, no. 4, pp. 459–483, 2016.

[58] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, ACM, 2004.

[59] A. Bagheri, M. Saraee, and F. de Jong, "An unsupervised aspect detection model for sentiment analysis of reviews," in *International Conference on Application of Natural Language to Information Systems*, pp. 140–151, Springer, 2013.

[60] J. Zhu, H. Wang, M. Zhu, B. K. Tsou, and M. Ma, "Aspect-based opinion polling from customer reviews," *IEEE Transactions on Affective Computing*, vol. 2, no. 1, pp. 37–49, 2011.

[61] A.-M. Popescu and O. Etzioni, "Extracting product features and opinions from reviews," in *Natural language processing and text mining*, pp. 9–28, Springer, 2007.

[62] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1533–1541, Association for Computational Linguistics, 2009.

[63] J. Yu, Z.-J. Zha, M. Wang, and T.-S. Chua, "Aspect ranking: identifying important product aspects from online consumer reviews," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 1496–1505, Association for Computational Linguistics, 2011.

[64] C.-P. Wei, Y.-M. Chen, C.-S. Yang, and C. C. Yang, "Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews," *Information Systems and E-Business Management*, vol. 8, no. 2, pp. 149–167, 2010.

[65] F. L. Cruz, J. A. Troyano, F. Enríquez, F. J. Ortega, and C. G. Vallejo, "'long autonomy or long delay?'the importance of domain in opinion mining," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3174–3184, 2013.

[66] P. Jiang, C. Zhang, H. Fu, Z. Niu, and Q. Yang, "An approach based on tree kernels for opinion mining of online product reviews," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pp. 256–265, IEEE, 2010.

[67] S. Li, R. Wang, and G. Zhou, "Opinion target extraction using a shallow semantic parsing framework," in *Twenty-sixth AAAI conference on artificial intelligence*, 2012.

[68] F. Hemmatian and M. K. Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis," *Artificial Intelligence Review*, pp. 1–51, 2017.

[69] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[70] C. Alfaro, J. Cano-Montero, J. Gómez, J. M. Moguerza, and F. Ortega, "A multistage method for content classification and opinion mining on weblog comments," *Annals of Operations Research*, vol. 236, no. 1, pp. 197–213, 2016.

[71] M. S. Hajmohammadi, R. Ibrahim, and A. Selamat, "Cross-lingual sentiment classification using multiple source languages in multi-view semi-supervised learning," *Engineering Applications of Artificial Intelligence*, vol. 36, pp. 195–203, 2014.

[72] N. Claypo and S. Jaiyen, "Opinion mining for thai restaurant reviews using k-means clustering and mrf feature selection," in *Knowledge and Smart Technology (KST), 2015 7th International Conference on*, pp. 105–108, IEEE, 2015.

[73] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[74] A. De and S. K. Kopparapu, "Unsupervised clustering technique to harness ideas from an ideas portal," in *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pp. 1563–1568, IEEE, 2013.

[75] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, “Lexicon-based methods for sentiment analysis,” *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[76] B. Liu, “Sentiment analysis and opinion mining,” *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[77] G. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[78] S. Baccianella, A. Esuli, and F. Sebastiani, “Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining.,” in *Lrec*, vol. 10, pp. 2200–2204, 2010.

[79] J. C. de Albornoz, L. Plaza, and P. Gervás, “Sentisense: An easily scalable concept-based affective lexicon for sentiment analysis.,” in *LREC*, pp. 3562–3567, 2012.

[80] F. Å. Nielsen, “A new anew: Evaluation of a word list for sentiment analysis in microblogs,” *arXiv preprint arXiv:1103.2903*, 2011.

[81] S. Neslin, S. Gupta, W. Kamakura, J. Lu, and C. Mason, “Defection detection: Improving predictive accuracy of customer churn models,” *Tuck School of Business, Dartmouth College*, 2004.

[82] O. Celik and U. O. Osmanoglu, “Comparing to techniques used in customer churn analysis,” *Journal of Multidisciplinary Developments*, vol. 4, no. 1, pp. 30–38, 2019.

[83] J. Kuang, Z. W. Raś, and A. Daniel, “Personalized meta-action mining for nps improvement,” in *International Symposium on Methodologies for Intelligent Systems, LNAI, Vol. 9384*, pp. 79–87, Springer, 2015.

[84] C. J. Hutto and E. Gilbert, “Vader: A parsimonious rule-based model for sentiment analysis of social media text,” in *Eighth international AAAI conference on weblogs and social media*, 2014.

[85] M.-C. De Marneffe and C. D. Manning, “Stanford typed dependencies manual,” tech. rep., Technical report, Stanford University, 2008.

[86] D. Chen and C. D. Manning, “A fast and accurate dependency parser using neural networks,” in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 740–750, 2014.

[87] K. Tarnowska, Z. Ras, and L. Daniel, *Recommender System for Improving Customer Loyalty*. Studies in Big Data, Vol. 55, Springer, 2020.

[88] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[89] M. Simunek, "Academic kdd project lisp-miner," in *Intelligent Systems Design and Applications*, pp. 263–272, Springer, 2003.

[90] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[91] W. Bancken, D. Alfarone, and J. Davis, "Automatically detecting and rating product aspects from textual customer reviews," in *Proceedings of the 1st international workshop on interactions between data mining and natural language processing at ECML/PKDD*, vol. 1202, pp. 1–16, 2014.

[92] C. Gilbert and E. Hutto, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf*, vol. 81, p. 82, 2014.