MASTER

Identifying and utilizing contextual information for banner scoring in display advertising

Kliger, V.P.

Award date:
2012

MASTER'S THESIS

Identifying and Utilizing Contextual Information for Banner Scoring in Display Advertising

Author: Vitaly Kliger (0755759)
v.kliger@student.tue.nl

Supervisors: Dr. Jeroen de Knijf
Dr. Mykola Pechenizkiy

SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE

AT
EINDHOVEN UNIVERSITY OF TECHNOLOGY
EINDHOVEN 2012

Abstract

The goal of the current project is to find a way of incorporating contextual information into web analytics. Web-analytics is a relatively new area, which does not have an established definition of context. Hence, literature survey of state of the art technologies as in the area of web-analytics, as in related areas of data mining and recommendation systems was conducted. We propose a definition of context allows for flexibility with defining contextual features from management, and evaluating them with data mining tools.

Then, a framework for studying contextual features and applying them in web-application is proposed. The framework represents a seamless line of data processing with website logs as input and a model as output. The major part of the work is devoted to subgroup discovery algorithm, suitable for web-analytics task. The algorithm is capable of building a model, describing significant regions of the weblog data, describing rather unusual behavior with respect to a property of interest.

The case study is devoted to Kliknieuws.nl, a Dutch online local news company whose underlying business model implies generating revenue by publishing banners on its web pages. Currently, Kliknieuws.nl charges advertisers for impressions only. However, it has an ambition to adjust its business model and charge for clicks too. Hence, the goal of the subgroup discovery algorithm developed is find a set of setting, relevant both to website itself and to external environment, under with probability of click on a banner is higher or lower. Then, Kliknieuws' banner placement algorithm will be adjusted in such a way that pay-per-click banners will be shown to visitors when they are more likely to be clicked, and pay-per-impression banners will be shown in the opposite situation.

We proposed some improvements, relevant to performance and quality of the model, as well as evaluation framework. We also discuss and evaluate diverse phenomena why in this or that situation, people adopted certain behavior with respect to clicking on banners.

Table of Content

List of Figures

List of Tables

1. Introduction

1.1. Motivation

The emergence of the Internet as a both distributional and communicational channel has created the opportunity for a range of online interactions between organizations and customers. These interactions occur during customer activities such as information search for company or product details, using online services such as banking, online purchase or engaging in social networking or participating in online communities or leisure pursuits. The Internet as a business medium continues to rise, as adoption and penetration levels of Internet technology continuously increase. At the same time, advances in mobile technology have created new opportunities for online communication in terms of when and where customers are able to interact online with an organization. These trends have become a driver to growing attention to online customer experience, according to Rose et al. [41], which included measuring website quality and performance; monitoring online customer behavior, particularly in relation to the linked activities of online search and online purchase; and investigating services delivered through the Internet such as online banking, news and weather, travel bookings, education programs and knowledge communities. Since consumers interact with Internet organizations across a diverse range of activities, a number of different behavioral patterns have emerged, leading to different user experiences. All the more, Moynagh and Worsley [42] suggested that the benefits afforded to the online customer change the balance of power within the organization – customer relationship, creating a more powerful and proactive customer. These trends pose to new challenges for online organizations striving to develop and maintain their effectiveness.

The above-mentioned challenges are addressed by web analytics, which is defined by the Web Analytics Association [27] as “the measurement, collection, analysis and reporting of Internet data for purposes of understanding and optimizing web usage”. Being an inherent part of Business Intelligence, web analytics aims at supporting management decisions using browsing and buying user behavior as a source of data. Human behavior as well as factors affecting it have very complicated nature, whereas their reflection on the web, and consequently, available data for web-analytics is very limited. A collection of factors influencing visitor behavior such as location, time, access device, weather, holidays are external towards web analytics applications, yet web applications are sensitive to them. In the literature, this collection of factors is typically referred to as context. User behavior may depend on the context and potentially vary within the context. Thus, complementing the web analytics tools with context management mechanisms are expected to make predictive analytics decisions for web applications more accurate.

Currently, recommendation systems are gaining increasing popularity on the Internet since they allow for providing mass customization in inexpensive way [44]. Two major classes of customization systems include collaborative filtering, which predicts a person’s preferences as a linear, weighted combination of other people’s preferences, and as content filtering, which makes recommendations on the basis of consumer preferences for product attributes. Thus, traditional recommender systems take into consideration two classes of entities, namely humans and recommended items, which can be products or services, and do not analyze any information from the external environment. In various applications, however, it may be important to weight circumstances, relevant to a user, an item, or a combination of them, under which the recommendation transaction is taking place. For instance, a recommendation system of online travel agency may give a vacation package recommendation in the winter which will be very different from the one in the summer if it examines temporal circumstances in which the recommendation process occurs. In the situation of personalized online content recommendation, it may be critical to identify the type of required content and the optimal time for recommendation or delivery. For example, a user may have preference to read local news in the morning on weekdays, financial reports in the evenings, and fashion reviews and to do online shopping on weekends. Hence, incorporating contextual information into the recommendation system can improve its performance.

One of the major sources of revenue of Internet organizations is advertisement. According to Archak et al. [22], the Internet has become a major advertisement medium, while interactive nature of online communication is the factor distinguishing online advertisement from traditional offline. Measuring effectiveness of a particular advertising campaign and allocating the advertising budget optimally was and still remains a very challenging task, yet the Internet made the task easier by connecting ad impressions to tangible user actions and artifacts such as posing a search query, clicking on an ad or converting. The simplicity of measuring and attributing user clicks has established the click-through rate (CTR) and conversion rate (CV) as the current de-facto standard of ad quality.

The current work is focused on optimization of banner placement algorithms. By complementing web-analytics data with contextual information, and applying data mining and machine learning methods, we are striving to build a prediction model, which will estimate the probability of a click on a banner for a given page view. Enriching banner placement algorithms with predicted probabilities of click will allow web applications to automatically select banners for displaying based on desirable CTR from commercial point of view. For example, web applications can show banners with a high click price if predicted CRT is high, and show pay-per-impression banners or banners with low click price when predicted CRT is low.

As a case study, we solve a task of optimization of the banner placement system of Kliknieuws.nl, a Dutch local news website whose underlying business model implies generating revenue by online advertisement. Currently, Kliknieuws.nl charges advertisers for every banner impression. Kliknieuws.nl plans to extend its business model with placement of pay-per-click banners. Thus, it is beneficial for the company to recognize circumstances under which the probability of a click will be higher or lower than average. Supplied with this knowledge, banner selection algorithm will be able to select for displaying a paid-per-click banner in case of high expected probability of click, and paid-per-impression banner in case of low expected probability of click. Kliknieuws.nl has imposed additional restrictions on banner placement system with regards to the minimal and maximal number of impressions per time interval for each banner. If the number of impressions is not met, Kliknieuws has to compensate for remaining impressions. The current banner placement system supports these constrains, while changes, introduced by our algorithm do not affect constraints significantly.

1.2. Research Questions

The current work aims at developing new techniques and tools for business intelligence, namely a framework and corresponding techniques for integrating predictive analytics and context awareness. This will allow online organizations to manage budgets and optimize profits more efficiently and effectively, and reduce chances that their customers are exposed to undesired or irrelevant content. Context awareness is needed in predictive web analytics, since the circumstances under which decisions are made are not static. Integration external explanatory information into the learning process will lead to reducing uncertainty for the learning models.

The case study aims at developing a predictor for optimization of the banner placement system of Kliknieuws.nl, incorporating web-analytics tools with contextual information under current constraints. By predicting if a page view will have high or low probability of click on a banner, the banner placement system will be able to select a banner of the desired payment scheme.

Thus, the main goals of the project are to develop a framework for designing context-aware prediction techniques for web analytics and develop an algorithm for optimization of banner placement systems. We formulate two research questions as follows:

> **Research question 1.** *How can we integrate context awareness into predictive web analytics in order to achieve better user(s) behavior prediction accuracy?*

To be able to integrate context awareness into predictive modeling we have to address the following sub-questions:

1.1. How to define the context (form and maintain contextual categories) in web analytics?
1.2. How to connect context with the prediction process in predictive web analytics?

These questions will be investigated by literature research, and then, validated by the case study. The second research question is devoted to the case study:

> **Research question 2.** *How can we improve the banner selection system by predicting probability of click on a banner using contextual information and quantify the effect?*

To be able to integrate context awareness into predictive modeling we have to address the following sub-question:

2.1. How to improve the banner selection system with a predictive model?
2.2. How to incorporate contextual information into the predictive model?

1.3. Thesis Structure

The work is organized as follows. Preliminaries with overviews of web analytics area, data mining, recommendation systems, and use of context in these areas is presented in chapter 2. Then, a methodology is proposed in chapter 3, explaining details about the algorithm that was developed in this work, while parameters and choices made are discussed at length. Evaluation framework, tailored to the case study is then presented. Details about the case study are discussed in chapter 4 with detailed description of the dataset. Results of experiments are explained in chapter 5. Chapter 6 presents the conclusion.

2. Preliminaries

Techniques used by web analytics are adopted from the data mining field. Hence, the current work aims at studying context in the field that embraces two areas of research: web analytics, and data mining.

2.1. Context

The use of the word “context” tends to be vague because all processes in the world are not isolated, but occur in a certain context. The standard definition given in the modern generic dictionaries is “conditions or circumstances which affect something” [1] or “the interrelated conditions in which something exists or occurs” [24]. Context is a multidisciplinary notion that embraces different areas of research besides computer science. Such disciplines as psychology, philosophy, and linguistics involve concept of context in the academic research in their own areas [2]. While each research area has its own focus of attention and has a tendency to give a preference to its own view that might be different from the other areas, there exist numerous definitions of context depending on the discipline and its subfields.

More than hundred diverse definitions of context taken from various fields were studied and presented in the report of Bazire and Brezillon [2]. In particular, they observe that in contrast to areas where the object of research is usually a person acting in a specific situation such as psychology, in computer science it is rather difficult to determine which context should be considered for the study. The possible alternatives are the contexts of the task, of the person, of the interaction, or of the situation. Depending on the implementation of the recommender system, engagement of these context definitions may be mutually exclusive. It is also typically not trivial to determine beginning and end of the context, as well as real interrelations between the context and cognition. There is no generic solution to the problem formulated by Bazire and Brezillon, hence, each researcher attempts to adopt a custom approach depending on the particular application.

2.2. Web Analytics

2.2.1. Overview

Web analytics has been defined by the Web Analytics Association [27] as “the measurement, collection, analysis and reporting of Internet data for purposes of understanding and optimizing web usage”. This definition encapsulates three main tasks that every Internet organization has to address while doing web analytics, namely measuring quantitative and qualitative data, continuously improving website, and aligning measurement strategy with your business strategy [30].

Every website has some goals, either commercial or not. A goal is any action or engagement that builds a relationship with visitors such as the completion of a feedback form, a subscription request, leaving a comment on a blog post, or viewing a special offers page. This action is more valuable to you than a standard page view. The percentage of visitors completing a target action is called conversion rate [28]. According to Clifton [31] quoting research of E-tailing group (2007) and Nielsen Online (2009), most e-commerce websites fit a model depicted in Figure 1. The figure illustrates that the majority of websites have 2-3% conversion rates, which suggests that there is a high potential for improvement from a user-experience point of view. Clifton also notices that Amazon, with a conversion rate of 17.2 percent reported in January 2009, is often cited as the benchmark standard for optimizing the conversion of visitors to customers. Ultimately, if value (in terms of quality, functionality, etc.) of firm's products or services is not the bottleneck, the online user experience is the main indicator of the success of the firm's website, and web analytics tools provide the means to investigate such user experience.

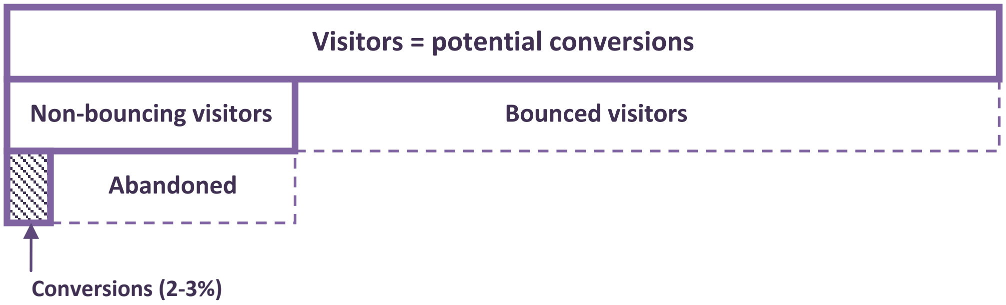


Figure 1 Conversion rates averages of e-commerce websites

After business community realized a commercial potential of the Internet, companies became willing to invest in their online presence. Such investments led to the need for developing a methodology for defining the amount financial and human resources that should be put into online activities. Decision makers are interested in if their website should cater to foreign languages, accept different currencies, develop a separate mobile version, and which channels of reaching the goals are most effective. In order to do online business effectively, an organization needs to continually refine and optimize online marketing strategy, site navigation, page content, and align them with offline business activities. A low-performing website may decrease return on investment or damage firm's brand. Detecting reasons of poor performance of a website is the task of web analytics.

Web analytics operates primarily with metrics which are based rather on technical information that have to be interpreted from firm's business goals point of view. In fact, large number of page views or website engagement metric, often computed as the number of sessions divided by the number of unique visitors, may mean that visitors come repeatedly because they constantly cannot find necessary information or because high quality of content or web services. To answer this questions, and to explain why customers adopt a certain model of behavior or which improvements on the web site have to be made, multiple online and offline feedback tools need to be employed, including surveys, customer ratings, feedback, blog comments, and discussions in social networks.

Web analytics tools can be seen as intermediaries between technical data recorded in website logs and decision-makers of the organization. These tools convert raw log data into metrics aligned with key performance indicators (KPIs), which reflect business goals of the organization. Thus, reported metrics supply management with objective information allowing them to increasing accuracy, efficiency and effectiveness of their decisions. The list of first-level metrics reported by a typical web analytics tool relevant to any, even to noncommercial, website may include the following:

- Number of daily (weekly, monthly) visitors
- The average conversion rate (to sales, registrations, downloads, etc.)

- Most visited pages
- The average duration of visits
- The average visit page depth
- Geographic distribution of visitors
- Bouncing rate (percentage of incidental visitors, leaving the website after viewing one page)

Web analytics tools for e-commerce websites may add metrics reflecting business performance, for example:

- The revenue the site is generating
- Top-selling products
- Referring sources
- The average order value of the top-selling products

Furthermore, the functionality of a typical commercial web analytics tool allows for monitoring more advanced metrics, reflecting key performance indicators of the organization, which may include the following:

- Value of a visitor and its variation on referring source (for example, an organic search, paid search or a referral program)
- Value of a web page
- Comparison of behavior of new and existing customers
- Variation of visits and conversions by referrer type or campaign source
- Variation of bounce rate by page viewed or referring source
- Website engagement (the average number of sessions per visitor)
- Effect of internal site search on hindering conversions
- Number of visits and time span necessary for a visitor to become a customer

All the above-mentioned metrics, as well as many others, are supported by commercial and some free web analytics tools such as Google Analytics, as reported by Clifton [31]. Similarly, Kaushik [30] notices that almost every modern web analytics tool provides a couple of hundred metrics in its default functionality, thus making available an increasingly wide array of complex data, including comparison with industry averages.

Web-analytics can be seen as a black box receiving collected technical data as input and producing metrics aligned with organization’s KPIs as the output. Best practices for collecting the data and reviewing the metrics are explained below.

2.2.2. Data Collection

Most studies on web analytics discuss two common techniques for collecting web visitor data, namely page tags and log files [29], [30], [31]. Log files refer to the data collected by a web server independently of a visitor’s web browser: the web server records browser’s activity to a text file stored locally. Page tags collect the data via the visitor’s web browser and send information to remote data collection servers. This information is usually captured by java script code, known as tags or beacons, and placed on each page of the website. Such technique is known as client-side data collection and is used mostly by outsourced Software as a Service Web analytics tools. In the recent years, page tags get more attention as method for collecting visitor data. It is caused by a larger number of information about visitor which is possible to collect with java script code as opposite to server log files, java script support by mobile devices, as well as the possibility to outsource data management to an external web analytics provider. Each technique has advantages and limitations.

One of the main challenges of web-analytics is visitor tracking. Page tags and log files methods approach this task differently: the first does it by using cookies; the second assumes that all queries from the same IP address with the same browser signature come from the same user. Each method has its limitations affecting data accuracy.

Precision of the log files method is bounded by dynamic IP address assignment technology, possibility of sharing the same IP addresses by multiple computers, caching web pages on the client side, and activities of web crawlers. In fact, Internet Service Providers can assign different IP addresses even throughout one session. Abraham et al. [32] showed that a typical home PC averages 10.5 different IP addresses per month while the number of diverse web browser signatures is limited. Those visits will be counted as 10 unique visitors by a log file analyzer. As a result, the number of visitors may be vastly overcounted.

Client-side caching implies that a previously visited page is stored on a visitor’s computer. In this case, repetitive visit of the same page is not recorded at the web server. Server-side caching can come from any web accelerator technology that caches a copy of a website and serves it from their servers to speed up delivery. This means that all subsequent site requests come from the cache and not from the site itself, leading to a loss in tracking. Currently, most of the Web is in some way cached to improve performance.

Robots, or web crawlers, are used by search engines to fetch and index pages, to check server performance such as uptime or download speed, to do page scraping, including price comparison, e-mail harvesting, and competitive research. These activities affect web analytics data because log files methods are not always capable of distinguishing human actions from actions imitated by a robot. Thus, counting the visitor numbers may not be accurate since robots can constitute a significant proportion of page view traffic. Filtering known IP addresses of web crawlers reduce the problem, but there is no unified solution to remove page views made by the robots. For this reason, log files methods are likely to overcount the number of visitors.

Precision of the page tag method is bounded by possible halting page loading due to java script errors, blocking page tags by firewalls, mobile browsers without java script support, visitors rejecting or deleting cookies. An error in other java script on the page halts the browser scripting engine at that point, so a page tag placed below it does not execute. Depending on configurations, firewalls can also reject or delete cookies automatically. Having analyzed reports of Sun Microsystems Forum statistics, Clifton [29] noticed that among advanced Internet users, one to five percent block tracking cookies, 20%

delete cookies at least once per month, and 83% own or share multiple computers. A mobile web audience study by Abraham et al. [32] showed that in 2007 in the USA, 30 million (or 19%) of the 159 million Internet users accessed the Internet from a mobile device. Before emergence of iPhone and Android, mobile devices did not support java script which restrained extensive use of page tags methods.

User tracking is one of tasks of web analytics. Cookies and IP addresses can be used to determine how many first-time or repeat visitors a site has received, how many times a visitor returns each period, and how much time elapsed between the visits. Furthermore, web servers can also use this information for content personalization. However, an issue of owning or sharing multiple computers remains unresolved for both log files and page tags methods. The following scenarios are possible: one user utilizes multiple computers causing web analytics tools to count each of these anonymous user sessions as unique, different users share the same computer which sometimes implies sharing cookies, different computers share the same IP address, or a mobile user may change IP address during one browsing session. In these scenarios, both log files and page tags methods employed either in isolation or together do not provide a robust solution to track or identify the visitors.

In the scope of the current work, data collection techniques reflect two important limitations, related to quality of source data, and to features of the external environment and the human visitor which can be captured for analysis by the web-application. As mentioned above, web-analytics tools use raw website logs as source data. Consequently, quality of data affects quality of metrics produced by these tools. Problems, referred to incorrect or incomplete raw website logs may be located and pre-processed by a human expert. However, van der Aalst states [45] that this correction can be limited and at some stage one needs to assume that the log contains information of what really happened. Then, not all of the features of the external environment which are desirable to analyze can be easily measured and quantified. This is also true about the human, whose behavior has a complex nature, whereas technical possibilities to grasp it, and consequently, available data for the web-analytics, are limited. Some background information, however, can be obtained such as gender, age, and location may be done by attaching a social network profile to the web-application. Page tagging technique allows for capturing more information about the visitor, however, data obtained from server log files are more accurate and complete. Hence, in the scope of the current work, both techniques have limitations influencing quality of the results.

Latency in making purchase decision and offline activities skew data collection and leave room for inaccuracy. The time it takes for a visitor to be converted into a customer, so called latency, can have a significant effect on accuracy. Higher-value items such as cars, loans, and mortgages may require a longer consideration time before purchase during which there is a risk of the user deleting cookies, reinstalling the browser, upgrading the operating system, or buying a new computer. Any of these occurrences will result in users being seen as new visitors when they finally make their purchase. Offsite factors such as seasonality, adverse publicity, offline promotions, or published blog articles or comments can also affect latency. High-value purchases may also be first researched online and then purchased offline. Connecting offline purchases with online visitor behavior is virtually impossible for web analytics tools. Current best practices overcome this limitation by using online vouchers that customers can print and take with them to claim discount or a free gift. Depending on business area and subject area of the web site, inaccuracy introduced by offline user activities must be taken in consideration while assessing precision of web-analytics data.

In conclusion, both page tagging and log file analysis techniques have their limitations while considered in isolation. The differences are summarized in Figure 2. A common opinion is that page tags are technically superior to other methods since it allows for more flexibility. According to Clifton [29], however, it depends on metrics or KPIs which e-commerce enterprise are interested in. Kaushik [30] argues that combining both techniques in one hybrid method to eliminate limitations of each one is a reasonable solution.

Figure 2 Summary of page tagging log file data collection methods

Page Tagging	Log file Analysis
Advantages	Advantages
• Higher accuracy of session tracking due to insensitivity to proxy and caching servers • Tracking client-side activities executed within web-browsers by JavaScript, Flash, or Ajax technologies • Capturing client-side e-commerce data which cannot be accessed on the server-side (e.g., browser's history, visits of other websites) • Nearly real-time collection and processing visitor data • Possibility of outsourcing the service to an external web analytics provider	• Simplicity of historical data reproduction • Insensibility to network traffic filters • Tracking bandwidth and completed downloads, differentiating between completed and partial downloads • Tracking search engine spiders and robots • Tracking mobile users with no Java Script support
Disadvantages	Disadvantages
• Data loss due to JavaScript errors • Filtering page tags by firewalls • Limitedness of tracking bandwidth or completion of downloads • Incapability of checking search engine spiders	• Missing page view counts due to caching • Impossibility of tracking client-side actions executed within web-browsers by JavaScript, Flash, or Ajax technologies • Outsourcing limitedness of data storage, management and archiving services • Limitedness of eliminating search engines and robots from web-analytics data

Besides log files and page tags, there are alternative methods for collecting web visitor data. For example, web traffic data can be gathered by network sniffers, a web server API or plug-in modules such as Flash or toolbars for browsers. These are programs that extend the capabilities of the web server by enhancing or extending properties that are logged.

2.2.3. Data Presentation

Innovation in the web analytics area continues with newer and easier tools for visualization complex data sets with sensitive information about website interactions. Traditional and common for other industries data representation schemes consisting of dashboards, interactive tables and diagrams, exhibiting summary data by segments are typically supported by web analytics tools. New data representation schemes, which include site overlay and heat map, have emerged [30]. Success of these tools has been enforced by their integration in the visual the context of the website. Site overlay adds small progress bars near the navigation links on website. A web-analyst can click individual progress bars to learn exactly how many times a link was activated during the selected time range. A heat map illustrates the clusters of clicks on a web page and their density by using colors, thus reflecting focus of attention of website visitors.

Furthermore, a traditional way of representing business-sensitive data is data cubes. A data cube is a three- or higher dimensional array of values, commonly used to describe a time series of image data. A cube can be seen as a generalization of a two-dimensional spreadsheet. For example, a company doing online sales might wish to summarize financial data by product, by time-period, by region to compare actual and budget expenses. Product, time, region and scenario (actual and budget) are the data's dimensions. Each cell of the cube holds a number that represents some measure of the business, such as sales, profits, expenses, budget and forecast. Data is typically stored in a star or snowflake schema in a warehouse or in a special-purpose data management system supporting such operations as slice and dice, drill down, roll up, and pivot.

A data warehouse is an integrated and time-varying database primarily used for the support of management decision making [38]. This database often integrates heterogeneous data from multiple and distributed information sources and contains historical and aggregated data. In case of online e-commerce application, a sales data warehouse may contain information on the products sold, the time, and the place of sale. Such a data warehouse is typically much larger than an operational database. From data modeling perspective, a data warehouse can be seen as dimensional model which is composed of a central fact table and a set of surrounding dimension tables, each corresponding to one of the components or dimensions of the fact table. In the example of e-commerce application, the fact table models the actual sales data and each dimension, such as the product detail, the time of sale, the geographical place of the buyer, is modeled by a separate dimension table. In relational database terms the fact table contains all the necessary foreign key attributes referencing the primary keys of the constituent dimension tables. Conceptually, this leads to a star schema, which can be further refined into snowflake schemas providing support for attribute hierarchies by allowing the dimension tables to have subdimension tables. For example, the dimension table storing the sold items may have a subdimension table containing their specifications.

2.2.4. Metrics

According to Clifton [31], selection of right metrics among hundreds possible as well as frequency of reporting must be aligned with stakeholder needs and the business sector. Then, segmentation for hierarchical KPIs can be performed, to attain a tradeoff between clarity about visitor behavior and information overload. For example, a chief marketing officer of a retail site may want to see the average conversion rate, average order value, and cost per acquisition. A marketing strategist would like to see this same information segmented by referral medium type to compare paid search with organic search and email marketing with display banners. Commercial web analytics tools typically offer viewing predefined KPIs spitted by segmentation as standard functionality. Regardless the business sector of the web site, best practices of data preprocessing typically include segmenting visitors by geographical location, campaign, medium, or referrer source, content and eliminating certain known visitors and web crawlers [31]. Reported metrics may include the number of page views, the number of page views per visit, the number of single-page visits, CTR, and the average amount of time visitors spent viewing a specified page or set of pages. Categorization dimensions may include visitor hardware such as device type, software such as browser, and the geographical location. Common metrics used in web analytics tools and categorization dimensions with levels of granularity is summarized in Appendix 2.

2.3. Data Mining

Data mining is a tool for the Knowledge Discovery in massive datasets defined as "the non-trivial extraction of implicit, unknown, and potentially useful information from data" [43]. Data mining techniques can be divided into two groups: predictive induction and descriptive induction. In predictive induction models are typically induced from class labeled data and used to predict the class value of previously unseen examples. Descriptive induction aims at finding comprehensible patterns, typically induced from unlabeled data. Classification, regression, or temporal series are examples of predictive induction techniques, whereas summarization and association rules illustrate descriptive induction techniques. Difference between predictive and descriptive induction can be demonstrated on a machine manufacturer example. If the manufacturer wants to know how many machines may break down in the upcoming period, he must conduct an analysis based on predictive induction techniques. If he wants to know in what circumstances his machines may break down, to understand underlying factors to avoid them he must use descriptive induction methods.

Currently, several techniques can be seen as an intersection of descriptive and predictive data mining. Supervised Descriptive Rule Induction [49] is a trending paradigm which includes techniques combining the features of both types of induction. These techniques use supervised learning to solve descriptive tasks. The following data mining techniques constitute this paradigm: subgroup discovery, contrast set mining, and emerging pattern mining. Subgroup discovery task aims at finding a set of subgroups that are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest. A contrast set is a conjunction of attribute-value pairs, defining a pattern that best discriminates the instances of different user-defined groups. In case of two contrasting groups, the algorithm attempts to find characteristics of one group discriminating it from the other. Emerging pattern mining works with two datasets, and aims at discovering itemsets whose support increases significantly from one data set to another. The definitions of contrast set mining, emerging pattern mining and subgroup discovery appear different: contrast set mining searches for discriminating characteristics of groups, emerging pattern mining aims at discovering itemsets whose support increases significantly from one data set to another, while subgroup discovery searches for subgroup descriptions with unusual distributions of target variable. Whereas contrast set mining and emerging pattern mining are based on measures of coverage and accuracy, subgroup discovery is also focused on novelty and unusualness measures.

For the web-analytics area both descriptive and predictive data mining represents some interest. Hence, the current work is focusing on subgroup discovery algorithm since the analysis is done with respect to a particular target variable, probability of a click on a banner, and without considering time sequences.

2.3.1. Recommendation Systems

Recommendation systems are software agents that use behavioral or preference information to filter alternatives and make suggestions to a user [44]. Recommendation systems provide a type of mass customization that is becoming increasingly popular on the Internet. Such customization decreases the search effort for users and promises companies greater customer loyalty, higher sales, more advertising revenues, and the benefit of targeted promotions. Furthermore, typical functionality of recommendation systems provides estimates of their accuracy, explains reasons behind recommendations, and incorporates dynamic learning improving performance with growing amount of data available for learning.

Current customization systems fall into two classes that use different information sources to make recommendations. The first class comprises collaborative filtering which predicts a person's preferences as a linear, weighted combination of other people's preferences. The second class, known as content filtering, makes recommendations on the basis of consumer preferences for products. Both types of filtering methods have limitations. Collaborative filtering requires dense data sets and at least a few people that have evaluated a product. It does not reflect uncertainty in predictions and typically provides too few reasons for a recommendation. Attribute-based content filtering systems can recommend entirely new items but do not necessarily incorporate the information in preference similarity across individuals. Similar to collaborative filtering, content filtering methods cannot make recommendations for people who provide no preference information because of privacy concerns or lack of time. Thus, limitations of collaborative and content filtering methods are associated with types of information they use.

Modern recommendation systems may employ additional information sources such as person's expressed preferences or choices among alternative products, preferences for product attributes, preferences or choices of other people, expert judgments, and individual characteristics such as age or civil status, that may predict preferences. According to Ansari et al. [44], such expanded configuration was superior in the sense that it increased number of people contributing their preferences, expert opinion, or considered additional product details. However, it was still based on two classes of entities, humans and recommended items, which can be products or services, and do not analyze any information from the external environment. In various applications, it may be important to weight external circumstances, relevant to a user, an item, or a combination of them, under which the recommendation transaction is taking place. In the situation of personalized online content recommendation, it may be critical to identify the type of required content and the optimal time for recommendation or delivery. Incorporating contextual information into recommendation systems can improve their performance.

2.3.2. Expanding Recommendation Systems with Context

As discussed in Section 2.1, there is no agreement among different research fields about the definition of the context. Literature survey suggested that the business community and the computer science community have different views on both defining the context and using the contextual information. These views are explained below.

2.3.2.1. Computer Science View

A survey on typical heuristic strategies for handling context-sensitive features in supervised machine learning was conducted by Turney [35]. These heuristic strategies can be employed by more advanced hybrid algorithms as building blocks. He also made a review of methods for discovering implicit contextual information, if the context is not given explicitly. For that, he proposed to use clustering and time sequence analysis. Turney discovered synergetic effect of combining strategies into hybrid, claiming that hybrid strategies tend to perform better than the sum of the component strategies.

A typical supervised machine learning algorithm represents examples as vectors in a multidimensional feature space. By using training examples it aims at finding a predictive function mapping feature set to a predefined set of classes. Turney divided all features on three categories depending on their ability to contribute into predictive model: primary, contextual, and irrelevant. A primary feature is the feature which is useful for classification on its own, without considering other features. A contextual feature is not useful in isolation, but is useful in combination with primary features. An irrelevant feature is not useful for classification on its own or in combination with other features, primary or contextual. Primary, contextual, and irrelevant feature categories constitute Turney's framework for studying context in machine learning environment.

Similarly to Turney, Widmer [36] conducted research on context-aware machine learning algorithms. He presented a method capable of automatic detection of contextual features in on-line learning settings and utilization this information during the learning, which is explained below. Widmer, however, differs from Turney in operational definitions by dividing all relevant features on predictive and contextual. A feature is predictive if there is correlation between the distribution of its values and the observed class distribution. A feature is contextual if there is a strong correlation between its temporal distribution of values and the times when certain other features are predictive. Intuitively, a contextual feature is one that could be used to predict which features are predictive at any point in time. To abstract from insignificant improvement of the prediction quality, Widmer used χ^2 goodness-of-fit test with a given threshold. Thus, predictive and contextual feature categories constitute Widmer's framework.

By analyzing definitions Turney [35] and Widmer [36], it can be seen that the authors differ in the naming convention. According to Turney, a contextual feature is the feature that is useful only in certain context, and according to Widmer, a contextual feature is the feature which creates the context for another feature. Figure 3 shows a pivoted dataset, illustrating differeence between the authors. It has two binary features, A and B, and a binary target class C. Two right columns contain weights as the number of examples, having given A and B values and falling to a given class C. For instance, there are three examples having $A = 0 \land B = 0 \land C = 1$.

Features		Target class (weight)	
A	B	C=0	C=1
0	0	1	3
0	1	4	3
1	0	3	1
1	1	3	4
Total		11	11

A=0	Target class (weight)	
B	C=0	C=1
0	1	3
1	4	3
Total	5	6

A=1	Target class (weight)	
B	C=0	C=1
0	3	1
1	3	4
Total	6	5

B=0	Target class (weight)	
A	C=0	C=1
0	1	3
1	3	1
Total	4	4

B=1	Target class (weight)	
A	C=0	C=1
0	4	3
1	3	4
Total	7	7

Figure 3 Example dataset, illustrating the difference between Turney's [35] and Widmer's [36] naming conventions

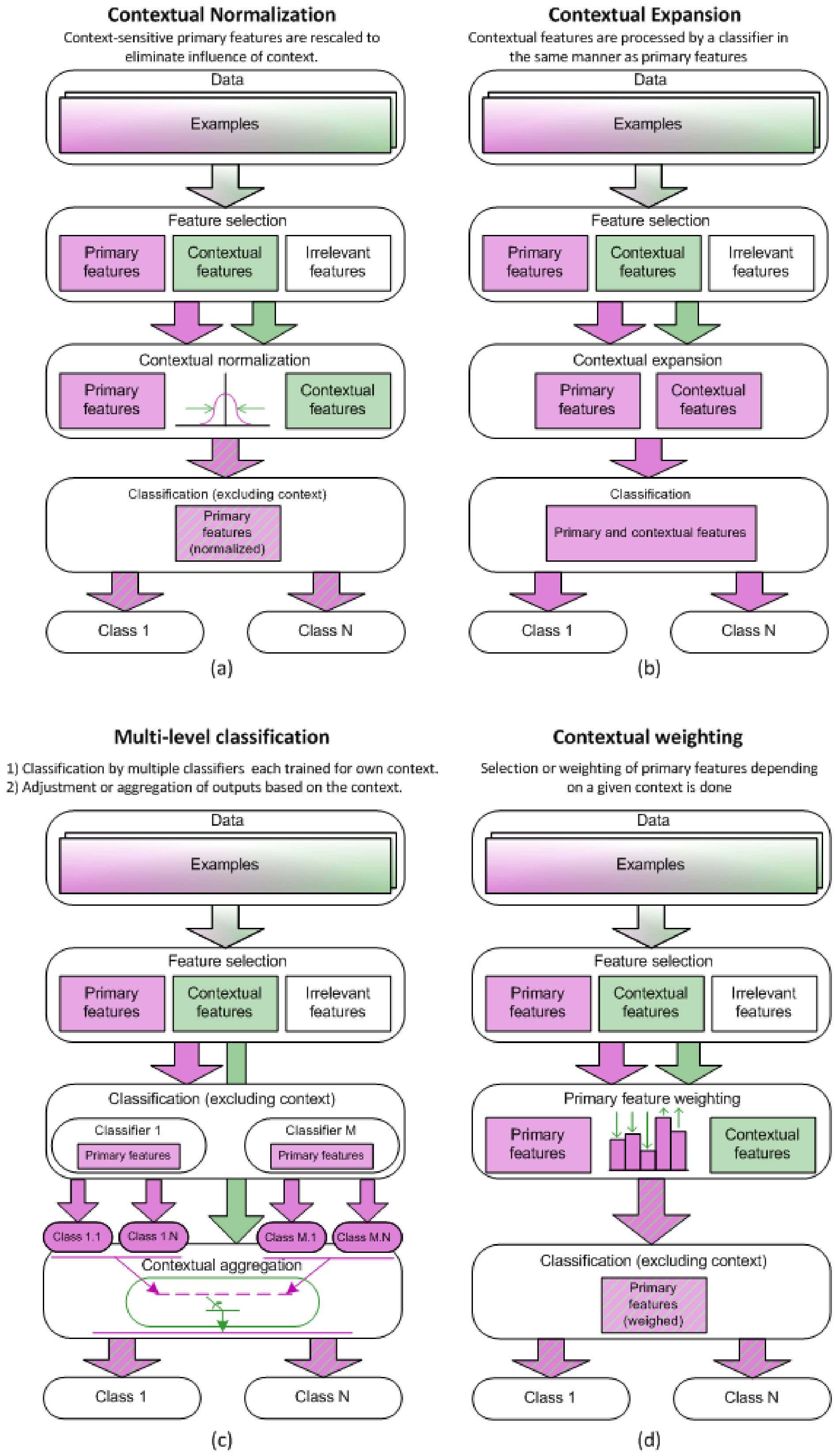
Feature A is useful on its own since fixing A leads to the different distribution of the target class, e.g. $P(C = 0|A = 0) \neq P(C = 0)$, which allows producing a model with rules $A = 0 \rightarrow C = 1$ and $A = 1 \rightarrow C = 0$, classifying correctly 12 examples out of 22 (accuracy=0.55). Thus, feature A is primary according to Turney, and predictive according to Widmer. Feature B is useless on its own, since fixing B does not lead to the different distribution of the target class, e.g. $P(C = 0 \mid B = 0) = P(C = 0) = 1/2$. However, fixing B given fixed A increases quality of prediction comparing to a model, using only feature A. Produced rules can be trivially derived from Figure 3, while the model correctly classifies 14 examples out of 22 (accuracy=0.64). Feature B is contextual according to Turney since it is useful in certain context, created by feature A. However, according to Widmer, both features A and B, are contextual, since fixing A makes B predictive, and fixing B increases "predictive power" of A. In other words, features A and B create context for each other.

Context-aware machine learning concept assumes that primary, or predictive, features are context-sensitive, that is the learning algorithm may perform better when it employs contextual features. For example, the task is to distinguish healthy people from sick people using a thermometer. Sick people tend to have higher temperatures than healthy people, but physical exercise also causes higher temperature. If the first context consists of temperature measurements made on people in the morning, after a good sleep, and the second context consists of temperature measurements made on people after heavy exercise, then correct diagnosis becomes more difficult if two contexts are mixed together. Any of heuristic strategies studied by Turney [35] can be helpful for separation of these two contexts. The strategies are the following: contextual normalization, contextual expansion, multi-level classification, and contextual weighting, as shown in Figure 4. Dataset records contain a mixture of primary, contextual and irrelevant features, depicted in red, green, and white colors respectively. During the first step, feature selection is done, where features are assigned to one of the categories. While irrelevant features are ignored by all the algorithms, contextual features are treated differently depending on the strategy:

- **Contextual normalization**: Context-sensitive primary features are rescaled to eliminate influence of context, as shown in Figure 4(a). This can be done, for example, by normalization whereas normalizing function must be context-specific.
- **Contextual expansion**: Contextual features are processed by a classifier in the same manner as primary features, thus expanding the feature space, as shown in Figure 4(b).
- **Multi-level classification**: Classification using only the primary features is done by multiple classifiers whereas each classifier has been trained for its context. Then, adjustment or aggregation of outputs such as voting or post-classification is done based on the contextual features, as shown in Figure 4(c).
- **Contextual weighting**: Selection or weighting of primary features depending on a given context is done as opposed to contextual selection of classifiers, as shown in Figure 4(d). The rational of this approach is to assign higher importance to primary features that, in a given context, are more important for classification. For example, for model predicting traffic jams, working hours in the particular area will have high weight on working days, and times of entertainment activities and events on holidays.

Each strategy results in a model, assigning examples to one of N classes.

Contextual expansion is the simplest strategy since it does not require any specific logic for processing context. Contextual normalization can be seen as preprocessing step of the machine learning algorithm, so that model itself does not contain any context-depended information. Contextual weighting can be seen as a simplified version of multi-level classification, where the model consists of a few sub-models, each trained for own context and differentiated weights of primary features. For regression models contextual weighting may be implemented trivially by assigning different coefficients to the attributes depending on the context.



Contextual Normalization

Context-sensitive primary features are rescaled to eliminate influence of context.

Contextual Expansion

Contextual features are processed by a classifier in the same manner as primary features

Multi-level classification

1) Classification by multiple classifiers each trained for own context.
2) Adjustment or aggregation of outputs based on the context.

Contextual weighting

Selection or weighting of primary features depending on a given context is done

Figure 4 Heuristic strategies for handling contextual information

The aforementioned strategies can be applicable if contextual features are obtainable from the dataset. However, contextual information is often not recorded due to either technical limitations or ignoring by domain experts who tend to presume that the context is known. Nevertheless, there are techniques capable of recovering contextual information.

Widmer [36] conducted research on learners that are capable of adapting to different contexts without explicit help from a teacher in the framework of incremental learning. In this framework, learning is done on-line from a stream of incoming labeled examples, and the concepts of interest depend on some, possibly hidden, context, whereas changes in this context can influences the target concepts. For example, weather prediction rules may vary drastically with the change of seasons. The visible effects of such changes are increased prediction error rates. Development of incremental learners that can trace a concept drift and keep track of changing contexts was the goal of the FLORA project [40]. The basic strategy in the FLORA algorithms is to continually monitor the success of incremental prediction and to make educated guesses at the occurrence of context changes and corresponding concept changes. FLORA does not assume that contexts are represented explicitly.

Then, Widmer suggests that in some domains, the data contains explicit clues that allow identification of the current context, and, technically, such clues may be attributes or combinations of attributes whose values are characteristic of the current context. Then, systematic changes in their values might indicate context change. For example, an auto traveler after crossing the border between two countries may notice systematic change in license plates attached to vehicles, which might lead him or her to suspect that in a different environment where some traffic rules may be different. Some other examples may include climate or season in weather prediction, lighting conditions or background color in automatic vision, or speaker nationality and sex in speech processing.

Widmer introduced a notion of contextual clues as a separate attribute corresponding to a combination of attributes defining the context. His algorithm, MetaL (Meta-Learner with underlying Bayes classifier or Instance-Based classifier), includes a base level learner that performs the regular on-line learning and classification task, and a meta-learner that attempts to identify attributes and features that might provide contextual clues. Context learning and detection occur during regular on-line learning, without separate training phases for context recognition. Perceived context changes are used to focus the incremental learner specifically on information relevant to the current context. The result is faster adaptation to changed concept definitions, and generally an increase in predictive accuracy in dynamically changing domains.

To recover missing contextual features Turney [35] proposed to use clustering algorithms. Assuming that if two instances are assigned to the same cluster, they likely share similar contexts, the author states that clusters that are generated by unsupervised clustering algorithms are capable of capturing this shared context. In other words, clustering cases by their primary features leads to grouping these instances by shared classes and shared contexts, and the likelihood that grouped instances belong to the same class and context is greater than the likelihood for the instances from the general population. Then, a discovered contextual feature can be introduced in the form of cluster – label pairs. However, the feature might not be purely contextual since clusters may be predictive of the class.

An alternative way to recover unrecorded contextual data is to analyze temporal information encoded in instances, or consider sequential order of instances in the assumption that the order of instances may correspond to their timing. Since instances can reflect events occurred closely in time in similar circumstances, they may share similar context. Depending on handling context strategy, it may be reasonable to convert temporal information into a discrete feature. Similarly to clustering, a new contextual feature can be introduced in the form “Time = Period” or “Number = Interval”.

Methods for recovering missing contextual features are not limited to unsupervised clustering and temporal and sequential analysis. Other approached may be dictated by specifics of application domains, and involvement of domain expert may be necessary.

2.3.2.2. Business View

Literature survey shows that a large number of projects incorporating contextual information into recommendation systems have been already executed. For example, designing restaurant recommender system based on machine learning techniques, Oku et al. [5] incorporate into the recommendation system such contextual dimensions as companion, time, and weather. The authors claim the system taking the context into consideration demonstrate higher accuracy and user satisfaction comparing to corresponding non-contextual recommendation systems.

Definition of the context used in the recommendation systems is left to developers’ discretion. Thus, Berry and Linoff [3], describing data mining applications, defined context as a set of events characterizing the life phases of a user and that can determine a transformation in his or her status or preferences. Such events as getting a new job, retirement, marriage, divorce, or birth of a child may constitute the context. In the literature focusing on mobile applications, context is defined as the location of the user, his or her surrounding environment (in terms of humans and objects), and changes of these elements over time, as it was implemented by Schilit and Theimer [6]. A number of other factors can also be taken into consideration. For instance, Brown et al. [7] use the season, the date, and the temperature as a context. Ryan et al. [8], [9] include user’s physical and conceptual statuses of interest. Dey et al. [10] include the social, physical, emotional, and informational state and extended the definition of the context to incorporate any information which can characterize the situation related to the interaction between humans, applications, and the surrounding environment. Hence, depending of the application domains, different definitions of context are employed by developers.

In the area of process mining, context was defined by Ploesser et al. [46] as a general term, addressing both the events and conditions in the environment and the specific properties of cases handled by the process. In line with this definition, van der Aalst and Dustdar [47] distinguished four types of context: instance, process, social, and external. Process instances (that is, cases) might have various properties that influence their execution. For example, the order’s size can influence the type of shipping the customer selects or the transportation time. Process context may include the number of instances being handled and resources available for the process. When predicting the expected remaining flow time for a particular case, for example, the analysis tool should consider not only the order’s status (instance context) but also the workload and resource availability (process context). Social context characterizes human work within a particular organization. Interpersonal relationships, not attributable to the business process, may influence speed at which people work. The external context captures factors that are part of an ecosystem that extends beyond an organization's control sphere. For example, the weather, the economic climate, and changing regulations might influence how organizations handle cases. In fact, changing oil prices can influence customer orders, as when the demand for heating oil increases as prices drop. Although external context can have a dramatic impact on the process being analyzed, selecting relevant variables is process-specific and requires domain knowledge. Taxonomy of the context proposed by Van der Aalst and Dustdar shown in Figure 5 is rather intuitive.

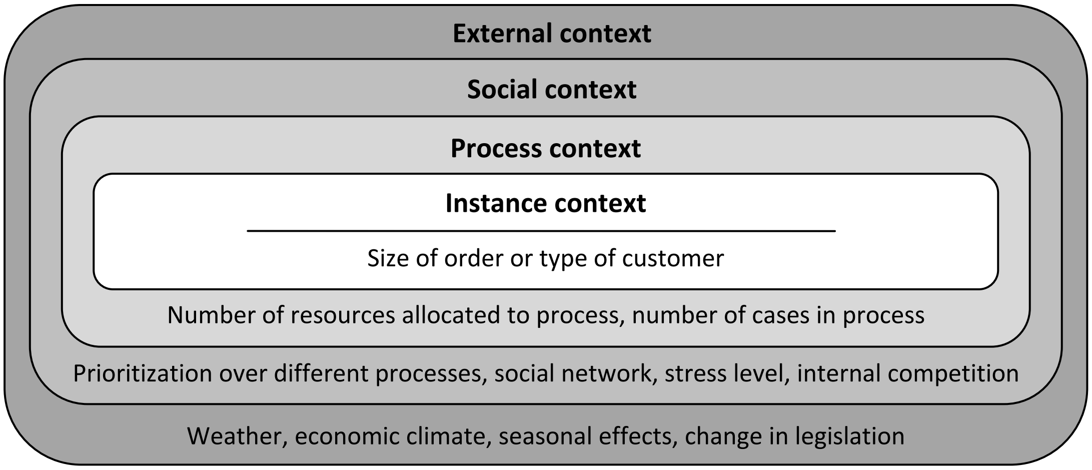

External context
Social context
Process context
Instance context
Size of order or type of customer
Number of resources allocated to process, number of cases in process
Prioritization over different processes, social network, stress level, internal competition
Weather, economic climate, seasonal effects, change in legislation

Figure 5 Context taxonomy used in the process mining according to van der Aalst and Dustdar [47]

It can be noticed, however, that the context definition and the taxonomy used in process mining are built around essential entities of workflow processes, namely case, task, and resource [48]. These essential entities may be roughly associated with instance context, process context and resource context, respectively. Since the notions of case, task, and resource are specific for the workflow area, and recommendations systems considered in this work have much broader application domain, it is not possible to apply directly or expand van der Aalst and Dustdar's taxonomy. It is, however, possible to define entities involved in the particular application and then, in a similar way, to associate context with these entities also considering domain knowledge.

Whereas Bazire and Brezillon [2] for the computer science area considered contexts of the task, of the person, of the interaction, or of the situation, some researches consider only one level of abstraction consisting of the user interacting with the application. Thus, some developers align the context with the user [10, 11], while others emphasize relations between the context and the application [12, 13]. A number of hybrid techniques using both user and application information have been developed for context-aware mobile applications [12, 14, 15]. Various Location-Based Services provided to mobile customers can illustrate these hybrid techniques. For instance, Schiller and Voisard [16] presented a case where a Broadway theater attempts draw more spectaculars to the upcoming shows by promoting discounted tickets to the general public found in New York's Time Square half an hour before beginning of the show. Not to waste the tickets after beginning of the shows the theaters send the offers to the mobile devices located in Time Square neighborhood. In this application the context is encoded by the location, time and the type of the communicator. Some personal information such as age, company and preferences may also be used by the service and represent the contextual data of the user. A similar service allowing tourists to interact with other tourists and remote users by sharing remarkable sights was described by Brown et al. [17]. This service also exploits both user and application context and proves that context-awareness can be useful in mediating social activities. Hence, for some applications consideration of contexts in the high level of abstraction consisting of the user and the application may result in improvement of the recommendation system.

Inclusion other types of the context such as interactional, or situational have been done by marketers, who attempted to study purchasing process in a specific context. Some researchers have come to the conclusion that the same individual may apply different decision-making methods and choose different items or brands depending on the situation and the context in which the transaction takes place. For example, Lilien et al. [18] stated that customers differ in their decision-making styles depending on the usage situation. The same customer would make a different purchasing decision if the article is intended for oneself, for family or as a gift, implying that the use of products and services influencing the decision; and situation of the purchase, which may be sales person assisted purchase, shelf selection in a department store, or market sale. Lilien et al. [18] conclude that the predictions of customer preferences should be based on the ability to discover and exploit the relevant information about his or her context at the moment of making the decision.

Some advanced studies include meta-level of context into analysis. Chen and Kotz [23] proposed to record the context across a time span in order to analyze context history, and exploit information about context evolution. Archak et al. [22] introduced a model including a feedback loop between the user and the context. While doing study on sponsored search and advertisement on the web, Archak et al. attempted to analyze structural patterns in visitors' online clicking behavior and visitors' trajectories on target e-commerce websites. The basic underlying assumption was that users tend to take specific trajectories in terms browsed websites and posed search queries. Advertisement server selects sequence of ads for display to visitors based on these trajectories; conversely, the sequence of ads shown to visitors' affects their browsing and search behavior. In other words, on the one hand, the user creates the context; on the other hand the user is affected by context. Similarly to Archak et al., Dourish [21] stated that certain users' actions may entail different types of related contexts, thus maintaining interconnection between actions and underlying contexts. Incorporating meta-level of context into analysis may be seen as a next step in the advancement of context-aware recommendation systems.

2.4. Taxonomy of Contextual Features

Variety of contextual features discussed by the business community leads to attempts to their classification. Schilit et al. [20] divided contextual features in the following dimensions: computing environment (available processors, devices accessible for user input and display, network capacity, connectivity, and costs of computing), user environment (location, collection of nearby people, and social situation), and physical environment (lighting and noise level). Chen and Kotz [23] extended Schilit's classification with the temporal aspect (time of a day, week, month, and season of the year). Prahalad [19] focused only on temporal, spatial, and technological dimensions of the contextual information. Kuniavsky [26] studying in depth user experience on the Web from the marketing perspective, divided human attributes on demographic (age, gender, level of income, location, time, culture, job title, goals, roles, needs, knowledge), technological (computer, monitor, network connection, level of experience, browser, operating system). Technological attributes are important for the analysis since they impose limitation into user's online behavior.

Dourish [21], looking at contexts classification from the data mining perspective, divided the context on representational and the interactional views. The representational view presumes prior specification of contextual attributes and their capturing and use within the context-aware applications during operation. The context may be defined with a given set of observable features, whose structure does not significantly deviate over time. In opposite, the interactional view presumes that behavior of the individual is stimulated by the context of his or her environment, but that the context itself is not necessarily observable.

It is obvious that not all the contextual information from the environment can be meaningful for the particular recommendation. For example, a recommendation system should maintain a distinction between work and family issues of a working individual: contextual information relevant to personal life should not affect decisions relevant to his or her work. Hence, a number of approaches to evaluating the relevance of a given contextual feature have been developed. Particularly, the relevance determination can either be done manually by using domain knowledge, or automatically, by using various existing feature selection procedures from such fields as machine learning, data mining, and statistics, based on existing ratings data obtained during the data preprocessing phase.

Table 1 gives a survey of studies with reference to specific contextual features.

Table 1 Overview of studies on contextual features

Contextual Features	Studies
Life stage and social status	[3], [10], [18]
User's intent (leading to different types of behavior)	[4], [18]
Companion / presence of other people or objects	[5], [6], [10], [20], [23]
Weather	[5]
Geographical region (from the scale of the world to the scale of city neighborhood)	[6], [9], [10], [16], [17], [19], [20], [23]
Indoor location	[6], [13], [20]
Date	[7], [10], [19], [20]
Time	[5], [9], [10], [19], [20], [23]
Season	[7]
Temperature	[7], [9]

Physical environment (lighting and acoustic environment)	[8], [10], [20]
User’s emotional status, current activity and focus of attention	[10], [23]
Hardware, infrastructure	[12], [20]
Meta-level (change over time, a feedback loop)	[6], [21], [22], [23]

Some contextual features given in Table 1 can be described hierarchically on different levels of abstraction. Considering the taxonomy introduced in [19], [20], and [23], Figure 6 gives a hierarchical representation of the contextual features described in the studies on context-aware systems gives such representation. Grouping of contextual features on user’s and application’s ones is proposed by Adomavicius and Tuzhilin [25]. Division of features on user’s, temporal, physical and technological contexts is proposed in [20] and [23]. Some researchers such as [10] and [23] do not distinguish between user’s and application’s contexts and relate all the features to the user’s context. Such taxonomy, however, according to Adomavicius and Tuzhilin [25], is not accurate since some objective and user-independent features such as location or weather can give an indirect hint about user’s current activity or social or emotional status. Considering an application’s context in isolation allows abstracting from user individual characteristics and analyzing behavior of groups of users acting under certain circumstances common for the entire group.

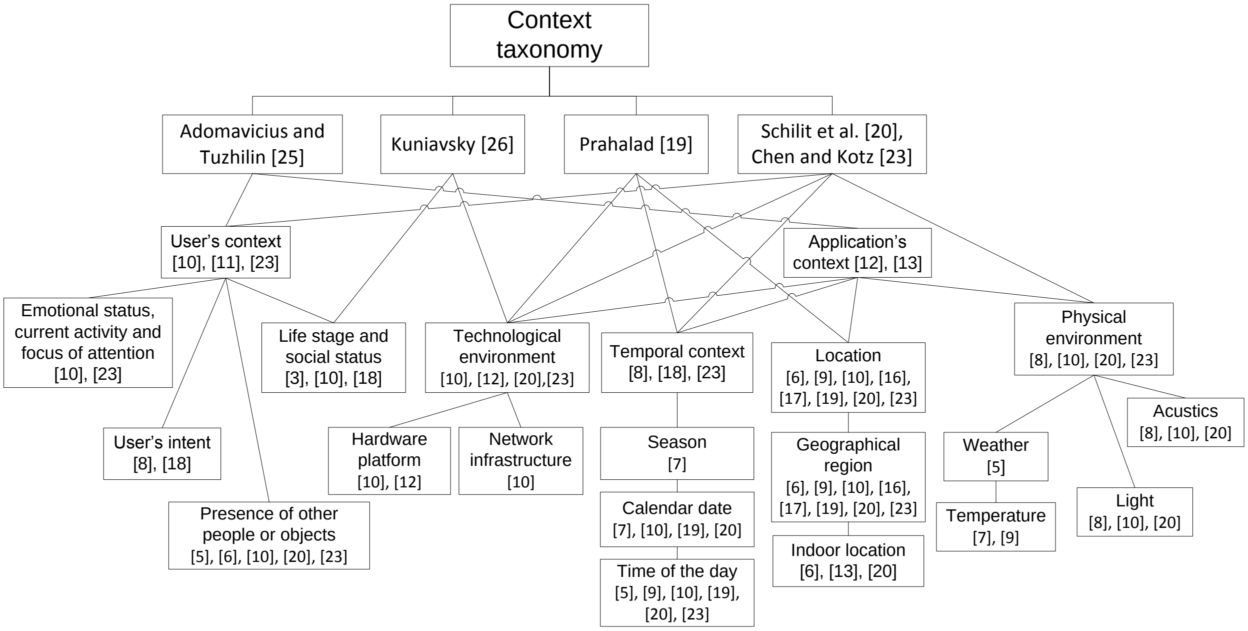

Context taxonomy
Adomavicius and Tuzhilin [25]
Kuniavsky [26]
Prahalad [19]
Schilit et al. [20], Chen and Kotz [23]
User’s context [10], [11], [23]
Application’s context [12], [13]
Emotional status, current activity and focus of attention [10], [23]
Life stage and social status [3], [10], [18]
Technological environment [10], [12], [20],[23]
Temporal context [8], [18], [23]
Location [6], [9], [10], [16], [17], [19], [20], [23]
Physical environment [8], [10], [20], [23]
User’s intent [8], [18]
Hardware platform [10], [12]
Network infrastructure [10]
Season [7]
Geographical region [6], [9], [10], [16], [17], [19], [20], [23]
Weather [5]
Acustics [8], [10], [20]
Presence of other people or objects [5], [6], [10], [20], [23]
Calendar date [7], [10], [19], [20]
Time of the day [5], [9], [10], [19], [20], [23]
Indoor location [6], [13], [20]
Temperature [7], [9]
Light [8], [10], [20]

Figure 6 Hierarchical representation of the contextual features: survey of the studies

3. Methodology

The current work aims at developing a framework and corresponding techniques for integrating predictive analytics and context awareness with the application to Kliknieuws.nl business case. Web-analytics uses data mining techniques for achieving business goals; hence, this field lies on the intersection of business and computer science domains. As discussed in Section 2.1, there is no agreement among researches concerning the definition of the context. Relevant literature study suggested that the business community and computer science community adopted different views. Business community focuses on underlying business processes and KPIs of the applications, and consider context as parameters and states of entities involved in the process, or external environment and influencing KPIs. The business community does not classify these parameters on primary (or predictive) and contextual, as accepted in the computer science community [35], [36]. Researches, belonging to the computer science community, see their task as improving performance of recommendation systems. Consequently, they view contextual attributes as capable of improving this performance, if the recommendation systems use them in a proper way, possibly different from the regular attributes. In the current work, we will reflect both business and computer science views.

For the business view, we adopt the following definition of context, partially derived from the process mining area [46]:

Definition (Context, business view): For a given KPI of the organization, and the process associated with this KPI, the context is a general term addressing

1. properties of parties (or entities) involved in the process,
2. properties of interactions (or transactions) between the parties,
3. events and conditions of the external environment

if they affect the KPI of the organization.

Selecting relevant attributes especially from events and conditions of the external environment is a nontrivial task, which must be done by a domain expert. Selecting too many even relevant attributes may lead to curse of dimensionality, discussed in Section 3.1.3. In the settings of a web-application such as Kliknieuws.nl, entities may be human visitors, web-crawlers, and the website, while page views and clicks constitute the interactions. Context of web-crawlers is not studied in the current work. Using taxonomy of contextual features in Figure 6 as a reference, it is possible to get insight which contextual features might make sense to consider. Table 2 summarizes the findings. It is necessary to note that the website's contextual features have to reflect the website's KPIs. Not all of the contextual features discussed in Table 2 can be easily measured and quantified. This is especially true for the human, whose behavior as well as factors affecting it have very complicated nature, whereas its reflection on the web, and consequently, available data for the web-analytics, is very limited. In fact, an emotional status and focus of attention may be easily quantified and measured by smart home applications, but they can hardly be grasped by a web-application. Background information about the visitor such as age and gender, however, may be retrieved by attaching his or her social network profile to the target web-application.

Table 2 Contextual features relevant to a web-application

Entity (or interaction)	Relevant contextual features
Human visitor	• Emotional status, current activity and focus of attention • User's intent • Presence of other people or objects • Life stage and social status • Location • Weather
Website	• Content category, URI • Presents of supportive content such as photo or video
Page view and click	• Hardware platform • Network infrastructure • Timing (season, calendar date, time of the day)

Definition of context according to Computer Science view we will give in Section 3.1.5, after introducing formal notations.

The framework, that has been developed in the scope of the current work, may be represented as a black box accepting website visitor logs as the input and producing a model as a set of rules as the output. Ideally, the framework can be implemented as a software module, pluggable into the target web-application. Since the current work focuses mostly on the subgroup discovery task, the framework represents a set of data processing modules, which may be connected as a seamless chain. Each module has to be run manually by an analyst, whereas output of one module serves as the input for the next module in the chain. The proposed methodology is shown in Figure 7.

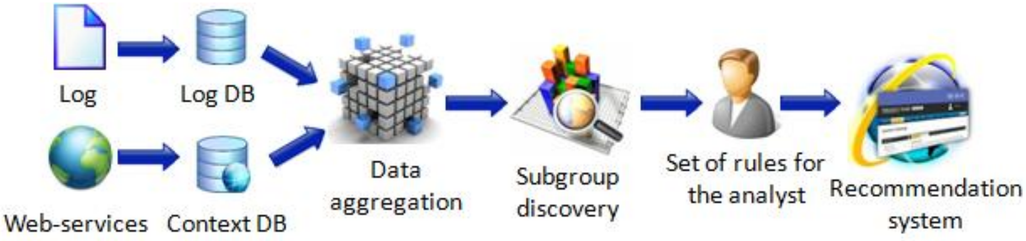


Figure 7 Methodology of study with application to Kliknieuws.nl banner placement system

The raw visitor logs, stored as text files are imported in the Log Database. Log information contains the data about visitors which can be extracted from HTTP-headers, or from visitors' browsers by Java Script tags. In parallel, external information is collected and stored in the Context Database. This information can reflect events of the external environment or store contextual information about the web-application itself. Next, data aggregation is done by joining visitor and contextual information in a database or in OLAP. Data aggregation typically implies that data combined from several measurements, and groups of observations are replaced with summary statistics based on those observations. As a result, high-level data is composed from a multitude or combination of other more individual data.

During the aggregation, data preprocessing such as discretization or resampling can be done as discussed in Section 3.1.2. The aggregated data are used as the input for a subgroup discovery algorithm, which produces a set of rules describing a property of interest such as click-through rate or conversion rate. This set of rules may be analyzed by the analyst and used as a model for recommendation system. For processing contextual features, Turney [35], contextual expansion approach is used, namely classifier treats them in the same manner as primary features. This is the simplest and domain independent strategy, which does not require any specific logic for processing context. Overall, the chain of data processing modules starting from raw log data and ending in recommendation system module is logically complete.

The central part of the current work in the subgroup discovery module, and it is presented below.

3.1. Subgroup Discovery Algorithm

3.1.1. Introduction

The task of subgroup discovery is to find a population of subgroups that are statistically "most interesting", namely they are as large as possible and have the most unusual distributional, or statistical, characteristics with respect to the property of interest. Thus, subgroup discovery is a technique for the extraction of patterns, with respect to a property of interest in the data, or target variable. The induced patterns are typically represented in the form of rules and called subgroups. Traditional data mining techniques have not been able to achieve this propose. For example, predictive techniques maximize accuracy in order to correctly classify new objects, and descriptive techniques solely search for relations between unlabeled objects. The need for obtaining simple models with a high level of interest led to statistical techniques which search for unusual relations. This technique combines features of both predictive and descriptive induction, and its goal is to generate in a single and interpretable way subgroups to describe relations between independent variables and a certain value of the target variable.

Figure 8 illustrates the main difference between descriptive and predictive induction. A precise model (classifier) for predictive induction divides the space in two determined regions with respect to the type of objects in the set. A model for descriptive induction describes groups of elements (clusters) in the set, without a target variable. As can be observed, the model of predictive induction has a different goal with respect to the model of descriptive induction. Therefore, different heuristics and evaluation criteria in both types of learning are employed. For predictive models precision is typically more important, whereas, descriptive models give preference to interpretability.

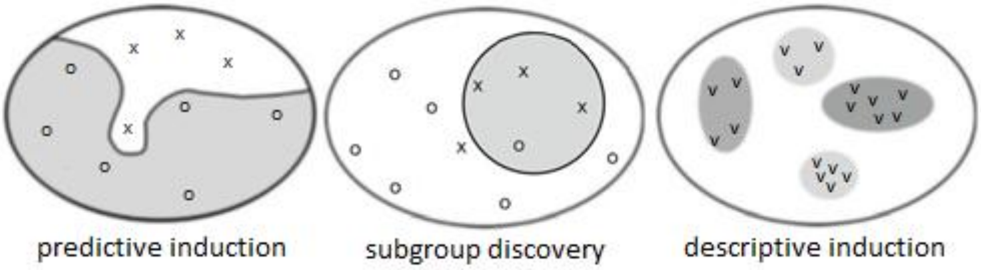


Figure 8 Models of different techniques of knowledge discovery

Figure 8 also shows an example of the rule for subgroup discovery, where two values for the target variable can be found. In this representation, a subgroup for the "x" value of the target variable can be observed, where the rule attempts to cover a high number of objects with a single function: a circle. As can be observed, the subgroup does not cover all the examples for the target value even the examples covered are not positive in all the cases, but the form of this function is uniform and very interpretable with respect others. In this way, the algorithm achieves a reduction of the complexity. Furthermore, the true positive rate for the value of the target variable is high, with a value of 75%.

The subgroup discovery task is differentiated from classification techniques because subgroup discovery attempts to describe knowledge for the data while a classifier attempts to predict it. Important limitation of such popular classification technique as decision tree is that some concepts are hard to learn, for instance XOR, parity or multiplexer problems, where correlated attributes are irrelevant for classification on their own, but relevant together. Decision trees do not express these problems easily. However, subgroup discovery algorithms are not sensitive to correlated attributes, and are free from this limitation. Furthermore, the model obtained by a subgroup discovery algorithm is usually simple and interpretable, while that obtained by a classifier is complex and precise.

Subgroup discovery attempts to search relations between different properties or variables of a set with respect to a target variable. Due to the fact that subgroup discovery is focused in the extraction of relations with interesting characteristics, it is not necessary to obtain complete but partial relations. These relations are described in the form of individual rules.

A rule (R), which consists of an induced subgroup description, can be formally defined [43] as:

$$R : Cond \rightarrow Target_{value}$$

where $Target_{value}$ is a value for the variable of interest (target variable) for the subgroup discovery task, and $Cond$ is commonly a conjunction of attribute-value pairs which is able to describe an unusual statistical distribution with respect to the $Target_{value}$.

As an example, let D be a data set with three variables $Age = \{young, middleaged, old\}$, Sex = {M, F} and Country = {USA, UK, NL}, and a variable of interest target variable $Money = \{Poor, Normal, Rich\}$. The model may consist of two rules containing the following subgroup descriptions:

$$\begin{cases} R1 : (Age = young \wedge Country = NL) \rightarrow Money = Rich \\ R2 : (Age = old \wedge Sex = F) \rightarrow Money = Normal \end{cases}$$

where rule $R1$ represents a subgroup of Dutch young people for which the probability of being rich is unusually high with respect to the rest of the population, and rule $R2$ represents that old women are more likely to have a normal economy than the rest of the population. As mentioned above, the model does not need to cover the whole example space.

3.1.2. Main Elements

For constructing or applying a subgroup discovery algorithm the following elements can be considered the most important, and must be aligned with specifics of the dataset and the task:

- Type of the target variable
- Description language of the subgroups
- Quality measure
- Search strategy
- Number of obtained subgroups
- Prediction mode

Subgroup discovery algorithms can be used with various types of the target variable: binary, nominal or numeric, whereas for each target variable type different analyses can be applied. In binary analysis, the variables have only two values, True or False, and the task is focused on providing interesting subgroups for each of the possible values. In nominal analysis, the target variable can take an undetermined number of values, but the idea for the analysis is similar to the binary, namely to find subgroups for each value. In numeric analysis, the variable can be studied different ways such as dividing the variable in two ranges with respect to the average, discretizing the target variable in a determined number of intervals, or searching for significant deviations of the mean among others.

The description language must be suitable for representing interesting rules. These rules must be simple and therefore are represented as attribute – value pairs in conjunctive or disjunctive normal form in general. Furthermore, the values of the variables can be represented as positive and/or negative, through fuzzy logic, or through the use of inequality or equality and so on.

Quality measure is a key factor for the extraction of knowledge because it defines the interestingness of subgroups found by the algorithm. Furthermore, quality measures provide the expert with the importance and interest of the subgroups obtained. There is no consensus about which quality measures are the most suitable for use in subgroup discovery [43], however, most of algorithms employ quality measures which incorporate subgroup size and the bias of the target variable such as weighted relative accuracy or its modifications.

Search strategy is important from performance point of view since the dimension of the search space has an exponential relation to the number of features and values considered. Strategies can use top-down or bottom-up approach and may include exhaustive search, beam search, or evolutionary algorithms. In addition, to limit the search space, pruning strategies can be employed such as based on minimal support, or on optimistic estimate of the quality measure of the current branch.

Depending on a dataset, task, and a quality measure, the number of rules learned by subgroup discovery algorithms can exponential in the size of the input. It might not represent a problem for recommender systems, but infeasible for a human expert. Consequently, subgroup discovery algorithm can have two modes of work, reporting either a given number of best rules ("top-k" approach), or all the rules above given quality measure value. The former make sense if the results are intended for a human expert, the latter – for machine learning algorithm. Limiting the number of obtained rules allows for additional pruning of the search space based on comparing optimistic estimate of a search path and quality measures of already discovered rules.

Subgroup discovery algorithm can induce overlapping subgroups. The same record of the dataset can be covered by a few rules, attributing it to different target classes. Hence, if the learned model is intended for use in a recommender system, it is necessary to employ an arbitration mechanism, defining how to classify records, covered by more than one rule. The possible alternatives include voting model, applying the most specific rule, or applying the rule with the highest quality measure.

3.1.3. Challenges

Describing state-of-the-art of data mining area, Hand et al. [37] highlight a number of its specifics: huge size of data sets possibly growing in real-time, presence of noise caused by incorrect logging or recording exceptional events, missing values, and possible large number of features, leading to so-called "curse of dimensionality". While first three challenges are properties of real-life data sets, curse of dimensionality is the fundamental theoretical problem, related also to ideal datasets. Curse of dimensionality means that with the growth of dimensionality, the volume of the space increases so fast that the available data become sparse. This sparsity is challenging for methods that require statistical significance and it leads to overfitting. Since organizing and searching data often relies on detecting areas where objects form groups with similar properties, the problem of in high dimensional data is that all objects appear to be sparse and dissimilar in many dimensions which prevents common data organization strategies from being efficient. In order to obtain a statistically sound and reliable result, the amount of data needed to support the result often grows exponentially with the dimensionality. In fact, with p features there are $2^p - 1$ possible subsets of features for a simple exhaustive search algorithm to consider. Most data mining tasks rapidly become computationally difficult as dimensionality increases.

Machine learning tasks involving learning from a finite number of data samples in a high-dimensional feature space with each feature having a number of possible values, a huge amount of training data is required to ensure that there are several examples with each combination of values. With a fixed number of training samples, the predictive power reduces as the dimensionality increases. A common strategy to treat the curse of dimensionality is to use a smaller subset of relevant features either by choosing most relevant or by transforming original feature space into feature space with lower dimensionality.

There are two types of challenges for subgroup discovery algorithms, relevant to performance and to a quality of obtained subgroups, respectively. Depending on a particular task and an application domain, these challenges can be solved. Performance issues can be caused by several aspects of the data, for example, size and dimensionality of real-life datasets, which may contain a huge number of rows as well as many attributes with high cardinality. Such complex data is challenging for existing discovery algorithms, primarily for reasons of computation time: all these aspects will have an impact on the time required for mining the data. If numeric attributes are concerned, detailed analysis of the data will imply high cardinalities on such attributes, blowing the search space. Generally, heuristics search strategy employing greedy approach can solve computational problem. However, in this case, not all interesting subgroups can be discovered; consequently, subgroup quality has to be compromised. Thus, a tradeoff must be attained between subgroup quality and computational complexity. Besides impossibility of discovering certain rules in case of using heuristics search strategy, quality of obtained subgroups is limited by subgroup redundancy, which is triggered by presence correlated attributes [43], [50]. Common best practices for increasing performance and quality of subgroups include preprocessing of the data, post-processing the rules, the discretization of continuous variables, and the use of domain knowledge.

Real-life problems usually have high dimensionality, unavoidable for most of the usual algorithms. There are two typical choices for applying a data mining algorithm if performance issues arise: redesigning the algorithm to run efficiently with huge input datasets by adjusting the search strategy or reducing the size of the data possibly with minimal change of result [43]. Stratified sampling is one of the techniques widely used in data mining to reduce the dimensionality of a data set and consists of the selection of particular instances of the data set according to some criterion.

It is very common that some of the variables collected in the data sets used to apply subgroup discovery techniques are continuous variables. Most of the subgroup discovery algorithms are not able to handle continuous variables. In this case, a prior discretization can be applied using various mechanisms, for example, based on fuzzy logic, or on clustering approach.

Using domain knowledge in data mining methods can improve the quality of data mining results. In subgroup discovery, it can help to focus the search on the interesting subgroups related to the target variable by restricting the search space. There are different approaches to include domain knowledge in subgroup discovery, for example, Semantic Subgroup Discovery, where obtained results have a complex structure which allows a human expert to see novel relationships in the data, or to use domain knowledge to identify potential causal relations and confounding subgroup patterns.

Quality of obtained subgroups is limited by subgroup redundancy which is often the result of dependencies between the non-target attributes, and lead to large numbers of variations of a particular finding. In case of using “top-k” rule generation mode, it will lead to the top of the rule list being populated with different variations on the same theme, and losing alternative rules. Removing redundancy can be easily done in a post-processing step of the algorithm. There are, however, attempts to integrate it in the beam search. For example, van Leeuwen and Knobbe [50] incorporate redundancy metrics based on subgroup descriptions, or subgroup covers in the beam search. CN2-SD algorithm use on-fly example removal [51].

3.1.4. Specifics of the Dataset and Task

There are a number of subgroup discovery algorithms currently available in the data mining community. However, specifics and challenges of a particular task and the dataset must be addressed. The dataset used in the current work has a huge size and a small number of attributes, comparing to what mostly presented in the academic research. For example, developers of CN2-SD algorithm [51] used as benchmarks datasets containing no more than 5000 examples, whereas the number of attributes for some datasets exceeds 50. In contrast, the dataset used in this work contains tens of millions of records, while every record represents a single page view, and the number of attributes in most of experiments does not exceed dozen. As mentioned above, for constructing or applying a subgroup discovery algorithm the following elements must be considered: type of the target variable, description language of the subgroups, quality measure, search strategy, number of obtained subgroups, and the prediction mode. Specifics of kliknieuws.nl dataset include the following:

- Size of the dataset is large, including approximately 100 000 page views per day
- Each page view contains typically 10 banners
- The target class variable is binary; it is either “click” (1) or “no-click” (0) on any banner during the page view
- Number of clicks constitutes approximately 0.5% of number of page views, thus making the class distribution extremely skewed
- Contextual attributes, provided by kliknieuws.nl, are binary (presents of photo, video, or forum on the viewed page) or nominal (page category, and visitor’s operating system and browser)

As it follows from the dataset description, the type of the target attribute is binary, and the input attributes are nominal. To outline other elements and optimization strategies, some preliminary definitions must be made. We introduce two modifications into the traditional subgroup discovery algorithms: dataset aggregated representation, and use of χ^2goodness-of-fit test as check for subgroup redundancy, explained in Section 3.1.3. We show that aggregated representation of the data increases performance of the algorithm without compromising rule quality, and χ^2goodness-of-fit test is a computationally inexpensive way for filtering redundant subgroups.

3.1.5. Formal Notations

Definition 1 (Dataset) A dataset consisting of n records, each having m attributes and with a binary target class C is defined in the following way:

Let $\{a_1, \dots, a_m\}$ be a set of attributes with a finite cardinality $dom(a_i) = \{v_{i,1}, \dots, v_{i,k_i}\}$. Let $X = dom(a_1) \times \dots \times dom(a_m)$. Let $C = \{0,1\}$ be the binary class label. We define a dataset $D = \left((x_i, c_i)\right)_{i=1}^{n} \subseteq X \times C$, where x is an antecedent (condition), c is consequence (target class). For short, we will use attribute-value pair notation whenever possible: $\boldsymbol{A_i} = a_i{:}\, v_{i,j}$. The notation assumes that in formulas attribute a_i may be initialized with only one value at the same time.

Definition 2 (Equivalence) Attribute-value pairs $\boldsymbol{A_i} = a_i{:}\, v_{i,i_a}$ and $\boldsymbol{A_j} = a_j{:}\, v_{j,j_a}$ are equivalent (denoted as $\boldsymbol{A_i} = \boldsymbol{A_j}$) if $a_i = a_j$ and $v_{i,i_a} = v_{j,j_a}$. Records (x_i, c_i) and (x_j, c_j) are equivalent (denoted as $(x_i, c_i) = (x_j, c_j)$) if $x_i = x_j$ and $c_i = c_j$.

Definition 3 (Rule) A rule R can be defined as a projection of a conjunction of attribute-value pairs to a target class $R{:} \bigwedge_i \boldsymbol{A_i} \rightarrow c$. Thus, with each rule, a set of attribute-value pairs may be associated which describes the antecedent of the rule $R{:} \{\boldsymbol{A_i}\}_{i=1}^{k} \rightarrow c$. The antecedent of the rule $\{\boldsymbol{A_i}\}_{i=1}^{k}$we call a subgroup descriptor, or simply a subgroup. We will omit curly brackets if it is clear form context, i.e. writing $\boldsymbol{A_1 A_2 A_3}$ instead of $\{\boldsymbol{A_1}, \boldsymbol{A_2}, \boldsymbol{A_3}\}$

To define Parent-Child relations between rules we will use operations over sets defined in traditional way, with the equivalence operation as defined above.

Definition 4 (Parent-Child subgroup relation) Let $\{\boldsymbol{A_i}\}_{i=1}^{k}$ and $\{\boldsymbol{B_j}\}_{j=1}^{l}$ be descriptors of two subgroups with $k < l$. If $\{\boldsymbol{B_j}\}_{j=1}^{l}$ is an extension of $\{\boldsymbol{A_i}\}_{i=1}^{k}$, i.e. $\forall \boldsymbol{A_i}\ \exists \boldsymbol{B_j} :\ \boldsymbol{B_j} = \boldsymbol{A_i}$, or using traditional notation of set theory $\{\boldsymbol{A_i}\}_{i=1}^{k} \subset \{\boldsymbol{B_j}\}_{j=1}^{l}$, then $\{\boldsymbol{A_i}\}_{i=1}^{k}$ is a parent of $\{\boldsymbol{B_j}\}_{j=1}^{l}$, and $\{\boldsymbol{B_j}\}_{j=1}^{l}$is a child of $\{\boldsymbol{A_i}\}_{i=1}^{k}$.

Consider, for example, two subgroups $x = 1$ and $x = 1 \wedge y = 1$. The first subgroup is the parent of the second, and the second is the child of the first. However, subgroups $x = 1$ and $x = 0 \wedge y = 1$ are not in the parent-child relation.

It can be noticed that the number of distinct subgroups depends on the number of attributes and their cardinalities and does not depend on the size of the dataset. The number of distinct subgroups does not exceed $\prod_i (|dom(a_i)| + 1) - 1$. This fact can be used for dataset compression without loss of the information in cases when a dataset size is large, but the number of attributes and their cardinalities are small.

We use the aggregation operation to achieve data compression: equivalent records of the original dataset are represented with one record of the aggregated dataset. The number of equivalent records is stored as the additional, so-called “weight” attribute, of the aggregating record.

Definition 5 (Aggregated Dataset) Consider a dataset $D = \left((x_i, c_i)\right)_{i=1}^{n}$ consisting of n records, each having m attributes and with a binary target class C as defined above. Let $w \in \mathbb{N}$ define a weight, i.e. the number of records in the original dataset D. We define mapping $f{:} D \rightarrow D'$ from original dataset D to aggregated dataset $D' = \left(\left(x_j, c_j, w_j\right)\right)_{j=1}^{n'} \subseteq X \times C \times \mathbb{N}$ in the following way: $\forall (x_i, c_i) \in D$ let $R \subseteq D$ represent a set of equivalent records in D, i.e. $R = \left\{\left(x_{j_1}, c_{j_1}\right), \dots \left(x_{j_h}, c_{j_h}\right)\right\}$ for which it holds that $(x_i, c_i) = (x_j, c_j)$. Then $f\left((x_i, c_i)\right) = \left(x_j, c_j, w_j\right)$, where $w_j = |R|$.

From the definition it follows that $\sum_{j=1}^{n'} w_j = n$. Compression of the aggregation operation can be calculated as $cr = \frac{|D|}{|D'|} = \frac{|n|}{|n'|}$.

An illustrative example is given in Figure 9. Records 1 and 2 are combined together, and also records 3 and 6 are combined together. Compression rate $cr = 6/4 = 1.5$. The original order of records is not preserved in the aggregated dataset.

Original dataset D

Index	Attributes		Class
i	A	B	C
1	0	0	0
2	0	0	0
3	0	1	1
4	1	1	0
5	1	1	1
6	0	1	1

Aggregated dataset D'

Index	Attributes		Class	Weight
i	A	B	C	w
1	0	0	0	2
2	0	1	1	2
3	1	1	0	1
4	1	1	1	1

Figure 9 Dataset aggregation

Property 1 Aggregation does not incur loss of information with respect to machine learning algorithms, which are insensitive to the order of records such as subgroup discovery.

Proof. Aggregation is a reversible operation with the exception of the order of records is not preserved. That is, given an aggregated dataset $D' = \left(\left(x_j, c_j, w_j\right)\right)_{j=1}^{n'} \subseteq X \times C \times \mathbb{N}$, non-aggregated dataset $D = \left((x_i, c_i)\right)_{i=1}^{n} \subseteq X \times C$ can be constructed by adding to it $\left(x_j, c_j\right)$ records $w_j$times■.

Property 2 Compression rate $cr \geq 1$, i.e. after the aggregation, dataset size does not increase.

Proof. From equation $\sum_{i=1}^{n'} w_i = n$, given that $w_i \geq 1$ since $w_i \in \mathbb{N}$, it follows that $n' \leq n$, thus $cr \geq 1$■.

These two properties of the aggregation operation show that it is possible to improve algorithm's performance even without using heuristic search strategy and compromising rule quality.

Definition 6 (positive, negative and total weights) For a given aggregated dataset D' with binary target class we will denote total weight $N = \sum_i w_i$ as the number of records in the non-aggregated dataset D, total weight of positive records $N^+ = \sum_{i,c_i=1} w_i$ and total weight of negative records $N^- = \sum_{i,c_i=0} w_i$. Thus, $N = N^+ + N^-$.

For each subgroup $\boldsymbol{A}$ we associate its weight $n_{\boldsymbol{A}} = \sum_{i,x_i \ni \boldsymbol{A}} w_i$, its positive weight $n_{\boldsymbol{A}}^+ = \sum_{i,x_i \ni \boldsymbol{A},c_i=1} w_i$, and its negative weight $n_{\boldsymbol{A}}^- = \sum_{i,x_i \ni \boldsymbol{A},c_i=0} w_i$. Similarly, $n_{\boldsymbol{A}} = n_{\boldsymbol{A}}^+ + n_{\boldsymbol{A}}^-$.

In the case of kliknieuws.nl dataset, N^+denotes the total number of clicks on banners, N denotes the total number of page views; $n_{\boldsymbol{A}}^+$ denotes the total number of clicks on banners contained in subgroup $\boldsymbol{A}$, $n_{\boldsymbol{A}}$ denotes the number of page views contained in subgroup $\boldsymbol{A}$. Coverage of the subgroup (or a rule) can be calculated as $Cov_{\boldsymbol{A}} = n_{\boldsymbol{A}}/N$.

In the current work, we define click through rate with regards to page view. Thus, terms CTR and probability of click on a banner are interchangeable. We adopt the following formal definition of click through rate:

Definition 7 (CTR) $CTR_{av} = N^{+}/N$ is the average probability of click on a banner, i.e. the prior distribution. Similarly, $CTR_{\boldsymbol{A}} = n_{\boldsymbol{A}}^{+}/n_{\boldsymbol{A}}$ is the probability of click on a banner for a given subgroup $\boldsymbol{A}$.

We will use χ^2 test of independence to give the definition of context according to Computer Science view. We adopt Turney's [35] division of attributes on primary, contextual, and irrelevant, and will use χ^2 test of independence, proposed by Widler [36], but in a modified form.

We give definition of primary, contextual, and irrelevant attributes. Let a be a nominal attribute, $a = \{v_{a,1}, \dots, v_{a,s}\}$.

Definition 8 (Primary attribute, Computer Science view): Attribute a is primary if a target attribute depends on a according to χ^2 test of independence.

In other words, the attribute is primary if fixing its values leads to significantly different distribution of the target variable comparing to the prior distribution, measured according to χ^2 test of independence. The test is constructed as follows (Figure 10):

	clicks, $\sum_{C=1} w$	no-clicks, $\sum_{C=0} w$	page views, $\sum w$
$a = v_{a,1}$	$n_{a=v_{a,1}}^{+}$	$n_{a=v_{a,1}}^{-}$	$n_{a=v_{a,1}}$
…	…	…	…
$a = v_{a,s}$	$n_{a=v_{a,s}}^{+}$	$n_{a=v_{a,s}}^{-}$	$n_{a=v_{a,s}}$
$\sum$	N^{+}	N^{-}	N

Figure 10 χ2 test for primary attributes

Null Hypothesis is that attribute a and the target attribute are independent. Alternative Hypothesis means that attribute a and the target attribute are related. The test rejecting the Null Hypothesis indicates that attribute a is primary.

Definition 9 (Contextual attribute, Computer Science view): Attribute a is contextual if it is not primary, and there is a subgroup of $\{\boldsymbol{F_i}\}_{i=1}^{k}$ for which it holds that a target attribute depends a it according to χ^2 test of independence (Figure 11). It may be noted that according to Widmer [36], in such situation, set of attributes $\{f_i\}_{i=1}^{k}$, but not attribute a will be contextual.

	clicks, $\sum_{C=1} w$	no-clicks, $\sum_{C=0} w$	page views, $\sum w$
$\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a = v_{a,1}\}$	$n^{+}_{\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a=v_{a,1}\}}$	$n^{-}_{\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a=v_{a,1}\}}$	$n_{\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a=v_{a,1}\}}$
…	…	…	…
$\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a = v_{a,s}\}$	$n^{+}_{\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a=v_{a,s}\}}$	$n^{-}_{\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a=v_{a,s}\}}$	$n_{\{\boldsymbol{F_i}\}_{i=1}^{k} \cup \{a=v_{a,1}\}}$
$\sum, \{\boldsymbol{F_i}\}_{i=1}^{k}$	$n^{+}_{\{\boldsymbol{F_i}\}_{i=1}^{k}}$	$n^{-}_{\{\boldsymbol{F_i}\}_{i=1}^{k}}$	$n_{\{\boldsymbol{F_i}\}_{i=1}^{k}}$

Figure 11 χ2 test for contextual attributes

If Null Hypothesis is rejected, then attribute a is contextual, i.e., it is useful in the context created by $\{\boldsymbol{F_i}\}_{i=1}^{k}$.

Definition 10 (Irrelevant attribute, Computer Science view): Attribute a is irrelevant if it is neither primary, nor contextual.

Definitions of primary, contextual, or irrelevant are somewhat similar to those used by Widmer in [36] in the way of employing χ^2 test. The difference is that Widmer used by applying χ^2 goodness-of-fit test to check each attribute-value pair separately, and then, summarized these tests to derive a conclusion about the attribute. We, however, use one χ^2 test of independence to check if there is a correlation between the tested attribute and the target variable and if this correlation is intact for the entire dataset or exists only under some attributes fixed.

We will be using contextual classification of attributes, to evaluate Kliknieuws.nl dataset. It is also important to note that discretization may affect an attribute's classification: depending on discretization, the attribute may be classified as primary, contextual, or irrelevant.

3.1.6. Quality Measures

Interestingness of discovered subgroups is defined by the quality measures. Hence, the choice of the quality measures employed to extract and evaluate the rules is one of the most important aspects in subgroup discovery. Quality measures are task-dependent, and there are a number of measures available in various algorithms. The most common quality measures used in subgroup discovery are described here, classified by their main objective such as complexity, generality, precision, and interest. For generalization, to avoid repeating notations for positive and negative rules separately, we will use '$*$' character; in formulas below it may be replaced by '$+$' for positive rules and '$-$' for negative rules.

Measures of complexity are related to the interpretability of the subgroups, namely to the simplicity of the knowledge extracted from the subgroups. These measures are the following:

- Number of subgroups in the learned model, which is equivalent to the number of rules: (N_R)
- Length of subgroup descriptor, or the number of variables of the antecedent of the rule: (n_V). The number of variables for a set of rules is computed as the average of the variables for each rule of the set: (N_V).

Measures of generality are used to quantify the quality of individual rules according to the individual patterns of interest covered. These measures are the following:

- Coverage, which measures the percentage of examples covered on average: $Cov = \frac{n}{N}$
- Support, which measures the frequency of correctly classified examples covered by the rule: $Sup(R^{*}) = \frac{n^{*}}{N}$

Measures of precision show the precision of the subgroups and are widely used in the extraction of association rules and classification. These measures are the following:

- Confidence, which measures the relative frequency of examples satisfying the complete rule among those satisfying only the antecedent. This can be computed with different expressions, for example, $Cnf(R^{*}) = \frac{n^{*}}{n}$
- Bias, or confidence gain of the rule is $Bias(R^{*}) = \frac{n^{*}}{n} - \frac{N^{*}}{N}$
- Precision, which measures the tradeoff between the true and false positives covered in a lineal function: $Q(R^{*}) = n^{*} - c \cdot (n - n^{*})$ where $(n - n^{*})$ are false positives, i.e. the examples satisfying the antecedent, but not the target variable, and the parameter c is used as a generalisation parameter. This quality measure is easy to use because of the intuitive interpretation of this parameter.

Measures of interest are intended for selecting and ranking patterns according to their potential interest to the human expert. These measures are the following:

- Interest, which measures the interest of a rule determined by the antecedent and consequent: $Int(R) = \frac{\sum_{i=1}^{n_v} Gain(A_i)}{n_v \cdot \log_2(|dom(G_k)|)}$, where $Gain$ is the information gain, $|dom(G_k)|$ is the cardinality of the target variable, and A_i is the number of values or intervals of the variable.
- Novelty, which measures unusualness of subgroups comparing to average distribution across the dataset: $Nov(R^{*}) = n^{*} - N^{*} \cdot n$
- Significance, which indicates the significance of a finding, if measured by the likelihood ratio: $Sig = 2 \cdot \left(n^{+} \cdot log \frac{n^{+}}{N^{+} \cdot \frac{n}{N}} + n^{-} \cdot log \frac{n^{-}}{N^{-} \cdot \frac{n}{N}} \right)$

There are a large number of hybrid quality measures since subgroup discovery attempts to obtain a tradeoff between generality, interest and precision in the results obtained. These measures are the following:

- Sensitivity, which is the proportion of actual matches that have been classified correctly: $Sens(R^{*}) = \frac{n^{*}}{N^{*}}$. This quality measure may be used to evaluate the quality of the subgroups in the Receiver Operating Characteristic (ROC) space [43]. Sensitivity combines precision and generality related to the target variable.

- False Alarm, also known as the false-positive rate, which covers the examples which are not in the target variable: $FA(R^*) = \frac{n-n^*}{N-N^*}$. Similar to sensitivity, this quality measure is used for evaluating the quality of the subgroups in the ROC space.
- Unusualness, also known as weighted relative accuracy of the rule: $WRAcc(R^*) = \frac{n}{N} \cdot \left(\frac{n^*}{n} - \frac{N^*}{N}\right)$. The unusualness of a rule can be described as the balance between the coverage of the rule and its accuracy gain. There are modifications of $WRAcc$ measure, for example, for handling multiple values of a nominal target class, or balancing weights between coverage of the rule and its accuracy gain. In the current work, we will use a modification of weighted relative accuracy, which values bias twice higher than accuracy gain: $Binomial(R^*) = \frac{n}{N} \cdot \left(\frac{n^*}{n} - \frac{N^*}{N}\right)^2$

The goal for the algorithm is to find regions in the dataset where the number of records with positive (n^+) and negative (n^-) class labels has distribution significantly different from the prior. This distribution is represented as $CTR = \frac{n^+}{n^+ + n^-} = \frac{n^+}{n}$. Since we strive to build a model for a recommendation system, it is justified to aim at discovering a large number of small subgroups, which have unusually high or unusually low click through rate. Thus, we give more preference to the difference of the target property rather than to a size of the region. The size of the region is still considered by the algorithm to prune statistically insignificant regions and prevent overfitting. Figure 12 depicts the example, where the average CTR across the dataset is 0.5%, whereas we attempt to find small, but significant regions having click-through rates as high as 0.90%, or as low as 0.20%.

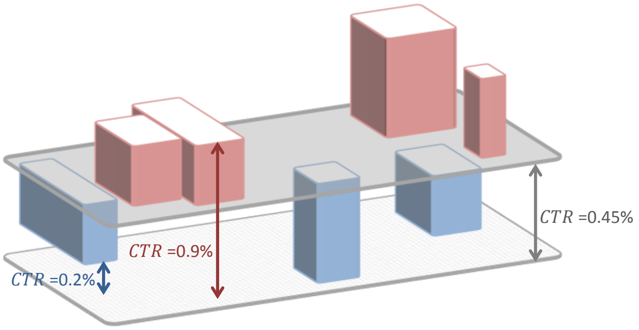


Figure 12 Regions of interest shown as red and blue bars

Then, as a quality metric we choose $Bias(R^+) = \frac{n^+}{n} - \frac{N^+}{N}$ for rules with the positive target class, and $Bias(R^-) = \frac{n^-}{n} - \frac{N^-}{N}$ for rules with the negative target class. $Bias(R^+)$ may be interpreted as difference between CTR of the subgroup (n^+/n) and CTR of the entire dataset (N^+/N).

The implemented algorithm also supports other quality measures such as $WRAcc$ and its variants. Using $WRAcc$ or other quality measure employing subgroup's coverage $Cov(R) = \frac{n}{N}$ allows for adaptive pruning strategies since coverage is inversely proportional to rule's length, and during top-down search it does not increase.

3.1.7. Search strategies

Our algorithm uses top-down breadth-first search strategy, which enables both χ^2 removing redundant rules, pruning based on minimal coverage. As it was noted in Section 3.1.3, one of the major challenges of subgroup discovery algorithms is computational complexity caused by both dimensionality of the attribute space and dataset size. In fact, a bottle neck of a typical subgroup discovery algorithm is iteration over example set and matching every example with a subgroup descriptor (the antecedent of the rule), thus assigning such basic statistical scores as coverage, positive and negative weights to each subgroup. In case of exhaustive search, the attribute space defines the number of subgroups as $\prod(|dom(a_i)|+1)-1$. For calculating subgroup's positive (n^+) and negative (n^-) class label distribution, it is necessary to match subgroup descriptor with every record of the dataset to determine if the record is covered by the subgroup. Thus, the number of calls of the comparison function on each iteration of the breadth-search algorithm is proportional to $O_{RECORDS} \cdot O_{SUBGROUPS}$. In settings of the current work, we encounter a situation when both $O_{RECORDS}$ and $O_{SUBGROUPS}$ exceed 50'000, which explodes the number of call of the comparison function to $2.5*10^9$, making the algorithm computationally infeasible.

The most effective way of reducing computational complexity is pruning the search space based on the minimal coverage (n/N), when the small and statistically insignificant subgroups are expanded by the algorithm to prevent overfitting. There are, however, other methods for reducing complexity. Firstly, we propose a method which significantly decreases $O_{RECORDS}$ by introducing the auxiliary example weight attribute and aggregating examples with the equal signatures. Secondly, we employ limiting search $Depth$, which is a common technique of pruning the search space. Finally, we try to adopt heuristics for limiting $O_{SUBGROUPS}$, number of generated subgroups, recently developed by the academic community for pruning the search space.

Let subgroup $\boldsymbol{A}$ be described by size $n_{\boldsymbol{A}}$, positive weight $n_{\boldsymbol{A}}^+$, and negative weight $n_{\boldsymbol{A}}^-$. Then, quality measures, assigning this subgroup to positive and negative classes are as follows: $Bias(R_{\boldsymbol{A}}^+) = \frac{n_{\boldsymbol{A}}^+}{n_{\boldsymbol{A}}} - \frac{N^+}{N}$, and $Bias(R_{\boldsymbol{A}}^-) = \frac{n_{\boldsymbol{A}}^-}{n_{\boldsymbol{A}}} - \frac{N^-}{N}$, where $R_{\boldsymbol{A}}^+ : \boldsymbol{A} \to 1$ and $R_{\boldsymbol{A}}^- : \boldsymbol{A} \to 0$, and N^+, N^-, N are properties of the dataset. Consider child subgroup $\boldsymbol{AB}$. Using similar notation, $Bias(R_{\boldsymbol{AB}}^+) = \frac{n_{\boldsymbol{AB}}^+}{n_{\boldsymbol{AB}}} - \frac{N^+}{N}$ and $Bias(R_{\boldsymbol{AB}}^-) = \frac{n_{\boldsymbol{AB}}^-}{n_{\boldsymbol{AB}}} - \frac{N^-}{N}$. Since N^+, N^-, N are constants for the given dataset, $Bias(R_{\boldsymbol{AB}}^+)$ is a function of $\frac{n_{\boldsymbol{AB}}^+}{n_{\boldsymbol{AB}}}$, which is in fact, probability of a click on a banner, and $Bias(R_{\boldsymbol{AB}}^-)$ is a function of $\frac{n_{\boldsymbol{AB}}^-}{n_{\boldsymbol{AB}}}$. Since size, positive and negative weights of child subgroups are anti-monotone, and cannot exceed those of the parental subgroup, the known facts about relations of parent-child metrics are as follows: $n_{\boldsymbol{AB}} \le n_{\boldsymbol{A}}$, $n_{\boldsymbol{AB}}^+ \le n_{\boldsymbol{A}}^+$, and $n_{\boldsymbol{AB}}^- \le n_{\boldsymbol{A}}^-$. Let $n_{\boldsymbol{A}}^+ > 0$, and $n_{\boldsymbol{A}}^- > 0$, i.e. there are both clicks and no-clicks in $\boldsymbol{A}$. Then, probability of click of $\boldsymbol{AB}$ subgroup, $\frac{n_{\boldsymbol{AB}}^+}{n_{\boldsymbol{AB}}}$, may be as high as 1 if $n_{\boldsymbol{AB}}^+ = n_{\boldsymbol{AB}}$, or as low as 0, if $n_{\boldsymbol{AB}}^+ = 0$. Known facts $n_{\boldsymbol{AB}} \le n_{\boldsymbol{A}}$, $n_{\boldsymbol{AB}}^+ \le n_{\boldsymbol{A}}^+$ are not helpful in prediction of the used quality measures.

Another useful observation is that probability of a click (or no-click for negative rules) of a parental subgroup represents a weighted sum of probabilities of a click of child subgroups: $\frac{n^+}{n^+ + n^-} = \frac{\sum n'^+}{\sum n'^+ + \sum n'^-}$.

So, each parental subgroup has at least one child with higher or equal probability of click and one child with lower or equal probability of click. Using this observation, it would be possible to construct a beam search, choosing for further expansion on every step subgroups with the highest biases, knowing that they may contain child subgroups with even higher biases. However, in the setting of the current work, we do not use this heuristics since we are also interested in studying contextual attributes defined in the Data Mining community as “not useful in isolation, but with fixed other attributes”. It draws our interest to subgroups with low bias, however, producing child subgroups with high bias. Since these contextual attributes may be excluded from the search space by a heuristics, employing this strategy is not desirable.

3.1.8. Pruning strategies

Pruning techniques is commonly based on the fact, that upper-bound of a quality measure of the child subgroup can be predicted based on properties of the parent subgroup. For example, size (n), positive (n^+) and negative (n^-) weights of child subgroup are anti-monotone, and hence, they cannot exceed those of the parental subgroup. Metrics, employing these properties can use these inequalities to give the upper estimates of quality measure of child subgroups. Consider, for example, $WRAcc = \frac{n}{N} \cdot \left(\frac{n^+}{n} - \frac{N^+}{N}\right)$ which is a popular quality measure of subgroup discovery algorithms. First multiplier of a child subgroup $\boldsymbol{AB}$ does not exceed parental one: $\frac{n_{\boldsymbol{AB}}}{N} \le \frac{n_{\boldsymbol{A}}}{N}$; $\frac{N^+}{N}$ is a constant for a given dataset, and $\frac{n_{\boldsymbol{AB}}^+}{n_{\boldsymbol{AB}}} \le 1$. Thus, the upper estimates of positive and negative rules of child subgroups of subgroup $\boldsymbol{A}$ are $WRAcc(R_{\boldsymbol{AB}}^+) \le \frac{n_{\boldsymbol{A}}}{N} \cdot \left(1 - \frac{N^+}{N}\right)$, $WRAcc(R_{\boldsymbol{AB}}^-) \le \frac{n_{\boldsymbol{A}}}{N} \cdot \left(1 - \frac{N^-}{N}\right)$, if an upper estimate of $WRAcc$ for a given subgroup is less than a certain threshold, it is possible to prune it with all its children.

Morishita and Sese [53], while developing association rules learning algorithm, used χ^2-values for assessment of rule interestingness. They proved that χ^2-value of child subgroup can be limited, which they effectively used for pruning search space. χ^2-values are calculated in the following way (Figure 13):

	Positive observations	Negative observations	Total
Subgroup	n^+	n^-	n
Subgroup’s addition	$N^+ - n^+$	$N^- - n^-$	$N - n$
Dataset total	N^+	N^-	N

Figure 13 Using χ^2-values for assessment of rule interestingness according to Morishita and Sese [53]

χ^2-value of the subgroup is a function of four parameters, namely n^+, n^-, N^+, N^-, the last two of which are constant for a given dataset. Thus, $\chi^2(\boldsymbol{A}) = \chi^2(n_{\boldsymbol{A}}^+, n_{\boldsymbol{A}}^-)$. Morishita and Sese’s theorem states that χ^2-value of any child subgroup does not exceed maximum of the following χ^2-values obtained from their parental subgroup: $\chi^2(\boldsymbol{AB}) \le max\{\chi^2(n_{\boldsymbol{A}}^+, 0), chi^2(0, n_{\boldsymbol{A}}^-)\ \}$. In other words, we assume that there is a child subgroup which isolates all positive observations from all negative observations, thus maximizing χ^2-value of this child subgroup. Then, comparing estimated maximum with a given threshold, it is possible to prune search space taking into account that $n^+$and n^- are anti-monotone and will decrease from parental to child subgroups. It can be noticed, that both $\chi^2(n_{\boldsymbol{A}}^+, 0)$ and $\chi^2(0, n_{\boldsymbol{A}}^-)$ are functions of one variable. Another important for us observation is that since in our settings distribution of $n^+$and n^- is considerably skewed; their influence on χ^2-value is not symmetric. Thus, it makes sense to assess $\chi^2(n^+, 0)$ and $\chi^2(0, n^-)$ for positive and negative rules independently, each with its own thresholds: $\chi^2(n^+, 0) > \tau^+$ and $\chi^2(0, n^-) > \tau^-$. In these expressions χ^2 is a monotonic function of one variable, hence, we can substitute $n^+ \geq n_\tau^+$ and $n^- \geq n_\tau^-$.

The heuristic explained above, can work well if subgroup discovery algorithm induces only positive or only negative rules. In this case, search space for positive rules will be pruned when $n^+ < n_\tau^+$. In case of both positive and negative class learning, however, this type of search space pruning is not possible since the subgroups that are useless for positive rules can be useful for negative rules, and the other way around. Stronger condition placing restrictions on both positive and negative observations, $n^+ + n^- \geq n_\tau^+ + n_\tau^-$, represents minimal coverage of the hypothesis. This is another common pruning metric used by data mining algorithms, also employed by our subgroup discovery algorithm.

3.1.9. Redundancy Removing Strategies

Presence of correlated attributes leads to large numbers of variations of a particular finding. In case of using “top-k” rule generation mode, it results in the top of the rule list being populated with different variations on the same rules, and losing alternative rules. Best practices for removing redundancy include removing or reweighting examples covered by the rule during the search, implemented in CN2, and RIPPER algorithm [51], or incorporating redundancy metrics based on subgroup descriptions and subgroup covers into search heuristics employed by van Leeuwen and Knobbe [50]. It can also be implemented as a post-processing step of the subgroup discovery algorithm.

In subgroup discovery algorithms using heuristic search, example removing or reweighting implies the following. As soon as an interesting subgroup is found, examples covered by the subgroup are reweighted, namely their positive weight is decreased linearly or exponentially depending on the implementation, or removed from the dataset. Thus, their contribution into subgroups which may be discovered further will be minimized or nullified. If examples of a parent subgroup are removed, then child subgroups will not be discovered by the further search. If examples of the parent subgroup are reweighted, then quality measures of child subgroups will be biased. This strategy can work well with quality measures, capable of giving upper estimates of quality measure of child subgroups such as $WRAcc$. In this case, the algorithm can guarantee that child subgroups will not have quality metrics better than the current parental subgroup, whose examples are removed or reweighted. Since our primary quality measure is $Bias$, and search strategy top-down breath-first approach, as discussed in Section 3.1.7 it is likely that child subgroups in many cases will have higher quality measures that the parental subgroups. Thus, applying this redundancy removing strategy is undesirable.

If on fly reweighting or removing examples discussed above is not possible, the “fair” strategy would look as follows: after the search run, the best subgroup is added to the model, and examples covered by it are reweighted or removed. Then, the search runs the second time, and the best subgroup is, again, added to the model, and examples covered by it are reweighted or removed, etc. This strategy ensures that after each cycle the best subgroup is added to the model. However, the number of necessary runs of the algorithm will be equal to the number of subgroups, which make the strategy infeasible.

The alternative approach that we use in our algorithm is as follows. The algorithm has only one search run that induces a subgroup space, storing all subgroups meeting minimal coverage ($Cov(R)$) and minimal quality measure ($Bias(R)$) criteria. Then, the post-processing step consists of a number of iterations, in which the best subgroup is chosen, examples covered by it are removed and all other subgroups are reweighted. If after reweighted subgroups do not meet minimal coverage criterion, it is removed from the subgroup space. Meeting minimal quality bias threshold is optional in our implementation. Test results described below demonstrate that minimal quality bias threshold option in many cases affects metrics of the learned model significantly.

Separating learning the subgroups on two phases, namely inducing the subgroup space during the first phase, and filtering redundant subgroups during the second phase, allows for separation of quality measures. Our algorithm uses minimal bias threshold for subgroup space induction; however, depending on analyst's preference, a different quality measure for filtering redundant subgroup can be employed. This quality measure can be, for example, rule length $Length(R)$ if we are interested in a model having smaller average length of the rules, or $WRAcc(R)$ which gives equal significance to rule coverage and the bias if we are interested in a model having smaller number of rules. Test results described below demonstrate that choosing different quality measure result in significantly different models.

In addition to this method, removing redundancy as a post-processing step, we propose to use χ^2goodness-of-fit test is a computationally inexpensive way for filtering redundant subgroups on fly, during the search phase. The idea of the method is as follows. In the top-down search strategy, parental subgroups are generated before child's subgroups. Thus, subgroup $\boldsymbol{ABC}$ is evaluated after metrics of $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{AB}, \boldsymbol{AC}, \boldsymbol{BC}$ subgroups have been evaluated. By using χ^2 goodness-of-fit test with a given significance level we check if the child subgroup has significantly different distribution of the target variable comparing to its parental subgroups presenting in the model. If the test indicates that the difference is not significant, then the child subgroup is not added to the model regardless its quality measures.

Let given subgroup $\boldsymbol{A}$ as attribute-value pair $a = v_a$. With χ^2 test we attempt to check if subgroup $\boldsymbol{AB}$: $a = v_a \wedge b = v_b$ has a significantly different distribution of the target class. In other words, fixing attribute b with value v_b, given fixed attribute $a = v_a$ leads to a significantly different distribution of the target class. Thus, distribution $n_{\boldsymbol{AB}}^{+}/n_{\boldsymbol{AB}}^{-}$is tested against the prior distribution $n_{\boldsymbol{A}}^{+}/n_{\boldsymbol{A}}^{-}$. We construct χ^2 test as follows (Figure 14):

	clicks, $\sum_{c=1} w$	no-clicks, $\sum_{c=0} w$	page views, $\sum w$
$\boldsymbol{AB}$: $a = v_a \wedge b = v_b$	$n_{\boldsymbol{AB}}^{+}$	$n_{\boldsymbol{AB}}^{-}$	$n_{\boldsymbol{AB}}$
$\boldsymbol{A}$: $a = v_a$	$n_{\boldsymbol{A}}^{+}$	$n_{\boldsymbol{A}}^{-}$	$n_{\boldsymbol{A}}$
Σ	$n_{\boldsymbol{AB}}^{+} + n_{\boldsymbol{A}}^{+}$	$n_{\boldsymbol{AB}}^{-} + n_{\boldsymbol{A}}^{-}$	$n_{\boldsymbol{AB}} + n_{\boldsymbol{A}}$

Figure 14 χ^2 goodness-of-fit test for removing redundant rules

The Null and the Alternative Hypotheses can be formulated in the following way:

H_0: Two nominal variables are independent: n^+/n^- distribution is not correlated by feature $b = v_b$, given $a = v_a$.

H_a: Two nominal variables are related: n^+/n^- distribution depends on feature $b = v_b$, given $a = v_a$.

The advantage of this scheme is that filtering of redundant subgroups is done on fly, during the search phase of the algorithm, meaning that there is no necessity of storing possibly a huge number of redundant rules until the post-processing phase where removing redundant rules is done.

3.1.10. Pseudo-code of the algorithm

Pseudo-code of the algorithm is presented in Figure 15.

```
Given: ExampleSet, MaxLength, MinCoverage, MinBias, QualityMeasure;

Iterate over ExampleSet
        calculate $N$, $N^+$,$N^-$;
Generate initial SubgroupList of length 1;
Repeat MaxLength
        Iterate over ExampleSet, SubgroupList
                calculate $n$, $n^+$,$n^-$ for every Subgroup;
        Iterate over SubgroupList
            If ($n$<MinCoverage OR max(($n^+$/n-$N^+$/N),($n^-$/n-$N^-$/N)) <MinBias)
                Delete Subgroup from SubgroupList;
        Iterate over Model, SubgroupList
            If (Model.Subgroup isParent(SubgroupList.Subgroup) AND
                chi2test(Model.Subgroup,SubgroupList.Subgroup) == true)
                Delete Subgroup from SubgroupList;
        Add SubgroupList to Model;
        Generate SubgroupList of length+1;

BestSubgroup = Model.getSubgroupWithHighestQualityMeasure(QualityMeasure);
Delete BestSubgroup  from Model;
Add BestSubgroup  to FilteredModel;
Repeat
        Iterate over ExampleSet, Model
                If (Example covered by BestSubgroup AND Example covered by Model.Subgroup)
                        Reweight Model.Subgroup;
        Iterate over Model
                If ($n$<MinCoverage OR max(($n^+$/n-$N^+$/N),($n^-$/n-$N^-$/N)) <MinBias)
                        Delete Subgroup from Model;
        BestSubgroup = Model.getSubgroupWithHighestQualityMeasure(QualityMeasure);
        Delete BestSubgroup  from Model;
        Add BestSubgroup  to FilteredModel;
until Model is empty;
Return FilteredModel;
```

Figure 15 Pseudo-code of the subgroup discovery algorithm

3.1.11. Applying the rules

As mentioned in Section 3.1.2, a subgroup discovery algorithm can induce overlapping subgroups. The same record of the dataset can be covered by a few rules, attributing it to different target classes. Hence, if the learned model is intended for use in a recommender system, it is necessary to employ an arbitration mechanism, defining how to classify records, covered by more than one rule. CN2-SD algorithm [51] uses a voting model, in which quality measure of the rule is used as a score. For classification of a new record, scores of the rules covering it are summarized for each target class. The record is assigned to the class which has highest total score. The possible alternatives include applying the most specific rule, or applying the rule with the highest quality measure.

Since the target class distribution is extremely skewed and $Bias$ is the primary measure, then contribution of positive and negative rules into the voting model will not be symmetric. If CTR of a subgroup may theoretically vary from 0 to 1, and but the prior $CTR = 0.005$, then the maximal $Bias$ of the negative rule cannot exceed 0.005, whereas subgroups of significant size with $CTR > 0.025$ are found in the dataset, thus having $Bias > 0.02$. In this case, at least four negative rules must vote to compensate for one positive rule. Hence, we use most specific rule as a prediction mode. However, voting model is also implemented and can be employed optionally.

3.1.12. Summary of Settings Used

The choices made for construction of the algorithm, are summarized in Table 3. These parameters are hardcoded and cannot be customized by a user.

Table 3 Hardcoded Settings

Choice element	Value
Types of attributes	• target class: binary • regular: nominal
Description language of the subgroups	• conjunction of attribute-value pairs
Quality measure	• search phase: Bias • redundancy removal phase: Bias, WRAcc, Length, or others
Search strategy	• top-down, breadth-first
Pruning	• post-processing phase • χ^2 goodness-of-fit test
Number of obtained subgroups	• not limited by the user explicitly, depends on the dataset, quality measure and coverage thresholds

3.2. Implementation

The algorithm has been implemented as a plug-in for Rapidminer, one of the world-leading open-source systems for data mining [54]. Figure 16 illustrates the use of the plug-in. Dataset is read from a remote database by the first module. Then, three modules in the middle convert numerical attributes into nominal and binominal target class, and assign roles of special attributes to weight and class label. The last module in the chain implements the subgroup discovery algorithm.

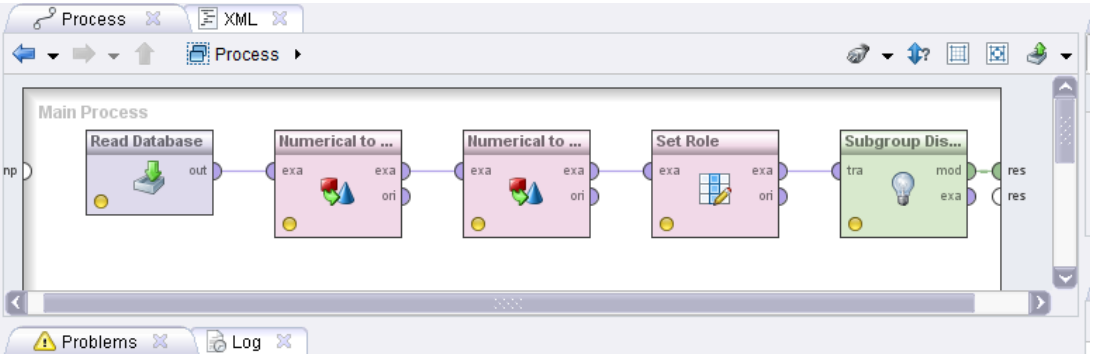
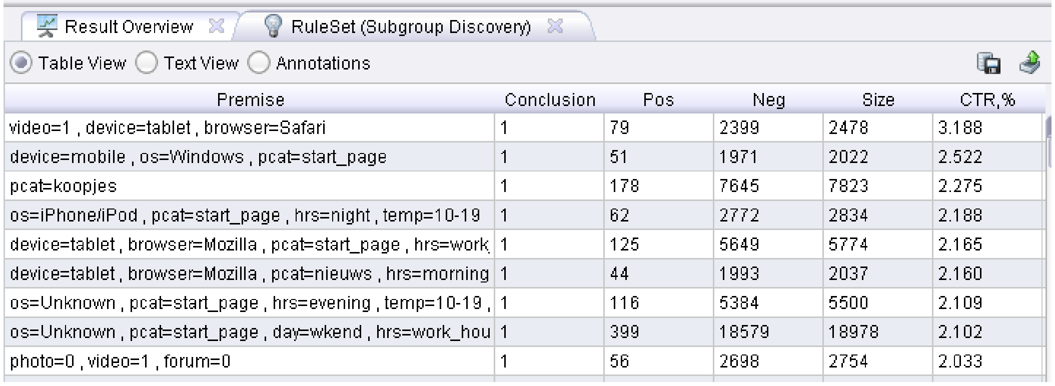


Figure 16 Example of subgroup discovery plug-in connection scheme

Figure 17 shows an example of a model learned by a subgroup discovery algorithm.



● Table View ○ Text View ○ Annotations

Premise	Conclusion	Pos	Neg	Size	CTR,%
video=1 , device=tablet , browser=Safari	1	79	2399	2478	3.188
device=mobile , os=Windows , pcat=start_page	1	51	1971	2022	2.522
pcat=koopjes	1	178	7645	7823	2.275
os=iPhone/iPod , pcat=start_page , hrs=night , temp=10-19	1	62	2772	2834	2.188
device=tablet , browser=Mozilla , pcat=start_page , hrs=work	1	125	5649	5774	2.165
device=tablet , browser=Mozilla , pcat=nieuws , hrs=morning	1	44	1993	2037	2.160
os=Unknown , pcat=start_page , hrs=evening , temp=10-19 ,	1	116	5384	5500	2.109
os=Unknown , pcat=start_page , day=wkend , hrs=work_hou	1	399	18579	18978	2.102
photo=0 , video=1 , forum=0	1	56	2698	2754	2.033

Figure 17 Example of a set of rules produced by a subgroup discovery algorithm

Customizable settings are summarized in Table 4.

Table 4 Customizable settings

Choice element	Value
Number of rules	• **Above minimum utility**: all the rules satisfying minimal Bias and minimal Coverage conditions will be in the output • **K-best rules**: only k rules having the best quality measure, k is specified separately
Utility function for post-processing	These are the quality measures used for ranking rules during the post processing step for removing redundancy: • Bias

	• WRAcc • Binomial, etc.
Rule generation mode	Specifies rule of which target class are learned: • Positive • Negative • Both
Prediction model	• Most specific rule • Voting model
Minimal Bias, **Minimal Coverage of the subgroup,** **Depth of search**	Open fields, where a numeric value must be specified
Redundancy filter (χ^2 filter) **Redundancy filter (post-processing)** **Bias check during the post-processing**	Checkboxes, where yes/no choice must be made

3.3. Evaluation Framework

The set of rules produced by the subgroup discovery algorithm is used for the prediction task. In the settings of Kliknieuws.nl it means that the algorithm predicts if a page view will have high or low CTR, or default CTR is it is not covered by the rules. Thus, as for classification task, we will use confusion matrix as a tool for evaluating algorithms. We also introduce some metrics, derived from the confusion matrix.

The dataset has two target classes, Click (1) and No-Click (0). Our algorithm produces a classification model assigning to the examples one of three target labels: Click (1), No-Click (0) and Undefined (-1). Thus, the confusion matrix has size 3x2 (see Figure 18).

		Actual	
		No-Click (0)	Click (1)
Predicted	No-Click (0)	a	b
	Click (1)	c	d
	Undefined (-1)	g	h

Figure 18 Confusion Matrix

CTR, which is one of our primary metrics, assessing performance of the algorithm, can be directly derived from confusion matrix:

- $CTR_{low} = CTR_{class(No\ Click)} = \frac{b}{a+b}$
- $CTR_{high} = CTR_{class(Click)} = \frac{d}{c+d}$
- $CTR_{not\ covered} = CTR_{class(Undefined)} = \frac{h}{g+h}$

Then, we develop specific metrics shaped specifically for our task: *weighted bias* and *coverage*.

- $W_{bias} = W_{low} \cdot (CTR_{av} - CTR_{low}) + W_{high} \cdot \left(CTR_{high} - CTR_{av}\right)$, where $W_{low} = (a + b)$ is the weight of CTR_{low} class, $W_{high} = (c + d)$ is the weight of CTR_{high} class, and $CTR_{av} = \frac{b+d+h}{a+b+c+d+g+h}$ is the average CTR
- $Cov_{low} = \frac{W_{low}}{W_{total}} = \frac{a+b}{a+b+c+d+g+h}$
- $Cov_{high} = \frac{W_{high}}{W_{total}} = \frac{c+d}{a+b+c+d+g+h}$

It's noticeable that weighted bias and coverage would produce the same results if they have been applied ranking performance model. Thus, using ranking performance model complicates the testbed, but does not produce additional value.

Weighted bias is an integral metric, which combines difference of CTRs, i.e. between CTR_{high} and CTR_{av}, CTR_{low} and CTR_{av}, with coverage Cov_{low} and Cov_{high}. Thus, it defines performance of our algorithm.

The last group of metrics assesses quality of rules:

- N_{rules} – the number of rules in the model
- $L_{av} = \frac{\sum_{i=1}^{N} L_i}{N}$ - average length of the rule
- $L_{W,av} = \frac{\sum_{i=1}^{N} L_i \cdot w_i}{\sum_{i=1}^{N} w_i}$ - weighted average length of the rule, where w_i is the number of page views covered by the rule
- $Cov_{Rule} = \frac{\sum_{i=1}^{N} w_i}{W_{high}+W_{low}}$ - measure of rule redundancy, calculated as the average number of page views covered by the rule

4. Case Study

4.1. Kliknieuws Business Case

Kliknieuws.nl is a Dutch online local news company whose underlying business model implies generating revenue by publishing banners on its web pages. Banners are placed by advertisers pursuing own, typically commercial goals such as increasing brand awareness, promoting products and services, online sales, or building relationships with visitors. The goal that advertisers usually aim at is more valuable than a standard banner view. A life cycle of the online advertising model consists of three stages: showing banners on a website of the publisher, clicking on banner by the publisher's visitor, and completing a goal of the target website of advertiser. The percentage of visitors clicking on a banner is called click-through rate (CTR). The percentage of visitors completing a target action is called conversion rate (CR). Currently, Kliknieuws.nl charges advertisers for impressions only. However, it has an ambition to adjust its business model and charge for clicks too (see Figure 19).

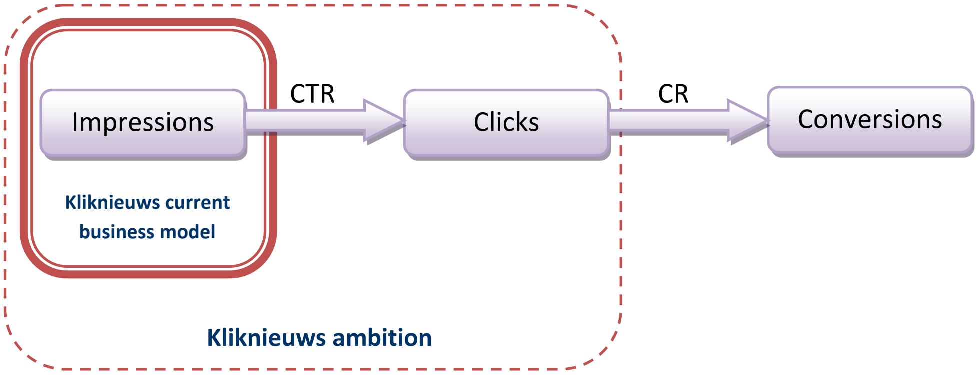


Figure 19 Current business model and ambition of banner placement

Since CTR of banners depends on many parameters relevant as to the banner itself, as to visitor or webpage context, Kliknieuws.nl wants to optimize a banner placement algorithm in such a way that it will show pay-per-click banners in situations when banners are more likely of being clicked, and to avoid showing these banners when probability of a click on a banner is low. The algorithm should not interfere with the current model, where banners are given upper and lower bounds of impressions within the advertisement period.

4.2. Dataset Description

Kliknieuws dataset consist of page view and click logs stored independently in separate data bases. As from June 2012, page view data collection method was changed from server log files to java script page tagging. Currently, page view data are currently stored in JSON format [56]. Susilo [55] conducted work for merging page view and banner click logs, by adding a random number, so-called "cb-number", to every banner impression, which is recorded in the click log is the banner was clicked. By matching these random numbers, it is possible to track page views which resulted in a banner click. Since random numbers can repeat themselves, correctness of matching was enforced by comparing IP addresses of banner impression and banner click records, or considering a limited time window between banner display and banner click which can be, for example, two hours. The former method was used for the log spanning from April 2011 to May 2012, the latter was used for the log of the new format since visitor IP addresses is not recorded in the new log.

The page view data for the period of April 2011 May 2012 were provided as server logs which need to be parsed to MySQL data base. One record of the log corresponded to one banner impression. A typical page contains ten banners; hence, size of the dataset was almost ten times larger than the number of page views. Besides merging data related to one page view into together, a challenge of elimination activity of crawlers must be solved. IP address of known crawlers [33] was used as reference in addition to analysis of User Agent browser field. In fact, 6.3% of all clicks between April 2011 and May 2012 were made by crawlers.

A page tag based method that have been employed for logging starting from June 2012, combines banner impressions of one page view into one record, thus, reducing size of the dataset. Since most of crawlers do not support Java Script, their activity is not recorded. Clicks on banners are still made by crawlers; however, by matching click with page view dataset, it is possible to exclude those clicks from statistics.

Detailed database structure for the period of April 2011 May 2012 is given in Appendix 3. Data transformation algorithm is given in Appendix 4. Database structure for a new logging method is approximately the same, with the exception of missing IP address field, and different discretization of operating systems and browsers.

Essential specific of the dataset is that approximately 25% records are duplicates, i.e. all the fields of the records are the same besides the timestamps which appear to be slightly different. The difference may constitute an interval from a few seconds to a few minutes. This phenomenon is characteristic of both data logging methods. We assume that it is caused by web pages caching, which causes browser to consequently retrieve banners with the same parameters when visitor retrieves the same page more than once. A method for overcoming a problem is to retain only the earliest occurrence of the duplicate entry. In this work, it was done by adding a unique index to the table of the database on a combination of fields which are supposed to by unique for every page view, for example, a set of cb-numbers, or a combination of the cb-number with IP address. Adding this unique index removes all the duplicate records besides the first.

The experiments were conducted on winter 2011-2012 dataset, consisting on monthly data of November 2011, December 2011, and January 2012, as well as on summer 2012 dataset spanning from June 13th to August 22nd. The daily number of page views after filtering crawlers and duplicate records, is approximately 100 thousand, 0.5% of which result in a banner click.

4.3. Defining View on the Data

As the input, the algorithm takes 11 variables, related to visitor, webpage, and external data's properties and predicts if page view will have high, low or "not covered" CTR. Variables $nieuws\_foto$, $nieuws\_video$, $nieuws\_forum$, $dl\ (URL)$, $userAgent$, $OSName$, $browser$ are contained in the log of summer 2012 dataset. An alternative way of obtaining them for November 2011 – January 2012 datasets is discussed Appendix 4. Table 5 summarizes a view of the data.

Table 5 View of the data

Variable category	Variable	Values	Data source and calculation
Webpage	photo	– 0 – 1	**nieuws_foto** variable contains 1 if a photo is present on the page, and 0 otherwise
	video	– 0 – 1	**nieuws_video** variable contains 1 if video is present on the page, and 0 otherwise
	forum	– 0 – 1	**nieuws_forum** variable contains 1 if forum is present on the page, and 0 otherwise
	pcat	– start_page – nieuws – sport – foto – forum – zoeken – selecteer-regios – pagina – agenda – koopjes – login – poll – other	Parsing **dl** value (requested URL), extracting root catalog right after domain name (see Appendix 4)
Visitor	device	– desktop – tablet – mobile – tv – unknown	Parsing **userAgent** value (see Appendix 4)
	os	– Windows – Linux – iPhone/iPod – Mac – unknown OS	Use **OSName** value
	browser	– Explorer – Firefox – Chrome – Opera – Safari – Mozilla – Netscape – unknown browser	Use **browser** value
External data	day	– wrkday – wkend	If a weekday is Saturday or Sunday, or the date is in a list of holidays, then wkend, otherwise wrkday. Both weekday and the calendar date can be extracted from the timestamp field

<table>
<tr>
<td></td>
<td>hrs</td>
<td>– morning
– work_hours
– lunch
– evening
– night</td>
<td>Times are given in the following table:
<table>
<tr><th>Period</th><th>From</th><th>To</th></tr>
<tr><td><b>morning</b></td><td>07.00</td><td>08.59</td></tr>
<tr><td><b>lunch</b></td><td>12.00</td><td>12.59</td></tr>
<tr><td><b>evening</b></td><td>23.59</td><td>18.00</td></tr>
<tr><td><b>night</b></td><td>00.00</td><td>06.59</td></tr>
<tr><td><b>work_hours</b></td><td colspan="2">Otherwise</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td>temp</td>
<td>Summer dataset:
– “<0”
– “0-9”
– “10-19”
– “20-29”
– “30+”
Winter dataset:
– cold
– chilly
– warm
– hot</td>
<td>Use wunderground.com service for reference (Appendix 4). Weather information is taken from meteorological station located near Eindhoven Airport.
TemperatureC value from CSV-file is used. Summer dataset discretization is intuitive.
For the winter dataset: cold, if the temperature is below 0°C; chilly, if the temperature is between 0°C and 10°C; warm, if the temperature is between 11°C and 20°C; hot, the temperature is above 21.</td>
</tr>
<tr>
<td></td>
<td>cond</td>
<td>Summer dataset:
– overcast
– rain
– normal

Winter dataset:
– fog
– light precipitations
– normal precipitations
– heavy precipitations
– normal</td>
<td>Use wunderground.com service for reference (see Appendix 4). Weather information is taken from meteorological station located near Eindhoven Airport.
Discretization for the summer dataset:
– Rain: if $Conditions$ value contains at least one of the following words ‘Drizzle’, ‘Hail’,’ Rain’,’ Thunderstorms’ (it can be checked by regexp function)
– Overcast: if $Condition$ value is ‘Overcast’ or ‘Mostly Cloudy’
– Normal: otherwise

Discretization for the winter dataset:
– Fog: if $Condition$ contains words ‘fog’, ‘smoke’ or ‘haze’
– Light precipitations: if $Condition$ contains words ‘light’ or ‘small’
– Heavy precipitations: if $Condition$ contains words ‘heavy’
– Normal: if $Condition$ contains words ‘Mist’, ’Cloudy’, ’Unknown’, ’Clouds’,’Overcast’
– Normal precipitations: otherwise</td>
</tr>
</table>

5. Experiments

5.1. Testbed Setup

For the experiments, a testbed was created within Rapiminer environment. The testbed reported metrics, discussed in Section 3.3 and Appendix 6. Three 10 cross-fold validation runs were done to ensure 30 measurements for each set of settings. Then, the averages and standard deviations were calculated. A separate cross-fold validation module supporting aggregated representation of the data was implemented. Some parameters of the algorithm such as those discussed in Table 3 are hardcoded and were fixed for all the tests. Customizable parameters listed in Table 4 were varied with the goal of finding an optimal set of settings or to prove certain claims. Here, the following choices were made:

- **Number of rules**: above minimum utility since the model is intended for use in recommendation system, but not for a human expert
- **Utility function for post-processing step**: experiments with Bias, WRAcc, Binomial and Rule Length quality measures were made to compare performance of models and quality of rules
- **Rule generation mode**: both target classes since it was desirable to implement arbitration of conflicting rules within the model
- **Prediction model**: most specific rule since the target class distribution is extremely skewed as discussed in Section 3.1.11.
- **Minimal Bias**: 0.3% was used in most of test; typically, the optimal value must be found experimentally, considering also desired coverage of the model, which decreases with growth of the Minimal Bias
- **Minimal Coverage of the subgroup**: the value was fixed to 2000 page views for all the tests. For finding interesting subgroups, it is desirable to keep the number small. However, if CTR is on around 0.5%, then the number of clicks in an average subgroup is 10. Further decreasing this number will make the subgroup very sensitive to individual clicks occurred by chance, and thus, cause overfitting
- **Depth of search**: it was chosen to use 7 in most of test, and decreased it to 5 and 3 to check sensitivity of the model to this parameter.
- **Redundancy filter (χ^2 filter)**: this parameter was alternated in most of tests to prove the claim about improving quality of rules without compromising on performance of the model.
- **Redundancy filter (post-processing)**: this parameter was alternated in most of tests to check its influence on rules' quality and performance of the model
- **Bias check during the post-processing**: this parameter was alternated in some test to check its influence on rules' quality and performance of the model

The experiments were run on remote computer systems of Eindhoven University of Technology. Each experiment depending on settings and the dataset took up to a few hours. Detailed results with some additional metrics are presented in Appendix 6. Below, we will consider only a subset of results for support of our claims.

5.2. Results

We will evaluate performance of models in terms of CTR and Coverage of regions CTR_{low}, CTR_{high}, Cov_{low}, Cov_{high}, and quality of rules terms of the number of rules in the model N_{rules}, average length of the rule L_{av}, weighted average length of the rule $L_{W,av}$, and rule redundancy Cov_{Rule}.

Claim 1 (effect of χ^2 filter for on-fly redundancy removal) χ^2 filter considerably improves quality of rules without compromising on performance of the model: performance of the model does not change or changes insignificantly. Figure 20 shows three tables with six different sets of setting demonstrating a proof of this claim. In most of the experiments, 95% confidence intervals of the coverage and CTRs of the regions are intersecting. In the rest of the experiments, i.e., CTR_{high} which is in fact the most sensitive parameter to the experimental settings. However, for CTR_{high} parameter, 99%-confidence intervals are intersecting, demonstrating that the difference is not significant.

	Dataset: December, Depth=7							
	minBias=0.003				minBias=0.002			
Filtering type	**No filtering**		**Chi-squared filter**		**No filtering**		**Chi-squared filter**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**Average**	**stdev**
Coverage$_{L}$	19.56	0.39	19.55	0.37	26.82	0.72	28.14	0.59
Coverage$_{H}$	18.90	0.72	18.47	0.53	33.31	0.47	31.51	0.66
CTR$_{L}$	0.11	0.01	0.11	0.01	0.18	0.02	0.20	0.04
CTR$_{H}$	0.74	0.03	0.75	0.03	0.66	0.03	0.67	0.07
N$_{Rules}$	16077.87	245.62	162.92	9.14	24120.97	270.48	295.93	9.70
Length$_{ave}$	5.45	0.01	3.38	0.07	5.45	0.01	3.72	0.05
Length$_{w,ave}$	5.04	0.01	2.6	0.07	5.05	0.01	2.69	0.03
Coverage$_{Rule}$	263.71	6.51	3.57	0.2	223.07	3.61	3.65	0.12

(a)

	Dataset: December, Depth=7, minBias=0.003, removing redundant rules with check for minBias							
	Removing redundant rules **WRAcc quality measure**				**Removing redundant rules** **Rule Length quality measure**			
Filtering type	**No filtering**		**Chi-squared filter**		**No filtering**		**Chi-squared filter**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**Average**	**stdev**
Coverage$_{L}$	19.26	0.27	19.64	0.34	18.69	0.35	18.74	0.35
Coverage$_{H}$	15.55	0.45	14.88	0.58	16.95	0.84	16.35	0.68
CTR$_{L}$	0.10	0.03	0.10	0.04	0.10	0.01	0.10	0.01
CTR$_{H}$	0.81	0.09	0.85	0.11	0.76	0.04	0.78	0.04
N$_{Rules}$	37.4	2.22	36.43	2.30	41.46	3.41	38.30	3.11
Length$_{ave}$	4.17	0.18	3.42	0.10	3.16	0.11	3.16	0.18
Length$_{w,ave}$	2.34	0.11	1.92	0.05	1.94	0.07	1.89	0.09
Coverage$_{Rule}$	1.15	0.02	1.15	0.03	1.33	0.04	1.26	0.03

(b)

	Dataset: December, Depth=7, minBias=0.003, Removing redundant rules with Bias quality measure							
	with check for minBias				**without check for minBias**			
Filtering type	**No filtering**		**Chi-squared filter**		**No filtering**		**Chi-squared filter**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**Average**	**stdev**
Coverage$_{L}$	17.29	0.53	17.20	0.64	19.49	0.29	19.82	0.46
Coverage$_{H}$	7.77	0.59	6.57	0.55	14.19	0.63	14.59	0.81
CTR$_{L}$	0.09	0.02	0.09	0.02	0.10	0.01	0.10	0.01
CTR$_{H}$	1.02	0.05	1.11	0.07	0.83	0.04	0.83	0.05
N$_{Rules}$	110.23	4.27	41.55	3.62	152.87	3.72	69.00	4.00
Length$_{ave}$	4.72	0.09	3.61	0.13	4.72	0.08	3.51	0.10
Length$_{w,ave}$	4.2	0.16	2.68	0.18	4.18	0.15	2.51	0.11
Coverage$_{Rule}$	1.83	0.11	1.38	0.12	2.47	0.17	2.03	0.13

(c)

Figure 20 Effect of $\chi 2$ filter for on-fly redundancy removal

In the above experiments, in case of not using post-filtering (Figure 20(a)), both redundancy and number of rules is reduced in tens of times, and average rule length is significantly smaller (approximately by two units). In case using post-filtering, gain in redundancy and number of rules is not is not so large. In case of WRAcc or Rule Length quality measures for removing redundancy (Figure 20(b)), differences of all the parameters are insignificant; in case of Bias quality measure (Figure 20(c)), number of rules decreases by more than two times, redundancy and the average rule length are significantly smaller, while model performance varies within 95% confidence interval. It is important to notice that even though there is no apparent increase in performance and rule quality in the case of using WRAcc or Rule Length quality measures, the advantage of on-fly redundancy removal of χ^2 filter is still intact. For example, in case of minBias=0.003 search phase of the algorithm will generate on average 16077.87 rules, which will be used as the input for the post-processing, redundancy removing phase (Figure 20(a)). If χ2 filter is employed, only 162.92 rules will be generated during the search phase.

Claim 2 (effect of redundancy removal during post-processing step) Using different quality measure during the second phase of subgroup discovery leads to significantly different results with coverage, while difference in CTRs preserves above the minimal Bias threshold. Figure 20(b,c) demonstrates this effect, as it shows performance results of the algorithm employing different quality measures for redundancy removing phase, namely WRacc, Rule Length, and Bias, while all other parameters remain unchanged. Some compromise on model performance has to be made. In fact, CTR_{high} demonstrates the biggest sensitivity to settings and quality measure. However, change in rule quality is significant.

5.3. Contextual Evaluation of Attributes

Notions of primary and contextual attributes used in the current work were introduced in Section 3.1.5. In this section, we evaluate attributes of Kliknieuws.nl dataset according these notions. As presented in Table 5, the dataset consists of 11 attributes. It should be also noticed that some attributes have different values in summer and winter datasets: temperature, conditions, browser and OS.

We apply χ^2 test of independence to each attribute as shown in Figure 21. All the tests with the resulting p-values are given in Appendix 7.

Winter dataset

device	Click	No Click	CTR,%
desktop	25451	6167235	0.4110
mobile	1660	286373	0.5763
tablet	3284	292333	1.1109
tv	1	847	0.1179
unknown	10	759	1.3004
Total	30406	6747547	0.4486

p-value < 2.2e-16

Summer dataset

device	Click	No Click	CTR,%
desktop	29594	5966195	0.4936
tablet	7162	633416	1.1181
mobile	2965	376695	0.7810
tv	4	2199	0.1816
unknown	2	2535	0.0788
Total	39727	6981040	0.5658

p-value < 2.2e-16

Figure 21 χ^2 test of device attribute

Summary of χ^2 tests of the attributes is given in Table 6. The test were conducted in R [57], and in the Apache Commons Mathematics Library [58]. in both software packages tests yielded identical results.

Table 6 Summary of χ^2 tests of the attributes

	P-values	
Attribute	**Winter dataset**	**Summer dataset**
Photo	< 2.2e-16	< 2.2e-16
Video	0.1822	2.975e-16
Forum	< 2.2e-16	< 2.2e-16
Page category	< 2.2e-16	< 2.2e-16
Device	< 2.2e-16	< 2.2e-16
OS	< 2.2e-16	< 2.2e-16
Browser	< 2.2e-16	< 2.2e-16
Day of the week	< 2.2e-16	< 2.2e-16
Time of the day	0.04581	0.001723
Temperature	< 2.2e-16	< 2.2e-16
Conditions	5.697e-13	9.420e-10

For most of attributes χ^2 test return p-value < 2.2e-16 (this is the lowest non-zero positive number that R can return), meaning that the corresponding Null Hypotheses, that attributes and the target class variable are not correlated have to be rejected, and indeed, there is a correlation between the attributes and the target visitor's clicking behavor. So, according to our notion, attributes "Photo", "Forum", "Page category", "Device", "OS", "Browser", "Day of the week", "Temperature" are primary. Attributes "Video" and "Time of the day" in Wintes dataset may not be primary if significance level alpha is set lower than 4.5%, wich corresponds to 95.5% confidence for rejecting Null-Hypothesis. It is also noticeable that attributes “Video”, “Time of the day” and “Conditions” have p-values greater than zero for both Winter and summer datasets. It may demonstrate that in bigger datasets, covering longer time spans, these attributes might not be primary with bigger significance level alpha.

Let’s look in details into the attributes, which are not primary according to our definition. Figure 22 shows “video” attribute within the entire dataset and under fixed primary attribute “photo”=0. The rule $“video” = 1\ \wedge "photo" = 0 \rightarrow 1$ was found by the subgroup discovery algorithm.

Winter dataset, prior distribution

Video	Click	No Click	CTR,%
0	30359	6734756	0.4488
1	47	12791	0.3661
Total	**30406**	**6747547**	**0.4486**

p-value = 0.1822

Winter dataset, photo=0

photo	Video	Click	No Click	CTR,%
0	**0**	21237	4962334	0.4261
0	**1**	23	2575	0.8853
Total		**15037**	**2978493**	**0.5023**

p-value = 0.0005812

Figure 22 Comarison of χ2 test of “video” attribute within the entire dataset and under fixed primary attribute

Similarly, Figure 23 shows “Hour of the day” attribute within the entire dataset and under fixed primary attribute “Day of the week”=weekend.

Winter dataset, prior distribution

Hours	Click	No Click	CTR,%
evening	10289	2332919	0.4391
lunch	2101	473783	0.4415
morning	1739	373346	0.4636
night	1194	262824	0.4522
work_hours	15083	3304675	0.4543
Total	**30406**	**6747547**	**0.4486**

p-value = 0.04581

Winter dataset, “Day of the week”=weekend

Day	Hours	Click	No Click	CTR,%
wkend	**evening**	2957	697658	0.4221
wkend	**lunch**	643	119269	0.5362
wkend	**morning**	394	66181	0.5918
wkend	**night**	396	85335	0.4619
wkend	**work_hours**	4734	910724	0.5171
Total		**9124**	**1879167**	**0.4832**

p-value < 2.2e-16

Figure 23 Comparison of χ2 test of “time of the day” attribute within the entire dataset and under fixed primary attribute

In both cases p-values became significantly smaller, meaning that the attributes become primary in the certain context: attribute “Video” becomes primary in context “Photo”=0, whereas attribute “Time of the day” becomes primary in context “Day of the week”=weekend. Hence, according to the definition, adopted in this work, these attributes are contextual. Remarkably, according to the definition of Widmer[36], in this situation attributes “Photo” and “Day of the week” are contextual since the create context, in which “Video” and “Time of the day” become primary.

Since subgroup discovery algorithms combine predictive and descriptive logic, it is possible to get some interesting inside about website visitors’ clicking behavior. For example, Figure 21 demonstrates dependency between CTR and device type of the visitor. It is seen that CTR and CTR and the device type are strongly correlated: Visitors using tablets generate CTR more than twice higher than visitor from desktops, and visitors from mobile devices have CTR 50% larger than visitor from desktops. This is true for both Winter and Summer datasets It can be explained that tablets and mobile devices are used for leisure activities or during the leisure time, whereas desktops are used mostly for work, and people are not so eager to devote their time to exploring advertisements. Difference between mobiles and tablets can be explained lower speed of the Internet connection, high cost of traffic, more complicated interaction with the web-page, which causes people to make reasonable decision before clicking on a banner.

Another noticeable observation is that “sport” page category has extremely low CTR, more than three times lower than the average (see Figure 36 in Appendix 7). This may be caused by the fact that visitors interested in the sport news, are focused on content of the news itself, and skip the content, irrelevant to it. The opposite is true about “advertisement” (“koopjes” in Dutch) page category, where CTR is four times larger than the average, approximately 2%. Here, the visitors are specifically looking for the advertisements, and perceive the banners, as advertisement too.

An interesting observation may be done from the atmospheric conditions, which is an external attribute for the kliknieuws.nl system (see Figure 35 in Appendix 7). During the raining in summer time CTR is slightly higher. In winter time, the stronger are precipitations, the higher is CTR. Possibly, precipitations cause people to postpone “going out” and spend some time on leisure activities.

It is also important to note that some attributes are correlated such as temperature and time of the day, page category and presence of photos, videos, or forums. To make an informed decision about significance of usefulness of attributes, expert’s judgment and insight into system may be needed.

6. Conclusion

During the current work the following results have been achieved.

First, a framework for studying contextual features has been developed. It represents a seamless line of data processing with website logs as input and a model as output. Integration of contextual information is done according to basic Turneys [35] scheme. However, the framework allows for more advanced approaches to incorporating the context. For example, some logic can be applied during the data aggregation step.

Giving definition of context suitable for web analytics was a challenge: diverse literature sources and research domains do not agree on with each other, whereas many researchers attempt to adopt own definitions tailored to the tasks they are working on. Albeit hundreds of definitions already in use, based on literature research, we systematized them, proposing a division into a “technical view” and a “business view”. This allows us to bring to different views on contextual features to a common denominator. This division provides flexibility as with defining contextual features for business owners and management, as analyzing and evaluating them with data mining tools. Since feature extraction is one of the tasks of the data mining, the approach assumes that contextual features reflecting the business view are chosen first, after which, they are categorized on primary, contextual and irrelevant according to the technical view. Then, all non-primary features can be eliminated.

While developing own subgroup discovery algorithm, two features were introduced: using compressed data representation and using chi^2-goodness of fit test for checking subgroup significance. Compressed data representation allows for increasing performance without compromise on subgroup quality. As test results show, using chi^2-goodness of fit test can replace traditional removing redundancy methods:

performance results are virtually indistinguishable, whereas quality of rules, which is measured in number of rules in the model, average length of the rules, and rules' redundancy, often increases significantly. Besides, χ^2 filter, removing redundant rules on fly, during the search, has lower computational complexity than traditional techniques and reduces requirements to memory since redundant rules do not need to be stored until the second, redundancy removing phase.

The subgroup discovery algorithm that was implemented in this work consists of two phases, whereas each phase can use its own quality measure. In our setup Bias is used as a primary quality measure employed in the search phase, while another quality measure can be used for ranking subgroup during post-processing step. It allows for having more flexibility while producing rules: for example, it becomes possible to combine requirements of having both high Bias of the rules, and rule with large coverage, or low number of rules in the model.

Finally, evaluation of attributes presented in the dataset form contextual framework prospective was done. It was discovered that most of attributes are primary. "Video" and "Time of the day" are not primary but contextual considering winter dataset. We showed examples of contexts when these attributes become primary. It is noticeable, that p-values are of the same attributes are distinct from zero for both Winter and Summer datasets, which may imply that for larger datasets covering longer time span, averaged data can hint that these attributes are, in fact, contextual for bigger significance level alpha. We also attempted to explain some phenomena related to correlation of CTR with some attributes such as device type, page category, or atmospheric conditions.

7. Limitations and Future Work

In the current study, data for short time intervals were analyzed. The dataset consisted of data for November, December 2011, January 2012, and from June, 13 to August, 22, 2012. Consequently seasonal and yearly effects could hardly be tackled. Analyzing data for longer time spans will allow incorporating more contextual features for the analysis. In the current work the only external contextual information that was added is the weather. More features relevant to the events, news, macroeconomic indicators can be added in the future. Attaching offline activities of advertisers will also give some inside about visitors' clicking behavior.

Then, the current implementation of the system is rather research framework. It would make sense to implement it as an adaptive system, pluggable into target web-applications as a commercial software module. Such system must also be capable of monitoring own performance and detecting changes automatically.

Under larger number of attributes and larger datasets the algorithm can start experiencing performance problems. Thus, suitable heuristics, compromising as little quality of the model as possible, need to be developed. Performance improvement techniques developed in this work do not trade-off model quality; hence, gain in performance is rather limited.

Concerning the case study, the current analysis did not tackle statistics of individual banners, or zones where banners are placed. Measurements were done by analyzing probability of click on any banner during the page view. It makes sense to consider separate statistics by banners, by zone, even by user since different users have different habits with regards to banner clicking behavior.

Finally, live A/B testing on kliknieuws.nl servers has to be done to ensure viability of the approach and the generated models.

Literature

[1] N. Webster. Webster's new twentieth century dictionary of the English language. Springfield, MA: Merriam-Webster, Inc., 1980.

[2] M. Bazire and P. Brezillon. Understanding context before using it. In A. Dey and et al., editors, Proceedings of the 5th International Conference on Modeling and Using Context. Springer-Verlag, 2005.

[3] M. J. Berry and G. Linoff. Data mining techniques: for marketing, sales, and customer support. John Wiley & Sons, Inc. New York, NY, USA, 1997.

[4] C. Palmisano, A. Tuzhilin, and M. Gorgoglione. Using context to improve predictive modeling of customers in personalization applications. IEEE Transactions on Knowledge and Data Engineering, 20(11):1535–1549, 2008.

[5] K. Oku, S. Nakajima, J. Miyazaki, and S. Uemura. Context-aware SVM for context-dependent information recommendation. In Proceedings of the 7th International Conference on Mobile Data Management, page 109, 2006.

[6] B. N. Schilit and M. M. Theimer. Disseminating active map information to mobile hosts. IEEE network, 8(5):22–32, 1994.

[7] P. J. Brown, J. D. Bovey, and X. Chen. Context-aware applications: from the laboratory to the marketplace. IEEE Personal Communications, 4:58–64, 1997.

[8] D. Smith, L. Ma, and N. Ryan. Acoustic environment as an indicator of social and physical context. Personal and Ubiquitous Computing, 10:241-254, March 2006.

[9] N. Ryan, J. Pascoe, and D. Morse. Enhanced Reality Fieldwork: the Context-Aware Archaeological Assistant. Gaffney, V., van Leusen, M., Exxon, S.(eds.) Computer Applications in Archaeology. British Archaeological Reports, Oxford, 1997.

[10] A. K. Dey, G. D. Abowd, and D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction, 16(2):97–166, 2001.

[11] D. Franklin and J. Flachsbart. All gadget and no representation makes jack a dull environment. In Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments, pages 155 – 160. AAAI Press, 1998.

[12] T. Rodden, K. Cheverst, K. Davies, and A. Dix. Exploiting context in hci design for mobile systems. In Workshop on Human Computer Interaction with Mobile Devices, pages 21–22, 1998.

[13] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. IEEE Personal Communications, 4(5):42–47, 1997.

[14] F. Ricci and Q. N. Nguyen. Mobyrek: A conversational recommender system for on-the-move travelers. Destination Recommendation Systems: Behavioural Foundations and Applications, pages 281–294, 2006.

[15] W. Woerndl, C. Schueller, and R. Wojtech. A hybrid recommender system for context-aware recommendations of mobile applications. In Proceedings of the 3rd International Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces, pages 871–878, 2007.

[16] J. H. Schiller and A. Voisard. Location-based services. Morgan Kaufmann, 2004.

[17] B. Brown, M. Chalmers, M. Bell, M. Hall, I. MacColl, and P. Rudman. Sharing the square: collaborative leisure in the city streets. In H. Gellersen, K. Schmidt, M. Beaudouin-Lafon, and W. E. Mackay, editors, Proceedings of the ninth conference on European Conference on Computer Supported Cooperative Work, pages 427–447. Springer, 2005.

[18] G. L. Lilien, P. Kotler, and K. S. Moorthy. Marketing models. Prentice Hall, 1992.

[19] C. K. Prahalad. Beyond CRM: CK Prahalad predicts customer context is the next big thing. American Management Association MwWorld, 2004.

[20] Schilit, B., Adams, N., & Want, R. Context-aware computing applications. Proceedings of the 1st International Workshop on Mobile Computing Systems and Applications. Los Alamitos, CA: IEEE.

[21] P. Dourish. What we talk about when we talk about context. Personal and ubiquitous computing, 8(1):19–30, 2004.

[22] N. Archak, V. S. Mirrokni, and S. Muthukrishnan. Mining advertiser-specific user behavior using adfactors. In Proceedings of the 19th international conference on World wide web (WWW '10). ACM, New York, NY, USA, 31-40, 2010

[23] G. Chen and D. Kotz. A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381

[24] Merriam-Webster’s Collegiate Dictionary. http://www.merriam-webster.com/

[25] G. Adomavicius and A. Tuzhilin. Context-Aware Recommender Systems. Handbook on Recommender System, Chapter 7, Springer, 2011.

[26] M. Kuniavsky, Observing the user experience: a practitioner’s guide to user research, Morgan Kaufmann, 2003

[27] Web Analytics Association, The official WAA definition of Web analytics, 2012, http://www.digitalanalyticsassociation.org/?page=aboutus

[28] J. Burby, A. Brown and WAA Standards Committee. Web Analytics Definitions – Version 4.0, http://www.digitalanalyticsassociation.org/resource/resmgr/PDF_standards/WebAnalyticsDefinitionsVol1.pdf

[29] B. Clifton. Advanced Web Metrics: Understanding Web Analytics Accuracy. Whitepaper. Version 2.0, 2010

[30] A. Kaushik. Web Analytics: An Hour a Day. Sybex,. 2007

[31] B. Clifton. Advanced Web Metrics with Google Analytics, 2nd Edition. Wiley Publishing, 2010

[32] M. Abraham, C. Meierhoefer and A. Lipsman. The Impact of Cookie Deletion on the Accuracy of Site-Server and Ad-Server Metrics: An Empirical ComScore Study. Whitepaper, 2007 http://www.comscore.com/Press_Events/Presentations_Whitepapers/2007/Cookie_Deletion_Whitepaper

[33] IP Addresses of Search Engine Spiders. http://www.iplists.com/

[34] Categorizr device detection script. https://github.com/bjankord/Categorizr/blob/master/categorizr-redirect.php

[35] P. Turney. The management of context-sensitive features: A review of strategies. In Proc. of the ICML-96 Workshop on Learning in Context-Sensitive Domains, pages 60-65, 1996

[36] G. Widmer. Tracking context changes through meta-learning. Machine Learning, 27(3):259–286, 1997

[37] D.J. Hand, H. Mannila, P. Smyth. Principles of Data Mining. Adaptive Computation and Machine Learning Bradford Books. MIT Press, 2001

[38] M. Levene, G. Loizou. Why is the snowflake schema a good data warehouse design? Information Systems 28 (3), pages 225-240, 2003

[39] OLAP and OLAP Server Definitions. The OLAP Council, 1995 http://www.olapcouncil.org/research/glossaryly.htm

[40] G. Widmer, M. Kubat. Learning in the presence of concept drift and hidden contexts. Machine Learning, 23(1), 69–101, 1996

[41] S. Rose, N. Hair, M. Clark. Online customer experience: a review of the business-to-consumer online purchase context. International Journal of Management Review, 13, pp. 24–39, 2011

[42] M. Moynagh, R. Worsley. Tomorrow's consumer – the shifting balance of power. Journal of Consumer Behaviour, 1, pp. 293–301, 2002

[43] F. Herrera, C.J. Carmona, P. González, M.J. del Jesus, An overview on Subgroup Discovery: Foundations and Applications. Knowledge and Information Systems, vol. In press, 2011

[44] A. Ansari, S. Essegaier, R. Kohli. Internet Recommendation Systems, Journal of Marketing Research, 37(3), 363–375, 2000

[45] W. M. P. van der Aalst. Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer-Verlag, 2011

[46] K. Ploesser, M. Peleg, P. Soffer, M. Rosemann, J. Recker. Learning from Context to Improve Business Processes. BPTrends, January 2009

[47] W.M.P. van der Aalst, S. Dustdar. Process Mining Put into Context, Internet Computing, IEEE , vol.16, no.1, pp.82-86, Jan.-Feb. 2012

[48] W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers, 8(1):21–66, 1998

[49] P.K. Novak, N. Lavrac, G.I. Webb. Supervised Descriptive Rule Discovery: A Unifying Survey of Contrast Set, Emerging Pattern and Subgroup Mining. Encyclopedia of Machine Learning 2010: 938-941

[50] M. van Leeuwen, A.J. Knobbe. Non-redundant Subgroup Discovery in Large and Complex Data. ECML/PKDD 3, Vol. 6913, Springer, 2011

[51] N. Lavrac, B. Kavsek, P. Flach, L. Todorovski. Subgroup discovery with CN2-SD. The Journal of Machine Learning Research 5, 153–188, 2004

[52] R. Kohavi, R. Longbotham, D. Sommerfield, R.M. Henne, Controlled experiments on the web: survey and practical guide. Data Min. Knowl. Discov. 18(1), 140–181, 2009

[53] S. Morishita and J. Sese. Traversing Itemset Lattices with Statistical Metric Pruning. In Symposium on Principles of Database Systems, pages 226–236, 2000

[54] Rapid - I – RapidMiner. http://www.rapidminer.com/

[55] A. W. Susilo. Framework for Studying Context-aware Banner Selection in Topical Online Portal Advertising. Masters Thesis, Eindhoven University of Technology, Eindhoven, August 2011

[56] Introducing JSON, http://www.json.org/

[57] The R Project for Statistical Computing, http://www.r-project.org/

[58] The Apache Commons Mathematics Library, http://commons.apache.org/math/

Appendix 1 Glossary

Common terms used in the Diploma with their definitions are listed in Table 7. The definitions are given according to the current version of Web Analytics Association's glossary [28].

Shortly after its foundation, Web Analytics Association Standards committee embarked on an effort to define what was agreed upon as the three most important metrics – *Unique Visitors*, *Visits/Sessions* and *Page Views*. The Standards committee determined that these three metrics make up the foundation for most web analytics definitions. In addition, since many other metrics rely on an understanding of these three, the decision was made to focus on these. In total, it was established that terms *Page, Page Views, Visits/Sessions*, and *Unique Visitors* would constitute building block terms; terms *Entry Page*, *Landing Page*, *Exit Page*, *Visit Duration*, *Referrer*, *Click-through*, *Click-through Rate*, and *Page Views per Visit* would provide visit characterization; terms *Page Exit Ratio*, *Single Page Visits*, *Single Page View Visits (Bounces)*, and *Bounce Rate* would provide content characterization; and terms *Event* and *Conversion* would define conversion metrics.

All measures and metrics assume that they relate to an action by a human visitor. The types of non-human "visitors" include robots, spiders and, website crawlers that periodically scan or methodically download content from a website. Programs typically identify themselves via the user agent field of HTTP request that allows the website to provide a different version of the content. However, there are many that do not identify themselves and can be confused with human traffic. The decision of identification of such traffic is left to the discretion of web analytic application.

There are three main types of Web analytics metrics – counts, ratios, and KPIs. A fourth type of definition, sometime referred as dimension in general, is included for terms that describe concepts instead of numbers. Count is the most basic unit of measure, a single number. Ratio is typically a count divided by a count, although a ratio can use either a count or a ratio in the numerator or denominator. The name of such metric normally contains word "per", for example, *Page Views per Visit*. KPI (Key Performance Indicator) is frequently a ratio which is infused with business strategy. Therefore the set of appropriate KPIs naturally varies across different E-commerce applications. The forth type, dimension, refers to a general source of data that can be used to define various types of segments or counts and represents a fundamental dimension of visitor behavior or site dynamics. Some examples are *event* and *referrer*. They can be interpreted the same as counts above, but typically they must be further qualified or segmented to be of actual interest. Therefore these define a more general class of metrics and represent a dimension of data that can be associated with each individual visitor.

A metric can apply to three different universes: aggregated, segmented, individual. Aggregated implies the total site traffic for a defined period of time. Segmented denotes a subset of the site traffic for a defined period of time, filtered in some way to gain greater analytical insight, for example, by *campaign*, *banner*, *affiliate*, by visitor type (*new* vs. *returning*, *repeat buyers*, *high value*), or by *referrer*. Individual refers to activity of a single Web visitor for a defined period of time.

Table 7 Glossary

Term	Definition
Page	Analyst definable unit of content.
Page Views	The number of times a page was viewed.
Visits/Sessions	An interaction, by an individual, with a website consisting of one or more requests for an analyst-definable unit of content (i.e. "page view"). If an individual has not taken another action (typically additional page views) on the site within a specified time period, the visit session will terminate.
Unique Visitors	The number of inferred individual people (filtered for spiders and robots), within a designated reporting timeframe, with activity consisting of one or more visits to a site. Each individual is counted only once in the unique visitor measure for the reporting period.
New Visitor	The number of Unique Visitors with activity including a first-ever Visit to a site during a reporting period.
Repeat Visitor	The number of Unique Visitors with activity consisting of two or more Visits to a site during a reporting period.
Return Visitor	The number of Unique Visitors with activity consisting of a Visit to a site during a reporting period and where the Unique Visitor also Visited the site prior to the reporting period.
Entry Page	The first page of a visit.
Landing Page	A page intended to identify the beginning of the user experience resulting from a defined marketing effort.
Exit Page	The last page on a site accessed during a visit, signifying the end of a visit/session.
Visit Duration	The length of time in a session. Calculation is typically the timestamp of the last activity in the session minus the timestamp of the first activity of the session.
Referrer	The referrer is the page URL that originally generated the request for the current page view or object.
Internal Referrer	The internal referrer is a page URL that is internal to the website or a web-property within the website as defined by the user.
External Referrer	The external referrer is a page URL where the traffic is external or outside of the website or a web-property defined by the user.
Search Referrer	The search referrer is an internal or external referrer for which the URL has been generated by a search function.

Visit Referrer The visit referrer is the first referrer in a session, whether internal, external or null.

Original Referrer The original referrer is the first referrer in a visitor's first session, whether internal, external or null.

Click-through Number of times a link was clicked by a visitor.

Click-through Rate The number of click-throughs for a specific link divided by the number of times that link was viewed.

Page Views per Visit The number of page views in a reporting period divided by number of visits in the same reporting period.

Page Exit Ratio Number of exits from a page divided by total number of page views of that page.

Single Page Visits Visits that consist of one page regardless of the number of times the page was viewed.

Single Page View Visits (Bounces) Visits that consist of one page-view.

Bounce Rate Single page view visits divided by entry pages.

Event Any logged or recorded action that has a specific date and time assigned to it by either the browser or server.

Conversion A visitor completing a target action.

Appendix 2 Metrics Overview

Table 8 summarizes common metrics used in web analytics. Table 9 brings categorization dimensions with levels of granularity.

Table 8 Common metric grouped by categories

Metric group	Metric	Description
Content	**Entrances per Page views**	The percentage of page views which were entrances to the site.
	Entrances	The number of times visitors entered the site through a specified page or set of pages.
	Avg. Value	The average value of each event.
	Event Value	The total value of an event or set of events, calculated by multiplying the per-event value by the number of times the event occurred.
	Search Depth	The number of pages visited after the search and before the next one or end of session.
	Time after Search	The time spent on the website from the start of the current search until session ended or another search started.
	Search Exits	The percentage of searches that resulted in an immediate exit from the website.
	Search Refinements	The percentage of searches that resulted in another search (i.e. a new search using a different term).
	Visits with Search	The number of visits during which at least one site search occurred.
	Time on Page	
	Total Events	Total Events is the number of times events occurred.
	Unique Events	The number of visits during which one or more events occurred.
Visitors	**Avg. Time on Page**	The average amount of time visitors spent viewing a specified page or set of pages.
	Pages per Visit	The average number of pages viewed during a visit to the website. Repeated views of a single page are counted.

	Avg. Visit Duration	The average time duration of a session.
	Bounce Rate	The percentage of single-page visits (i.e. visits in which the person left the website from the entrance page).
	Bounces	The number of single-page visits.
	% Exit	The percentage of site exits that occurred from a specified page or set of pages.
	Exits	The number of times visitors exited your site from a specified page or set of pages.
	New Visits	The number of first-time visits (from visitor, never entered the website before).
	Page views	The total number of pages viewed. Repeated views of a single page are counted.
	Results Page views per Search	The average number of times visitors viewed a search results page after performing a search.
	% New Visits	The percentage of visits that were first-time visits (from visitor, never entered the website before).
	% Search Exits	The percentage of searches that resulted in an immediate exit from the website.
	% Search Refinements	The percentage of searches that resulted in another search (i.e. a new search using a different term).
	Total Unique Searches	The number of times people searched the website. Duplicate searches within a single visit are excluded.
	Visit Duration	The average time duration of a session.
	Unique Visitors	The number of unduplicated (counted only once) visitors to your website over the course of a specified time period.
	Unique Page views	The number of visits during which the specified page or pages are viewed at least once.
	Visits	The number of visits to your site. The terms visit and session have the same meaning and are used interchangeably.
Traffic Sources	**Organic Searches**	The number of organic searches that occurred within

		a session.
Conversions	**Goal Starts**	The total number of starts for all goals.
	Goal Value	The total value produced by goal conversions on the website. This value is calculated by multiplying the number of goal conversions by the value assigned to each goal.
	Per Visit Goal Value	The average value (based on goal value) of a visit to the website. Calculated as Total Goal Value divided by Visits.
	Quantity	The number of units sold in ecommerce transactions.
	Product Revenue	The total revenue from product sales. Excludes tax and shipping.
	Average Price	The average ecommerce revenue per product.
	Average Quantity	The average quantity of the product (or group of products) sold per transaction.
	Unique Purchases	The total number of times a specified product (or set of products) was a part of a transaction.
Advertising	**AdSense Ad Units Viewed**	The total number of AdSense ads that were viewed by visitors of the website.
	AdSense Ads Clicked	The number of times AdSense ads on the website were clicked.
	AdSense CTR	The percentage of page impressions that resulted in a click on an ad.
	AdSense eCPM	The estimated cost per thousand page impressions. It is the AdSense Revenue per 1000 page impressions.
	AdSense Page Impressions	The number of page views during which an ad was displayed.
	AdSense Revenue	The revenue from AdSense ads.
	AdSense Ads Viewed	AdSense Ads Viewed.
Social	**Social Actions**	The number of social actions that occurred.
	Actions Per Social Visit	Total Social Actions divided by Unique Social Actions.

	Unique Social Actions	The number of visits during which the specified social actions occurred at least once.
Other	**Abandoned Funnels**	The number of times visitors entered a goal funnel without converting.

Table 9 Common dimensions with the levels of granularity

Dimension group	Dimension	Description and levels of granularity
Visitor’s hardware	**Screen resolution**	The screen resolutions of visitor's monitor
	Screen colors	The screen color depths of visitor' monitor
	Mobile	Indicates whether visits were from mobile devices (Yes) or not (No).
	Mobile device branding	Manufacturer or Branded name (examples: Samsung, HTC, Verizon, T-Mobile).
	Mobile device info	The Branding, model, and marketing name used to identify the device (e.g., Acer A501 Picasso, Samsung GT-I9001)
	Mobile input selector	Selector used on device (examples: touchscreen, joystick, clickwheel, stylus)
Visitor’s software	**Browser**	Name of the browser used by the visitor (e.g. Chrome, Firefox, Opera)
	Version of the browser	The browser version used by the visitor
	Operating system platform	The operating systems of the visitor including mobile (e.g., Windows, Linux, Android, Macintosh, Playstation 3)
	Operating system version	The operating system version (e.g. XP, 2000, NT, 7); intended to be used as secondary dimension together with Operating system platform
	Language	The language preference settings of visitor’s browser (e.g., en-us, de-de)
	Java support	Browser with and without (Yes or No) Java enabled

	Flash version support	The versions of Flash supported by visitor's browser, including minor versions (e.g., 11.2 r202)
Visitor's location	**IP address**	IP address of the visitor
		The ISP organization registered to the IP address of the user
	Continent	Geographic continent location obtained by information registered with the IP address (e.g., Europe, Asia)
	Sub continent region	Geographic region or state location, obtained by information registered with the IP address (e.g., Southern Europe, Northern America)
	Country (territory)	The country (territory) from which visit originated, based on IP address (e.g., Netherlands, Germany)
	City	The city from which visit originated, based on IP address (e.g., Eindhoven, Utrecht)
Visitor's time	**Date**	The dates of the active date range
	Day of week	Weekday ranging from 0 to 6
	Hour	Hour of the day ranging from 1 to 24
	Hour of day	Combination of a calendar day with hour of the day, (e.g. 2012012616, 2012011420)
	Month of year	Combination of a year with month number (e.g., 201201, 201202)
	Week of year	Combination of a year with week number (e.g. 201201, 201215)
Content	**Landing page**	The page through which visitor entered the website
	Exit page	The page visitor viewed last on the website
	Hostname	The full domain name of the page requested
	Page	Relative URL (the piece of the URL after the

		hostname)
Traffic Sources	**Full URL of the referral page**	The URLs that referred traffic
	Traffic type	The type of traffic to the website: search, referral, direct, and other.
	Medium	The medium used to generate the request (e.g., organic search, referral, paid search, advertisement)
	Source	The sources which referred traffic (e.g., google, bing, direct)
	Keywords	The keywords, both paid and unpaid, used by a user to reach your site
Visitor behavior	**Visitor type**	Either new visitor or returning visitor
	Time on Page	Amount of time visitors spent viewing a specified page or set of pages
	Visit Duration	Time duration of a session
	Page depth	The number of pages viewed by visitors in a session
	Days since last visit	The number of days elapsed since visitors last visited the site

Appendix 3 Database Structure

A starting point for analysis is Kliknieuws's logs with recorded information about impressions and clicks. Impressions data contain 10 to 15 million records per month leading to slowing execution even simple "select" SQL queries, time of which can take a few hours. Fields containing URL addresses and user agents cover the most valuable information about the visitor, namely web page address, title and category, and browser, operating system and device type. However, parsing these fields on-fly is computationally expensive. Our goal was to propose optimal data structure of the database which will improve proficiency without significant loss of information. For that, we allocated to tables some additional columns with which respect to which we attempting to analyze visitors' behavior. The main working tables containing information about impressions and clicks are the following:

- ctr_sep_2011
- ctr_oct_2011
- ctr_nov_2011
- ctr_dec_2011
- ctr_jan_2012

Structure of these tables is identical and is represented in Table 10.

Table 10 CTR table structure

Column	Index (Primary key)	Type	Content and role
id	**PK**	INT	A unique key of the entry stored for reference to original unfiltered dataset in case of need for additional information
os	✓	SET	A predefined set of 35 possible types of visitor operating system
ua		VARCHAR(200)	User agent string representing source information for detecting a type of visitor's browser, operating system and device; stored for only reference
ip_address	✓	INT	Four-byte integer representing visitor's IP address
banner_id	✓	SMALLINT	ID of the banner
unix_timestamp	✓	BIGINT	Timestamp as the number of milliseconds elapsed from 1-1-1970, representing source information for calculating day and time of the visit; together with cookie used for grouping multiple impressions into one page view

cookie	✓	BIGINT	Unique identifier of the visitor; used for tracing visitor activities and together with Unix timestamp for grouping multiple impressions into one page view
viewer_id		VARCHAR(32)	Unique alternative identifier of the visitor; stored for reference only
screen_width	✓	SMALLINT	Visitor screen's width
screen_height	✓	SMALLINT	Visitor screen's height
color_depth	✓	TINYINT	Visitor screen's color depth
referrer		VARCHAR(1000)	URL address of page that originally generated the request for the current page view
url		VARCHAR(400)	URL address of page
referrer_domain		VARCHAR(200)	Domain name of the referrer
campaingn_ids	✓	MEDIUMINT	ID of campaign
zone_ids	✓	SMALLINT	ID of field of the page where banner is shown
cb_num	✓ **unique, composite with ip_address**	BIGINT	Identifier of banner within pageview – a random number intended for referencing clicked banner with a table containing information about clicks on banners; due to not guaranteed uniqueness, used in combination with IP address column (an assumption is made that IP address doesn't change in a short time slot between load of the page and click on a banner)
date	✓	DATE	Date of the impression
time	✓	TIME	Time of the impression
page_category	✓	SET	Set of 24 most frequent categories (sub-catalogs of Kliknieuws's website), covering more than 98.5% impressions
page_code	✓	MEDIUMINT	Integer code of the internal page of website; used for analysis with regards to page categories and placement of photo, video or forum on the same page

device	✓	SET	Set of the following values • unknown • desktop • tablet • mobile • tv • bot The division is done by parsing user agent string of the browser
click_timestamp		DATETIME	Unix timestamp of the click if it occurred, or NULL otherwise

Index covering cb_num and ip_address is unique and composite, having cb_num as a first column and ip_address as second. The rationale behind this index is the following. Random number stored in cb_num column is an identifier of banner within pageview must be unique for every impression to be used for matching impressions with clicks stored in the click log. However, since randomness cannot guarantee uniqueness this numbers may repeat themselves across time span of analysis for different banners and user sessions. To retain entries with duplicated cb_num, we must exploit additional information from other columns of table with impression and click logs and such as banner_id, zone_id, or ip_address. Since the range of values of ip_address column is the widest, we use combination of cb_num and ip_address as a unique key for matching impression and click data.

The following fields are unique for each impression within one pageview:

- id
- banner_id
- campaingn_ids
- zone_ids
- cb_num
- click_timestamp

Other fields are common for all the impressions within the same pageview.

Furthermore, there are two reference tables, click_log_f containing information about clicks, and categories containing information about web page categories and placement of photo, video or forum. These tables are described in Table 11 and Table 12 respectively.

Table 11 Structure of click_log_f table

Column	Index (Primary key)	Type	Content and role
viewer_id		VARCHAR(32)	
date_time	✓	DATETIME	Unix timestamp of the click
banner_id		INT(10)	ID of the banner
zone_id		INT(10)	ID of field of the page where banner is shown
ip_address	✓	VARCHAR(16)	IP address of visitor, together with cb_num is used for matching clicks with impression data
cb_num	✓	VARCHAR(100)	Random number intended for referencing clicked banner with a table containing information about clicks on banners; due to not guaranteed uniqueness, used in combination with IP address column
dest		VARCHAR(255)	URL of the target page
referer		VARCHAR(255)	URL address of page that originally generated the request for the current page view
ua		VARCHAR(255)	User agent string representing source information for detecting a type of visitor's browser, operating system and device

Table 12 Structure of categories table

Column	Index (Primary key)	Type	Content and role
nid	**PK**	INT(11)	ID of the page; used for referencing with impression data
nslug	✓	TINYTEXT	Title of the page
nfoto		TINYINT(3)	Boolean flag (zero or one) indicating presence of photo on the page
nvideo		TINYINT(3)	Boolean flag (zero or one) indicating presence of video on the page
nforum	✓	TINYINT(3)	Boolean flag (zero or one) indicating presence of forum on the page

ccategorie	VARCHAR(45)	Category of the page

Appendix 4 Data Retrieval and Transformation

Since not all the fields were given in format currently presented in the table, some transformation logic was applied (see Table 13). Changing of data types was necessary since all columns in source table were of type VARCHAR. `CAPA`.`table_name` must be replaced with the real name of the table, for example, `CAPA`.`ctr_sep_2011`.

Table 13 Data transformation

Column	Transformation algorithm
os	ALTER IGNORE TABLE `CAPA`.`table_name` CHANGE COLUMN `os` `os` SET('Amiga OS','Android','CentOS','FreeBSD','Linux Debian','Linux Fedora','Linux Gentoo','Linux Mandriva','Linux Red Hat','Linux SUSE','Linux Unknown Version','Macintosh (iPhone)','Macintosh OS X','Media Center 2004','Media Center 2005','Nintendo Wii','NULL','OS/2','Playstation 3','SunOS','Ubuntu 10.04','Ubuntu 7.10','Ubuntu 8.04','Ubuntu 9.04','Windows 2000/NT 5','Windows 3.x','Windows 7','Windows 98','Windows CE','Windows ME','Windows NT','Windows NT 4','Windows Server 2003 and XP x64 Edition','Windows Vista','Windows XP') NULL DEFAULT NULL ;
ip_address	UPDATE `CAPA`.`table_name` SET ip_address = INET_ATON(ip_address) ; ALTER ignore TABLE `CAPA`.`table_name` CHANGE COLUMN `ip_address` `ip_address` INT UNSIGNED NULL DEFAULT NULL ;
unix_timestamp	ALTER ignore TABLE `CAPA`.`table_name` CHANGE COLUMN `unix_timestamp` `unix_timestamp` BIGINT UNSIGNED NULL DEFAULT NULL ;
cookie	UPDATE `CAPA`.`table_name` SET cookie = CONV (cookie, 16, 10) ; ALTER ignore TABLE `CAPA`.`table_name` CHANGE COLUMN `cookie` `cookie` BIGINT UNSIGNED NULL DEFAULT NULL ;
cb_num	UPDATE `CAPA`.`table_name` SET cb_num = CONV (cb_num, 16, 10) ; ALTER ignore TABLE `CAPA`.`table_name` CHANGE COLUMN `cb_num` `cb_num` BIGINT UNSIGNED NULL DEFAULT NULL ;
date	ALTER TABLE `CAPA`.`table_name` ADD COLUMN `date` DATE NULL DEFAULT NULL AFTER `cb_num` ; UPDATE `CAPA`.`table_name` SET date = DATE(FROM_UNIXTIME(unix_timestamp/1000)) ;
time	ALTER TABLE `CAPA`.`table_name` ADD COLUMN `time` TIME NULL DEFAULT NULL AFTER `date` ; UPDATE `CAPA`.`table_name` SET time = TIME(FROM_UNIXTIME(unix_timestamp/1000)) ;

page_category	ALTER TABLE `CAPA`.`table_name` ADD COLUMN `page_category` SET('_start_page', 'nieuws', 'sport', 'foto', 'selecteer-regios', 'forum', 'zoeken', 'pagina2', 'agenda', 'koopjes', 'pagina3', 'poll', 'miss-uden', 'pagina4', 'login', 'pagina5', 'epaper', 'prive-berichten', 'registreren', 'pagina6', 'colofon', 'mobiel', 'pagina7', 'fancy-image') NULL DEFAULT NULL AFTER `time` ; UPDATE `CAPA`.`table_name` SET page_category = page_category(url) ;
page_code	ALTER TABLE `CAPA`.`table_name` ADD COLUMN `page_code` MEDIUMINT NULL DEFAULT NULL AFTER `page_category` ; UPDATE `CAPA`.`table_name` SET page_code = page_code (url) ;
device	ALTER TABLE `CAPA`.`table_name` ADD COLUMN `device` SET('unknown', 'desktop', 'tablet', 'mobile', 'tv', 'bot') NULL DEFAULT NULL AFTER `page_code` ; UPDATE `CAPA`.`table_name` SET device = device_type (ua) ;
click_timestamp	ALTER TABLE `CAPA`.`table_name` ADD COLUMN `click_timestamp` DATETIME NULL DEFAULT NULL AFTER `device` ; UPDATE `CAPA`.`table_name` AS t1 INNER JOIN click_log_f AS t2 on (t1.cb_num = t2.cb_num and t1.ip_address = t2.ip_address) SET t1.`click_timestamp` = t2.date_time ;

In real setting aforementioned queries were combined together in one ‘ALTER TABLE’ or one ‘UPDATE’ statements. Listing of functions page_category (url), page_code (url) and device_type (ua) is given below.

Function page_category() extracts page subcategory from the URL passed.

```
USE `CAPA`;
DROP function IF EXISTS `page_category`;
DELIMITER $$
USE `CAPA`$$
CREATE DEFINER=`admin`@`%` FUNCTION `page_category`(url varchar(200)) RETURNS varchar(50)
CHARSET latin1
BEGIN
declare delim int default 0;
if url not regexp '^http://.*kliknieuws\.nl'
then return '_unknown';
end if;
set url:= SUBSTRING_INDEX(url,'kliknieuws.nl/',-1);
if trim(both '/' from url) = ''
```

```
then return '_start_page';
end if;
set delim := locate('/',url);
if (delim = 0) then set delim := locate('?',url); end if;
if (delim = 0) then return url; end if;
return substring(url,1,delim-1);
END
$$
DELIMITER ;
```

Function page_code() extracts page code from URL of the article or returns NULL for start and subcategory pages:

```
USE `CAPA`;
DROP function IF EXISTS `page_code`;
DELIMITER $$
USE `CAPA`$$
CREATE DEFINER=`admin`@`%` FUNCTION `page_code`(url varchar(400)) RETURNS int(8)
BEGIN
declare delim, len int default 0;
if url not regexp '^http://.*kliknieuws\.nl/[^/]+/[[:digit:]]+/'
then return NULL;
end if;
set url:= trim(both '/' from SUBSTRING_INDEX(url,'kliknieuws.nl/',-1));
set delim = locate ('/',url,1)+1;
set len = locate ('/',url,delim) - delim;
return SUBSTR(url,delim,len);
END
$$
DELIMITER ;
```

Function device_type() used to assign one of device types 'unknown', 'desktop', 'tablet', 'mobile', 'tv', or 'bot' to each entry based on user agent string. The algorithm is based on classification is based on Categorizr open source script [34].

```
USE `CAPA`;
DROP function IF EXISTS `device_type`;
DELIMITER $$
USE `CAPA`$$
```

```
CREATE DEFINER=`admin`@`%` FUNCTION `device_type`(ua varchar(200)) RETURNS
set('unknown','desktop','tablet','mobile','tv','bot') CHARSET latin1
BEGIN
IF ua REGEXP
'GoogleTV|SmartTV|Internet.TV|NetCast|NETTV|AppleTV|boxee|Kylo|Roku|DLNADOC|CE\-HTML'
THEN RETURN 'tv';
END IF;
IF ua REGEXP 'Xbox|PLAYSTATION.3|Wii'
THEN RETURN 'tv';
END IF;
IF ua REGEXP 'iP(a|ro)d' OR ua REGEXP 'tablet' AND ua NOT REGEXP 'RX-34' OR ua REGEXP 'FOLIO' OR ua
REGEXP 'Transformer'
THEN RETURN 'tablet';
END IF;
IF ua REGEXP 'Linux' AND ua REGEXP 'Android' AND ua NOT REGEXP
'Fennec|mobi|HTC.Magic|HTCX06HT|Nexus.One|SC-02B|fone.945'
THEN RETURN 'tablet';
END IF;
IF ua REGEXP 'Kindle' AND ua REGEXP 'Mac.OS' AND ua REGEXP 'Silk'
THEN RETURN 'tablet';
END IF;
IF ua REGEXP 'GT-P10|SC-01C|SHW-M180S|SGH-T849|SCH-I800|SHW-M180L|SPH-P100|SGH-
I987|zt180|HTC(.Flyer|\_Flyer)|Sprint.ATP51|ViewPad7|pandigital(sprnova|nova)|Ideos.S7|Dell.Streak
.7|Advent.Vega|A101IT|A70BHT|MID7015|Next2|nook' OR ua REGEXP 'MB511' AND ua REGEXP
'RUTEM'
THEN RETURN 'tablet';
END IF;
IF ua REGEXP
'BOLT|Blackberry|Fennec|HTC|Iris|Maemo|Minimo|Mobi|mowser|NetFront|Novarra|Prism|RX-
34|Skyfire|Tear|XV6875|XV6975|Google.Wireless.Transcoder|BrowserNG|NokiaBrowser'
THEN RETURN 'mobile';
END IF;
IF ua REGEXP 'Opera' AND ua REGEXP 'HTC|Xda|Mini|Vario|SAMSUNG\-GT\-i8000|SAMSUNG\-SGH\-i9'
THEN RETURN 'mobile';
END IF;
IF ua REGEXP 'Windows.(NT|XP|ME|9)'AND ua NOT REGEXP 'Phone' OR ua REGEXP 'Win(9|.9|NT)'
THEN RETURN 'desktop';
END IF;
IF ua REGEXP 'Macintosh|PowerPC'AND ua NOT REGEXP 'Silk'
THEN RETURN 'desktop';
END IF;
```

```
IF ua REGEXP 'Linux'AND ua REGEXP 'X11'
THEN RETURN 'desktop';
END IF;
IF ua REGEXP 'Solaris|SunOS|BSD'
THEN RETURN 'desktop';
END IF;
IF ua REGEXP 'Bot|Crawler|Spider|Yahoo|ia_archiver|Covario-
IDS|findlinks|DataparkSearch|larbin|Mediapartners-Google|NG-Search|Snappy|Teoma|Jeeves|TinEye'
THEN RETURN 'bot';
END IF;
RETURN 'unknown';
END
$$
DELIMITER ;
```

The following query was used to create indexes:

```
ALTER TABLE `CAPA`.` table_name`
ADD UNIQUE INDEX `joinind` (`cb_num` ASC, `ip_address` ASC)
, ADD INDEX `banneridind` (`banner_id` ASC)
, ADD INDEX `osindex` (`os` ASC)
, ADD INDEX `timestampind` (`unix_timestamp` ASC)
, ADD INDEX `cookieind` (`cookie` ASC)
, ADD INDEX `screenwind` (`screen_width` ASC)
, ADD INDEX `screenhind` (`screen_height` ASC)
, ADD INDEX `colorind` (`color_depth` ASC)
, ADD INDEX `campidind` (`campaingn_ids` ASC)
, ADD INDEX `zoneidind` (`zone_ids` ASC)
, ADD INDEX `pcatind` (`page_category` ASC)
, ADD INDEX `pcodeind` (`page_code` ASC)
, ADD INDEX `deviceind` (`device` ASC)
, ADD INDEX `ipaddressind` (`ip_address` ASC);
```

The following filtering algorithm was applied to the initial data provided:

- Transforming data by using UPDATE and ALTER TABLE queries as described above
- Removing duplicate entries adding unique composite index on cb_num and ip_address columns by using ALTER TABLE … ADD UNIQUE INDEX queries as described above
- Removing bots based on device column and known IP addresses published on iplists.com [33] (stored in table ip_robot), and removing entries with banner_id equals to zero by the following query:

```
DELETE QUICK t1 from `CAPA`.` table_name` t1 LEFT OUTER JOIN ip_robot t2 on
(t1.ip_address=t2.ip) WHERE t2.ip IS NOT NULL OR t1.banner_id=0 OR t1.device='bot';
```

Information about temperature and conditions is taken from www.wunderground.com’s CSV-file http://www.wunderground.com/history/airport/EHEH/2012/08/24/DailyHistory.html?format=1, which can be retrieved every 5-10 minutes (the date should be adjusted accordingly). The last row contains information about temperature and atmospheric conditions. Then, data can be stored locally on the server. We are interested in 2nd and 12th values ($TemperatureC$ and $Conditions$). Their exact location in CSV-file can be updated in the future. Consequently, it make sense to scan second row (first row is empty) with titles of fields and find locations of these our fields).

Appendix 5 Example of a Rule Set

Rules are given in Table 14. Comma must be interpreted as logical AND operation.

Table 14 List of rules

Rule	Class 1= $\mathrm{CTR}_{\mathrm{HIGH}}$ 0= $\mathrm{CTR}_{\mathrm{LOW}}$	CTR, %	Cove-rage, %
device=tablet	1	1.118	9.124
pcat=sport	0	0.169	9.678
pcat=foto	0	0.041	7.288
forum=0, device=mobile	1	0.871	3.873
forum=1, pcat=nieuws, day=wkend, hrs=work_hours	1	0.970	2.379
pcat=nieuws, day=wkend, hrs=work_hours, cond=Rain	1	1.099	0.477
pcat=koopjes	1	2.275	0.111
pcat=poll	1	1.967	0.102
browser=Explorer, pcat=nieuws, day=wkend, hrs=lunch	1	0.880	0.411
device=desktop, os=Windows, pcat=zoeken	0	0.262	0.375
os=Linux, browser=Chrome, pcat=nieuws	0	0.134	0.245
pcat=nieuws, day=wkend, hrs=night, cond=Overcast	1	0.942	0.236
day=wkend, hrs=lunch, cond=Rain	1	1.155	0.149
pcat=agenda	1	1.350	0.107
browser=Firefox, pcat=start_page, day=wkend, hrs=work_hours, temp=10-19	1	0.923	0.210
photo=0, forum=0, day=wrkday, hrs=work_hours, temp=30+	0	0.258	0.237
os=iPhone/iPod, cond=Rain	1	1.001	0.164
photo=0, video=1	1	1.356	0.086
device=desktop, os=Linux, browser=Firefox, day=wrkday	0	0.250	0.200
forum=1, hrs=morning, temp=20-29	1	0.975	0.153
video=1, day=wkend	1	1.199	0.099
device=desktop, browser=Safari, pcat=start_page, temp=20-29, cond=Overcast	0	0.264	0.199
forum=1, browser=Explorer, pcat=nieuws, temp=20-29, cond=Rain	1	0.888	0.178
day=wkend, hrs=morning, cond=Rain	1	1.001	0.130
photo=1, browser=Chrome, hrs=work_hours, cond=Rain	1	0.941	0.147
os=Mac, browser=Safari, pcat=start_page, temp=10-19, cond=normal	0	0.259	0.148
photo=0, forum=0, browser=Firefox, hrs=lunch, cond=normal	0	0.240	0.125
device=desktop, browser=Explorer, pcat=pagina	0	0.264	0.129
device=desktop, browser=Mozilla, pcat=nieuws	1	1.224	0.056
forum=0, os=Mac, pcat=nieuws, day=wrkday, hrs=evening	0	0.245	0.099
device=desktop, day=wrkday, hrs=night, temp=20-29, cond=Overcast	0	0.212	0.087
photo=0, forum=1, os=Windows, hrs=night, cond=normal	1	0.871	0.098
forum=0, os=Mac, browser=Safari, hrs=night	0	0.193	0.074
pcat=login, temp=10-19	0	0.209	0.061
device=desktop, os=Linux, browser=Safari	0	0.000	0.034
forum=1, device=mobile, browser=Mozilla, hrs=lunch	0	0.196	0.051
photo=1, forum=1, browser=Chrome, hrs=morning, cond=normal	1	0.943	0.048

device=desktop, pcat=forum, day=wkend, temp=20-29	0	0.236	0.054
photo=1, browser=Firefox, hrs=lunch, temp=10-19, cond=Overcast	0	0.266	0.059
browser=Firefox, day=wkend, hrs=morning, cond=Overcast	1	1.024	0.033
os=Mac, browser=Chrome, pcat=start_page, temp=20-29	1	0.903	0.043
browser=Explorer, pcat=other, hrs=work_hours, temp=10-19	1	0.900	0.043
os=Mac, browser=Firefox, pcat=start_page, day=wrkday, cond=normal	0	0.255	0.045
os=Windows, browser=Safari, temp=10-19, cond=Overcast	0	0.192	0.037
browser=Chrome, pcat=forum, hrs=work_hours, cond=Overcast	0	0.227	0.038
os=Mac, browser=Firefox, pcat=nieuws, hrs=work_hours, cond=normal	1	0.954	0.033
os=Mac, pcat=start_page, hrs=morning, temp=10-19, cond=Overcast	0	0.245	0.035

The model operates with 47 rules, which are given in Table 14. Order of rules is important. For every page view, all the rules must be applied in the order which is in Table 14, and the rule with the highest length (“most specific rule”) must be chosen. If more than one rule of the same length are fulfilled for the page view, than the first rule must be applied. If no rules fulfill the page view than page view must be assigned to default CTR class.

Function’s pseudo code:

```
/** returns  1:CTR_HIGH, 0:CTR_LOW,( -1):CTR_DEFAULT */
function int getPediction(photo,video,forum,pcat,device,os,browser,day,hrs,temp,cond) {

int prediction=-1;
int ruleLenght=0;

if (device==”tablet”) then {ruleLenght=1; prediction=1};
if (pcat==”sport” && ruleLenght<1) then {ruleLenght=1; prediction=0};
if (pcat==”foto” && ruleLenght<1) then {ruleLenght=1; prediction=0};
if (forum==0 && device==”mobile” && ruleLenght<2) then {ruleLenght=2; prediction=1};
if (forum==1 && pcat==”nieuws” && day==”wkend” && hrs==”work_hours” && ruleLenght<4) then
{ruleLenght=4; prediction=1};
…
// remaining 42 rules
…
return prediction;
}
```

Appendix 6 Test Results

The evaluation framework has been defined above, namely the dataset has two target classes, Click (1) and No-Click (0). Our algorithm produces a classification model assigning to the examples one of three target labels: Click (1), No-Click (0) and Undefined (-1). Thus, the confusion matrix has size 3x2 (see Figure 18). The following metrics are produced by the test framework:

- CTR of CTR$_{\text{low}}$ class: $CTR_{low} = CTR_{class(No\ Click)} \ = \frac{b}{a+b}$
- CTR of CTR$_{\text{low}}$ class: $CTR_{high} = CTR_{class(Click)} \ = \frac{d}{c+d}$
- CTR of CTR$_{\text{not covered}}$ class: $CTR_{not\ covered} = CTR_{class(Undefined)} \ = \frac{h}{g+h}$
- Weighted bias: $W_{bias} = W_{low} \cdot (CTR_{av} - CTR_{low}) + W_{high} \cdot \left(CTR_{high} - CTR_{av}\right) = (a+b) \cdot \left(CTR_{av} - \frac{b}{a+b}\right) + (c+d) \cdot \left(\frac{d}{c+d} - CTR_{av}\right)$, where average CTR is calculated as $CTR_{av} = \frac{b+d+h}{a+b+c+d+g+h}$
- Coverage of CTR$_{\text{low}}$ class: $Cov_{low} = \frac{W_{low}}{W_{total}} = \frac{a+b}{a+b+c+d+g+h}$
- Coverage of CTR$_{\text{high}}$ class: $Cov_{high} = \frac{W_{high}}{W_{total}} = \frac{c+d}{a+b+c+d+g+h}$
- Number of rules: N_{rules}
- Average length of rules: $L_{av} = \frac{\sum_{i=1}^{N} L_i}{N}$
- Weighted average length of rules: $L_{W,av} = \frac{\sum_{i=1}^{N} L_i \cdot Cov_i}{\sum_{i=1}^{N} Cov_i}$

We use some additional metrics mostly to maintain compatibility with information retrieval theory, which evaluate classifiers with such metrics as recall, precision, F-measure and its variants. These metrics do not find direct use in our case study.

The produced confusion matrix shown in Figure 18 has skewed data: the number of actual No-Clicks exceeds the number of actual clicks approximately 200 times. In fact, if the average $CTR = 0.5\%$, then the ratio between clicks and no-clicks is $1{:}\,200$. To reweight the data, we multiply the number actual No-Clicks by average ratio between Clicks and No-Click, $\frac{b+d+h}{a+c+g} = \frac{N^+}{N^-}$, thus replacing the number of No-Clicks with corresponding to it number of Clicks. Adjusted confusion matrix is shown in Figure 24.

		Actual	
		NoClick (0)	Click (1)
Predicted	NoClick (0)	$a^* = a \cdot \frac{N^+}{N^-}$	b
	Click (1)	$c^* = c \cdot \frac{N^+}{N^-}$	d
	Undefined (-1)	$g^* = g \cdot \frac{N^+}{N^-}$	h

Figure 24 Reweighted confusion matrix for two class learning

All the metrics above are calculated for reweighted confusion matrix (asterisk sigh in formulas is skipped).

For one-class learning confusion matrices are shown in Figure 25. Similarly to two-class-learning case, reweighting here is also used.

No-Click:		Actual	
		No-Click	Click
Predicted	No-Click (No-Click + Undefined)	a*	b
	Click	c*	d

Click		Actual	
		No-Click	Click
Predicted	No-Click (NoClick + Undefined)	a*	b
	Click	c*	d

Figure 25 One-class learning confusion matrices

For one-class learning we use the following standard metrics:

- $Precision = \frac{d}{c+d}$
- $Recall = \frac{d}{c+d}$
- $F = \frac{2PR}{P+R}$
- $F_{0.5} = \frac{1.5\ PR}{0.5\ P+R}$ since we give more preference to precision than to recall

To calculate coverage we build a confusion matrix which groups classified examples together, thus comparing them with unclassified (see Figure 26).

		Actual	
		NoClick (0)	Click (1)
Predicted	Defined (0+1)	e=a+c	f=b+d
	Undefined (-1)	g	h

Figure 26 Confusion matrix groupping classified examples together

Settings: December 2011, maxDepth=7, minBias=0.003, Bias quality measure for removing redundant rules

Filtering type	**No filtering**		**Removing redundant rules Bias quality measure with check for minBias**		**Chi-squared filter**		**Chi-squared filter and removing redundant rules Bias quality measure with check for minBias**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2318.12	105.24	2013.08	77.21	2321.31	91.93	2012.28	93.13
Coverage	38.46	0.87	25.06	0.82	38.02	0.46	23.77	0.75
- Coverage$_L$	19.56	0.39	17.29	0.53	19.55	0.37	17.20	0.64
- Coverage$_H$	18.90	0.72	7.77	0.59	18.47	0.53	6.57	0.55
CTR$_L$	0.11	0.01	0.09	0.02	0.11	0.01	0.09	0.02
CTR$_H$	0.74	0.03	1.02	0.05	0.75	0.03	1.11	0.07
Precision$_L$	81.28	1.55	83.2	2.28	81.23	1.54	84.09	2.29
Precision$_H$	62.03	0.83	69.26	1.15	62.29	0.61	71.06	1.43
Recall$_L$	19.63	0.39	17.35	0.53	19.62	0.37	17.27	0.64
Recall$_H$	30.79	1.21	17.39	0.95	30.43	1.13	16.02	1.05
F$_{0.5,L}$	39.7	0.46	36.72	0.64	39.67	0.39	36.70	0.80
F$_{0.5,H}$	46.34	1.11	34.71	1.29	46.17	1.03	33.09	1.56
N$_{Rules}$	16077.87	245.62	110.23	4.27	162.92	9.14	41.55	3.62
Length$_{ave}$	5.45	0.01	4.72	0.09	3.38	0.07	3.61	0.13
Length$_{w,ave}$	5.04	0.01	4.2	0.16	2.6	0.07	2.68	0.18
Coverage$_{Rule}$	263.71	6.51	1.83	0.11	3.57	0.2	1.38	0.12

Settings: December 2011, maxDepth=7, minBias=0.003, Bias quality measure for removing redundant rules

filtering	No filtering		Removing redundant rules Bias quality measure without check for minBias		Chi-squared filter		Chi-squared filter and Removing redundant rules Bias quality measure without check for minBias	
	average	stdev	average	stdev	average	stdev	average	stdev
Weighted bias	2318.12	105.24	2326.58	101.96	2321.31	91.93	2353.14	105.62
Coverage	38.46	0.87	33.68	0.64	38.02	0.46	34.42	0.93
- Coverage$_{L}$	19.56	0.39	19.49	0.29	19.55	0.37	19.82	0.46
- Coverage$_{H}$	18.90	0.72	14.19	0.63	18.47	0.53	14.59	0.81
CTR$_{L}$	0.11	0.01	0.10	0.01	0.11	0.01	0.10	0.01
CTR$_{H}$	0.74	0.03	0.83	0.04	0.75	0.03	0.83	0.05
Precision$_{L}$	81.28	1.55	82.26	1.99	81.23	1.54	81.65	2.03
Precision$_{H}$	62.03	0.83	64.73	0.91	62.29	0.61	64.65	1.11
Recall$_{L}$	19.63	0.39	19.56	0.29	19.62	0.37	19.89	0.46
Recall$_{H}$	30.79	1.21	25.96	1.13	30.43	1.13	26.59	1.24
F$_{0.5,L}$	39.7	0.46	39.76	0.39	39.67	0.39	40.11	0.41
F$_{0.5,H}$	46.34	1.11	43.20	1.20	46.17	1.03	43.75	1.28
N$_{Rules}$	16077.87	245.62	152.87	3.72	162.92	9.14	69.00	4.00
Length$_{ave}$	5.45	0.01	4.72	0.08	3.38	0.07	3.51	0.10
Length$_{w,ave}$	5.04	0.01	4.18	0.15	2.6	0.07	2.51	0.11
Coverage$_{Rule}$	263.71	6.51	2.47	0.17	3.57	0.2	2.03	0.13

Settings: December, maxDepth=7, minBias=0.003, WRAcc quality measure for removing redundant rules with check for minBias

filtering	**No filtering**		**Removing redundant rules WRAcc quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules WRAcc quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2318.12	105.24	2322.36	239.96	2321.31	91.93	2408.47	327.21
Coverage	38.46	0.87	34.82	0.52	38.02	0.46	34.51	0.61
- Coverage$_{L}$	19.56	0.39	19.26	0.27	19.55	0.37	19.64	0.34
- Coverage$_{H}$	18.90	0.72	15.55	0.45	18.47	0.53	14.88	0.58
CTR$_{L}$	0.11	0.01	0.10	0.03	0.11	0.01	0.10	0.04
CTR$_{H}$	0.74	0.03	0.81	0.09	0.75	0.03	0.85	0.11
Precision$_{L}$	81.28	1.55	81.67	3.35	81.23	1.54	81.8	5.15
Precision$_{H}$	62.03	0.83	63.9	2.23	62.29	0.61	64.99	2.2
Recall$_{L}$	19.63	0.39	19.33	0.27	19.62	0.37	19.71	0.34
Recall$_{H}$	30.79	1.21	27.56	2.58	30.43	1.13	27.66	2.7
F$_{0.5,L}$	39.7	0.46	39.35	0.54	39.67	0.39	39.86	0.85
F$_{0.5,H}$	46.34	1.11	44.35	2.90	46.17	1.03	44.77	3.01
N$_{Rules}$	16077.87	245.62	37.4	2.22	162.92	9.14	36.43	2.3
Length$_{ave}$	5.45	0.01	4.17	0.18	3.38	0.07	3.42	0.1
Length$_{w,ave}$	5.04	0.01	2.34	0.11	2.60	0.07	1.92	0.05
Coverage$_{Rule}$	263.71	6.51	1.15	0.02	3.57	0.20	1.15	0.03

Settings: December, maxDepth=7, minBias=0.003, Binomial quality measure for removing redundant rules with check for minBias

filtering	**No filtering**		**Removing redundant rules Binomial quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules Binomial quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2318.12	105.24	2310.48	317.29	2321.31	91.93	2332.96	324.85
Coverage	38.46	0.87	34.36	0.59	38.02	0.46	34.34	0.74
- Coverage$_{L}$	19.56	0.39	19.08	0.26	19.55	0.37	19.18	0.29
- Coverage$_{H}$	18.90	0.72	15.27	0.53	18.47	0.53	15.15	0.68
CTR$_{L}$	0.11	0.01	0.10	0.03	0.11	0.01	0.10	0.03
CTR$_{H}$	0.74	0.03	0.82	0.1	0.75	0.03	0.83	0.10
Precision$_{L}$	81.28	1.55	81.94	4.21	81.23	1.54	81.68	4.37
Precision$_{H}$	62.03	0.83	64.12	2.3	62.29	0.61	64.38	2.82
Recall$_{L}$	19.63	0.39	19.15	0.25	19.62	0.37	19.25	0.29
Recall$_{H}$	30.79	1.21	27.31	2.34	30.43	1.13	27.48	3.03
F$_{0.5,L}$	39.7	0.46	39.13	0.64	39.67	0.39	39.22	0.69
F$_{0.5,H}$	46.34	1.11	44.21	2.75	46.17	1.03	44.41	3.51
N$_{Rules}$	16077.87	245.62	51.7	2.79	162.92	9.14	45.63	3.29
Length$_{ave}$	5.45	0.01	4.58	0.09	3.38	0.07	3.43	0.10
Length$_{w,ave}$	5.04	0.01	2.78	0.08	2.60	0.07	2.03	0.08
Coverage$_{Rule}$	263.71	6.51	1.27	0.04	3.57	0.20	1.29	0.04

Settings: December, maxDepth=7, minBias=0.003, Rule length metric for removing redundant rules with check for minBias

filtering	**No filtering**		**Removing redundant rules Rule length quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules Rule length quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2318.12	105.24	2241.09	108.33	2321.31	91.93	2259.93	107.57
Coverage	38.46	0.87	35.64	0.90	38.02	0.46	35.10	0.74
- Coverage$_L$	19.56	0.39	18.69	0.35	19.55	0.37	18.74	0.35
- Coverage$_H$	18.90	0.72	16.95	0.84	18.47	0.53	16.35	0.68
CTR$_L$	0.11	0.01	0.10	0.01	0.11	0.01	0.10	0.01
CTR$_H$	0.74	0.03	0.76	0.04	0.75	0.03	0.78	0.04
Precision$_L$	81.28	1.55	82.10	1.67	81.23	1.54	82.60	2.04
Precision$_H$	62.03	0.83	62.68	0.88	62.29	0.61	63.07	0.96
Recall$_L$	19.63	0.39	18.79	0.37	19.62	0.37	18.81	0.35
Recall$_H$	30.79	1.21	28.33	1.41	30.43	1.13	27.84	1.04
F$_{0.5,L}$	39.7	0.46	38.66	0.42	39.67	0.39	38.76	0.48
F$_{0.5,H}$	46.34	1.11	44.61	1.29	46.17	1.03	44.34	1.05
N$_{Rules}$	16077.87	245.62	41.46	3.41	162.92	9.14	38.30	3.11
Length$_{ave}$	5.45	0.01	3.16	0.11	3.38	0.07	3.16	0.18
Length$_{w,ave}$	5.04	0.01	1.94	0.07	2.60	0.07	1.89	0.09
Coverage$_{Rule}$	263.71	6.51	1.33	0.04	3.57	0.20	1.26	0.03

Settings: December, maxDepth=7, minBias =0.002, Bias quality measure for removing redundant rules with check for minBias

filtering	**No filtering**		**Removing redundant rules Bias quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules Bias quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2672.24	245.33	2278.47	147.5	2617.15	445.56	2264.52	303.07
Coverage	60.13	0.97	35.43	0.96	59.65	0.66	32.46	0.71
- Coverage$_L$	26.82	0.72	22.67	0.38	28.14	0.59	22.86	0.55
- Coverage$_H$	33.31	0.47	12.76	0.83	31.51	0.66	9.59	0.70
CTR$_L$	0.18	0.02	0.15	0.02	0.20	0.04	0.14	0.04
CTR$_H$	0.66	0.03	0.85	0.06	0.67	0.07	0.96	0.14
Precision$_L$	71.48	2.73	75.82	2.12	69.86	3.73	76.43	4.31
Precision$_H$	59.23	0.94	65.22	1.55	59.35	1.73	67.73	3.02
Recall$_L$	26.89	0.72	22.74	0.38	28.21	0.59	22.94	0.55
Recall$_H$	48.33	1.96	23.84	1.39	46.04	3.56	20.19	2.42
F$_{0.5,L}$	45.99	0.44	42.63	0.45	46.77	0.96	42.96	1.01
F$_{0.5,H}$	55.08	1.36	41.29	1.63	54.10	2.56	37.89	3.41
N$_{Rules}$	24120.97	270.48	177.03	4.37	295.93	9.70	67.87	3.44
Length$_{ave}$	5.45	0.01	4.89	0.04	3.72	0.05	3.77	0.08
Length$_{w,ave}$	5.05	0.01	4.53	0.11	2.69	0.03	2.70	0.17
Coverage$_{Rule}$	223.07	3.61	0.64	0.03	3.65	0.12	1.64	0.17

Settings: December, maxDepth=7, minBias =0.002, WRAcc quality measure for removing redundant rules with check for minBias

filtering	**No filtering**		**Removing redundant rules WRAcc quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules WRAcc quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2672.24	245.33	2660.58	109.96	2617.15	445.56	2775.89	339.09
Coverage	60.13	0.97	56.71	1.66	59.65	0.66	56.02	0.77
- Coverage$_L$	26.82	0.72	26.31	0.96	28.14	0.59	26.44	0.48
- Coverage$_H$	33.31	0.47	30.39	1.57	31.51	0.66	29.59	0.69
CTR$_L$	0.18	0.02	0.18	0.02	0.20	0.04	0.17	0.04
CTR$_H$	0.66	0.03	0.68	0.02	0.67	0.07	0.70	0.06
Precision$_L$	71.48	2.73	72.06	1.60	69.86	3.73	72.71	3.93
Precision$_H$	59.23	0.94	59.89	0.65	59.35	1.73	60.61	1.70
Recall$_L$	26.89	0.72	26.39	0.96	28.21	0.59	26.51	0.48
Recall$_H$	48.33	1.96	45.25	1.85	46.04	3.56	45.52	3.17
F$_{0.5,L}$	45.99	0.44	45.66	0.68	46.77	0.96	45.95	1.02
F$_{0.5,H}$	55.08	1.36	54.04	0.91	54.10	2.56	54.55	2.40
N$_{Rules}$	24120.97	270.48	47.40	3.86	295.93	9.70	50.97	2.50
Length$_{ave}$	5.45	0.01	4.27	0.18	3.72	0.05	3.76	0.10
Length$_{w,ave}$	5.05	0.01	2.80	0.11	2.69	0.03	2.27	0.06
Coverage$_{Rule}$	223.07	3.61	1.28	0.03	3.65	0.12	1.26	0.02

Settings: December, maxDepth=7, minBias=0.002, Binomial quality measure for removing redundant rules

filtering	**No filtering**		**Removing redundant rules Binomial quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules Binomial quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2672.24	245.33	2712.84	284.83	2617.15	445.56	2697.00	321.71
Coverage	60.13	0.97	54.95	0.74	59.65	0.66	55.77	0.76
- Coverage$_L$	26.82	0.72	25.78	0.57	28.14	0.59	25.83	0.57
- Coverage$_H$	33.31	0.47	29.26	0.59	31.51	0.66	29.95	0.68
CTR$_L$	0.18	0.02	0.17	0.03	0.20	0.04	0.17	0.03
CTR$_H$	0.66	0.03	0.70	0.05	0.67	0.07	0.69	0.05
Precision$_L$	71.48	2.73	73.02	3.56	69.86	3.73	72.81	3.36
Precision$_H$	59.23	0.94	60.43	1.03	59.35	1.73	60.16	1.34
Recall$_L$	26.89	0.72	25.79	0.59	28.21	0.59	25.9	0.57
Recall$_H$	48.33	1.96	44.57	1.94	46.04	3.56	45.18	2.29
F$_{0.5,L}$	45.99	0.44	45.30	0.79	46.77	0.96	45.36	0.89
F$_{0.5,H}$	55.08	1.36	54.01	1.46	54.10	2.56	54.16	1.78
N$_{Rules}$	24120.97	270.48	68.79	3.30	295.93	9.70	62.47	2.98
Length$_{ave}$	5.45	0.01	4.57	0.08	3.72	0.05	3.62	0.09
Length$_{w,ave}$	5.05	0.01	3.07	0.11	2.69	0.03	2.33	0.07
Coverage$_{Rule}$	223.07	3.61	1.34	0.07	3.65	0.12	1.37	0.04

Settings: December, maxDepth=7, minBias=0.002, Rule length quality measure for removing redundant rules

filtering	**No filtering**		**Removing redundant rules Rule length quality measure**		**Chi-squared filter**		**Chi-squared filter and Removing redundant rules Rule length quality measure**	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2672.24	245.33	2559.71	104.36	2617.15	445.56	2582.22	116.97
Coverage	60.13	0.97	57.78	1.27	59.65	0.66	57.51	1.33
- Coverage$_{L}$	26.82	0.72	25.41	0.91	28.14	0.59	25.43	0.94
- Coverage$_{H}$	33.31	0.47	32.38	1.09	31.51	0.66	32.08	0.94
CTR$_{L}$	0.18	0.02	0.18	0.01	0.20	0.04	0.17	0.02
CTR$_{H}$	0.66	0.03	0.66	0.02	0.67	0.07	0.66	0.02
Precision$_{L}$	71.48	2.73	72.10	1.60	69.86	3.73	72.39	2.08
Precision$_{H}$	59.23	0.94	59.05	0.54	59.35	1.73	59.19	0.57
Recall$_{L}$	26.89	0.72	25.48	0.91	28.21	0.59	25.50	0.94
Recall$_{H}$	48.33	1.96	46.59	1.42	46.04	3.56	46.43	1.17
F$_{0.5,L}$	45.99	0.44	44.75	0.79	46.77	0.96	44.84	0.63
F$_{0.5,H}$	55.08	1.36	54.21	0.77	54.10	2.56	54.21	0.70
N$_{Rules}$	24120.97	270.48	56.47	4.49	295.93	9.70	55.30	3.70
Length$_{ave}$	5.45	0.01	3.34	0.15	3.72	0.05	3.40	0.16
Length$_{w,ave}$	5.05	0.01	2.30	0.08	2.69	0.03	2.28	0.10
Coverage$_{Rule}$	223.07	3.61	1.47	0.07	3.65	0.12	1.42	0.06

Settings: Summer 2012, maxDepth=7, bias=0.003, Chi-squared filter and removing redundant rules

filtering	removing redundant rules based on WRAcc		removing redundant rules based on WRAcc with double-checking bias		removing redundant rules based on Bias		removing redundant rules based on WRAcc with Bias	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	10933.42	279.31	10593.07	252.31	11653.91	295.58	10765.75	226.80
Coverage	42.63	0.75	35.26	0.42	41.05	0.69	27.84	0.76
- Coverage$_{L}$	17.81	0.25	17.31	0.22	19.03	0.32	17.32	0.65
- Coverage$_{H}$	24.82	0.70	17.95	0.40	22.02	0.73	10.52	0.40
CTR$_{L}$	0.16	0.01	0.14	0.01	0.16	0.01	0.13	0.01
CTR$_{H}$	0.90	0.02	1.00	0.02	0.97	0.02	1.31	0.03
Precision$_{L}$	78.52	0.93	80.03	1.20	78.35	0.81	81.32	1.23
Precision$_{H}$	61.45	0.39	63.91	0.35	63.17	0.50	69.96	0.51
Recall$_{L}$	17.88	0.25	17.38	0.22	19.11	0.32	17.39	0.65
Recall$_{H}$	39.43	0.89	31.66	0.67	37.61	0.78	24.32	0.70
F$_{0.5,L}$	36.86	0.27	36.35	0.21	38.52	0.41	36.52	0.84
F$_{0.5,H}$	51.80	0.54	47.70	0.56	51.50	0.48	43.02	0.73
N$_{Rules}$	77.43	3.59	57.60	3.52	228.13	7.93	140.73	5.34
Length$_{ave}$	4.31	0.13	4.33	0.11	4.36	0.06	4.46	0.07
Length$_{w,ave}$	2.33	0.08	1.89	0.06	3.11	0.07	3.14	0.11
Coverage$_{Rule}$	1.38	0.05	1.11	0.01	2.71	0.14	1.81	0.13

Settings: minBias=0.003, maxDepth=(5 and 7) Chi-squared filter

	November				December				January			
Max depth	7		5		7		5		7		5	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	2279.52	97.28	2227.58	103.34	2321.31	91.93	2291.91	103.51	2790.32	160.75	2750.32	139.18
Coverage	40.14	1.51	38.09	1.3	38.02	0.46	38.05	0.95	35.33	0.7	33.83	0.44
- Coverage$_{L}$	19.06	0.30	18.71	0.38	19.55	0.37	19.40	0.41	16.35	0.38	15.74	0.41
- Coverage$_{H}$	21.08	1.54	19.38	1.46	18.47	0.53	18.65	0.79	18.98	0.71	18.08	0.49
CTR$_{L}$	0.1	0.01	0.09	0.01	0.11	0.01	0.1	0.01	0.15	0.02	0.15	0.01
CTR$_{H}$	0.64	0.03	0.65	0.03	0.75	0.03	0.74	0.04	0.72	0.03	0.74	0.03
Precision$_{L}$	81.22	1.26	81.95	1.54	81.23	1.54	81.58	1.88	75.84	1.84	76.22	1.72
Precision$_{H}$	60.41	1.02	60.67	0.91	62.29	0.61	61.95	0.82	60.95	0.93	61.33	0.72
Recall$_{L}$	19.12	0.3	18.77	0.38	19.62	0.37	19.47	0.41	16.41	0.38	15.79	0.41
Recall$_{H}$	32.06	1.56	29.8	1.75	30.43	1.13	30.27	1.08	29.54	0.87	28.59	0.63
F$_{0.5,L}$	39	0.41	38.62	0.48	39.67	0.39	39.53	0.45	34.35	0.55	33.49	0.55
F$_{0.5,H}$	46.62	1.08	45.05	1.33	46.17	1.03	45.92	0.97	45	0.86	44.38	0.66
N$_{Rules}$	187.23	11.74	164.88	9	162.92	9.14	162.88	11.75	251.27	12.84	216.69	12.62
Length$_{ave}$	3.96	0.09	3.67	0.08	3.38	0.07	3.36	0.07	3.9	0.08	3.58	0.07
Length$_{w,ave}$	3.03	0.1	2.85	0.09	2.6	0.07	2.59	0.1	2.73	0.08	2.58	0.05
Coverage$_{Rule}$	3.1	0.18	3.04	0.16	3.57	0.2	3.59	0.22	3.68	0.15	3.63	0.14

Settings: minBias=0.003, maxDepth=(5 and 7) Chi-squared filter and Redundant Rule Filter (Bias)

Dataset	November				December				January			
Max depth	7		5		7		5		7		5	
	average	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**	**average**	**stdev**
Weighted bias	1892.97	115.29	1845.93	108.55	2012.28	93.13	2029.28	89.72	2422.17	104.65	2417.32	120.04
Coverage	23.21	1.69	22.13	1.65	23.77	0.75	23.57	0.65	21.13	0.69	20.43	0.66
- Coverage$_{L}$	15.55	1.52	14.94	1.62	17.20	0.64	17.08	0.41	12.98	0.52	12.62	0.57
- Coverage$_{H}$	7.66	0.64	7.19	0.56	6.57	0.55	6.49	0.52	8.36	0.50	7.81	0.56
CTR$_{L}$	0.08	0.01	0.07	0.01	0.09	0.02	0.08	0.01	0.12	0.01	0.11	0.01
CTR$_{H}$	0.90	0.04	0.91	0.06	1.11	0.07	1.13	0.08	0.99	0.03	1.01	0.04
Precision$_{L}$	84.46	2.05	85.7	2.58	84.09	2.29	84.84	2.14	79.36	1.69	80.58	1.9
Precision$_{H}$	68.09	0.90	68.41	1.26	71.06	1.43	71.37	1.49	68.10	0.80	68.66	0.92
Recall$_{L}$	15.6	1.53	14.99	1.63	17.27	0.64	17.14	0.41	12.93	0.45	12.66	0.57
Recall$_{H}$	16.27	1.25	15.53	1.44	16.02	1.05	16.06	0.91	17.53	1.17	17.03	1.22
F$_{0.5,L}$	34.09	2.21	33.22	2.43	36.70	0.80	36.61	0.55	29.24	0.70	28.88	0.82
F$_{0.5,H}$	32.98	1.73	31.99	2.14	33.09	1.56	33.2	1.37	34.68	1.63	34.12	1.67
N$_{Rules}$	49.28	3.78	44.43	3.48	41.55	3.62	40.61	3.71	67.00	3.72	58.33	3.75
Length$_{ave}$	3.87	0.16	3.52	0.11	3.61	0.13	3.49	0.12	4.00	0.10	3.56	0.08
Length$_{w,ave}$	2.84	0.22	2.68	0.19	2.68	0.18	2.63	0.17	2.94	0.13	2.72	0.12
Coverage$_{Rule}$	1.29	0.12	1.29	0.11	1.38	0.12	1.35	0.09	1.37	0.13	1.46	0.15

Conditions:

- Depth 3
- Min coverage 2000 pviews
- Post-filtering with deleting examples
- Min bias 0.001 (CTR$_{AVE}$ +/-0.10%)

			Model					
			November		December		January	
			Total rules:172 Average length:2.936 Weighted average length:2.963 Average covering:2.185		Total rules:171 Average length:2.895 Weighted average length:2.893 Average covering:2.230		Total rules: 226 Average length: 2.938 Weighted average length: 2.930 Average covering: 2.062	
Dataset	November		weighted bias: 3363.716 coverage: 54.27%		weighted bias: 2384.971 coverage: 54.62%		weighted bias: 2198.495 coverage: 61.40%	
			Low CTR	High CTR	Low CTR	High CTR	Low CTR	High CTR
		CTR	0.12%	0.70%	0.22%	0.63%	0.24%	0.57%
		Precision	78.37%	62.59%	66.13%	59.96%	63.67%	57.63%
		Recall	26.92%	45.78%	31.66%	34.41%	38.19%	31.63%
		F1	40.07%	52.88%	42.82%	43.73%	47.74%	40.84%
		F0.5	47.87%	55.76%	48.52%	48.06%	52.08%	45.23%
	December		weighted bias: 2680.411 coverage: 59.73%		weighted bias: 3187.298 coverage: 48.88%		weighted bias: 2684.846 coverage: 51.88%	
			Low CTR	High CTR	Low CTR	High CTR	Low CTR	High CTR
		CTR	0.16%	0.65%	0.12%	0.81%	0.23%	0.41%
		Precision	74.39%	58.84%	79.89%	63.95%	66.74%	65.25%
		Recall	24.79%	49.95%	24.91%	42.50%	37.87%	26.36%
		F1	37.19%	54.03%	37.98%	51.07%	48.32%	37.55%
		F0.5	44.62%	55.54%	46.02%	54.74%	53.21%	43.73%
	January		weighted bias: 2945.127 coverage: 56.59%		weighted bias: 3004.511 coverage: 51.32%		weighted bias: 4348.575 coverage: 53.05%	
			Low CTR	High CTR	Low CTR	High CTR	Low CTR	High CTR
		CTR	0.22%	0.62%	0.23%	0.65%	0.23%	0.92%
		Precision	68.28%	57.14%	67.41%	58.34%	66.88%	66.57%
		Recall	20.28%	48.41%	24.78%	37.18%	38.46%	29.10%
		F1	31.27%	52.41%	36.24%	45.41%	48.84%	40.50%
		F0.5	38.16%	53.89%	42.84%	49.03%	53.66%	46.57%

Conditions:

- Depth 5
- Min coverage 2000 pviews
- Post-filtering with deleting examples
- Min bias 0.001 (CTR_{AVE} +/-0.10%)

		Model								
		November			December			January		
		Total rules:389 Average length:4.576 Weighted average length:4.450 Average covering:2.129			Total rules:357 Average length:4.510 Weighted average length:4.289 Average covering:2.2015			Total rules: 481 Average length: 4.636 Weighted average length: 4.606 Average covering: 1.842		
Dataset	November	weighted bias: 4096.964 coverage: 65.43%			weighted bias: 2369.706 coverage: 65.67%			weighted bias: 2465.060 coverage: 71.19%		
			Low CTR	High CTR		Low CTR	High CTR		Low CTR	High CTR
		CTR	0.13%	0.72%	CTR	0.26%	0.60%	CTR	0.25%	0.57%
		Precision	76.22%	63.30%	Precision	62.15%	58.86%	Precision	63.18%	57.64%
		Recall	34.07%	54.10%	Recall	39.84%	36.97%	Recall	37.32%	46.11%
		F1	47.09%	58.34%	F1	48.55%	45.41%	F1	46.92%	51.23%
		F0.5	53.96%	59.90%	F0.5	52.38%	49.16%	F0.5	51.32%	53.20%
	December	weighted bias: 2677.407 coverage: 66.25%			weighted bias: 4045.154 coverage: 66.40%			weighted bias: 2573.299 coverage: 65.68%		
			Low CTR	High CTR		Low CTR	High CTR		Low CTR	High CTR
		CTR	0.21%	0.65%	CTR	0.16%	0.82%	CTR	0.26%	0.67%
		Precision	68.26%	58.67%	Precision	73.49%	64.35%	Precision	64.10%	59.67%
		Recall	29.74%	51.84%	Recall	37.79%	51.65%	Recall	37.40%	41.85%
		F1	41.43%	55.04%	F1	49.91%	57.30%	F1	47.24%	49.20%
		F0.5	47.67%	56.20%	F0.5	55.89%	59.48%	F0.5	51.78%	52.26%
	January	weighted bias: 2923.451 coverage: 61.39%			weighted bias: 3306.631 coverage: 64.09%			weighted bias: 5433.274 coverage: 64.94%		
			Low CTR	High CTR		Low CTR	High CTR		Low CTR	High CTR
		CTR	0.26%	0.61%	CTR	0.28%	0.65%	CTR	0.20%	0.81%
		Precision	64.26%	56.91%	Precision	62.66%	58.33%	Precision	70.36%	63.68%
		Recall	25.26%	47.71%	Recall	34.68%	41.19%	Recall	37.55%	48.02%
		F1	36.26%	51.91%	F1	44.65%	48.28%	F1	48.97%	54.75%
		F0.5	42.43%	53.47%	F0.5	49.38%	51.22%	F0.5	54.49%	57.44%

Conditions:

- Depth 5
- Min coverage 2000 pviews
- Post-filtering with deleting examples
- Min bias 0.002 (CTR_{AVE} +/-0.20%)

<table>
<tr><td colspan="3"></td><td colspan="6">Model</td></tr>
<tr><td colspan="3"></td><td colspan="2">November</td><td colspan="2">December</td><td colspan="2">January</td></tr>
<tr><td colspan="3"></td><td colspan="2">Total rules:249
Average length:4.426
Weighted average length:4.342
Average covering:1.833</td><td colspan="2">Total rules:234
Average length:4.338
Weighted average length:3.934
Average covering:1.946</td><td colspan="2">Total rules: 319
Average length: 4.561
Weighted average length: 4.516
Average covering: 1.742</td></tr>
<tr><td rowspan="21">Dataset</td><td rowspan="7">November</td><td colspan="1"></td><td colspan="2">weighted bias: 3271.246
coverage: 40.43%</td><td colspan="2">weighted bias: 2114.902
coverage: 41.47%</td><td colspan="2">weighted bias: 2151.282
coverage: 44.63%</td></tr>
<tr><td></td><td>Low CTR</td><td>High CTR</td><td>Low CTR</td><td>High CTR</td><td>Low CTR</td><td>High CTR</td></tr>
<tr><td>CTR</td><td>0.07%</td><td>0.85%</td><td>0.20%</td><td>0.71%</td><td>0.19%</td><td>0.65%</td></tr>
<tr><td>Precision</td><td>85.53%</td><td>66.94%</td><td>67.75%</td><td>62.88%</td><td>68.71%</td><td>60.64%</td></tr>
<tr><td>Recall</td><td>23.61%</td><td>34.09%</td><td>29.14%</td><td>20.93%</td><td>27.95%</td><td>25.74%</td></tr>
<tr><td>F1</td><td>37.01%</td><td>45.17%</td><td>40.75%</td><td>31.41%</td><td>39.74%</td><td>36.14%</td></tr>
<tr><td>F0.5</td><td>45.64%</td><td>50.66%</td><td>46.99%</td><td>37.70%</td><td>46.23%</td><td>41.76%</td></tr>
<tr><td rowspan="7">December</td><td></td><td colspan="2">weighted bias: 2336.483
coverage: 43.28%</td><td colspan="2">weighted bias: 3282.937
coverage: 39.73%</td><td colspan="2">weighted bias: 2259.008
coverage: 40.02%</td></tr>
<tr><td></td><td>Low CTR</td><td>High CTR</td><td>Low CTR</td><td>High CTR</td><td>Low CTR</td><td>High CTR</td></tr>
<tr><td>CTR</td><td>0.13%</td><td>0.70%</td><td>0.09%</td><td>1.01%</td><td>0.18%</td><td>0.78%</td></tr>
<tr><td>Precision</td><td>77.61%</td><td>60.77%</td><td>83.67%</td><td>68.98%</td><td>72.23%</td><td>63.30%</td></tr>
<tr><td>Recall</td><td>21.42%</td><td>33.90%</td><td>24.63%</td><td>33.57%</td><td>24.47%</td><td>26.86%</td></tr>
<tr><td>F1</td><td>33.57%</td><td>43.52%</td><td>38.06%</td><td>45.16%</td><td>36.56%</td><td>37.72%</td></tr>
<tr><td>F0.5</td><td>41.40%</td><td>48.07%</td><td>46.51%</td><td>51.04%</td><td>43.76%</td><td>43.59%</td></tr>
<tr><td rowspan="7">January</td><td></td><td colspan="2">weighted bias: 2520.644
coverage: 38.44%</td><td colspan="2">weighted bias: 2751.116
coverage: 39.59%</td><td colspan="2">weighted bias: 4246.604
coverage: 36.52%</td></tr>
<tr><td></td><td>Low CTR</td><td>High CTR</td><td>Low CTR</td><td>High CTR</td><td>Low CTR</td><td>High CTR</td></tr>
<tr><td>CTR</td><td>0.20%</td><td>0.68%</td><td>0.22%</td><td>0.72%</td><td>0.12%</td><td>1.00%</td></tr>
<tr><td>Precision</td><td>69.74%</td><td>59.46%</td><td>67.96%</td><td>60.91%</td><td>79.96%</td><td>68.36%</td></tr>
<tr><td>Recall</td><td>17.20%</td><td>31.15%</td><td>21.92%</td><td>27.54%</td><td>22.57%</td><td>30.14%</td></tr>
<tr><td>F1</td><td>27.59%</td><td>40.88%</td><td>33.15%</td><td>37.93%</td><td>35.20%</td><td>41.83%</td></tr>
<tr><td>F0.5</td><td>34.56%</td><td>45.64%</td><td>39.98%</td><td>43.39%</td><td>43.28%</td><td>48.05%</td></tr>
</table>

Conditions:

- Depth 5
- Min coverage 2000 pviews
- Post-filtering with deleting examples and chi^2-filtering
- Min bias 0.002 (CTR$_{AVE}$+/-0.20%)

			Model					
			November		December		January	
			Total rules: 130 Average length: 3.831 Weighted average length: 2.687 Average covering:1.740		Total rules: 103 Average length: 3.699 Weighted average length: 2.977 Average covering:1.659		Total rules: 155 Average length: 3.845 Weighted average length: 3.149 Average covering: 1.548	
Dataset	November		weighted bias: 2938.682 coverage: 38.33%		weighted bias: 2114.014 coverage: 41.50%		weighted bias: 2126.913 coverage: 42.93%	
			Low CTR	High CTR	Low CTR	High CTR	Low CTR	High CTR
		CTR	0.09%	0.84%	0.20%	0.72%	0.20%	0.69%
		Precision	81.74%	66.71%	67.64%	63.10%	67.56%	62.12%
		Recall	23.66%	29.44%	30.06%	19.62%	29.29%	22.40%
		F1	36.70%	40.85%	41.62%	29.93%	40.87%	32.93%
		F0.5	44.95%	46.91%	47.74%	36.29%	47.06%	39.04%
	December		weighted bias: 2425.166 coverage: 42.74%		weighted bias: 3002.249 coverage: 38.16%		weighted bias: 2255.769 coverage: 39.13%%	
			Low CTR	High CTR	Low CTR	High CTR	Low CTR	High CTR
		CTR	0.12%	0.72%	0.12%	0.81%	0.17%	0.80%
		Precision	79.67%	61.31%	79.89%	63.95%	72.52%	63.79%
		Recall	20.62%	35.07%	24.91%	42.50%	24.95%	25.03%
		F1	32.76%	44.62%	37.98%	51.07%	37.12%	35.95%
		F0.5	40.76%	49.07%	46.02%	54.74%	44.34%	42.07%
	January		weighted bias: 2722.666 coverage: 40.08%		weighted bias: 2738.667 coverage: 41.32%		weighted bias: 3757.781 coverage: 35.07%	
			Low CTR	High CTR	Low CTR	High CTR	Low CTR	High CTR
		CTR	0.18%	0.68%	0.22%	0.70%	0.23%	0.92%
		Precision	72.07%	59.46%	67.70%	60.13%	66.88%	66.57%
		Recall	17.31%	33.40%	23.77%	26.50%	38.46%	29.10%
		F1	27.91%	42.78%	35.18%	36.78%	48.84%	40.50%
		F0.5	35.07%	47.18%	41.89%	42.25%	53.66%	46.57%

Conditions:

- Depth 5
- Min coverage 2000 pviews
- Post-filtering with deleting examples and chi^2-filter
- Min bias 0.003 (CTR$_{AVE}$ +/-0.30%)

<table>
<tr><td></td><td></td><td colspan="3">Model</td></tr>
<tr><td></td><td></td><td>November</td><td>December</td><td>January</td></tr>
<tr><td></td><td></td><td>Total rules: 70
Average length: 3.914
Weighted average length: 2.825
Average covering:1.288</td><td>Total rules: 69
Average length: 3.754
Weighted average length: 2.986
Average covering:1.485</td><td>Total rules: 102
Average length: 3.765
Weighted average length: 2.930
Average covering: 1.441</td></tr>
<tr><td rowspan="3">Dataset</td><td>November</td>
<td>weighted bias: 2264.751
coverage: 23.84%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.05%</td><td>1.03%</td></tr><tr><td>Precision</td><td>89.65%</td><td>71.09%</td></tr><tr><td>Recall</td><td>16.13%</td><td>18.98%</td></tr><tr><td>F1</td><td>27.34%</td><td>29.96%</td></tr><tr><td>F0.5</td><td>35.58%</td><td>37.12%</td></tr></table></td>
<td>weighted bias: 1931.869
coverage: 27.04%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.11%</td><td>0.82%</td></tr><tr><td>Precision</td><td>79.79%</td><td>66.03%</td></tr><tr><td>Recall</td><td>19.15%</td><td>15.40%</td></tr><tr><td>F1</td><td>30.89%</td><td>24.98%</td></tr><tr><td>F0.5</td><td>38.81%</td><td>31.50%</td></tr></table></td>
<td>weighted bias: 1880.011
coverage: 32.90%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.09%</td><td>0.62%</td></tr><tr><td>Precision</td><td>83.00%</td><td>59.78%</td></tr><tr><td>Recall</td><td>16.77%</td><td>24.01%</td></tr><tr><td>F1</td><td>27.90%</td><td>34.26%</td></tr><tr><td>F0.5</td><td>35.83%</td><td>39.94%</td></tr></table></td></tr>
<tr><td>December</td>
<td>weighted bias: 1914.136
coverage: 25.46%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.07%</td><td>0.88%</td></tr><tr><td>Precision</td><td>86.02%</td><td>65.94%</td></tr><tr><td>Recall</td><td>15.36%</td><td>19.58%</td></tr><tr><td>F1</td><td>26.07%</td><td>30.19%</td></tr><tr><td>F0.5</td><td>33.96%</td><td>36.85%</td></tr></table></td>
<td>weighted bias: 2572.379
coverage: 27.81%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.07%</td><td>1.14%</td></tr><tr><td>Precision</td><td>87.41%</td><td>71.58%</td></tr><tr><td>Recall</td><td>18.46%</td><td>23.56%</td></tr><tr><td>F1</td><td>30.48%</td><td>35.46%</td></tr><tr><td>F0.5</td><td>38.93%</td><td>42.62%</td></tr></table></td>
<td>weighted bias: 2103.537
coverage: 26.68%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.09%</td><td>0.96%</td></tr><tr><td>Precision</td><td>84.18%</td><td>67.87%</td></tr><tr><td>Recall</td><td>17.18%</td><td>20.10%</td></tr><tr><td>F1</td><td>28.54%</td><td>31.01%</td></tr><tr><td>F0.5</td><td>36.60%</td><td>37.87%</td></tr></table></td></tr>
<tr><td>January</td>
<td>weighted bias: 2014.641
coverage: 21.76%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.16%</td><td>0.83%</td></tr><tr><td>Precision</td><td>75.01%</td><td>64.18%</td></tr><tr><td>Recall</td><td>12.18%</td><td>17.17%</td></tr><tr><td>F1</td><td>20.96%</td><td>27.09%</td></tr><tr><td>F0.5</td><td>27.58%</td><td>33.55%</td></tr></table></td>
<td>weighted bias: 2417.403
coverage: 23.68%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.15%</td><td>0.91%</td></tr><tr><td>Precision</td><td>76.25%</td><td>66.37%</td></tr><tr><td>Recall</td><td>14.75%</td><td>17.62%</td></tr><tr><td>F1</td><td>24.72%</td><td>27.85%</td></tr><tr><td>F0.5</td><td>31.91%</td><td>34.52%</td></tr></table></td>
<td>weighted bias: 3040.636
coverage: 23.11%<table><tr><td></td><td>Low CTR</td><td>High CTR</td></tr><tr><td>CTR</td><td>0.08%</td><td>1.08%</td></tr><tr><td>Precision</td><td>84.94%</td><td>70.12%</td></tr><tr><td>Recall</td><td>14.14%</td><td>21.05%</td></tr><tr><td>F1</td><td>24.24%</td><td>32.38%</td></tr><tr><td>F0.5</td><td>31.82%</td><td>39.45%</td></tr></table></td></tr>
</table>

Conditions:

- Depth 7
- Min coverage 2000 pviews
- Post-filtering with deleting examples and chi^2-filter
- Min bias 0.003 (CTR$_{AVE}$ +/-0.30%)

		Model								
		November			December			January		
		Total rules: 89 Average length: 4.584 Weighted average length: 3.214 Average covering:1.253			Total rules: 82 Average length: 4.183 Weighted average length: 3.168 Average covering:1.399			Total rules: 136 Average length: 4.463 Weighted average length: 3.430 Average covering: 1.466		
Dataset	November	weighted bias: 2496.832 coverage: 26.97%			weighted bias: 1967.072 coverage: 28.45%			weighted bias: 1963.310 coverage: 35.50%		
			Low CTR	High CTR		Low CTR	High CTR		Low CTR	High CTR
		CTR	0.05%	0.96%	CTR	0.11%	0.79%	CTR	0.10%	0.63%
		Precision	89.66%	69.62%	Precision	78.84%	65.26%	Precision	80.17%	59.84%
		Recall	16.44%	24.12%	Recall	19.36%	17.12%	Recall	17.97%	26.16%
		F1	27.34%	29.96%	F1	31.09%	27.12%	F1	29.36%	36.40%
		F0.5	36.09%	42.74%	F0.5	38.96%	33.68%	F0.5	37.22%	41.87%
	December	weighted bias: 1965.536 coverage: 28.93%			weighted bias: 2674.625 coverage: 29.52%			weighted bias: 2161.802 coverage: 30.60%		
			Low CTR	High CTR		Low CTR	High CTR		Low CTR	High CTR
		CTR	0.07%	0.79%	CTR	0.07%	1.10%	CTR	0.11%	0.88%
		Precision	85.98%	63.52%	Precision	87.18%	70.78%	Precision	80.10%	65.93%
		Recall	15.45%	23.49%	Recall	18.95%	25.62%	Recall	18.56%	23.32%
		F1	26.19%	34.30%	F1	31.13%	37.62%	F1	30.14%	34.45%
		F0.5	34.09%	40.51%	F0.5	39.62%	44.58%	F0.5	38.05%	40.97%
	January	weighted bias: 2110.635 coverage: 24.99%			weighted bias: 2406.527 coverage: 24.99%			weighted bias: 3330.873 coverage: 25.99%		
			Low CTR	High CTR		Low CTR	High CTR		Low CTR	High CTR
		CTR	0.15%	0.76%	CTR	0.16%	0.87%	CTR	0.08%	1.04%
		Precision	75.36%	62.19%	Precision	74.86%	65.34%	Precision	84.69%	69.21%
		Recall	12.20%	21.03%	Recall	15.21%	18.46%	Recall	15.07%	24.54%
		F1	21.00%	31.43%	F1	25.28%	28.79%	F1	25.59%	36.23%
		F0.5	27.65%	37.64%	F0.5	32.44%	35.38%	F0.5	33.34%	43.08%

Appendix 7 Contextual Evaluation of the Attributes

Here are the χ^2 tests of attributes in the winter and the summer datasets:

Winter dataset

device	Click	No Click	CTR,%
desktop	25451	6167235	0.4110
mobile	1660	286373	0.5763
tablet	3284	292333	1.1109
tv	1	847	0.1179
unknown	10	759	1.3004
Total	**30406**	**6747547**	**0.4486**

p-value < 2.2e-16

Summer dataset

device	Click	No Click	CTR,%
desktop	29594	5966195	0.4936
tablet	7162	633416	1.1181
mobile	2965	376695	0.7810
tv	4	2199	0.1816
unknown	2	2535	0.0788
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 27 χ^2 test of “device” attribute

Winter dataset

OS	Click	No Click	CTR,%
Windows	24994	6018038	0.4136
Amiga OS	0	2	0
Android	1191	183790	0.6438
Linux	80	40054	0.1993
Macintosh (iPhone)	637	89322	0.7081
Macintosh OS X	3384	387370	0.866
Nintendo Wii	0	90	0
NULL	120	28881	0.4138
Total	**30406**	**6747547**	**0.4486**

p-value < 2.2e-16

Summer dataset

OS	Click	No Click	CTR,%
Windows	29663	5917967	0.4987
Linux	3022	356523	0.8405
Mac	803	191237	0.4181
iPhone/iPod	1028	120332	0.8471
Unknown	5211	394981	1.3021
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 28 χ^2 test of “OS” attribute

Winter dataset

Photo	Click	No Click	CTR,%
0	21260	4964909	0.4264
1	9146	1782638	0.5104
Total	**30406**	**6747547**	**0.4486**

p-value < 2.2e-16

Summer dataset

Photo	Click	No Click	CTR,%
0	27138	5116824	0.5276
1	5116824	1864216	0.6708
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 29 χ^2 test of “photo” attribute

Winter dataset

Video	Click	No Click	CTR,%
0	30359	6734756	0.4488
1	47	12791	0.3661
Total	**30406**	**6747547**	**0.4486**

p-value = 0.1822

Summer dataset

Video	Click	No Click	CTR,%
0	39389	6942924	0.5641
1	338	38116	0.8790
Total	**39727**	**6981040**	**0.5658**

p-value = 2.975e-16

Figure 30 χ^2 test of “video” attribute

Winter dataset

Forum	Click	No Click	CTR,%
0	21423	4909552	0.4345
1	8983	1837995	0.4864
Total	**21423**	**4909552**	**0.4345**

p-value < 2.2e-16

Summer dataset

Forum	Click	No Click	CTR,%
0	26906	5082680	0.5266
1	12821	1898360	0.6708
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 31 χ^2 test of “forum” attribute

Winter dataset

Hours	Click	No Click	CTR,%
evening	10289	2332919	0.4391
lunch	2101	473783	0.4415
morning	1739	373346	0.4636
night	1194	262824	0.4522
work_hours	15083	3304675	0.4543
Total	**30406**	**6747547**	**0.4486**

p-value = 0.04581

Summer dataset

Hours	Click	No Click	CTR,%
morning	2483	439684	0.5616
work_hours	19267	3367828	0.5688
lunch	2805	501095	0.5567
evening	13319	2373290	0.5581
night	1853	299143	0.6156
Total	**39727**	**6981040**	**0.5658**

p-value = 0.001723

Figure 32 χ^2 test of “time of the day” attribute

Winter dataset

Day	Click	No Click	CTR,%
wkend	9124	1879167	0.4832
wrkday	21282	4868380	0.4352
Total	**30406**	**6747547**	**0.4486**

p-value < 2.2e-16

Summer dataset

Day	Click	No Click	CTR,%
wkend	11116	1674921	0.6593
wrkday	28611	5306119	0.5363
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 33 χ^2 test of “day of the week” attribute

Winter dataset

Temperature	Click	No Click	CTR,%
chilly	24672	5419789	0.4532
cold	1960	494851	0.3945
warm	3774	832907	0.4511
Total	**30406**	**6747547**	**0.4486**

p-value = 2.287e-08

Summer dataset

Temperature	Click	No Click	CTR,%
0-9	38	10188	0.3716
10-19	23754	4031898	0.5857
20-29	15123	2816840	0.5340
30+	812	122114	0.6606
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 34 χ^2 test of “temperature” attribute

Winter dataset

Conditions	Click	No Click	CTR,%
fog	1600	400445	0.3980
heavy prec	526	98428	0.5316
light prec	2736	583144	0.4670
no	23895	5337440	0.4457
normal prec	1649	328090	0.5001
Total	**30406**	**6747547**	**0.4486**

p-value = 5.697e-13

Summer dataset

Conditions	Click	No Click	CTR,%
Rain	4417	707817	0.6202
Overcast	21046	3737348	0.5600
normal	14264	2535875	0.5593
Total	**39727**	**6981040**	**0.5658**

p-value = 9.420e-10

Figure 35 χ^2 test of “conditions” attribute

Winter dataset

Pcat	Click	No Click	CTR,%
agenda	87	8289	1.0387
colofon	13	546	2.3256
epaper	17	1490	1.1281
fancy-image	0	2	0.0000
forum	252	90112	0.2789
foto	75	317794	0.0236
koopjes	164	7950	2.0212
login	16	2086	0.7612
miss-uden	31	3193	0.9615
mobiel	5	588	0.8432
nieuws	15037	2978493	0.5023
pagina	96	17290	0.5521
poll	78	4243	1.8051
prive-berichten	9	926	0.9626
registreren	16	2067	0.7681
selecteer-regios	214	40770	0.5222
sport	885	801305	0.1103
unknown	52	7737	0.6676
zoeken	105	42799	0.2447
_start_page	13254	2419867	0.5447
Total	**30406**	**6747547**	**0.4486**

p-value < 2.2e-16

Summer dataset

Pcat	Click	No Click	CTR,%
start_page	17292	2672692	0.6428
pagina	64	14198	0.4487
nieuws	19809	2920784	0.6736
sport	1146	678338	0.1687
foto	211	511435	0.0412
forum	301	65221	0.4594
zoeken	109	32590	0.3333
selecteer-regios	225	37583	0.5951
agenda	101	7379	1.3503
koopjes	178	7645	2.2753
login	26	7580	0.3418
poll	141	7026	1.9674
other	124	18569	0.6633
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16

Figure 36 χ^2 test of “page category” attribute

Summer dataset

Browser	Click	No Click	CTR,%
Chrome	2536	640118	0.3946
Chromium	8	928	0.8547
Firefox 10	5	2596	0.1922
Firefox 11	1	129	0.7692
Firefox 12	0	70	0.0000
Firefox 3	321	87558	0.3653
Firefox 9	439	130170	0.3361
Firefox other	1244	353177	0.3510
MSIE 6	253	71544	0.3524
MSIE 7	3682	942413	0.3892
MSIE 8	9673	2336535	0.4123
MSIE 9	7041	1517962	0.4617
MSIE other	15	699	2.1008
Opera 8	0	3	0.0000
Opera 9	65	17974	0.3603
Opera other	0	4	0.0000
Safari	4977	617498	0.7996
Seamonkey	0	67	0.0000
unknown	146	28102	0.5169
Total	**30406**	**6747547**	**0.4486**

p-value < 2.2e-16

Summer dataset

Browser	Click	No Click	CTR,%
Firefox	2721	558245	0.4851
Explorer	23184	4645972	0.4965
Chrome	4043	782813	0.5138
Opera	109	19303	0.5615
Safari	6784	658384	1.0199
Mozilla	2879	312392	0.9132
An unknown browser	7	3873	0.1804
Netscape	0	54	0.0000
Konqueror	0	1	0.0000
Camino	0	3	0.0000
Total	**39727**	**6981040**	**0.5658**

p-value < 2.2e-16