A Data Mining Framework for Automatic Online Customer Lead Generation

by

Md. Abdur Rahman

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at
Dalhousie University
Halifax, Nova Scotia
March 2012

DALHOUSIE UNIVERSITY

FACULTY OF COMPUTER SCIENCE

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled "A Data Mining Framework for Automatic Online Customer Lead Generation" by Md. Abdur Rahman in partial fulfillment of the requirements for the degree of Master of Computer Science.

Dated: March 23, 2012

Supervisors: _____

_____

Reader: _____

DALHOUSIE UNIVERSITY

DATE: March 23, 2012

AUTHOR: Md. Abdur Rahman

TITLE: A Data Mining Framework for Automatic Online Customer Lead Generation

DEPARTMENT OR SCHOOL: Faculty of Computer Science

DEGREE: MCSc CONVOCATION: May YEAR: 2012

_____

Signature of Author

Table of Contents

List of Tables

List of Figures

Abstract

Customer lead generation is a crucial and challenging task for online real estate service providers. The business model of online real estate service differs from typical B2B or B2C e-commerce because it acts like a broker between the real estate companies and the potential home buyers. Currently, there is no suitable automatic customer lead generation system available for online real estate service providers. This thesis aims at developing a systematic solution framework of automatic customer lead generation for online real estate service providers. This framework includes data modeling, data integration from multiple online web data streams, as well as data mining and system evaluation for lead pattern discovery and lead prediction. Extensive experiments were conducted based on a case study. The results demonstrate that the proposed approach is able to empower online real estate service providers for lead data analysis and automatically generate targeting customer leads.

List of Abbreviations and Symbols Used

B2B	Business 2 Business
B2C	Business 2 Customer
FP	Frequent Pattern
SaM	Split and Merge
DB	Database
ETAP	Electronic Trigger Alert Program
TidList	Transactions id list
IP	Internet Protocol
LP	Landing Page
JDBC	Java Database Connectivity Driver
AD	Advertisement

Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. Hai Wang and Dr. Qigang Gao for their continual support, encouragements and patience. It would not be possible for me to complete my work without their valuable suggestion and proper guidance.

I also would like to thank Dr. Vlado Keselj for reviewing the thesis and giving me valuable suggestions.

Finally, I would like to extend my greatest gratitude to my parents, Younus Ali and Safaly Begum for their endless encouragement and support.

Chapter 1 **Introduction**

1.1 Background and Problem Statement

Customer leads generation is a crucial and challenging task for online businesses to attract customers and improve their services. A *customer lead*, also referred to as a *lead*, is defined as a potential sales contact, i.e., a potential customer who expresses an interest in the company's products or services [4]. Currently, the leads are produced by a "brute-force" method which requires tedious manual effort. There is no suitable automatic leads-generation system readily available which could empower the service provider to quickly and easily capture targeted leads.

A recent study shows that almost 80% of home buyers start their home search online [38]. Most users visit an online real estate service provider's website through search engines or third-party websites where the provider's advertisements are posted. While visiting the provider's website, users are exposed to a landing page. A *landing page*, also referred to as a *lead capture page*, is a web page that is displayed in response to clicking on an online advertisement, either on the website of an online real estate service provider or on a third-party website. These landing pages may generate leads for online real estate service providers, as potential home buyers may leave contact information on them. If a user visits the landing page and leaves his/her contact information, the user is then considered a lead. On the other hand, if a user leaves the landing page without submitting contact information, the user is considered a non-lead. To attract users who turn into leads, landing pages are carefully designed regarding text size, style, color, graphics, and so on. They also contain various business offers, rewards and information. Generally, several landing pages are used to attract various user groups with different interests. The biggest challenge is providing the user with the right landing page so that the user can become a lead.

1.2 Current Solutions and Challenges

Currently, there are no automatic lead generation tools available for real estate service providers. One of the main difficulties is the large volume of data which is distributed into several sources. This data needs to be integrated into a single source before conducting any analysis. The erroneous data need to be cleaned out and the remaining data transformed into a suitable format for data mining tasks that generate leads. Online real estate service providers differ from other typical B2B or B2C e-commerce companies in that they act as brokers between real estate companies and potential home buyers.

1.3 Proposed Solution

Our main goal in this thesis is to provide automatic customer lead generation tools which will assist online real estate service providers to quickly and easily capture targeted leads in a timely manner. To achieve this goal, a data mining framework is developed to analyze customer behavior and the effectiveness of the landing page in converting users into leads.

As the first step towards this goal, the current online real estate service provider's business models are studied to establish proper data models, explore possible solution options, and develop a framework for data integration and data mining. This system automatically generates classification and association rules from the integrated data set of the users' web clicks data on a provider's website and other associated ads. These rules can then be interpreted as guidelines for updating a website's various landing pages in order to increase lead conversions.

The proposed system will be based on advanced machine learning and data mining techniques, such as ensemble classification for predicting targets, association rule mining on an integrated click-stream, and quantization for improving the predictability of lead generation.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 presents the background studies

and literature review. Chapter 3 presents the system architecture of the lead generation engine. Chapter 4 describes the data model design and integration based on the activity flow of users who use online services for home buying/selling purposes. Chapter 5 describes the data mining framwork, results and evaluation. Finally, Chapter 6 concludes the thesis and describes possible future work.

Chapter 2 **Literature Review and Background**

2.1 Customer Leads

A *customer lead* or *sales lead* is defined as a person or entity that shows interest in purchasing a product or service from a provider. It is not a direct sale contact, but rather a sales process which usually identifies a potential customer for closing a sale. It is typically generated by a business owner or a lead generation company through marketing campaigns such as trade fairs, internet marketing, online rewards programs, etc. To qualify as a valid customer lead or sales prospect, the lead must fulfill certain conditions defined by the provider and need to be evaluated carefully. If a lead eventually makes a purchase, this is considered a conversion. In general, there are two types of organizations that generate leads. The first type wants to increase sales of their own products and organization, so their target is to sell potential leads to business owners. The second type of organization that generates leads is generic (i.e., a person signs up for a type of offer, instead of a particular provider or brand). In this thesis, the studied business models produce generic leads that are eventually sold to prospective buyers.

Automatic lead generation has been studied in the literature for typical B2B and B2C types of e-commerce. This includes studies on automatic sales lead generation based on web page content for B2B scenarios [28], and studies on lead generation based on online customers' purchasing patterns [24, 42]. An online real estate service provider differs from typical B2B or B2C e-commerce companies because it acts like a broker between the real estate companies and potential home buyers. To our best knowledge, there is no previous research on customer lead generation for online real estate service companies.

2.2 Business Model of Online Real Estate Service Providers

An online real estate service provider acts as a broker between potential home buyers/sellers and real estate companies. It maintains a website to provide various services to home buyers/sellers, and in turn generates customer leads, which it sells to real estate companies. Figure 2.1 describes the general business model of an online real estate service provider:

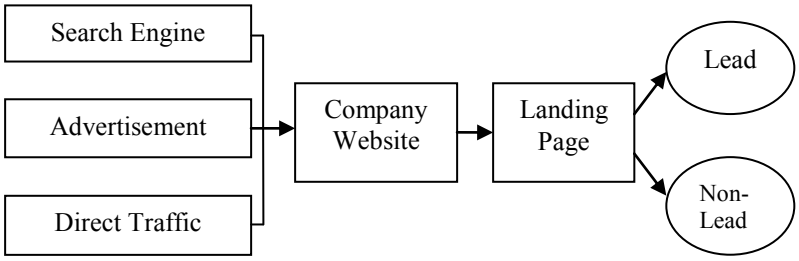


Figure 2.1 General business model for an online real estate service provider.

Most users visit the online real estate service providers' website through search engines or third-party websites where the providers' advertisements are posted. While visiting the providers' website, the users are exposed to a landing page. If a user visits the landing page and leaves his/her contact information, the user is then considered a lead. On the other hand, if a user leaves the landing page without submitting his/her contact information, the user is considered a non-lead.

Real estate service providers strive to achieve their mission of increasing the efficiency of a customer's home buying/selling process. They do so by providing useful neighborhood information, such as businesses (e.g., grocery stores, coffee shops, restaurants, retail stores), childcare services, schools, and local transit information for areas in which the customers are considering buying a property. This information also helps determine the most appropriate neighborhoods for a customer. The current business model of most real estate service providers identifies three types of customers: private potential home buyers/sellers, real estate agents, and advertisers. When it comes to real estate agents, the real estate service providers equip them with online tools such as neighborhood reports, targeted advertising based on search locations, and virtual tours of homes and neighborhoods to help them market homes and network with potential online homebuyers.

One of the current revenue sources for real estate service providers is advertising. These advertisers can be classified into three categories: large housing-related advertisers (e.g.,

banks, home hardware retailers, mortgage brokers), hyper-local advertisers (e.g., Tim Horton's, Subway) and businesses (e.g., dry cleaners, plumbers), and real estate agents. Real estate agents are by far the biggest and most important advertisers on the website. However, the cumulative revenue generated from these three sources is much lower than what could be gained through lead generation, particularly if real estate companies are willing to purchase as many leads as the real estate service providers can produce by predicting whether online homebuyers have an intention to perform a revenue-generating action.

2.3 Data Mining of Web Click-Streams

Web click-stream is the record of a user's activity when he/she clicks, navigates or performs any activity on the website. The user's actions are logged in the web server along with the activity information and the timestamp. The web server logs contain various types of information, such as the user's ip location, browser information, request method (i.e., get, post), access time, user's system information, accessed page information, and so on.

Web click-stream data mining, also known as web usage mining in general, involves automatically extracting and analyzing useful information and patterns in click-stream data. The discovered information can be used to analyze the behavioral patterns and profiles of users interacting with a website [20]. The patterns are usually represented as associations of pages, objects, or resources which are frequently accessed by groups of users with common needs or interests.

In this thesis, web click-stream data is analyzed to create leads and non-leads profiles. Association patterns between pages, resources, user activities and time stamps are considered to analyze the behavior of leads and non-leads. Associations between landing pages and certain lead user groups indicate which landing pages have higher lead conversion rates for which group of users.

In [35], the web usage mining tool "WebSwift" is described. This tool collects web data from three sources: 1. Server Level Collection; 2. Client Level Collection; and 3. Proxy Level Collection. The server level collection is the same as the web server data collection described earlier. The client level collection and proxy level collection both depend on client preference regarding data sharing. After data collection, it goes through three stages of data preprocessing: 1. Usage Preprocessing; 2. Content Preprocessing; and 3. Structure Preprocessing. Finally, data mining algorithms are applied to discover patterns within the datasets. The system can be used for general purpose data usage, including personalization, system improvement, site modification and usage characterization.

In [24], a customer purchasing behavior predictor for online stores is described that uses web click-stream data. The system uses variables from four different categories in predicting online-purchasing behavior: (1) general click-stream behavior; (2) more detailed click-stream information; (3) customer demographics; and (4) historical purchase behavior. The results indicate that click-stream behavior is significant when deciding the trend to buy. The research also indicates that detailed click-stream variables play vital roles in classifying customers according to their online purchase behavior. However, the model is designed to target B2C types of consumer, as it predicts only the purchasing behavior of a customer.

In [10], a model for website browsing behavior analysis using web click-stream data is described. Two major aspects of user behavior are analyzed. The first is the visitor's decision either to continue browsing by submitting an additional page request or to exit the site. The second aspect of user behavior that is analyzed is the length of time spent viewing each page. It models each browsing decision as a function of user and site covariates. The paper also investigated how visit depth (such as duration of page view) and repeat visits affect how the users use the site. These two covariates empower the model to record the changes in browsing behavior that take place within and across site visits. The model also handles some of the limitations of server log data such as identifying unique visitors and their multiple sessions.

In [42], a system is described that uses web mining for customer behavior pattern discovering. The system mainly considers data from web logs and customer-related information from an e-commerce site. It includes a web mining template that combine both web logs and business data to accomplish data-driven decisions that positively influence e-commerce business. The pre-processing of web-click stream data includes filtering, selecting, arranging and often creating new variables with the web logs. The data mining tasks mainly generate customer buying patterns and association rules of visitor traffic among the web pages, and also predict model generation for potential customers. To avoid the information avalanche that is characteristic of wide-ranging web data, a semantic taxonomy method is introduced to group web pages and discovered patterns.

2.4 Survey on Customer Lead Generation

A web-based text mining system for customer lead generation from web data is shown in [31]. The mining system consists of a data acquisition process that extracts textual data from different internet sources, a database server for storing the extracted data, a text mining engine that executes query-based requests by users, and an output repository. The system searches the prospects matching the business owner's criteria from the internet source, then forwards the gathered prospects' contact information to the business owner, with a link to the original document that verifies the prospects' interest. If the business owner accepts the prospective buyers, the system automatically creates personalized sales scripts and direct marketing materials that appeal to the prospects' stated or implied interests. It then verifies inward sales and marketing components by applying data and text mining to generate profiles of the business's most profitable customers. Finally, it cross-references and matches the customer profiles with generated leads to create more efficient marketing and sales presentations.

In [32], a web-based lead generator system is developed for business enterprises that maintain e-commerce web sites. The system is described as an application service system that uses a pre-emptive profiling process to search the Internet to obtain email address of potential buyers. It uses the client-provided criteria to search Internet postings for

prospective buyers who are discussing products or services that are related to the client's business offerings. The major data sources include Internet news groups, web-based discussion forums, email lists, web content of various websites, etc. This data is used to create purchasing profiles of a customer. The profile information and the capture information are used to direct sales or permission-based email opt-in. A general overview of the system is provided in Figure 2.2.

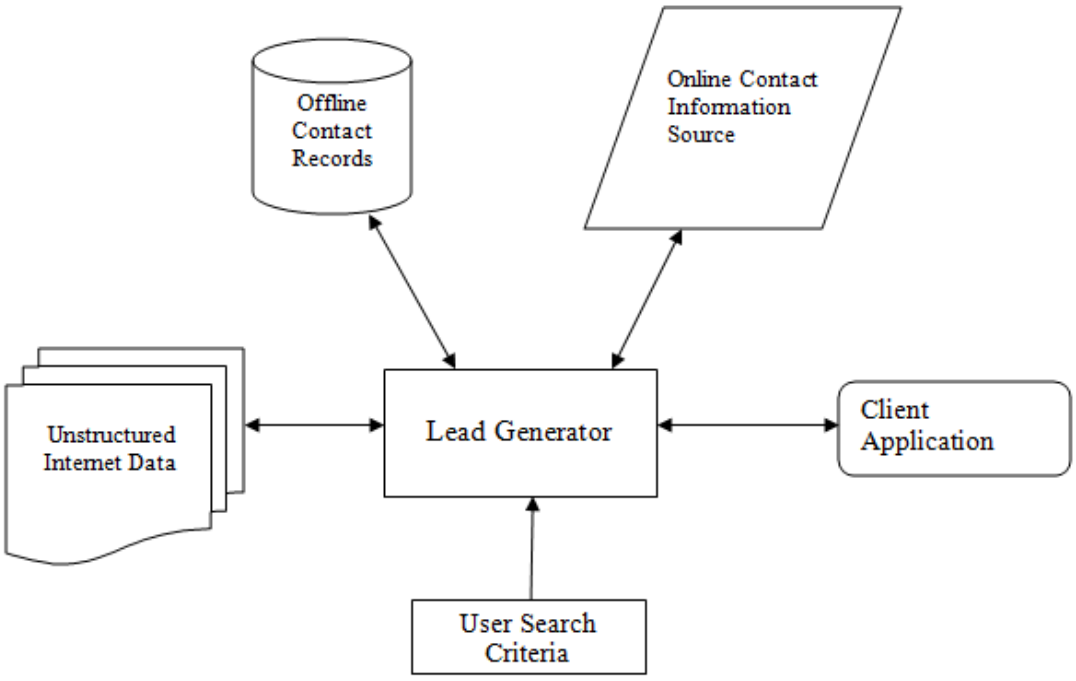


Figure 2.2 Overview of a web-based lead generator system.

In [28], an automatic sales lead generation process from web data is described. The paper presents a system called ETAP (Electronic Trigger Alert Program) that extracts trigger events from web data to discover prospective buyers or leads. Trigger events are defined according to corporate relevance and company's interest to purchase new products associated with these events, such as change in management, revenue growth, and mergers and acquisitions. Unlike other lead generation methods, the system is suited for identification of prospective companies as opposed to the identification of individual buyers, making it suitable for B2B (business to business) scenarios. This system gathers information or identifies trigger events from web pages data/content, instead of analyzing

the user activity while visiting a site. Thus, ETAP assists in prioritizing target companies that need to be approached, rather than targeting a customer. A general overview of the system is provided in Figure 2.3.

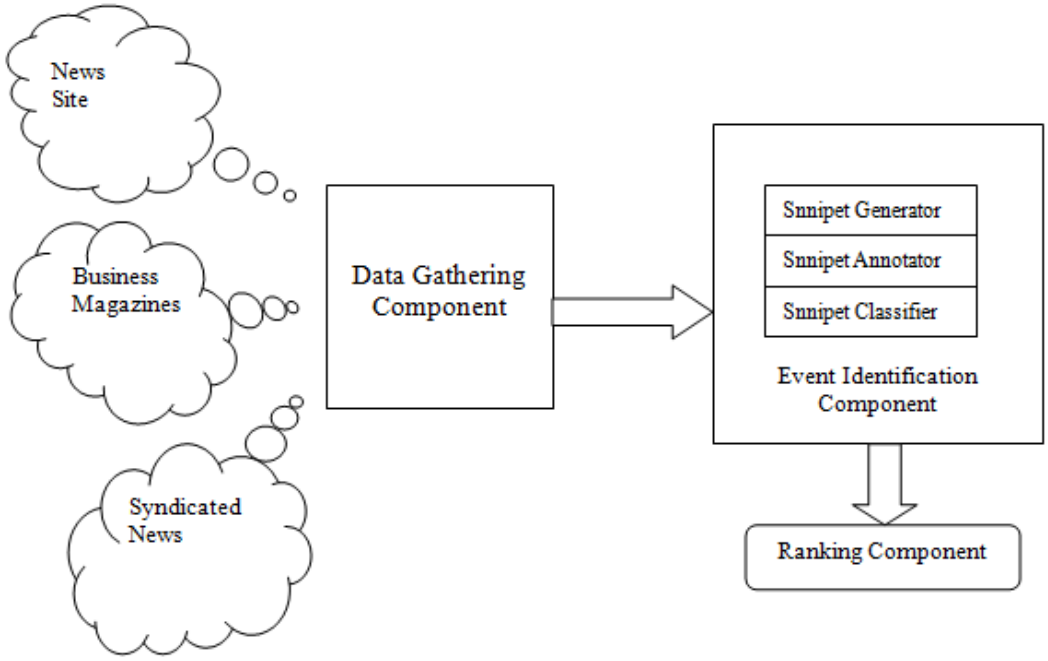


Figure 2.3 Overview of ETAP system.

BuyerZone [49] is an online marketplace for B2B buying. It provides a lead generation system to connect buyers that are searching for particular products or services with various businesses that will be able to satisfy their requirements. Customers provide details about their interests and the purchases that they aim to finalize, and BuyerZone then forwards them to a set of available vendors considered the most appropriate for the customer. From there, the customer and possible vendors can work out a finalized offer, which includes a list of service and other fees. Advantages for customers include price quotes, and shopping options for the best prices as they receive multiple offers from various vendors. In turn, vendors get a source for proved and fresh targeted leads.

There are many CRM software packages that also support customer lead generation. Oracle's PeopleSoft Enterprise Online Marketing [43] includes a lead generation system for marketing organizations. It mainly uses web forms and surveys to obtain customer

information and update customer profiles in order to personalize direct marketing communications. It also obtains, increases, and detains the most imperative customers, and communicates via email and web-based services. Its automated content for more pertinent offers and messages increases user response and conversion rates. Ultimately, the product offers to improve the quality of leads through imitated, prescheduled contact with prospects and customers.

SAP CRM 7.0 [44] provides a lead generation process in order to predict the level of significance shown by business partners, with an aim to convert these leads into real customers. It provides mainly two aspects for generating leads: outbound and inbound. In an outbound scenario, the company is industrious in building a connection with business partners. Leads are usually produced after a marketing campaign and as the result of activities targeted for the campaign. In an inbound scenario, a prospect is communicated with via media such as telephone, E-mail, or fax. If this prospect presents any interest in a certain product, the associate employee can create a lead. If leads need to be produced immediately from a campaign, an additional campaign module needs be added, to which are also allocated the target groups comprising those prospects for which leads are to be generated.

SAS CRM [45] solution provides lead prediction tools to: enhance profit by providing the business to cross-sell and up-sell more efficiently; predict the most prospects and customers who are interested in buying a particular product or service; discover potential customers and grow long-term relationships; and augment brand awareness by providing well-focused communications. In addition, the solutions minimize the costs by providing the business with the ability to target campaigns more precisely and minimize customer support requirements. As well, the solutions also remove the cost of acquiring new customers and provide different types of communication to high-value and low-value customers.

IBM Unica Lead [46] captures, authorizes, orders and assigns leads. It enables marketers to effectively notify and provide leads to relevant departments and agents. It also provides

a dashboard to monitor their disposition and measure results from past leads, and assists marketers to enhance lead close rates to increase the return on investment (ROI) of lead generation marketing. As well, it enhances the point-to-point lead tracking process and minimizes the crucial gap between marketing and sales.

ActiveConversion [47] provides qualified leads to sales forces, producing significantly increased sales. Its marketing intelligence presents sales teams to prospects and visitors, discovering business and individual contacts. It has a scoring system to rank the leads by analyzing certain factors set by the marketers. Periodic updates of important leads help sales teams to learn about the most active prospects discovered from the website. It accelerates the sales cycle and removes the necessity of cold calling. It also integrates the website with Salesforce.com, Jigsaw, Google, and so on, to enhance demand generation, lead nurturing, and management.

Marketo [48] provides on-demand lead generation software for B2B marketing professionals to discover leads that are ready for sales. It also automatically brings up the rest of the prospects that represent future possibility but are not yet ready to be involved in an active buying process. It transforms website traffic into leads providing targeted pages and content. It ranks fresh inquiries to automatically discover the most important customer leads, and also develops new inquiries into prompted sales leads using appropriate and personalized campaigns.

All of the discussed lead generation systems are built targeting B2C or B2B scenarios. This targeting facilitates the lead generation process by improving marketing and sales strategies. To our best knowledge, there is no suitable automatic lead-generation system readily available which could empower the online real estate service provider to quickly and easily capture targeted leads.

2.5 Related Data Mining Techniques

2.5.1 Classification Algorithms

Classification is the task of mapping a new data object into one of several predefined classes of a predetermined target. In the web application domain, this technique can be used to identify whether a user's profile belongs to a particular class or category. In this thesis, the goal of classification is to build user profiles belonging to lead user groups or non-lead user groups.

Because of its human readable and easily interpretable output, the most commonly used classification algorithm is decision tree learning. Decision tree generates a model for predicting a value of a target variable by analyzing other relevant input variables. This algorithm can be divided into two types. The first is a classification tree that classifies the target variable into several classes such as, for instance, weather forecast. The second type of algorithm is the regression tree, which predicts numerical values for the target variable, such as the price of a product. Figure 2.6 shows an example of a decision tree that predicts playing conditions based on the provided weather forecast.

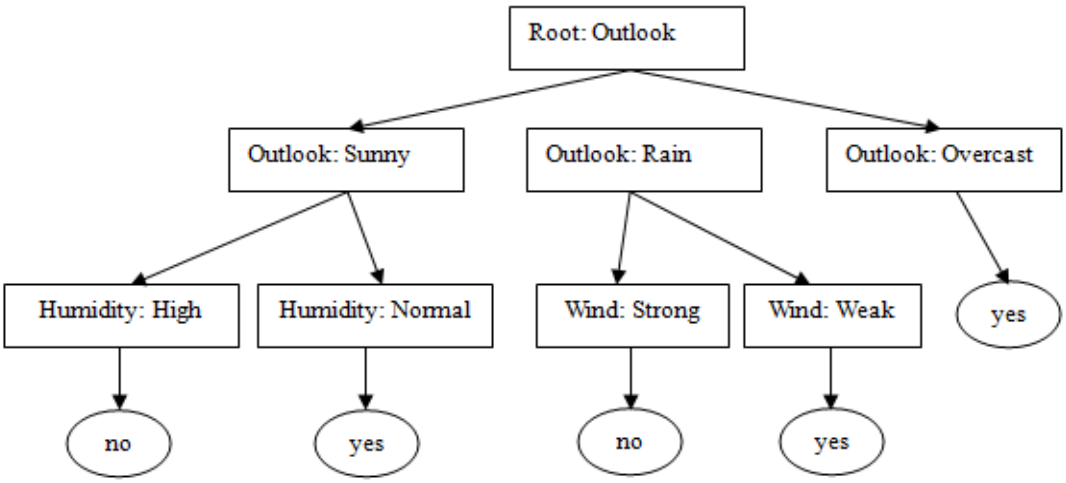


Figure 2.4 Example of Decision Tree.

Decision tree algorithms have advantages over other classification algorithms like k-

nearest neighbor, neural networks or Bayesian networks. For instance, decision tree takes minimal data preparation time and can handle both numerical and categorical data, making it suitable for the dataset used in this thesis.

There are many decision tree algorithms available, most notably ID3 [26], C4.5 [25] and C5.0. C4.5, which originated from ID3, is probably the most commonly used decision tree learning algorithm, with C5.0 being its successor. Surveys in [39] show slight differences but negligible improvements between C4.5 and C5.0. These two algorithms show similar performance, except the size of the induced trees is significantly reduced when a pruning strategy is adopted, but the same effect can often be achieved by adjusting C4.5 pruning parameters.

2.5.1.1 ID3 Algorithm

ID3 is a simple decision tree learning greedy algorithm which generates decision trees from a fixed set of data in a top-down, recursive, divide-and-conquer manner. The major disadvantages of this algorithm are that the attributes must be categorical and continuous-valued attributes must be discretized in advance. The general algorithm properties building decision trees can be presented in the following paragraph [15]:

- The tree begins with a single node consisting of all the training samples.
- It uses an entropy-based measure, information gain [13] as a heuristic to find the attribute with most information to separate the samples into individual classes. This attribute is then set as the test attribute at the node.
- It uses the same procedure iteratively to create the sub-trees based on the samples at each partition. Any attribute selected for a node should be removed from the test-attribute-list so that it will not be considered in any of the node's descendents.
- The iteratively partitioning procedure halts only when any of the following conditions occur:
    - If all of the samples for a given node pertain to the same target class.
    - If there are no more attributes by which the samples may be further divided.
    - If there are no more samples for the branch test-attribute.

2.5.1.2 C4.5 Algorithm

C4.5 [25] is the most widely used decision tree learning algorithm and is an extension of the ID3 [26] algorithm. Both C4.5 and ID3 are divide-and-conquer greedy algorithms that use hill-climbing search techniques and Information Gain [13] measures to search through the decision tree's space. The major advantages of C4.5 over ID3 are listed below:

- It can handle both continuous and discrete attributes. To handle continuous attributes, it creates a maximum value and then divides the list into attribute values that are above the maximum value and those that are less than or equal to it [27].
- Handling training data with missing attribute values.
- It also shows robust performance in the presence of noisy data, and avoids over fitting.
- Post pruning: Once the decision tree has been created, it goes back through the tree and removes non-performing subtrees by replacing them with leaf nodes.
- It converts decision trees to rules.

2.5.2 Association Rule Mining Algorithms

In data mining, association rule generation is a method for discovering interesting relationships among various item sets in a dataset. In this thesis, the expected output from association rule mining is relationships among the item sets of the user activities, landing pages, demographic information, and the company's advertisements.

Association rules are presented in the form of X => Y where X and Y are two disjointed subsets of all available items. X is called the antecedent or LHS (left-hand side), and Y is called the consequent or RHS (right-hand side).

The association rule mining procedure can be divided into two parts: generating frequent item sets, and then applying the interestingness measure to generate final rules. The support rate is the most commonly used measure to generate frequent item sets.

- **Support rate of frequent item sets (X, Y):**
  - The percentage of instances in datasets containing both X and Y.
  - The support rate, also called *supp*, is the probability that a transaction contains both X and Y, i.e., P(X∪Y).

Different algorithms for generating frequent itemsets are described as follows:

2.5.2.1 Apriori

Apriori is a classic data mining algorithm for generating association rules first described in [2]. The algorithm mainly follows one principle to generate frequent item sets, which is that any subset of a frequent item set is frequent. In other words, if there is any item set which is infrequent, its superset should not be generated for testing.

Method:

–generate length (k+1) candidate item sets from length k frequent item sets only, and

–test the (k+1) candidates against the DB.

Any subset of a frequent item set must be frequent

–If {A, B, C} is frequent, so is its subsets, such as {A,B},{A,C}, etc., since every transaction having {A, B, C} also contains {A,B},{A,C}, {B,C}, {A},{B},{C}.

Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. However, it also has a few inefficiencies. For example, the algorithm attempts to load up the candidate set with as many as possible before each scan to generate large numbers of subsets.

2.5.2.2 Eclat

The eclat association rule mining algorithm, described in [40, 41], uses the structural properties of frequent item sets to alleviate fast discovery. First, the items are organized into a subset lattice search space, which is decomposed into small independent sub lattices and can be solved in memory. Then the lattice traversal techniques are applied to quickly identify all of the long frequent item sets and their subsets. The algorithm also uses various database schemes merged with the decomposition and traversal techniques.

For every item, a list of transaction ids is stored where the item occurs (called a tidlist), and for every itemset's tidlist equals the intersection of the tidlists of two of its subsets. Figure 2.3 shows the tidlist generating procedure:

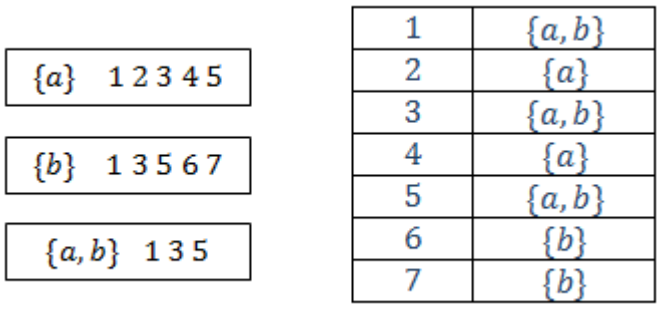


Figure 2.5 Generating tidlists for item sets.

2.5.2.3 FPGrowth

FP-growth was proposed in [16]. FP-growth is the first association rule mining algorithm that does not follow the candidate set generation and test approach, such as Apriori-based algorithms. The efficiency of the algorithm is addressed mainly based on three techniques. The first technique is the input database, which is compressed into a concentrated and smaller data structure called the FP-tree, as shown in Figure 2.4.

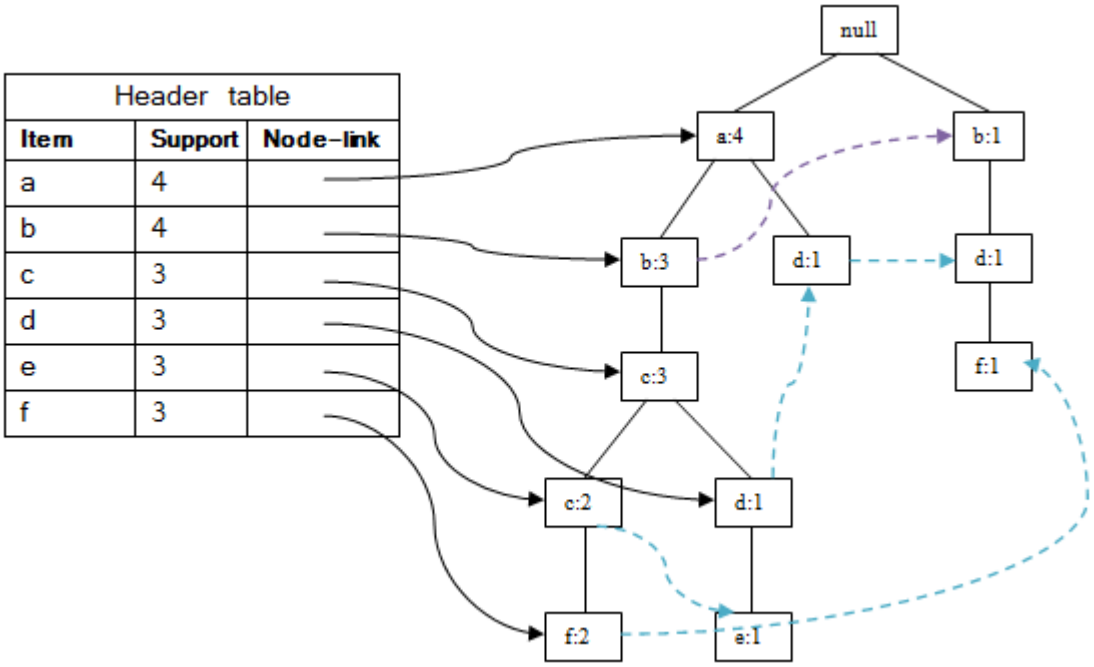


Figure 2.6 Database storing procedure in FP-tree.

This tree is used for subsequent database scans to avoid costly and repeated database scan of the original. The second technique is a frequent pattern growth method used to evade the expensive generation of a large number of candidate sets. The third and last technique is a partitioning-based, divide-and-conquer method to disintegrate the mining task into a set of smaller tasks for mining restricted patterns in conditional databases. This third technique proved to reduce the search space.

2.5.2.3 SaM

SaM, a split and merge algorithm for frequent item set mining, is described in [5, 6, 7]. One of the major advantages of this algorithm is the simple data structure and processing scheme which make it easy to execute, especially when a large volume of data sets cannot be loaded into the main memory. The algorithm first decide the frequencies of each items from the input database in order to remove infrequent items immediately. The frequent items in each transaction are then sorted, according to their frequency, in the database, since processing the items in the order of increasing frequency usually take the shortest execution times. After that, transactions are sorted lexicographically into descending order, the item with the higher frequency preceding the item with the lower frequency. Lastly, the data structure on which SaM functions is constructed by merging similar transactions and setting up an array, in which every item composed of two fields: an occurrence counter and a pointer to the sorted transaction. This data structure is then processed recursively to find the frequent item sets.

2.5.3 Interestingness Measures of Association Rules

Association algorithms can generate a large volume of patterns from the datasets, most of which are of no interest to domain experts. To remove the uninteresting rules, interestingness measure techniques are applied to the result sets [14]. The classic research contained in [23] describes the usages of different measures of interestingness in analyzing and presenting strong rules discovered in databases. Interestingness measures use statistically derived from datasets to determine whether a pattern is interesting or not.

Computing Interestingness Measure: Information needed to compute interestingness of a rule X → Y can be obtained from a contingency table, as shown in the following figure:

	Y	$\bar{Y}$
X	$P(X,Y)$	$P(X,\bar{Y})$
$\bar{X}$	$P(\bar{X},Y)$	$P(\bar{X},\bar{Y})$

Figure 2.7 A 2-way contingency table for variable X and Y.

Here, P(X, Y) is equivalent to the support of (X,Y). The contingency table can be used to define various interestingness measures such as Confidence, Lift, Gini, J-measure, etc.

Confidence [2] is a measure of the conditional probability that a transaction having X also contains Y. In probability notation, it can be expressed as $P(Y|X)$.

Φ-coefficient [3] is a measure of the correlation (linear dependence) between two variables, X and Y. It is mainly used as a measure of the strength of linear dependence between two variables. In probability notation, it can be expressed as

$$\frac{P(X,Y) - \big(P(X) * P(Y)\big)}{\sqrt{P(X)\,P(Y)\,P(\bar{X})\,P(\bar{Y})}}$$

IS Measure or Cosine similarity [37] is a measure of similarity between two sets by measuring the cosine of the angle between them. This is an alternative measure that is commonly used for handling asymmetric binary variables. In probability notation, it can be expressed as $\frac{P(X,Y)}{\sqrt{P(X)* P(Y)}}$

Lift [12] measure is equivalent to the ratio of the confidence of the rule and the expected confidence of the rule. In probability notation, it can be expressed as $\frac{P(X,Y)}{P(X)*P(Y)}$.

The odds ratio [22] measure is used to calculate the strength of association or non-independence between two binary data values. It is used as a descriptive statistic, and plays an important role in logistic regression. In probability notation, it can be expressed as $\frac{P(X,Y)\,P(\bar{X},\bar{Y})}{P(\bar{X},Y)\,P(X,\bar{Y})}$.

Cohen's kappa coefficient [36] is a statistical measure of inter-rater agreement or for qualitative (categorical) items. In probability notation, it can be expressed as $\frac{P(X,Y)+P(\bar{X},\bar{Y})-P(X)P(Y)-P(\bar{X})P(\bar{Y})}{1-\big(P(X)*P(Y)\big)-\big(P(\bar{X})*P(\bar{Y})\big)}$.

J-measure [34] is a single criterion which evaluates both applicability and accuracy with various desirable properties. In probability notation, it can be expressed as $P(X,Y)*Log\left(\frac{P(Y|X)}{P(Y)}\right)+\big(P(X\bar{Y})\big)*Log\left(\frac{P(\bar{Y}|X)}{P(\bar{Y})}\right)$.

Gini coefficient [8] is a measure of statistical dispersion or the inequality of a distribution. In probability notation, it can be expressed as $P(X)[\,P(Y|X)^2+\,P(\bar{Y}|X)^2\,]+\,P(\bar{X})[\,P(Y|\bar{X})^2+\,P(\bar{Y}|\bar{X})^2-P(Y)^2-P(\bar{Y})]$

Laplace [11] measure can be defined as the expected value of the absolute value of the difference between a random variable and its mean. In probability notation, it can be expressed as $\frac{NP(X,Y)+1}{NP(X)+2}$

Conviction [9] can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independently divided by the observed frequency of incorrect predictions. In probability notation, it can be expressed as $\frac{P(X)P(\bar{Y})}{P(X\,\bar{Y})}$

Piatetsky-Shapiro's [23] described measure is mainly based on three simple rules. Firstly, Rule Interestingness = 0 if Support (A,B) = Support (A) x Support(B). Secondly,

Rule Interestingness monotonically increases with Support (A,B) when other parameters are fixed. Thirdly, Rule Interestingness monotonically decreases with Support (A) and Support (B), when other parameters are fixed. In probability notation, it can be expressed as $P(X,Y) - P(X) * P(Y)$.

Certainty [33] measure is similar to conditional probabilities, but rather than representing the degree of probability of an outcome, they represent a measure of belief in the outcome. In probability notation, it can be expressed as

$$\frac{P(Y|X) - P(Y)}{1 - P(Y)}$$

The added value [30] of the rule X→Y is the measure of whether the proportions of transactions containing Y among the transactions containing X are greater than the proportion of transactions containing Y among all transactions. In probability notation, it can be expressed as

$P(Y|X) - P(Y)$.

Collective strength [1] measure uses the violation rate for an item set which is the fraction of transactions which contains some, but not all, items of the item set. The violation rate is compared to the expected violation rate under independence. Collective strength is downward closed. In probability notation, it can be expressed as

$$\left(\frac{P(X,Y)+P(\overline{X}\overline{Y})}{P(X)P(Y)+P(\overline{X})*P(\overline{Y})}\right) * \left(\frac{1-P(Y)P(X)-P(\overline{Y})*P(\overline{X})}{1-P(X,Y)-P(\overline{X}\overline{Y})}\right).$$

Jaccard's [29] measure, also known as the Jaccard similarity coefficient, is a statistic used for comparing the similarity and diversity of sample sets. In probability notation, it can be expressed as

$$\frac{P(X,Y)}{P(X) + P(Y) - P(X,Y)}$$

Chapter 3 **Methodology and System Architecture**

In this chapter, the methodology and the overall system architecture for customer lead generation is described. The approaches and algorithms are discussed in detail for each module of the system.

3.1 System Architecture

Figure 3.1 illustrates the overall work flow of the system, which consists of three major modules, as follows:

1. Web Click-Stream Data Pre-Processing:
    1. Data Cleaning.
    2. Session Identification.
    3. Data Modeling.
    4. Data Integration.
    5. Data Transformation.
2. Data Mining:
    1. Association Rules Generation.
    2. Classification.
3. Pattern Analysis:
    1. Pattern Filtering.
    2. Aggregation.
    3. Characterization.

3.2 Web Click-Stream Data Pre-Processing

The system first gathers all available data sources. The primary data sources used in this system are the server log files, which include web server access logs and application server logs. Additional data sources, essential for both data preparation and pattern discovery, include the site files and meta-data, operational databases, application templates, and domain knowledge. After gathering data from all of the required sources, the web data will go through a pre-processing phase in which it removes all irrelevant and

redundant data. Next, the user session is identified to group together all of a single user's information into a single object. Subsequently, data modeling is conducted, which mainly involves identifying relevant dataset(s) by analyzing user activity, dividing user groups, and performing data integration.

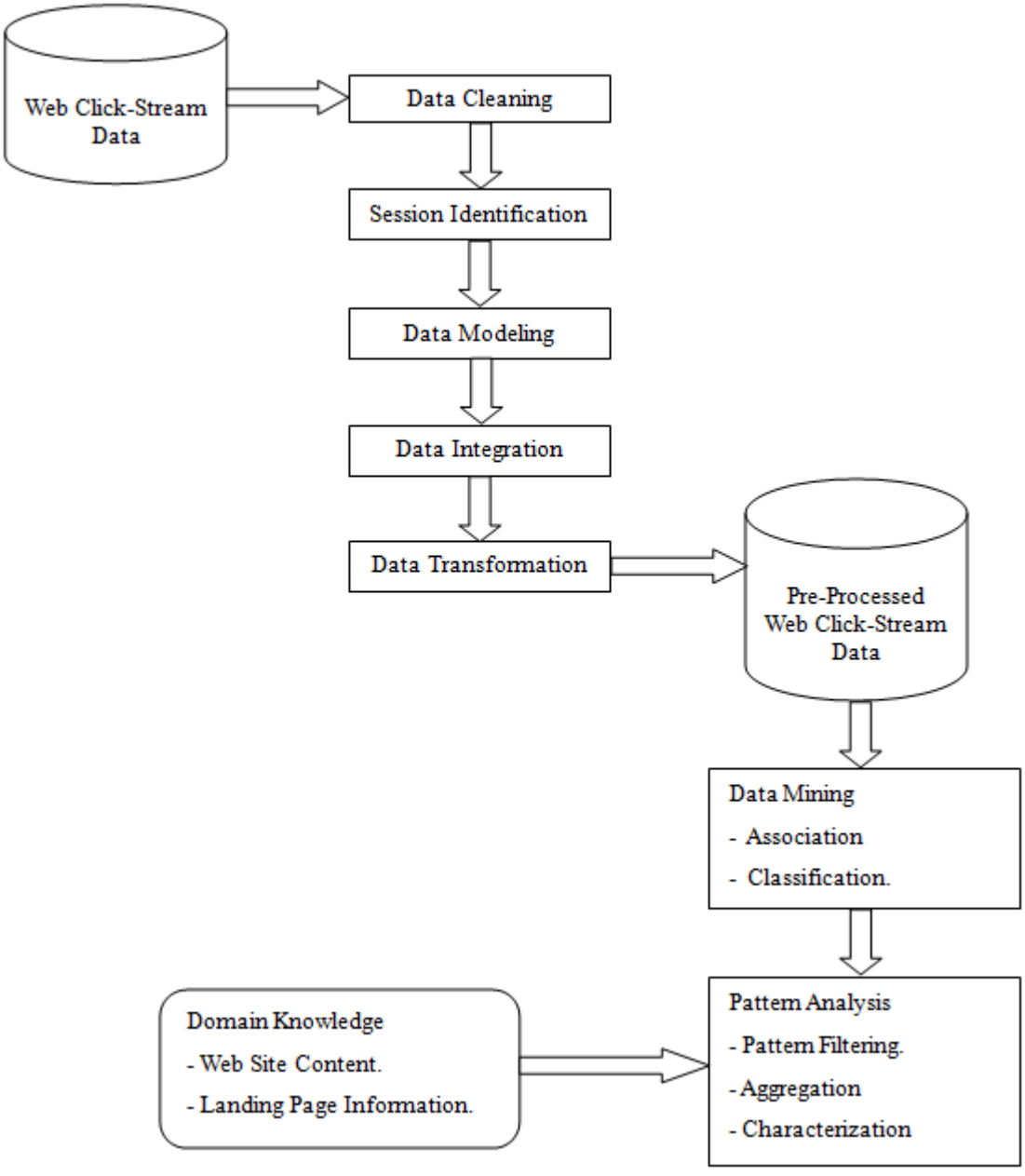


Figure 3.1 System architecture.

Following the pre-processing phase, the data will be formatted appropriately according to the application. Then, the log data will be converted into a form suitable for a specific data-mining task. This transformation will be accomplished according to the transaction model for that particular task (i.e., classification or association rule mining). The steps of data pre-processing, including data modeling and data integration, are described in detail in Chapter 4.

3.2 Data Mining

After data pre-processing, the final pre-processed data sets are stored in the database, and then the data mining algorithms are applied on the data sets to discover rules and patterns from item sets. In this thesis, two categories of data mining techniques are used: Classification and Association rule mining.

3.2.1 Classification Algorithm

C4.5 is the most widely used decision tree learning algorithm. It applies a divide-and-conquer, hill-climbing search technique using Information Gain [13] measures to search through the space of a decision tree. The general algorithm for building decision trees can be presented in the following pseudo code [19]:

Input: Training Samples
Output: A Decision Tree

1. Check for base cases
    - If all of the samples in the list belong to the same class, it simply creates a leaf node representing that class for the decision tree.
    - If none of the attributes provide any information gain, it creates a decision node at the top of the tree using the desired value of the class.
2. For each attribute att.
    - Find the normalized information gain from splitting on att.
3. Let att1 be the attribute with the highest normalized information gain.
4. Create a decision node that splits on att1.

5. Recurse on the sublists obtained by splitting on att1, and add those nodes as children of node.

3.2.2 Association Rule Mining Algorithms

Apriori: Apriori is the best known basic algorithm for generating frequent item sets from a set of transactions. It utilizes the breadth-first search technique and a hierarchical tree structure to generate frequent item sets. It mainly follows the key rule:

- Frequent Itemset Property: Any subset of a frequent itemset is frequent.
- Contrapositive: If an itemset is not frequent, none of its super sets is frequent.

In other words, if {A, B, C} is frequent, so should any of its subsets be, such as {A,B}, {A,C}, etc., since every transaction having {A, B, C} also contains {A,B}, {A,C}, {B,C}, {A}, {B}, {C}.

Apriori candidate generation and test approach:

1. First, scan DB once to get frequent 1-itemsets.
2. Generate length (k+1) candidate itemsets from length k frequent itemsets.
3. Test the candidates against DB (pruning).
4. Terminate when no frequent or candidate set can be generated.

The Apriori algorithm for generating frequent item sets can be presented in the following pseudo code [17]:

Input: Database D, min_supp
Output: L, frequent item sets in D

C_k : Candidate item set of size k;
L_k: frequent item set of size k;
L_1= frequent_1-item sets (D, min_supp);
for(k= 2; L_{k-1} !=Ø; k++) { // search in order
Ck= apriori_gen(Lk-1); // candidates generation
for each transaction t Ɛ D
{
  // scan D for counts (candidate testing)
  Ct= subset(Ck, t); // candidate subsets foreach candidate c Ɛ Ct
  c.count++;
}

Lk= {c Ɛ Ck \c.count⩾ min_supp}
}
return ⋃ k Lk;

Eclat: Eclat algorithms make use of the structural properties of frequent itemsets to alleviate fast discovery. At the beginning, the algorithm organizes the items into a subset lattice search space, which is decomposed into small independent sub lattices and can be solved in memory. After that, the lattice traversal techniques are applied to quickly identify all of the long frequent itemsets and their subsets [40, 41]. Before describing the details of the algorithm, some basic lattice theory terminology is explained here, which form the key techniques for itemset enumeration:

Let P be a set. A partial order on P is a binary relation ≤ , such that for all $X, Y, Z \in P$, the relation is:

1. Reflexive: $X \leq X$.
2. Anti-symmetric: $X \leq Y \; and \; Y \leq X, implies \; X = Y$.
3. Transitive: $X \leq Y \; and \; Y \leq Z, implies \; X \leq Z$.

The set P with the relation ≤ is called an ***ordered set***. Let L be an ***ordered set***. L is called a join (meet) semilattice if $X \vee Y (X \wedge Y)$ exists for all $X, Y \in L$. L is called a lattice if it is both a join and meet semilattice, i.e., if $X \vee Y$ and $X \wedge Y$ exist for all pairs of elements $X, Y \in L$. L is a complete lattice if $\wedge S$ and $\vee S$ exist for all subsets $S \in L$. An ordered set $M \in L$ is a sublattice of L, if $X, Y \in M$ implies $X \vee Y \in M$ and $X \wedge Y \in M$.

For set S, the ordered set $P(S)$, the power set of S, is a complete lattice in which join and meet are given by union and intersection, respectively

$$\bigvee \{A_i \mid i \in I\} = \bigcup_{i \in I} A_i \quad , \quad \bigwedge \{A_i \mid i \in I\} = \bigcap_{i \in I} Ai$$

Let us assume the power set lattice P(I) of the set of items in an example database $I = \{A, C, D, T, W\}$. It should be noted that the set of all frequent itemsets forms a meet semilattice, as it is closed under the meet operation, i.e., for any frequent itemsets X, and

Y, $X \cap Y$ is also frequent. Conversely, it does not form a join semilattice, since X and Y being frequent does not imply $X \cup Y$ is frequent. Thus, all subsets of frequent itemsets are frequent, which is a result of the closure under meet operation for the set of frequent itemsets. Consequently, all supersets of an infrequent itemset are infrequent. This remark build the foundation of a intense pruning strategy in a bottom-up search method for frequent itemsets. To be specific, only the frequent itemsets found at the previous level need to be extended as candidates for the current level. However, the lattice formulation indicates that it needs to be limited to a purely bottom-up search, as the maximum **frequent itemsets** exclusively **define all frequent itemsets**. This remark indicates that the main objective should be to devise a search procedure that quickly identifies the maximal frequent itemsets.

Lattice decomposition using Prefix-Based classes: If there were sufficient primary memory, all of the frequent itemsets could be enumerated by traversing the power set lattice, and itemset supports could be calculated by performing intersections. However, in practice, primary memory is limited, which might not be sufficient for all of the intermediate generated tidlists. To solve this problem, the original lattice is fractionated into smaller pieces such that each portion can be solved independently in the main-memory. The lattice decomposition process is described in the following paragraph:

Let P be a set. An ***equivalence relation*** on P is a binary relation $\equiv$ such that for all $X, Y, Z \in P$, the relation is:

1. Reflexive: $X \equiv X$.
2. Symmetric: $X \equiv Y$ implies $Y \equiv X$.
3. Transitive: $X \equiv Y$ and $Y \equiv Z$, implies $X \equiv Z$.

The equivalence relation partitions the set P into disjoint subsets called ***equivalence classes***. The ***equivalence class*** of an element $X \in P$ is given as $[X] = \{ Y \in P \mid X \equiv Y \}$. If a function is defined as $p : P(I) \times N \rightarrow P(I)$ where $p(X,k) = X[1:K]$ the k length prefix of X , then an equivalence relation θ_k on the lattice P(I) can be defined as follows:

$$\forall X, Y \in \rho(I), X \equiv \theta_K Y \Leftrightarrow p(X,k) = p(Y,k).$$

Here, θ_k is called a prefix-based equivalence relation. In practice, the one-level decomposition provoked by θ_1 is satisfactory. However, in some cases, a class may still be too large to be solved in primary memory. In that case, a recursive class decomposition is applied. For example, if $[A]$ is too large to fit in main memory, as $[A]$ is itself a binary lattice, it can be disintegrated using θ_2. The resulting set of classes are $\{AC, AD, AT, AW\}$, so each class can be solved independently, and it can be solved in reverse lexicographic order to enable subset pruning. Moreover, the large class can be recursively partitioned into smaller ones until each class is sufficiently small to be solved independently in primary memory.

Bottom-Up frequent itemsets search: The bottom-up search is based on a recursive disintegration of each class into smaller classes provoked by the equivalence relation θ_k. The equivalence class lattice can be traversed in either depth-first or breadth-first manner. To compute the support of itemset, the tidlists can be divide into two of its subsets at the previous level. Since the search is breadth-first, this technique enumerates all frequent itemsets.

FP Growth: FP-growth is the first association rule mining algorithm which does not follow the candidate set generation and test approach, like Apriori-based algorithms. At the outset, the algorithm builds a compressed data structure called the frequent pattern tree, which is an additional prefix-tree structure storing quantitative data regarding frequent patterns [16]. The properties of the FP-tree are as described in the following paragraph:

1. The FP tree is composed of mainly three components: A root with “null” value; a set of item prefix subtrees as the descendant of the root; and a header table for frequent items.

2. Each node contains three components: node link, item-name and count. Node link connects to the following node with the same item name, item-name shows which item this node represents, and count represents the total number of transactions that occurred by the segment of the path reaching this node.

3. Each object of the frequent-item header table contains two fields: item name and an index that points to the first node in the FP-tree carrying the item-name.

The tree nodes are designed in such a way that more frequently occurring nodes are preferred to share nodes than less frequently occurring ones**.** This method confirms that the tree structure is compressed and informative. The experiments show that this kind of tree is compressed, and that the relative size of the tree is often smaller than the original database. The mining algorithm only requires to operate on the FP-tree rather than the original database.

After generating the FP-tree, a pattern fragment growth mining procedure is constructed. The procedure begins from a frequent length-1 pattern that checks only its conditional pattern base; from there, it builds its FP-tree and applies mining recursively. The pattern growth is attained by joining the suffix pattern with the newly generated pattern growth from a conditional FP-tree. The pattern growth ensures the completeness of the result, as the frequent itemset in any object is always embedded in the associated path of the frequent-pattern trees. The main functions of mining are count collection and prefix path count fixing, which are generally less expensive than candidate generation and pattern matching functions conducted in most Apriori-based algorithms.

At the end, the search technique developed for mining is a partitioning-based, divide-and-conquer method instead of a candidate-generation of frequent itemsets. This technique impressively decreases the size of provisional pattern bases and the corresponding conditional FP-tree. In addition, it overcomes the issue of finding large frequent patterns by seeking smaller ones and then joining the suffix. Instead, it uses the minimum frequent items as suffix, which provides effective selectivity. All of these methods help in achieving considerable decrease of search costs.

SaM: One of the major advantages of this algorithm is the simple data structure and processing scheme which makes it easy to execute, especially when a large volume of data sets cannot be loaded into the main memory.

Figure 2.4 shows the basic operations of the SaM algorithm to build the simplified data structure from a transactional database. Step 1 shows the original database. In step 2, the algorithm first determines the total occurrence of each items from the input database in order to discard infrequent items immediately. In step 3, the frequent items in each transaction are sorted according to their frequency in the database, since processing the items in the order of increasing frequency generally takes the least execution times. In step 4, transactions are ordered lexicographically into descending order, with item comparisons again being decided by item frequency (items with higher frequency preceding items with lower frequency). Lastly, the data structure on which SaM functions is constructed by merging similar transactions and setting up an array, in which every element contains two fields: an occurrence counter and a pointer to the sorted transaction.

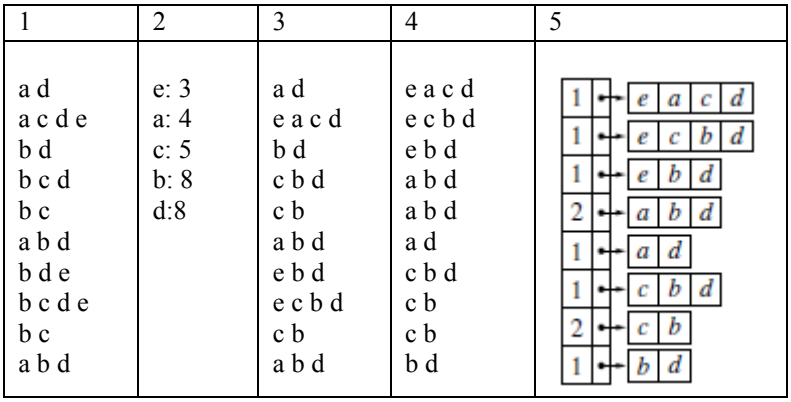


Figure 3.2 The basic operations of the SaM algorithm.

This data structure is then processed recursively to find the frequent item sets. The basic operations of the recursive process follow the general depth-first or divide-and-conquer scheme. In the split step, the data structure is split with respect to the leading item of the first transaction. All instances referring to transactions starting with this item are moved to a new array. In this stage, the pointer to the transaction is forwarded by one item, so

that the common leading item is “removed” from all transactions. Certainly, this new array substitute the provisional database of the first sub-problem, which is then function recursively to find all of the frequent items sets containing the split item.

The provisional database for frequent item sets which does not contain the split item can be generated with a simple merge step. The generated new array and the rest of the original array are merged with a method which is nearly similar to one phase of the well-known merge sort algorithm. Since both arrays are lexicographically sorted, one merging traversal is enough to create a lexicographically-sorted merged array. The only variance to a merge sort phase is that equal transactions are merged. As a result, there is constantly just one instance of each transaction, while its number of occurrences is kept in the frequency counter. The array for the divided database can be used again after the recursion for the split with respect to the next item. As a result, each recursion step only needs to allocate one new array, with a size that is limited to the size of the input array of that recursion step.

At the end, the algorithm uses only a simple array to construct the data structure, and the algorithm can be simply implemented to operate on any storage or a relational database system. There no need to load the transactions into a main memory and even the array may be easily stored as a simple table in a relational database.

3.3 Pattern Analysis

Finally, after discovering hidden common patterns among data items, the domain knowledge will be used to evaluate the newly discovered knowledge. Two main categories of knowledge are generated from the data mining: decision tree and association rule mining. The decision tree classifies the users into two groups: lead and non-lead. The rules generated from the decision tree will help delineate the behavioral pattern of lead and non-lead users. On the other hand, association rules mining algorithms discover interesting patterns among users’ demographic information, web click-stream data and visited pages. This pattern provides information about which landing pages are performing well in converting lead users and whether there is any particular group of

users who are attracted to any particular landing page. In Chapter 5, a broad explanation is provided of pattern analysis.

Chapter 4 **Data Model Design and Data Integration**

Data model design is a critical step for data mining. It involves several steps, such as analyzing user activity, identifying relevant dataset(s), dividing user groups, and integrating data. These procedures will be discussed in this chapter.

Section 4.1 gives an overview of a user's activity flow in an online real estate service provider; section 4.2 gives an overview of the datasets; section 4.3 describes the proposed data models; and section 4.4 presents the data integration engine design and implementation.

4.1 User Activity Flow

This section analyzes the activity flow of users who visit the online real estate service provider's website, including both leads and non-leads. It serves as the basis for designing the data models.

In order for users to be considered as potential leads, they must have fulfilled certain conditions. For example, web users must have clicked on the provider's rewards ad or on a Google AdWords ad. In general, rewards ads are posted on the homepage of a provider's website. Nevertheless, it should be noted that the ads shown depend on the ip location of the users; hence, an ad shown for a user from "Ottawa, Ontario" may not necessarily be shown for a user from Halifax, NS. Once a user clicks on a rewards ad, he or she is redirected to a "points program" selection page. For the purposes of this research, this page will be referred to as the rewards landing page. Finally, users must then have filled out the subsequent "Contact Form" after having made a "points program" selection, and also agree to be contacted via an agent. Under-performance of these forms has been a source of concern for companies, as not all users who reach this page successfully complete and submit the form. Figure 4.1 illustrates the assumed user navigation and task flow summarizing these conditions.

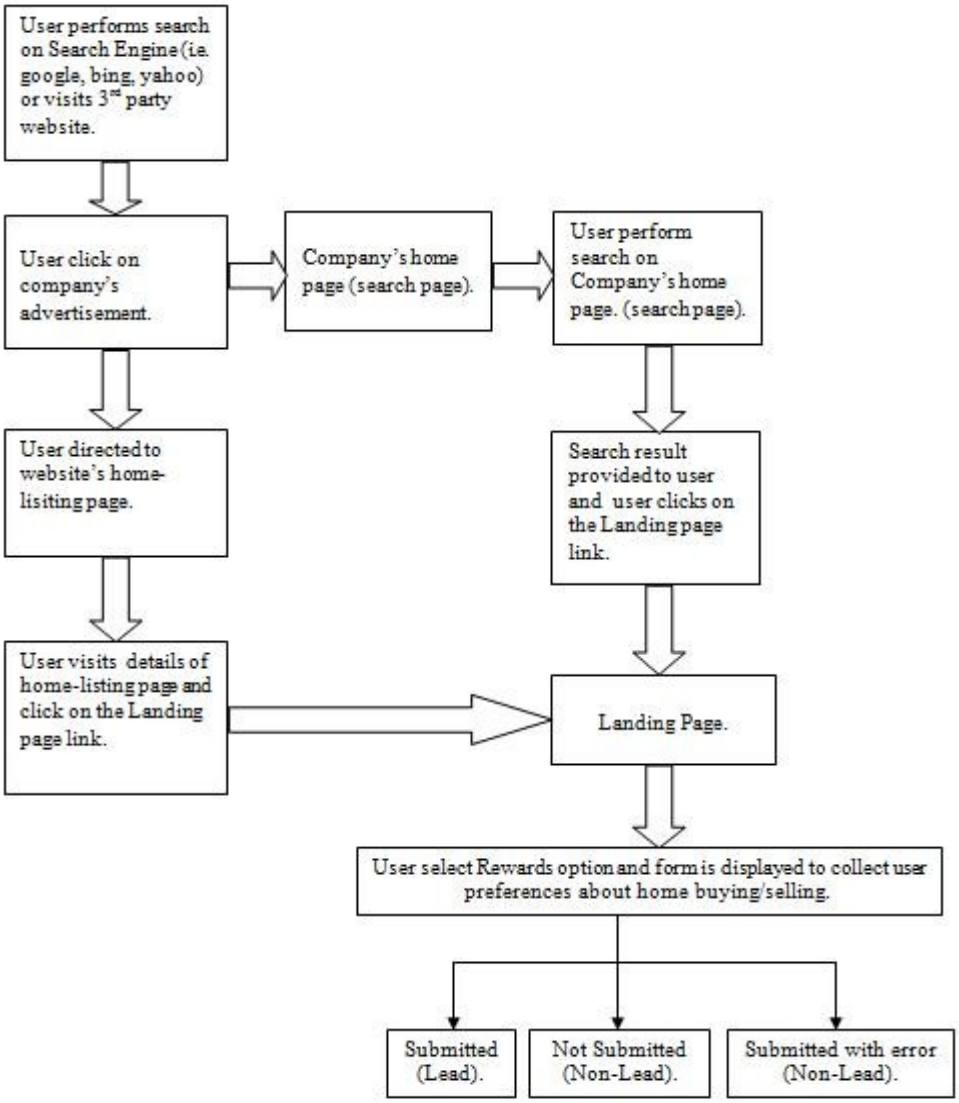


Figure 4.1 User navigation and Task flow.

Web users can also fill out forms in order to be contacted by realtors through ads on real estate agent pages, which re-direct the user to a "Realtor match" page. There are also ads for visitors to connect with a mortgage broker.

In summary, a user becomes a lead upon filling out a landing page form and successfully submitting it. These landing pages are manually generated. Table 4.1 shows examples of customer leads. All of these leads provide contact information and show an intention to buy or sell property. However, for the purposes of this research, we will be carefully

assessing the contact information provided. As noted earlier, a potential customer must provide a valid phone number and agree to be contacted by phone in order to qualify as a lead. As such, customers who provide non-valid phone numbers, such as "555 555 ----", would not be considered potential leads.

Table 4.1 Example of customer leads for online real estate service providers.

Full Name	E-mail	Phone	City	Price Range	Intent	Prov.	Request Type	Program Choice	IP
-	--@gmail.com	416 319 ----	Aurora	600-800k	buy	ON	Non-points Request	-	-
R. F.	--@----.ns.ca	902 660 ----	Amherst	-	buy	NS	Points Section Request	Rewards 1	NS
C. W.	--@yahoo.com	360 441 ----	Sechelt	200-400k	buy	BC	Non-points Request	-	-
J. L.	--@----.nb.ca	506 333 ----	Quispamsis	-	buy	NB	Points Section Request	Rewards 1	NB
S. K.	--@hotmail.com	905 685 ----	Hamilton	200-400k	buy	ON	Non-points Request	-	-
L. M.	--@aol.com	905 659 ----	Burlington	-	buy	ON	Points Section Request	Rewards 2	-
C. L.	--@live.com	604 719 ----	North Vancouver	200-400k	buy	BC	Points Section Request	Rewards 1	-
N. D.	--@hotmail.com	705 695 ----	Balmertown	200-400k	buy	ON	Non-points Request	-	-
S. F.	--@sympatico.ca	519 289 ----	Glencoe	-	buy	ON	Points Section Request	Rewards 1	ON
A. P.	--@yahoo.ca	555 555 ----	Sarnia	<200k	buy	ON	Non-points Request	-	ON
C. L.	--@----.com	289 237 ----	Hamilton	<200k	buy	ON	Non-points Request	-	ON
O. N.	--@yahoo.com	514 934 ----	Montreal	-	sell	QC	Points Section Request	Rewards 2	-
-	--@rogers.com	905 303 ----	Alliston	200-400k	buy	ON	Non-points Request	-	-
A. L.	--@hotmail.com	289 254 ----	Hamilton	200-400k	buy	ON	Non-points Request	-	ON
V. E.	--@hotmail.com	604 759 ----	Victoria	800-1000	buy	BC	Points Section Request	Rewards 2	BC

Certainly, there are many possible challenges that could be encountered during the span of this research, such as during the data preparation, cleaning, modeling and integration phases. Proper data integration of all available data sources, including various clicks, viewed pages and ad key words, and the order and time information of click sequences into a uniform structure for mining the statistics patterns, is crucial.

Figure 4.2 shows a finite state machine model of the user activity flow. It includes all

possible visiting paths for online customers.

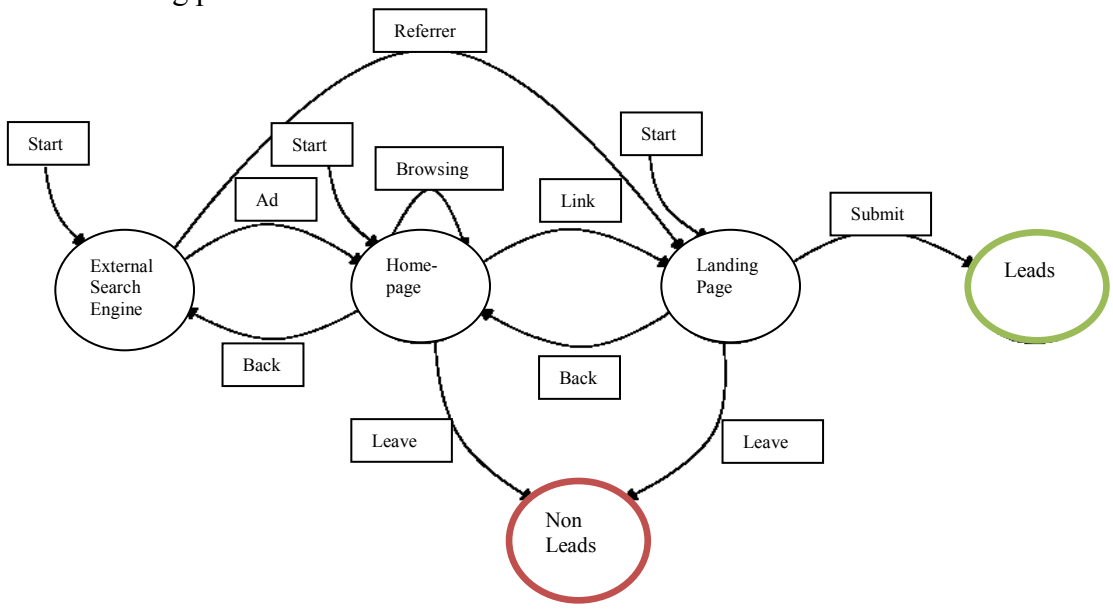


Figure 4.2 User activity flow.

As shown in Figure 4.2, five states are defined according to the page being visited: 1) External Search Engine; 2) Website's pages (excluding landing page, i.e., example pages are: Real Estate Agent, Real Estate Office, Mortgage Broker, Search page, and others); 3) Landing Page; 4) Leads and 5) Non Leads. Among the five states, the first three are the *start* states of a user's visiting path, and the final two states are termination states – represented by green circles – of a visiting path.

Users who enter the system from "External Search Engine" are considered being referred by the search engine, and users who enter from "Website's pages" or "Landing Page" are considered as direct traffic.

From "External Search Engine", users may reach one of the website pages by clicking on an *ad* shown on the homepage, or land on the landing page directly by clicking on the referral link shown on the search page. Users who are browsing website pages can reach the landing page by clicking on the landing page link shown on one of the website pages they visit.

Leads can only be generated by landing pages, as a result of users filling out the form and clicking on "submit". From any previous state, users may leave the system, as "Non-Leads".

4.1.1 Typical activity flow for lead users

Figure 4.3 shows the typical activity flow for leads. Users may enter from any of the three start states. However, in order to become leads, users must visit the landing page, fill out the form, and click "submit". Therefore, the "Landing Page" state (highlighted in the figure) is a required state for being leads. The other 2 states prior to the Landing Page estate (External Search Engine and Website's pages) are optional (shown as dashed circles in the figure).

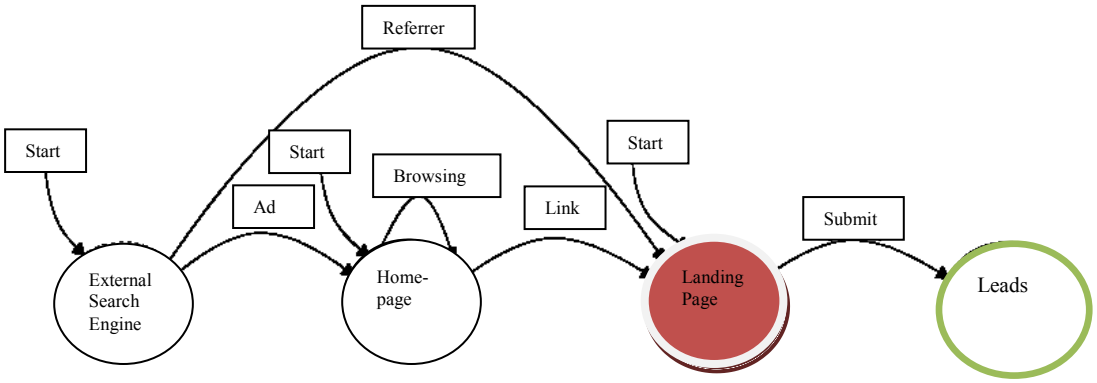


Figure 4.3 Typical activity flow for Leads.

4.1.2 Typical activity flow(s) for non-lead users

Figure 4.4 shows the typical activity flow for a subset of non-lead users who reached the landing page. However, these users may either leave without filling out the form, or abort after filling out part of the form, which make them non-leads.

This group of users is the closest to leads, because they did reach the landing page, which is the required state for becoming leads.

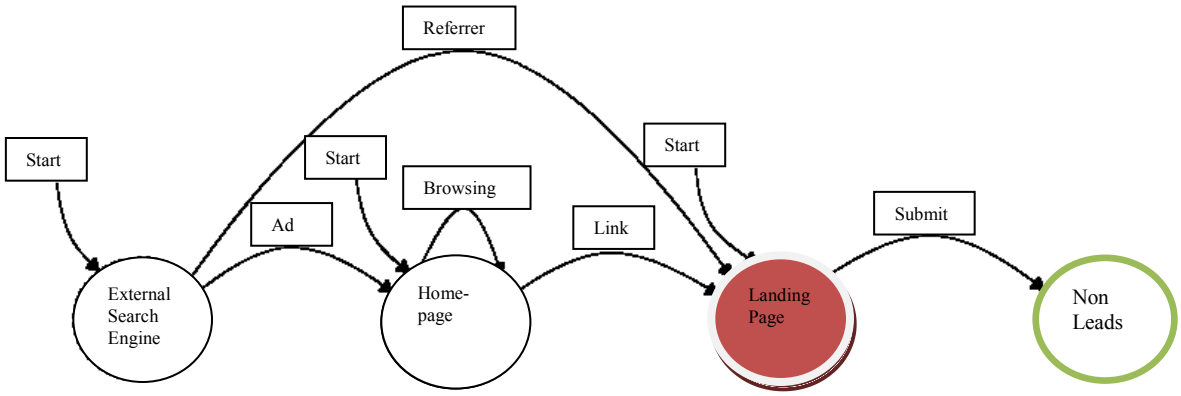


Figure 4.3 Typical activity flow for Non-Leads (Group 1: Reach landing page).

Figure 4.5 shows another subset of non-leads users who did not reach landing page. They visited the website either directly or from an external search engine, probably browsed a few or several pages on the website, but ended up leaving as non-leads.

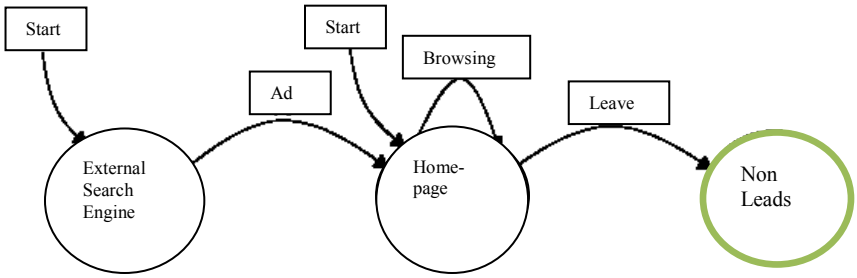


Figure 4.4 Typical activity flow for Non-Leads (Group 2: Do not reach landing page).

4.2 Description of the Datasets

In this thesis, a real-life online real estate service provider's datasets are used. The current data repository setup, for the purposes of this thesis, contains nine datasets, which hold information for the period January to June 2011. The "seoarchive" dataset contains data dealing with search strings users entered on Google and other search engines when searching for the company's website. The "new_tracked_search_items" and "new_tracked_searches" contain data pertaining to what users' actions were while using the website. The "new_tracked_search_items" dataset contains search type information such as whether a user searched for schools, banks, and other retailers while searching for

a home or neighborhood, or which ads were served during this time. The "new_tracked_searches" dataset contains more detailed information about each search (e.g., search strings, referrer pages, or user agent or browser used). The "agent_ads" dataset contains information regarding the ads displayed in the website. The "search_counts" dataset contains information regarding the number of searches for a particular city; this information is regularly updated but not generally used. The "search_engine_referrals" dataset keeps track of all the searches that come to the website. The "provinces" and "cities" datasets contain information related to the cities and provinces used in a search. In addition, there are two more datasets available: user_search_data and user_session_activity. A summary of these datasets is shown on Table 4.2.

Table 4.2 Summary of datasets.

#	Name	Size	#Cases	#Fields
1	Seoarchive	51.4M	100,180	4
2	search_engine_referrals	52M	163,904	7
3	search_counts	16k	1,365	2
4	new_tracked_search_items	81.8M	1,015,332	7
5	new_tracked_searches	109.1M	349,850	18
6	new_tracked_items	668.3M	2,197,320	10
7	agent_ads	53k	222	13
8	Provinces	12k	13	19
9	Cities	2M	32,747	6

Table 4.3 Description of fields of datasets.

Dataset	Fields	Description
seoarchive	search_engine	Google or other search engine.
	search_string	Search string users entered on the search engine.
	destination_page	The page in provider's website shown as a search result (and clicked on by user).
	timestamp	Time and date of access.
new_tracked_search_items	Id	Unique identifier (PK).

Dataset	Fields	Description
	track_search_id	Foreign Key from the new_tracked_searches table (id).
	type	Type of action taken. Values: - finder: show a selection from the side Finder Menu. - auto finer: means type was preselected via URL .
	item1	More detailed information related to the type.
	item2	More detailed information related to the type.
	city_id	Unique identifier for city city being searched.
	timestamp	Time and date of access.
new_tracked_searches	Id	Unique identifier (PK) – (used as FK in new_tracked_search_items).
	user_id	Unique identified for a user if they are registered.
	user_searched	Search string the user entered.
	address	Parsed address (if possible).
	City	Parsed city (if possible).
	province	Parsed province (if possible).
	country	Parsed country (if possible).
	accuracy	Map accuracy of data; see Google Geocoder Accuracy webpages.
	Lat	Latitude of the location being searched.
	Lng	Longitude of the location being searched.
	neighborhood_id	Foreign Key from neighborhood table (missing table).
	Ip	User's IP address.
	referrer	Referrer page in provider's website.
	user_agent	User's browser when accessing provider's website.
	source	Realtor partners that drive traffic to company's website.
	geo_source	Who geocoded the address (yahoo, google, etc)
	status	Records if address was geocoded properly or not.
	timestamp	Time and date of access.
agent_ads	Id	Unique identifier (PK).
	city_id	Unique identifier for city.
	province_id	Unique identifier for province.
	user_searched_substring	Search string entered by user. The rows for this field are empty as it is a work around field.

Dataset	Fields	Description
	name	This field contains unique names given to the ad on the web system.
	url	URL of page user is taken to after clicking on ad.
	image	File path to image used in click-ad.
	agent_id	Unique identifier for agents (FK).
	ad_type	Type of ad. Values: SKY, LREC, IPHONE.
	active	Whether ad is active or not, contains a value of 0 (not active) or 1 (active).
	pay_state	PAID or UNPAID.
	creation_date	Date ad was created.
	expiration_date	Date ad expires.
user_search_data	search_id	Unique id of search string (PK).
	user_search_string	Search string entered by the user on website's search page.
	City	City in which the user is located.
	province	Province in which the user is located.
	landing_page	URL of landing page the user visited.
	session_id	Foreign Key from user_session_activity.
	lp_version	Version of landing page rendered to the user.
	ad_clicked	Ad clicked by the user to arrive at landing page.
	init_timestamp	Timestamp of the user's entry to the site.
	ad_click_timestamp	Timestamp of the user entry to the rewards landing page.
user_session_activity	session_id	Unique identified for the user's session on the reward's page
	session_activity	Activity performed by user on the page.
	activity_info	Information related to the activity. - target area of the landing page. - the user choice upon entering the landing page. - the field, which was changed on the rewards page. - the scroll point when the user is finished (px). - the scroll point when the user is finished (px). - the field in which the error occurred.
	referrer	Referring page for the session (if available).
	timestamp	Time and date in which user performed activity.

Dataset	Fields	Description
province	Id	Unique identifies (PK)
	name	Name of province (being searched)
	short_name	Province name abbreviation
	code	Website's code for the province (not always the same as "short_name")
	zoom	zoom level google map is set to when user looks at data from prov. level
	centre_lat	Latitude coordinate for province's centre
	centre_lng	Longitude coordinate for province's centre
	icon_image	Name of province's flag image file
	icon_lat	
	icon_lng	
	capital	Name of province's capital city
	capital_lat	Latitude coordinate of province's capital city
	capital_lng	Longitude coordinate of province's capital city
	city_xml_file	Name of XML file containing the names of cities (of that province)
	population	Province's population
	national_percentage	
	area	Province's area (calculated in square kilometres)
	population_density	Province's population density
cities	Id	Unique identifier (PK)
	province_id	Unique identifier for provinces
	county_id	Unique identifier for county
	City	Name of city (being searched)
	Lat	Latitude coordinate of city
	Lng	Longitude coordinate of city
search_engine_referrals	Id	Unique identifier
	search_term	Search string users entered in the search engine
	search_domain	Search domain
	country_code	Country code
	landing_page	Landing page version
	referring_url	URL of the referring page (search result page at the search engine)
	timestamp	Time and date of access.

Dataset	Fields	Description
search_counts	city_id	Unique identifier used for city
	count	number of times a particular city has been searched for

4.2.1 Identifying relevant datasets

The current data repository contains 9 datasets. However, the user activity data are stored only in three tables, as shown in Table 4.4.

Table 4.4 Summary of user activity datasets.

#	Name	Size	#Cases	#Fields
1	new_tracked_items	668.3M	2,197,320	10
2	new_tracked_search_items	81.8M	1,015,332	7
3	new_tracked_searches	109.1M	349,850	18

Among the three datasets, "new_tracked_items" is the most important one. It holds user activity information for all users, from all of the website's pages, including landing pages and other pages. The datasets "new_tracked_search_items" and "new_tracked_searches" contain activities on the website search page, for users who did a search on the website. The schema of the selected three datasets is shown in Figure 4.7

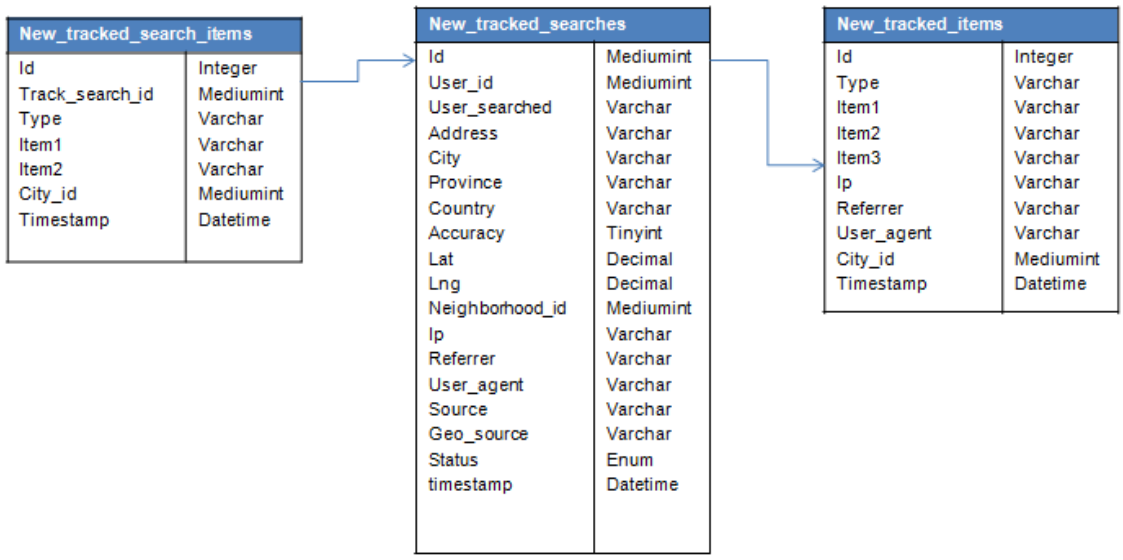


Figure 4.5 Schema of selected datasets.

One of the major challenges in designing the data model is that the "new_tracked_items" is dynamically generated in such a way that a new data instance is created for every different action performed by a user and the properties of a particular instance depends on the action of the user (i.e., submitting a form, doing a search, etc). In other words, the dataset contains many data instances for one user if the user performs more than one action. The "type" field contains the "activity", according to which "item1" "item2" and "item3" fields may contain different types of information.

Note that, in this thesis, we assume that activities that satisfy the following three conditions belong to a single user session:

1. IP addresses are the same.
2. User agents (browsers) are the same.
3. Occur within five minutes of time frame.

Based on the above conditions, we can extract user sessions from "new_tracked_items". A sample user's activities are shown in Table 4.4.

As shown in Table 4.5, the user came from an external search engine "bing" (shown in "referrer" field of row #1). A realtor text ad was served on the search result page, which directed the user to Website's Real Estate Agent page (row #2 and #3). After reaching the Real Estate Agent page, a "Realtor Match" ad was displayed (row #3), and the user eventually reached the landing page version 11 (row #4, #5). The user's ip location was recorded (row #6). Then the user left the landing page without further activities on the Website, and returned to the original "bing" search page he came from (row #7, #8, #9).

Table 4.5 Web-logs of a single user.

type	item1	item2	item3	ip	Referrer
realtor text ad served	Ontario	Barrie		99.238.233.18	http://www.bing.com/search?q=gail+wiltse+of+barrie...
realtor text ad clicked	Ontario	Barrie	top	99.238.233.18	http://www.website.ca/RealEstateAgent/Ontario/Ba...

type	item1	item2	item3	ip	Referrer
redirect ad	.../realtorMatch/Ontario/Barri...	text	top	99.238.233.18	.../RealEstateAgent/Ontario/Ba...
landing page instance		11	by random	99.238.233.18	.../RealEstateAgent/Ontario/Ba...
sem pa11 page view	32177	Barrie, Ontario		99.238.233.18	.../RealEstateAgent/Ontario/Ba...
sem pa11 iplocation	32177	location	Oro-Medonte, ON, Canada	99.238.233.18	.../realtorMatch/Ontario/Barri...
realtor text ad served	Ontario	Barrie		99.238.233.18	http://www.bing.com/search?q=gail+wiltse+of+barrie...
404	/404.html	.../RealEstateAgent/Ontario/Ba...		99.238.233.18	.../RealEstateAgent/Ontario/Ba...
realtor text ad served	Ontario	Barrie		99.238.233.18	http://www.bing.com/search?q=gail+wiltse+of+barrie...

Therefore, this dataset is a semi-structured table in which one field may contain different types of information, and for one user there might be various numbers of rows associated with him or her. For data mining purposes, we normalized this table, combined all information related to one user into one single row, and then only selected a fixed number of representative fields, which contain useful information for data mining.

4.3 Data Model Design

As mentioned in Section 4.1, user activities on a landing page create more attributes that do not exist for users who do not reach the landing page. In order to avoid keeping too many empty values in our integrated dataset, we propose two separated data models – one for general users (all users who visited any website pages) and one for landing page users only (users who reached one of the landing pages). The reasons are described in the following paragraph:

1. The “Landing Page” state is the most important state, because all leads are generated by this state. Users who reach this state have a much higher possibility of becoming leads.

2. Users who reached the landing page have more attributes than other users. For instance, they have information pertaining to “landing page version”; users’ preferences, such as “price range”, “reward program”, etc; users’ personal information, such as “location”, “phone number”, etc., and so on.

3. If we combine the landing page users with all other users who did not reach the landing page, we have to keep *empty* or *null* values for the landing page-related attributes for all of the users who did not reach the landing page, which is a HUGE group. Keeping too many empty values makes the dataset unbalanced, which may result in the data mining engine not working properly.

Therefore, it is helpful to construct a separate dataset for landing page users and perform additional data mining tasks on this dataset. Meanwhile, another dataset will be formed for all users, with simplified landing page related attributes, so that the unbalance of number of attributes can be avoided, and the unbalance of leads/non-leads can be alleviated.

4.3.1 General user data model

The general user data model covers all users including both the leads users and the non-leads users. The purpose of this data model is to form a single integrated dataset, which contains all activities of all users, i.e., including who did not reach the landing page, as well as the activities of the users prior to reaching any landing page.

Although all landing-page-related fields were removed, we generated an additional field “landing page related”, which indicated whether the user eventually reached the landing page, whether the user eventually became a lead. This added field was for the purpose of differentiating landing-page-users from non-landing-page-users, and differentiating leads

from non-leads. Specifically, we used an integer value for the "landing page related" field. We assigned the value "-1" for users who did not reach the landing page, "0" for non-leads users who reached the landing page, and "1" for leads.

The selected features for general user data model are shown in Table 4.6. There are nine features for this model. Among these, eight are extracted from the existing datasets, and the feature #7 is generated to distinguish all landing page users and leads. The selected features fall into four categories: activities before user reached website; activities on website; user info; and additional information, such as time stamp, etc.

Table 4.6 General user data model.

Category	#	Features	Description
Before reaching Website	1	referrer to Website	referrer page to Website
	2	ad served	ad served on referrer page
	3	ad name	ad name
During Visiting Website	4	page view on Website	Website page(s) being visited
	5	search address	address being searched
User info	6	IP location	IP location of user
	7	landing page related	whether the user reached LP, whether the user became a lead
Additional	8	error	whether there is an error
	9	time stamp	time stamp

4.3.2 Landing page user data model

For all users who reached the landing page (leads or non-leads), we designed a separate data model in order to accommodate the extra landing page related activities, as shown in Table 4.7.

Table 4.7 Landing page user data model.

Category	#	Features	Description
Before reaching Website	1	referrer to Website	referrer page to Website
	2	ad served	ad served on referrer page
	3	ad name	ad name
During Visiting Website	4	page view on Website	Website page(s) being visited
	5	search address	address being searched (Prov.)
	6	referrer to LP	referrer page to landing page
Landing page info	7	LP selection method	method of selecting the LP
	8	LP version	version of LP
User info	9	IP location	IP location of user (Prov.)
	10	full name	name of user
	11	phone	phone number of user
	12	email	email address of user
	13	intent	buy, sell, or don't know
	14	choice	reward programs user choose
	15	hood	hood information
	16	price range	price range
Leads or not	17	submit	whether user submitted form
Additional	18	error	whether there is an error
	19	time stamp	time stamp

A total of 19 features are selected for the landing page user data model, which falls into six categories: activities before reaching the website; activities on the website (before reaching the landing page); landing page information; user information; leads or not; and additional information.

As shown in Table 4.6, most of the features are landing page related features, which do not exist for users who did not reach the landing page. Having separate data models can a preclude performing data mining tasks on unbalanced datasets.

4.4 Data Integration Engine Design and Implementation

A data mining engine requires a single, integrated dataset. For each of the data models, we need to generate an integrated dataset. The integration mainly includes two steps:

1. Integrate features from different tables into one table.
2. Integrate multiple rows which are related to one user into one row.

For step 1, because the three selected tables (new_tracked_items, new_tracked_searches, and new_tracked_search_items) share a common id, we can use that common id to join all tables. [1]

For step 2, the activities from the same ip address, using the same browser, and within a five-minute time frame are considered to be from the same user.

An automated system has been developed using Java programming language, which takes the three "user activity" tables (new_tracked_items, new_tracked_search_items, and new_tracked_searches) as input and generates a single dataset for data mining purpose containing all attributes that we suggested in the data model. The program first connects to the website's local database using JDBC (Java Database Connectivity Driver). Next, it reads the input data and processes the data in the flow chart, as shown in Figure 4.8. A brief description of the flow chart is given below.

Input: User activity database (new_tracked_items, new_tracked_search_items, and new_tracked_searches).
Output: Integrated pre-processed datasets.

Step 1: All Landing Page visits are identified by querying "sem%page view" as a sub string in "type" field.

Step 2: For each Landing Page visit, "item3" attribute is checked to identify whether this user has activities on the Website's Search page or not. If the user has search activities on

[1] Note that only the users who did search on the website appear in the two search tables.

the Website's Search page, all search-related attributes are extracted for this user. If there is no search activity performed by the user, the program moves to the next step.

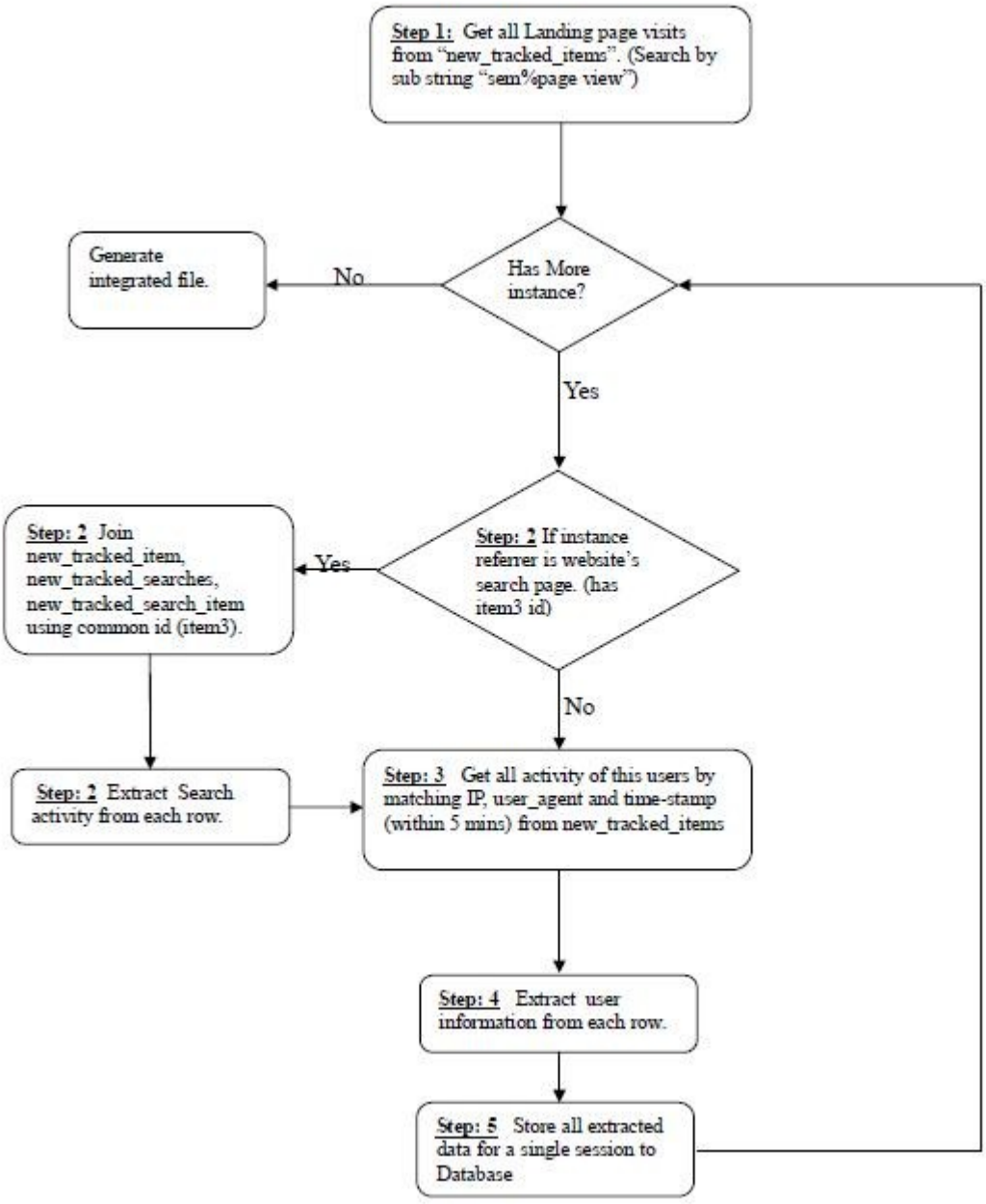


Figure 4.6 Flow chart of data integration (for landing page user dataset only).

Step 3: User session is identified from "new_tracked_items" using ip, user_agent and time_stamp (activities that occurred within a five-minute time frame, from the same ip, and using the same browser are considered to be in the same session).

Step 4: All required attributes for the data model are extracted from each row within the session.

Step 5: All extracted attributes from a session are stored in a new table within a single row, where each single row holds data for only one session.

At the end, if all sessions are identified and data extraction is completed, then the pre-processed integrated file is generated for data mining purposes.

Chapter 5 **Data Mining for Automatic Leads Generation**

5.1 A Data Mining Framework for Automatic Leads Generation

As Figure 3.1 shows, the customer lead generation system consists of three subsystems: data pre-processing, data mining, and output models evaluation. Data preprocessing has been discussed in Chapter 4. This chapter will introduce the core data mining framework. As Figure 5.1 shows, the proposed data mining framework consists of two major parts: classification and association rule mining. Association rule mining section is divided into two modules: Generating Frequent Item sets, and Applying interestingness measure techniques to filter uninteresting rules.

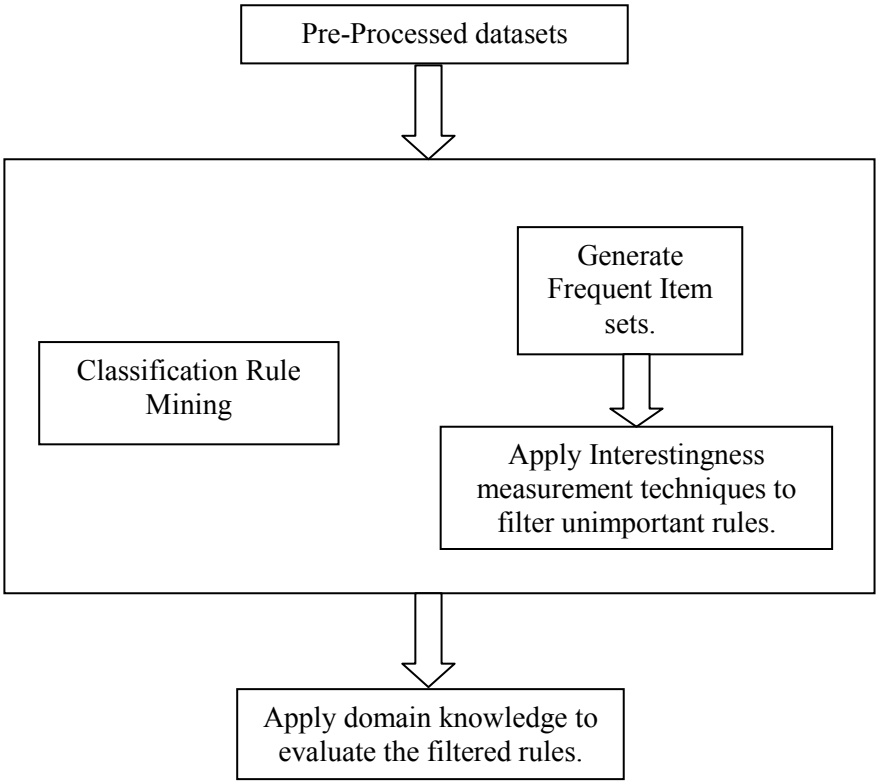


Figure 5.1 A data mining framework for automatic Leads generation.

5.2 Association Rules Mining

Association rules mining, which is one of the widely used, well-researched and proven data mining methods, which was first introduced in [2]. It is also used to discover interesting and frequent patterns, extract correlations, and find associations among sets of items from large scale transactional data. Today's association rules mining can serve a wide range of applications, such as inventory control, risk management, and intrusion detection [18].

In this research, one of the main challenges is to provide effective landing pages to the users so that the lead generation may increase. The available web usage data, which is generated from user activity during the visit of a website, contains user demographic information, landing page information, click-stream data, and users' final decisions about requesting service from the online company. In this case, association rule mining is the best-suited data mining method for producing associations between landing pages and user activities in order to analyze which landing pages are the most effective for which category of users.

At the outset, the framework takes the pre-processed dataset as input and then applies association rule mining algorithms to generate rules. The association rule mining algorithm generates a large volume of frequent item sets and rules; however, most of the rules are not important or interesting to the business owners. To filter out the uninteresting rules, interestingness measure techniques (i.e., Gini, Jaccard, Confidence, etc.) are applied. The major challenge is to make the framework more effective and efficient by finding the best association rule mining algorithm and the right interestingness measure technique to remove the uninteresting rules.

Before applying interestingness measure techniques, all possible association rules are generated from the frequent item sets with a user-defined threshold. There are many algorithms available for association rule mining. In this thesis, four different algorithms are used: Apriori, Eclat, FPGrowth and SaM, which are described in detail in Chapter 3.

Apriori is a classic algorithm for association rule mining which uses breadth-first search on the subset lattice and decides the support of item sets by subset tests. It uses a prefix-tree structure to count candidate item sets efficiently. While Apriori is quick to implement and historically significant, it has been outperformed on almost all datasets by depth-first algorithms such as Eclat and FP-Growth. Eclat uses a depth-first search on the subset lattice and decides the support of item sets by intersecting transaction lists. Unlike Apriori and Eclat, FPGrowth algorithm does not follow the candidate generation process but rather represents the transaction database as a prefix tree, which is improved using links that arrange the nodes into lists pointing to the same item. Another algorithm known as SaM (the shortened form for Split and Merge) uses a combination of a depth-first traversal of the subset lattice and a horizontal transaction representation. The strength of these algorithms are their simple structures. Basically, the frequent pattern sets can be generated by using just a single recursive function and a single array for data structure.

At the second stage of association rules generation, interestingness measure techniques are applied to filter the unimportant or uninteresting rules. Table 5.1 shows the list of interestingness measure techniques used in this experiment, which are described in detail in section 2.5.3. Here, P(X, Y) is equivalent to the support of (X,Y).

Table 5.1 List of interestingness measure techniques.

Measure	Formula	Ranges
Confidence	$\max(\ P(Y\|X), P(X\|Y)\ )$	[0, 1]
Φ-coefficient	$\frac{P(X,Y) - \big(P(X) * P(Y)\big)}{\sqrt{P(X)\ P(Y)\ P(\bar{X})\ P(\bar{Y})}}$	[-1, +1]
IS or Cosine Measure	$\frac{P(X,Y)}{\sqrt{P(X) * P(Y)}}$	[0, 1]

Measure	Formula	Ranges
Lift	$\frac{P(X,Y)}{P(X)*P(Y)}$	[0, ∞]
The odds ratio	$\frac{P(X,Y)P(\bar{X},\bar{Y})}{P(\bar{X},Y)P(X,\bar{Y})}$	[0, ∞]
Cohen's kappa coefficient	$\frac{P(X,Y)+P(\bar{X},\bar{Y})-P(X)P(Y)-P(\bar{X})P(\bar{Y})}{1-\big(P(X)*P(Y)\big)-\big(P(\bar{X})*P(\bar{Y})\big)}$	[-1, 1]
J-measure	$\max\Big(\ P(X,Y)*Log\left(\frac{P(Y\|X)}{P(Y)}\right)+\big(P(X\bar{Y})\big)*Log\left(\frac{P(\bar{Y}\|X)}{P(\bar{Y})}\right),$ $P(X,Y)*Log\left(\frac{P(X\|Y)}{P(X)}\right)+\big(P(\bar{X}Y)\big)*Log\left(\frac{P(\bar{X}\|Y)}{P(\bar{X})}\right)\Big)$	[0, 1]
Gini Coefficient	$\max\big(\ P(X)[\ P(Y\|X)^2+\ P(\bar{Y}\|X)^2\ ]+\ P(\bar{X})[\ P(Y\|\bar{X})^2+\ P(\bar{Y}\|\bar{X})^2-P(Y)^2-P(\bar{Y})],$ $P(Y)[\ P(X\|Y)^2+\ P(\bar{X}\|Y)^2\ ]+\ P(\bar{Y})[\ P(X\|\bar{Y})^2+\ P(\bar{X}\|\bar{Y})^2-P(X)^2-P(\bar{X})]\ \big)$	[0, 1]
Laplace	$\max\left(\ \frac{NP(X,Y)+1}{NP(X)+2},\frac{NP(X,Y)+1}{NP(Y)+2}\ \right)$	[0, 1]
Conviction	$\max\left(\ \frac{P(X)P(\bar{Y})}{P(X\,\bar{Y})},\quad\frac{P(X)P(\bar{Y})}{P(Y\,\bar{X})}\ \right)$	[0.5, ∞]
Piatetsky-Shapiro	$P(X,Y)-P(X)*P(Y)$	[-0.25, 0.25]
Certainty Factor	$\max\left(\ \frac{P(Y\|X)-P(Y)}{1-P(Y)},\frac{P(X\|Y)-P(X)}{1-P(X)}\right)$	[-1, 1]
The added value	$\max\big(\ P(Y\|X)-\ P(Y),\quad P(X\|Y)-\ P(X)\ \big)$	[-0.5, 1]
Collective strength	$\left(\frac{P(X,Y)+P(\overline{XY})}{P(X)P(Y)+P(\bar{X})*P(\bar{Y})}\right)*\left(\frac{1-P(Y)P(X)-P(\bar{Y})*P(\bar{X})}{1-P(X,Y)-P(\overline{XY})}\right)$	[0, ∞]
Jaccard	$\frac{P(X,Y)}{P(X)+P(Y)-P(X,Y)}$	[0,1]

5.2.1 Experiments of Association Rule Mining

In this section, experiments are carried out on the landing page dataset described in section 4.2. As described in Chapter 3, the raw dataset goes through four phases of pre-processing: Data cleaning, Session Identification, Data Integration, and Data Transformation. The final dataset contain 7350 instances and 19 features.

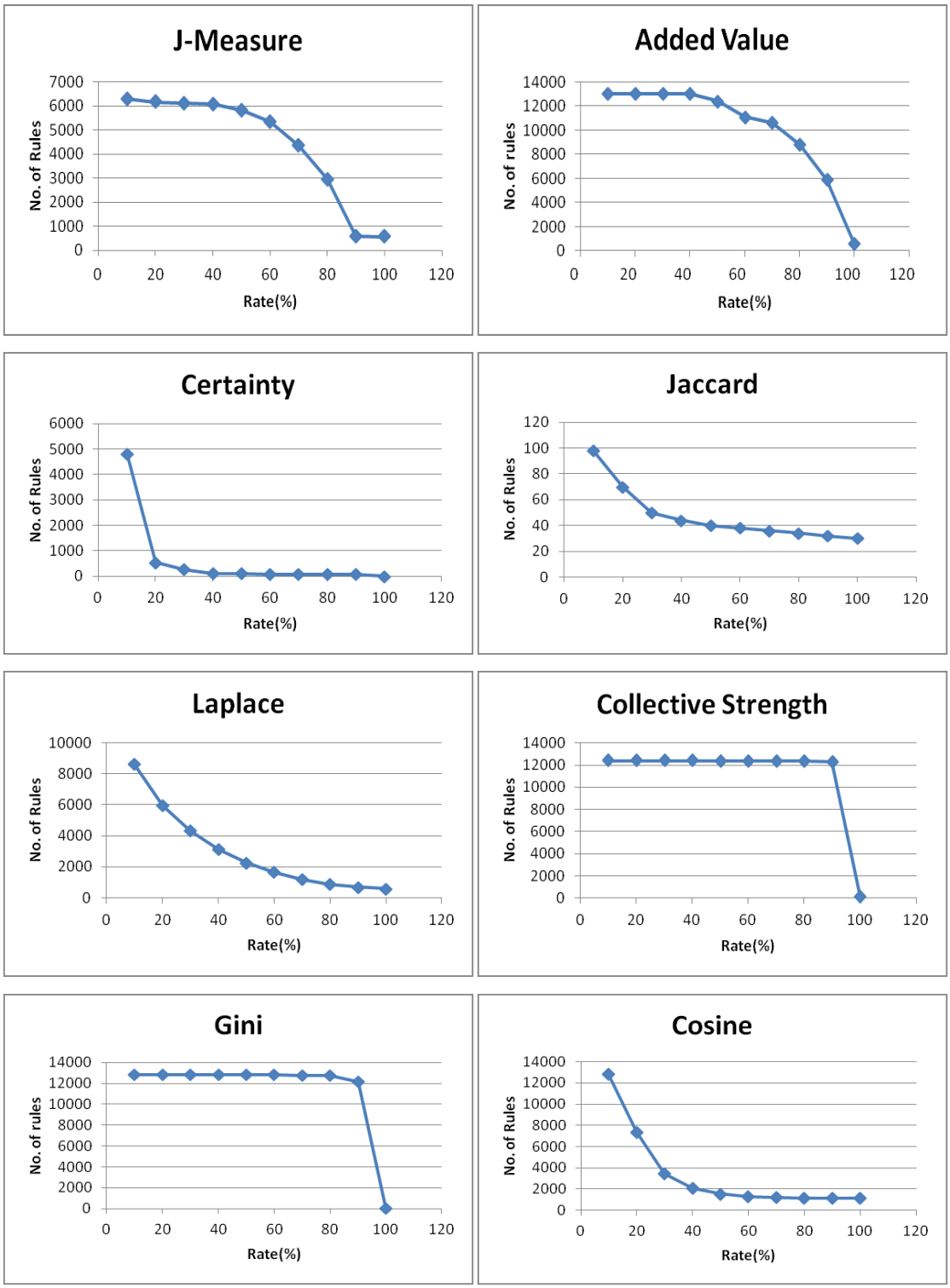
Two experiments are carried out to compare the effectiveness and efficiency of the data mining framework. The first experiment shows a comparison among interestingness measures in terms of generating rules at a certain threshold. The second experiment shows a comparison of the execution time among the combination of algorithms and interestingness measure techniques.

It should be noted here that all of the association rule mining algorithms generate all possible frequent item sets at certain support rates, so the output from the association rule mining algorithms are the same but vary in execution time.

Two parameters are used for association rule mining. The first parameter is “support rate” for generating frequent item sets, and the second parameter is “interesting measure rate” for filtering out uninteresting rules. The support rate is set to 10%, as rules with low support rates might possess high interesting rates. For interestingness measures, five measure rates are used: 5%, 10%, 25%, 50% and 75%.

5.2.2 Experimental result

Figures 5.3 and 5.4 illustrate comparisons among interestingness measure techniques in terms of total number of rules generated at certain thresholds.


J-Measure
No. of Rules
Rate(%)
Added Value
No. of rules
Rate(%)
Certainty
No. of Rules
Rate(%)
Jaccard
No. of Rules
Rate(%)
Laplace
No. of Rules
Rate(%)
Collective Strength
No. of Rules
Rate(%)
Gini
No. of rules
Rate(%)
Cosine
No. of Rules
Rate(%)

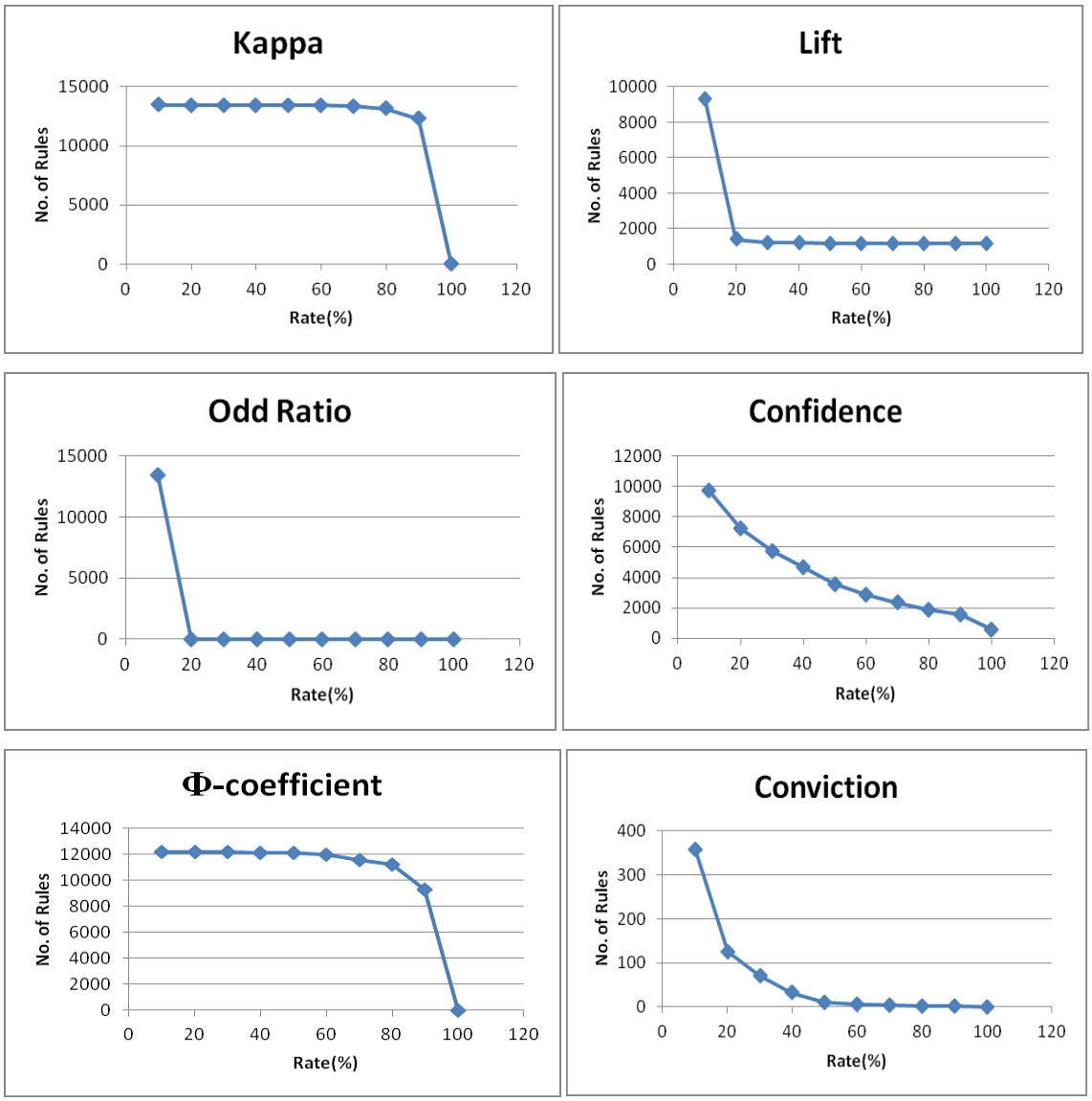


Figure 5.2 Comparision among interestingness measures.

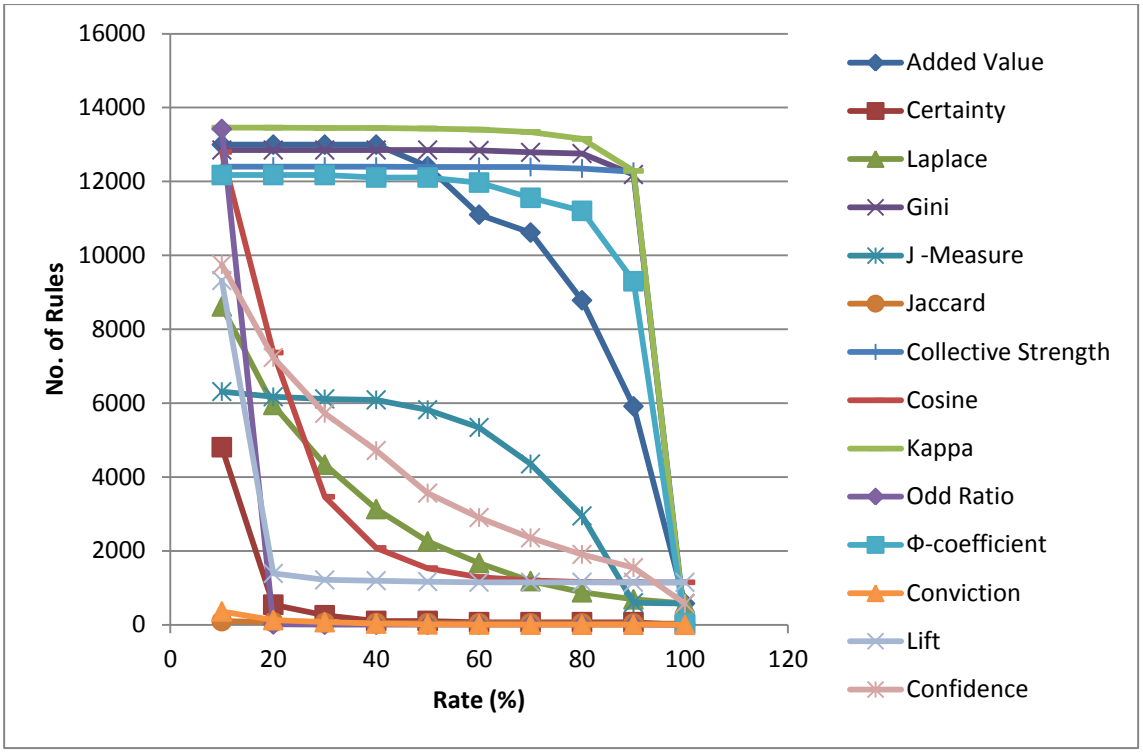


Figure 5.3 Comparisions among interestingness measures in a single graph.

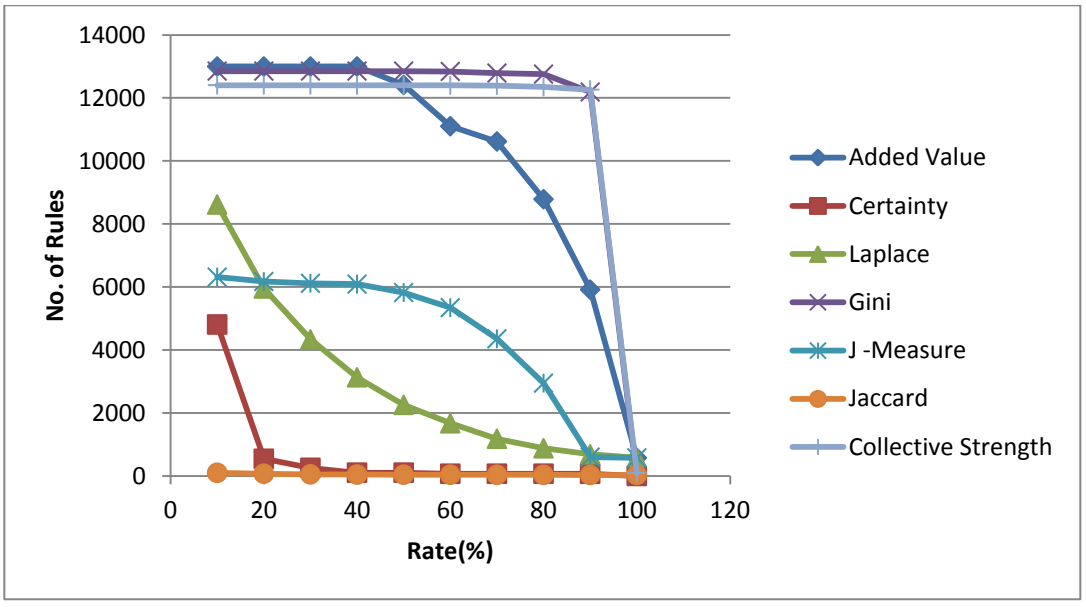


Figure 5.4 Comparisions among selective interestingness measures in a single graph.

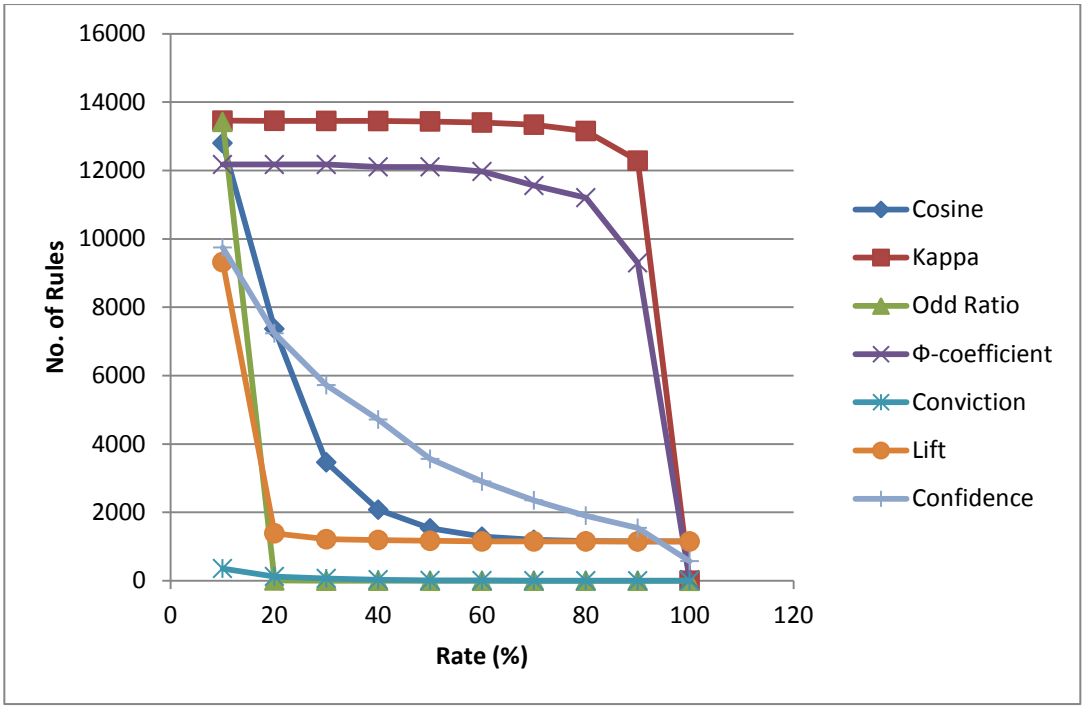


Figure 5.5 Comparisions among selective interestingness measures in a single graph.

The following figures show the comparison of the execution time among the combination of algorithms and interestingness measure techniques with interestingness measure rates 5%, 10%, 25%, 50% and 75% consecutively.

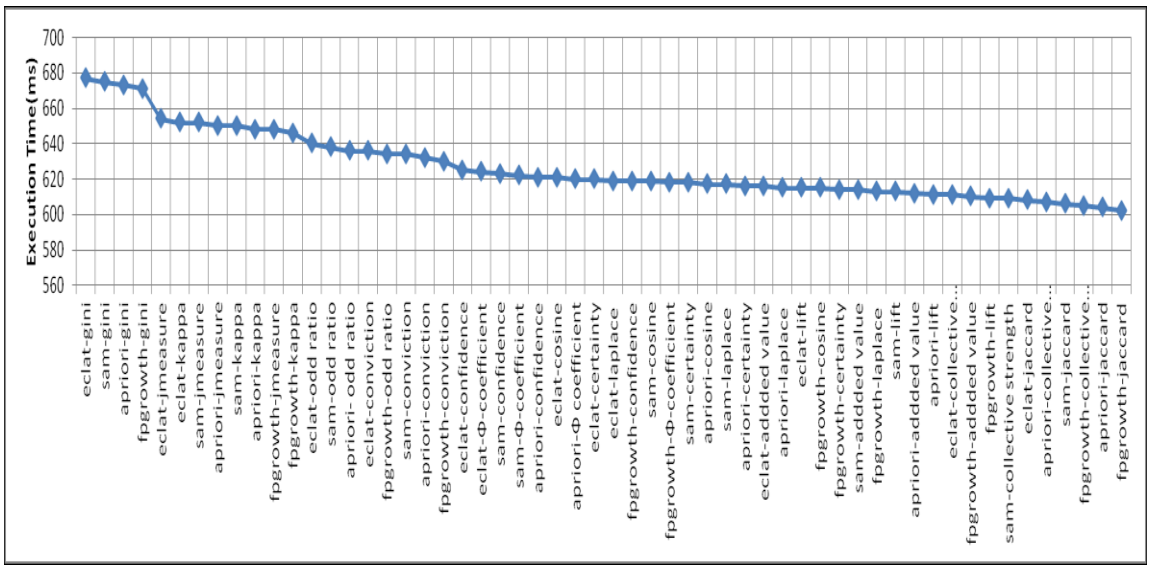


Figure 5.6 Comparison of execution time with a **5%** interestingness measure rate.

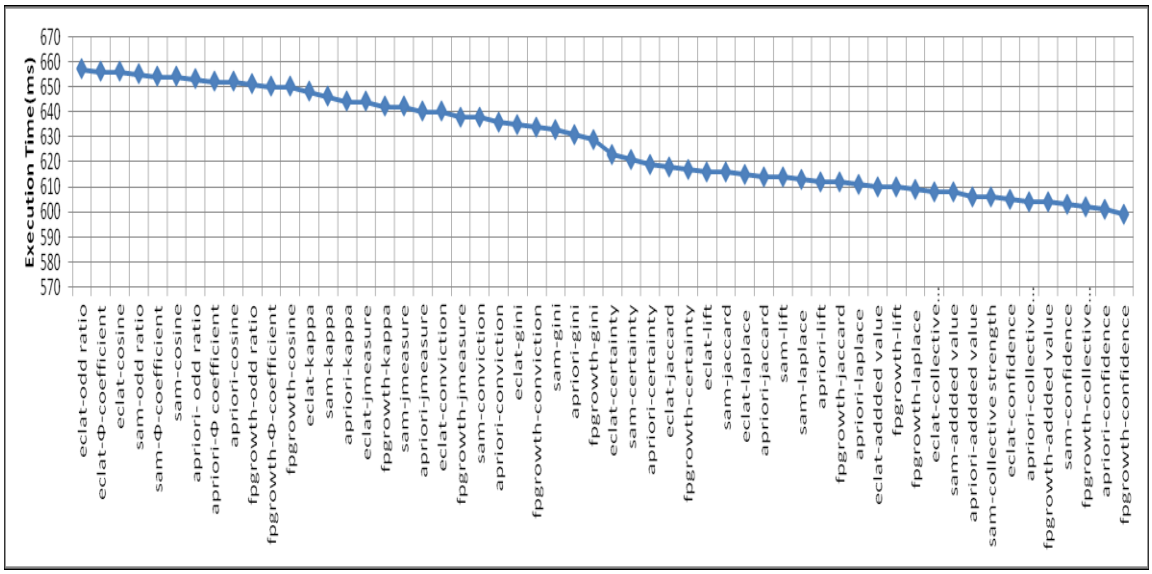


Figure 5.7 Comparison of execution time with a **10%** interestingness measure rate.

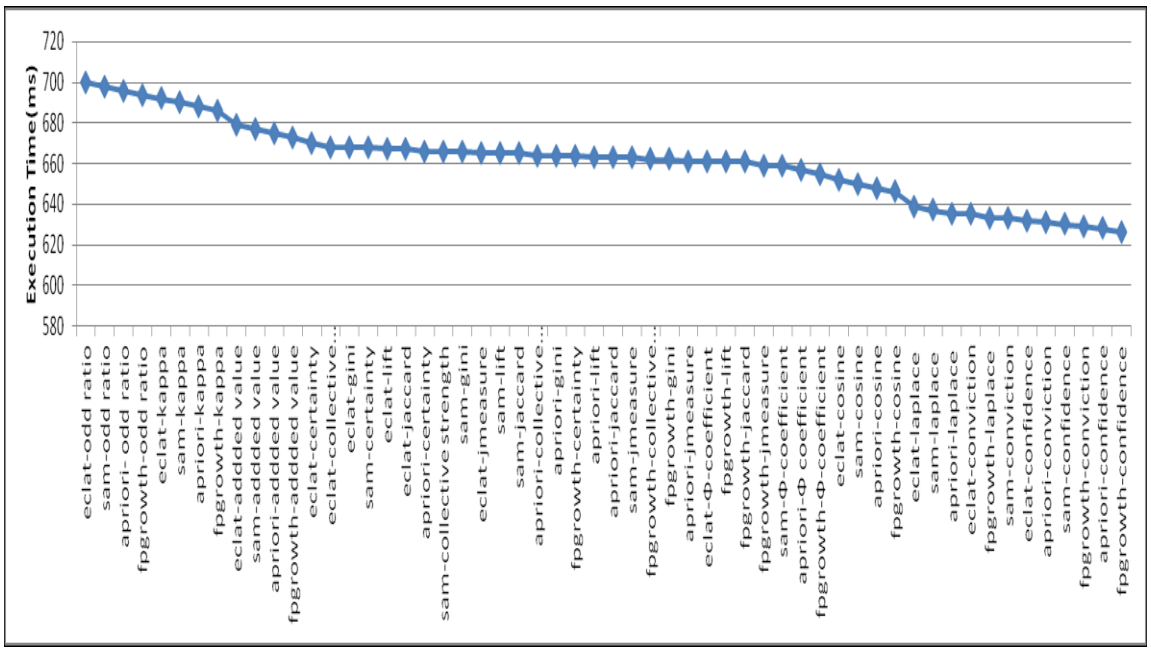


Figure 5.8 Comparison of execution time with a **25%** interestingness measure rate.

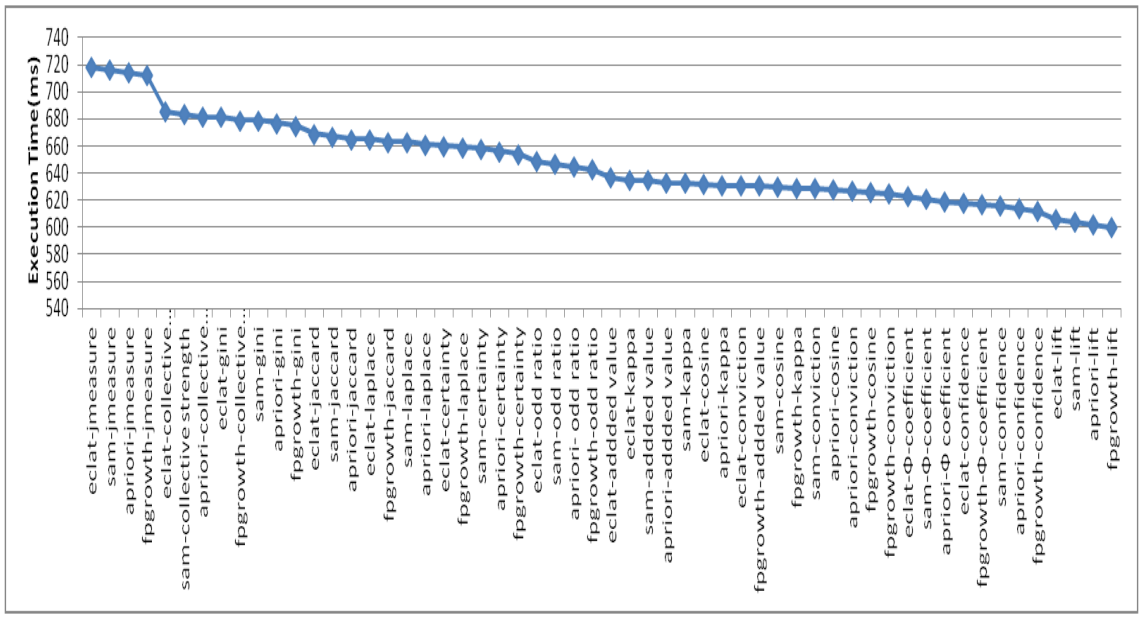


Figure 5.9 Comparison of execution time with a **50%** interestingness measure rate.

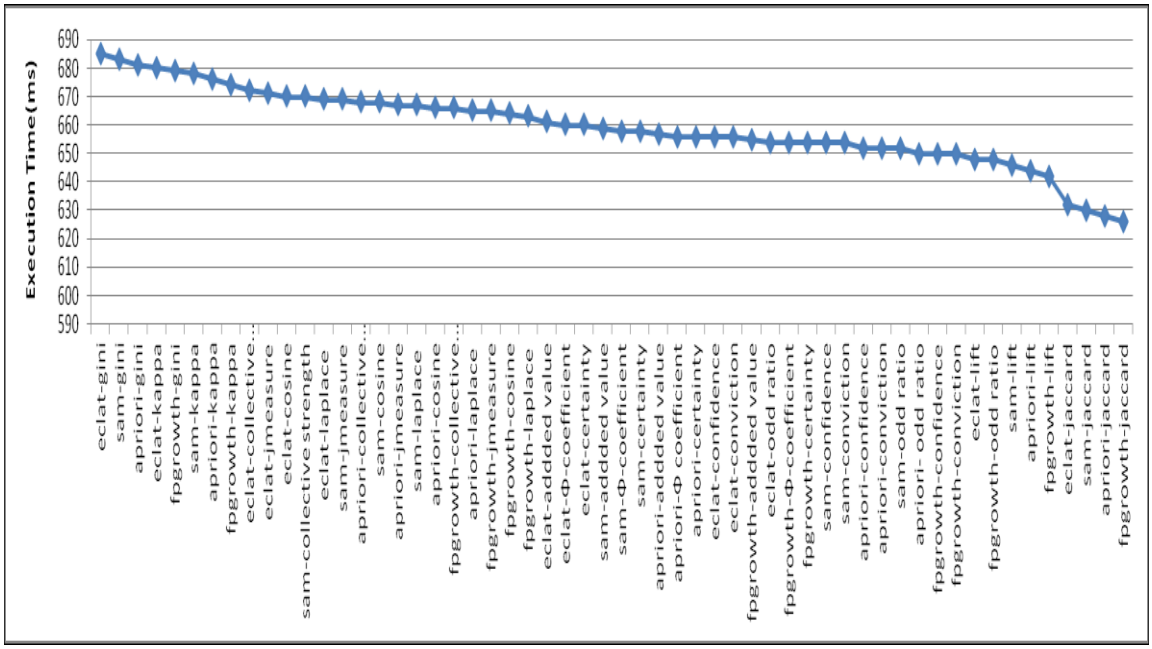


Figure 5.10 Comparison of execution time with a **75%** interestingness measure rate.

The first experiment in Figure 5.3, shows that the Kappa measure generates the most number of rules at a 50% threshold. The other measures that also show close results are Gini, Added Value, Φ-coefficient, and Collective Strength. These four measures will now be referred to as Group 1 for the purpose of discussion. On the other hand, Lift, Conviction, Laplace, Confidence, Odd ratio, Jaccard, and Certainty measures show sharp decreases in generating rules as the measure rate increases.

The second experiment shows the comparison of execution time among the combination of algorithms and interestingness measure techniques. Figure 5.6 illustrates that the combination of association rule mining algorithms with “Gini” measure have the highest execution time at the 5% threshold. Therefore, there is a trade-off between generating number of rules and execution time. The combination of FPgrowth and Kappa measure shows the lowest execution time among all combinations that include Kappa measure. The combination of FPgrowth and Collective strength has the lowest execution time among the Group 1 measures.

Figure 5.7 shows the comparison at the 10% threshold, illustrating slightly different results from the previous result at the 5% threshold. The combinations of Sam and Φ-coefficient measure have the highest execution rate. Otherwise, it has similar findings. For instance, the FPgrowth and Kappa measure shows the lowest execution time among all combinations that include Kappa measure. Also, the combination of FPgrowth and Collective strength has the lowest execution time among the Group 1 measures.

The rest of the figures show comparisons at the 25%, 50% and 75% thresholds. The results reveal that the combination of FPgrowth and Φ-Coefficient has the lowest execution time among the Group 1 measures, and that the combination of FPgrowth and Kappa measure shows the lowest execution time among all combination that include the Kappa measure.

In summary, the Group 1 measure, which includes the Kappa, Gini, Added Value, Φ-coefficient and Collective Strength, shows a better performance in terms of generating association rules from frequent item sets, and the Kappa measure shows the best performance among Group 1 measures.

At the 5% and 10% thresholds, the combination of Kappa and FPGrowth shows the best performance in terms of number of rules generation. On the other hand, the combination of FPgrowth and Collective strength shows a balanced performance in terms of execution time and number of rules.

At the 25%, 50% and 75% thresholds, the combination of Kappa and FPGrowth still shows the best performance in terms of number of rules generation. On the other hand, the combination of FPgrowth and Φ-coefficient shows a balanced performance in terms execution time and number of rules.

5.2.4 Analysis of Association Rules:

The following results are extracted from the Lead user data set. These rules show the

correlation among attributes for Lead user data sets. Table 5.2 summarizes the actions that could be suggested based on the association rules.

Table 5.2 Extracted association rules and the actions or recommendations.

	Association rules	Possible recommendations
1.	**pageview**=pa11 **==> page_view**=page1, **pricerange**=200, **intent**=buy supp:(1.13%), conf:(1.0)	Landing page "pa11" should be exposed to the users whose price range is below 200k with intention to buy as well as visit the "page1" before visiting Landing page.
2	**pageview**=pa11 ==> **address**=AB, **pricerange**=200-400,**ad_type**=ad3 supp:(3.06%), conf:(0.85)	Landing page "pa11" should be exposed to the users whose price range is 200-400k, address is "AB" and visit the website through "ad3" advertisement.
3	**pageview**=pa10 ==> **address**=ON, **page_view**=page1, **ad_type**=ad1 supp:(1.30%), conf:(0.95)	Landing page "pa10" should be exposed to the users whose address is "ON" and visit the "page1" through "ad1" advertisement.
4	**pageview**=pa10 ==> **address**=BC, **page_view**= page1 supp(1.67%), conf:(0.91)	Landing page "pa10" should be exposed to the users whose address is "BC"and visit the web page "page1".
5	**pageview**=pa10, **pricerange**=200-400, **intent**=buy **==> page_view**=page3 supp:(4.53%), conf:(0.80)	Landing page "pa10" should be exposed to the users whose price range is "200-400k" with buying intention and visit the web page "page3".
6	**pageview**=pa6 ==> **choice**=rewards1 **address**=BC supp:(0.75%), conf:(1.0)	Landing page "pa6" should be exposed to the users whose address is "BC" and choose the reward option "rewards1".
7	**pageview**=pa6, **address**=NL ==> **pricerange**=200 supp: (2.01%), conf:(0.91)	Landing page "pa6" should be exposed to the users whose address is "NL" and

		price range is below 200k.
	Association rules	Possible recommendations
8	**pageview**=pa5 ==> **address**=AB, **page_view**= page2, **ad_type**= ad2 supp: (0.63%), conf:(1.0)	Landing page "pa5" should be exposed to the users whose address is "AB" and visit the website through advertisement version "ad2".
9	**pageview**=pa5 ==> **choice**=rewards2 supp: (5.33%), conf:(0.73)	Landing page "pa5" should be exposed to the users who choose the reward option "rewards2".

The rules presented here are related to landing page information, as the main objective of this framework is to find the most effective landing pages for the various groups of users so that leads generation increase. Here, the "pageview" attribute represents the "landing page" version number. The first two rules show the information of landing page "pa11"; rules no. 3 to 5 show the information of landing page "pa10"; rules no. 6 to 7 show the information of landing page "pa6"; and rules no. 10 to 12 show the information of landing page "pa5".

The first rule shows that the landing page "pa11" has a 100% lead generation rate for the users whose intent is buying, whose price range is 200K or less, and who visit the website's page "page1". The second rule shows that it has a 85% lead generation rate for users whose address is "AB", price range is 200-400K and who visited the site through "ad3" advertisement.

The third and fourth rules show that the landing page "pa10" has an above 90% lead generation rate for users whose address is "BC" and "ON" and who visited the webpage "page1". The fifth rule shows that it has an 80% lead generation rate for users whose intent is buying, whose price range is 200-400K, and who visited the webpage "page3".

The sixth rule shows that the landing page "pa6" has a 100% lead generation rate for users whose address is "BC" and who choose the reward option "reward1". Rule no. 7

shows that the same landing page has a 91% lead generation rate for users whose address is “NL” and price range is 200K.

Rule no. 8 shows that the landing page “pa5” has a 100% lead generation rate for users whose address is “AB”, who viewed the webpage “page2” and visited the website through advertisement “ad2”. Rule no. 9 shows that it has a 90% lead generation rate for users whose intent is buying, whose price range is 200K, and who viewed the webpage “page1”. The last rule shows that it has a 73% lead generation rate for users who choose the reward option “rewards2”.

5.2.5 The Quality of Association Rules and The Interestingness Measures

The interestingness measure of an association rule can be divided into two categories: Subjective and Objective. The subjective measure completely relies on the users’ knowledge, goals and beliefs. Users apply their own knowledge to capture the novelty and surprisingness of the rules. On the other hand, objective measures are evaluated by numerical indexes which depend on data distribution. In this thesis, the scope of evaluating the quality of an association rule is limited to objective measures based on numerical index. According to experiments on the current datasets, the Φ-coefficient measure, which is mainly used as a measure of the strength of linear dependence between two variables, and the kappa measure, which is a statistical measure of inter-rater agreement or for qualitative (categorical) items, performs better than other interestingness measures. However, this observation is based on current data sets and might vary with different types of datasets.

5.3 Classification

Decision tree is a simple but powerful knowledge representation method of multiple variable analysis. It is also robust to noisy data and improves human readability through learning disjunctive expressions. Surveys in [21] show that decision tree learning is very effective for problems that possess the following characteristics: (1) Objects are represented by attribute-value pairs; (2) the target class are discrete values; (3) disjunctive

descriptions are required; (4) the training data may contains errors or be missing attribute values. This thesis contains all of the characteristics stated above. As a result, decision tree was chosen for the data mining framework.

As section 2.5.1.1 mentioned, C4.5 is a landmark in decision tree learning, and is robust in handling both continuous and discrete attributes, as well as missing values and pruning tree. On the other hand, according to surveys in [39], improvements of this algorithm, such as C5.0, are rarely recognized as being considerable improvements in precision over a wide spectrum of diverse datasets. Therefore, C4.5 is a good choice for this thesis.

5.3.1 Experiments and Results of Classification

In the first experiment, the decision tree algorithm C4.5 with 10-Fold Cross-validation [17] is applied on the user datasets which contain information about all landing page visits. The generated decision tree shown in Figure 5.11 is a top-down hierarchy tree, where the left-most nodes represent the top decision nodes and right-most nodes represent the classification results as leaf nodes. Each node in between ends stands for a particular attribute chosen by the algorithm for making an interim decision. Moreover, each branch of the tree represents a rule made up by sequential decisions. All of the nodes or attributes are marked as bold words followed by an equal sign and the value of each attribute. The right-most value represents the target class of each rule, which is either Lead or Non-Lead.

At each node, the data set is split into subsets based on the values of the attribute at the node. The top node contains the most information about the target classification, the second node contains the second most informative information about the classification at that level, and so on.

pricerange = 200 : Non Lead (0.70)
pricerange = 200-400
| **pageview** = pas5 : Non Lead (0.95)
| **pageview** = pas6 : Lead (0.8)
| **pageview** = pas7 : Lead (1.0)
| **pageview** = pa11
| | **ad_name** = ON : Non Lead (1.0)

```
|       |       ad_name = AB : Non Lead   (0.75)
|       |       ad_name = BC : Lead   (1.0)
|       |       ad_name = NL : Non Lead   (1.0)
|       |       ad_name = unknown : Lead   (1.0)
|       pageview = pa10
|       |       intent = intent_not_available :  Non Lead  (1.0)
|       |       intent = buy-sell: Non Lead  (0.75)
|       |       intent = buy : Lead   (0.70)
|       |       intent = sell : Non Lead  (1.0)
|       |       intent = unknown : Non Lead  (1.0)
|       pageview = mort0 : Lead   (0.61)
pricerange = 400-600
|       iplocation = ON : Non Lead  (1.0)
|       iplocation = NS : Lead   (1.0)
pricerange = 600-800 : Lead   (1.0)
pricerange = 800-1000 : Non Lead  (0.66)
pricerange = 1000
|       error = no_error : Non Lead  (1.0)
|       error = yes : Lead   (1.0)
pricerange = pr_not_available : Non Lead  (0.99)
pricerange = unknown : Non Lead  (0.98)
```

Figure 5.11 Decision Tree for landing page users.

The examples classification rules for leads:

1. if **pricerange** = "200-400" and **pageview** = "pa10" and **intent** = "buy" then **Class** = "Lead" (0.70)

2. if **pricerange** = "200-400" and **pageview** ="mort0" then **Class** = "Lead" (0.61)

3. if **pricerange** = "200-400" and **pageview** ="pas6" then **Class** = "Lead" (0.8)

4. if **pricerange** = "200-400" and **pageview** ="pas7" then **Class** = "Lead" (1.0)

5. if **pricerange** = "200-400" and **pageview** ="pa11" and **ad_name** = BC then **Class** = "Lead" (1.0)

6. if **pricerange** = "400-600" and **iplocation** ="NS" then **Class** = "Lead" (1.0)

7. if **pricerange** = "600-800" then **Class** = "Lead" (1.0)

8. if **pricerange** = "1000" and **error** =" no_error" then **Class** = "Lead" (1.0)

The examples of classification rules for Non-Lead Users:

1. if **pricerange** = "200" then **Class** = "Non Lead" (0.70)
2. if **pricerange** = "200-400" and **pageview** = "pas5" then **Class** = "Non Lead" (0.95)
3. if **pricerange** = "200-400" and **pageview** = "pas5" then **Class** = "Non Lead" (0.95)
4. if **pricerange** = "200-400" and **pageview** = pa10 and **intent** = sell then **Class** = "Non Lead" (1.0)
5. if **pricerange** = "200-400" and **pageview** = pa10 and **intent** = buy-sell then **Class** = "Non Lead" (0.75)
6. if **pricerange** = "200-400" and **pageview** = pa10 and **intent** = intent_not_available then **Class** = "Non Lead" (1.0)
7. if **pricerange** = "200-400" and **pageview** = "pa11" and **ad_name** = ON then **Class** = "Non Lead" (1.0)
8. if **pricerange** = "200-400" and **pageview** = "pa11" and **ad_name** = AB then **Class** = "Non Lead" (0.75)
9. if **pricerange** = "400-600" and **iplocation=** "ON" then **Class** = "Lead" (1.0)
10. if **pricerange** = "800-1000" then **Class** = "Non Lead" (0.66)
11. if **pricerange** = "1000" and **error** = no_error then **Class** = "Non Lead" (1.0)
12. if **pricerange** = "pr_not_available" then **Class** = "Non Lead" (0.99)

5.3.2 Analysis of Classification Rules:

The decision tree provides rules that are related to landing page version, ad name, ip location, price range, user intent to buy or sell, etc., which can be used as guidelines for selecting more productive landing pages to convert a user into a lead. Here, the landing page version is used as an example to show how a decision tree can help design landing pages. Currently, there are nine landing pages available: pas5, pas6, pas7, pas8, pa11, pa10, mort0, real10 and real11. These landing pages can be divided into three sections to

serve different purposes: pas5, pas6, pas7, pas8 are designed to attract users who are interested in the company's rewards program; and pa11 and pa10 are designed to serve all categories of users. Finally, mort0, real10 and real11 are designed to attract users who are interested in mortgages.

More than half (61%) of the users who visited the mortgage landing page "mot0" became leads. However, the decision tree did not generate any rules for real10 and real11, which is also a mortgage-related landing page. This means that these two pages do not have any effect on generating lead users, which indicates that the mortgage landing page mort0 is performing well to convert mortgage users into leads, while real10 and real11 need to be reviewed or improved for better performance.

The two rules below indicate that landing page 10 performs very well (a 70% lead generation) for users who want to buy homes within a price range of $200K-$400K ("200-400"). On the other hand, the landing page performs very poorly (a 100% no-lead generation) for users within the same price range but with the intent to sell a home rather than to buy one.

1. if **pricerange** = "200-400" and **pageview** = "pa10" and **intent** = "buy", then **Class** = "Lead" (0.70)
2. if **pricerange** = "200-400" and **pageview** = pa10 and **intent** = sell, then **Class** = "Non Lead" (1.0)

The rules detailed below are related to landing page 11, which performs well with advertisements targeting users from "BC" (a 100% lead generation rate). On the other hand, the same landing page has a very poor performance for advertisements related to "ON" and "AB" users.

1. if **pricerange** = "200-400" and **pageview** ="pa11" and **ad_name** = BC, then **Class** = "Lead" (1.0)
2. if **pricerange** = "200-400" and **pageview** = "pa11" and **ad_name** = ON, then **Class** = "Non Lead" (1.0)
3. if **pricerange** = "200-400" and **pageview** = "pa11" and **ad_name** = AB, then **Class** = "Non Lead" (0.75)

These results indicate that for users from “AB” and “ON”, the landing page needs to be modified on a trial-and-error basis to increase lead generation.

Regarding choice rewards landing pages, version 7 performs the best, with 100% of visitors turning into leads. This landing page has detailed information about the rewards programs and also shows the actual amounts of points users can get if they buy or sell a house. Interestingly, landing page versions 5 and 6, which share similar designs, perform quite differently. Version 5 turns 95% of its visitors into non-leads, but version 6 turns 80% of its visitors into leads. The only difference between versions 5 and 6 is the instruction sentence. Instead of “Pick your points program to get started!”, as used in version 5, version 6 uses the sentence “What points program do you use?” as the instruction sentence. This fact shows that a small design pattern may affect the performance of a landing page in terms of lead productivity. The decision tree can help to identify which design pattern works better than others.

5.3.3 Important Factors for Lead Generations for the Real Estate Service Provider:

The decision tree provides several interesting factors which are important to determine whether a visit is a lead or not. One such important factor is price range of the visitor. The current business scenario contains various price ranges for example, <200k, 200-400k, 400-600k etc. The results show that some landing pages performed well with particular price range in converting visitors into leads. On the other hand some landing pages show poor performance with particular price range in converting visitors into leads. Another important factor that plays vital role in converting visitors into lead is the intention of visitor whether they are interested in buying property or selling property. For example, landing page version “pa10” works good for visitors who intent to buy rather visitors who intent to sell. The visitors location also plays important role in determining lead users for example, particular landing pages perform well with certain combination of users’ location and their price range. Lastly, the text of the various offer in the landing page should be considered as important factor while designing the landing pages, it is observed that two landing pages which only differ by the text of the offer perform differently in

terms of converting visitors into leads. The findings are only based on the datasets that is used in this research for experiments, If more data were to be used, other factors might be more important and some of the important factors that is described here might play less important roles.

Chapter 6 **Conclusion and Future Work**

6. 1 Conclusion

In this thesis, a symmetric approach and prototype of an automatic lead generation system for online real estate services is developed. The system has been evaluated by exploiting various data mining techniques and comparing them through experiments. At the end, best data mining techniques are adopted suitable for customer leads generation.

The study worked with a new type of business scenario – real estate service providers. The main business model for this type of company is to attract customers by providing effective landing pages with proper information. This customer lead generation system is capable of identifying effective landing pages on websites to maximize lead conversion. Data mining techniques such as classification and association rule mining are employed for identifying the proper landing pages. The results of classification rules can tell which groups of users are more likely turned into leads, so that the corresponding pages/ads on the website may be designed to attract those users. The association rules technique provides useful information for webpage designers, in that the togetherness of certain association patterns is important, as are correlations among attributes in terms of desired target analysis.

This framework includes data model analysis and design for online real estate service providers, automatic data integration methods for multiple online web data streams, and data mining solutions for lead pattern discovery and lead prediction. The experiments, carried out on a real-world online real estate service provider's dataset, demonstrate that the proposed system is able to empower online real estate service companies to quickly and effectively generate targeted customer leads, which may have significant potential on lead generation for other online businesses as well.

6.2 Future Work

The following area is of possible research interest to improve the automatic lead generation system.

- In this thesis, we have applied the data mining techniques only on web click-stream data. In the next phase, web page content and web click-stream data can be combined for lead generation.
- A knowledge base of lead generation can be developed and updated periodically to dynamically update the knowledge in order to best support business practices.
- This system can be extended to a real-time lead generation system that could select the most effective landing page by matching user requests with the existing knowledge base.

References

[1] C.C. Aggarwal and P.S. Yu. A new framework for itemset generation, in: Proceedings of the 17th Symposium on Principles of Database Systems, Seattle, WA, June 1998, pp. 18–24.

[2] R. Agrawal, T. Imielienski, and A. Swami. Mining association rules between sets of items in large databases. Proceeding of the SIGMOD Conference, pp.207–216. ACM Press, New York, NY, USA 1993.

[3] A. Agresti, Categorical Data Analysis, Wiley, New York, 1990.

[4] S. J. Bigelow. "Lead", August 2007. <http://searchitchannel.techtarget.com/definition/lead> [accessed October 1, 2011].

[5] C. Borgelt and X. Wang. SaM: A Split and Merge Algorithm for Fuzzy Frequent Item Set Mining. Proceeding of 13th Int. Fuzzy Systems Association World Congress and 6th Conf. of the European Society for Fuzzy Logic and Technology, pp.968-973 IFSA/EUSFLAT Organization Committee, Lisbon, Portugal 2009.

[6] C. Borgelt. Simple Algorithms for Frequent Item Set Mining. Advances in Machine Learning II (Studies in Computational Intelligence 263) , 351-369. Springer-Verlag, Berlin, Germany 2010.

[7] C. Borgelt and X. Wang. (Approximate) Frequent Item Set Mining Made Simple with a Split and Merge Algorithm. Scalable Fuzzy Algorithms for Data Management and Analysis: Methods and Design, Chapter 10 IGI Global, Hershey, PA, USA 2009.

[8] L. Breiman, J. Friedman, R. Olshen and C. Stone. Classification and Regression Trees, Chapman & Hall, New York, 1984.

[9] S. Brin, R. Motwani, J. Ullman and S. Tsur. Dynamic itemset counting and implication rules for market basket data, in: Proceedings of 1997 ACM-SIGMOD International Conference on Management of Data, Montreal, Canada, June 1997, pp. 255–264.

[10] R. E. Bucklin and C. Sismeiro. A Model of Web Site Browsing Behavior Estimated on Clickstream Data, Journal of Marketing Research, Volume 40, Issue 3, pp. 249-267, American Marketing Association, 2003.

[11] P. Clark and R. Boswell. Rule induction with cn2: some recent improvements, in: Proceedings of the European Working Session on Learning EWSL-91, Porto, Portugal, 1991, pp. 151–163.

[12] D. S. Coppock. "Data Modelling and Mining: Why Lift?", June 2002. <http://www.informationmanagement. com/news/5329-1.html> [accessed February 6, 2012].

[13] M. R. Fazlollah. (1961, 1994). An Introduction to Information Theory. Dover Publications, Inc., New York. ISBN 0-486-68210-2.

[14] L. Geng, H. J. Hamilton, Interestingness measures for data mining: A survey, ACM Computing Surveys (CSUR), Volume 38, Issue 3, Article 9, 2006.

[15] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann publishers, 2001.

[16] J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. Proceeding of Conf. on the Management of Data (SIGMOD'00, Dallas, TX), 1-12 ACM Press, New York, NY, USA 2000.

[17] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. pp. 1137–1143, 1995.

[18] S. Kotsiantis and D. Kanellopoulus. Association rules mining: A recent overview. International Transactions on Computer Science and Engineering Journal, 2006. 32, 1, pp. 71-82.

[19] S.B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, Informatica 31(2007) 249-268, 2007.

[20] B. Liu. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, Chapter 12, pp. 527-594, Springer Series on Data-Centric Systems and Applications, 2007.

[21] T. M. Mitchell. Machine Learning. WCB/McGraw-Hill, 1997.

[22] F. Mosteller. Association and Estimation in Contingency Tables. Journal of the American Statistical Association. pp. 1–28, 1968.

[23] G. Piatetsky-Shapiro. Discovery, analysis and presentation of strong rules, in: G. Piatetsky-Shapiro, W. Frawley (Eds.), Knowledge Discovery in Databases, MIT Press, Cambridge, MA, 1991, pp. 229–248.

[24] D. V. D. Poel and W. Buckinx. Predicting online-purchasing behavior, European Journal of Operational Research, Volume 166, Issue 2, 16 October 2005, pp.557-575.

[25] J. R. Quinlan. C4.5 Programs for Machine Learning. Morgan Kaufman Publishers, 1993.

[26] J. R. Quinlan. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106.

[27] J. R. Quinlan. Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996.

[28] G. Ramakrishnan., S. Joshi, S. Negi, R. Krishnapuram, and S. Balakrishnan. Automatic Sales Lead Generation from Web Data, Proceedings of the ICDE Conference, pp.101, 2006.

[29] C.J. van Rijsbergen, Information Retrieval, 2nd Edition, Butterworths, London, 1979.

[30] S. Sahar and Y. Mansour. An empirical evaluation of objective interestingness criteria, in: SPIE Conference on Data Mining and Knowledge Discovery, Orlando, FL, April 1999, pp. 63–74.

[31] J. C. Seibel, Y. Feng and R. L. Foster, inventor; 2006, Feb. 21. Web-Based system and method for archiving and searching participant-based internet text sources for customer lead data. United States patent US 7,003,517.

[32] J. C. Seibel, Y. Feng and R. L. Foster, inventor; 2006, May. 09. Web-Based customer lead generator system with pre-emptive profiling. United States patent US 7,043,531.

[33] E. Shortliffe and B. Buchanan. A model of inexact reasoningin medicine, Math. Biosci. 23 (1975) 351–379.

[34] P. Smyth, R.M. Goodman, Rule induction using information theory, in: Gregory Piatetsky-Shapiro, William Frawley (Eds.), Knowledge Discovery in Databases, MIT Press, Cambridge, MA, 1991, pp. 159–176.

[35] J. Srivastava, R. Cooley, M. Deshpande and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. ACM SIGKDD Exploration, Newsl. vol. 1, issue 2, pp. 12—23, 2000.

[36] J. Strijbos. R. Martens, F. Prins and W. Jochems. "Content analysis: What are they talking about?". Computers & Education 46: 29–48, 2006.

[37] P.N. Tan and V. Kumar, Interestingness measures for association patterns: a perspective, in: KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining, Boston, MA, August 2000.

[38] E. Weintraub. Writing Purchase Offers in a Buyer's Market, <http://homebuying.about.com/od/offersnegotiations/tp/BuyersMKTOffers.htm> [accessed October 2, 2011].

[39] I. H. Written and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann Publishers, 2000.

[40] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New Algorithms for Fast Discovery of Association Rules. Proceeding of 3rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'97, Newport Beach, CA), pp.283-296 AAAI Press, Menlo Park, CA, USA 1997.

[41] M. J. Zaki. Scalable algorithms for association mining. IEEE Transactions on Knowledge and Data Engineering, 12(3):372-390, May-June 2000.

[42] X. Zhang, W. Gong and Y. Kawamura. Customer behavior pattern discovering with web mining, Lecture Notes in Computer Science, vol. 3007, pp. 844–853. Springer-Verlag Berlin Heidelberg 2004.

[43] "Peoplesoft Enterprise Online Marketing", http://www.oracle.com/us/media1/057261.pdf, last accessed in 25 March 2012.

[44] "Lead Generation", http://help.sap.com/saphelp_crm40/helpdata/en/f7/60823a2e5d3469e10000000a114084/content.htm, last accessed in 25 March 2012.

[45] SaS White Paper, "Analytical CRM: Optimizing Your Customer Initiative for Maximum ROI", http://www.sas.com/offices/europe/sweden/pdf/Analytical_CRM.pdf, last accessed in 26 March 2012.

[46] IBM Company, "Unica Leads". http://www-142.ibm.com/software/products/us/en/lead-management/, last accessed in 26 March 2012.

[47] "Sales Lead Generation and Management". http://www.activeconversion.com/lead-generation.html, last accessed in 26 March 2012.

[48] Marketo Lead Generation, "Make Sales Happy with More Qualified Leads".http://www.marketo.com/b2b-marketing-software/lead-generation-software.php, last accessed in 26 March 2012.

[49] BuyerZone Company, "The leader in online B2B lead generation", http://www.buyerzone.com/pages/leads/index.html, last accessed in 26 March 2012.