FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

The Road to Enlightenment: Generating Insight and Predicting Consumer Actions in Digital Markets

Jorge Moreira da Silva

Mestrado Integrado em Engenharia Informática e Computação

Supervisors: Hugo Sereno Ferreira / João Mendes Moreira

Proponent: Rui Gonçalves

July 29, 2014

The Road to Enlightenment: Generating Insight and Predicting Consumer Actions in Digital Markets

Jorge Moreira da Silva

Mestrado Integrado em Engenharia Informática e Computação

July 29, 2014

Abstract

E-commerce platforms are growing without stop nowadays. Lots of scientific research have been employed into the evolution of these platforms, a simple example is the recommendation engines that are now standard in electronic commerce websites.
However, little effort has been made to determinate whether or not a given user is more or less prone to buy a product based on his previous actions.
The goal of this dissertation is to design and develop an automatic approach for identifying user behaviour, predicting their future actions throughout an website. Machine Learning techniques will be applied in order to learn from click-through logs data and predict user profitableness. Results will be studied and benchmarked in order to acknowledge the best approach to identify buying users.

Resumo

Plataformas de comércio digital não param de crescer nos dias de hoje. Muita investigação é feita com o objetivo de evoluir estas plataformas, um simples exemplo disto são os "motores de sugestão" que são padrão das mesmas atualmente.
No entanto, pouco esforço é feito no sentido de se tentar perceber se um utilizador é mais ou menos propício a comprar um produto com base na sua navegação.
O objetivo desta dissertação é desenvolver uma abordagem automática para identificar o comportamento de utilizadores, prevendo as suas ações futuras. Técnicas de *Machine Learning* serão aplicadas a registos de navegação de modo a prever a rentabilidade de utilizadores. Os resultados serão estudados e comparados de forma a perceber qual a melhor abordagem para identificar utilizadores compradores.

Acknowledgements

First of all, I am grateful to my family for everything, without them I wouldn't be who I am today.
I'd like to express my gratitude towards my supervisors Hugo Sereno Ferreira and João Mendes Moreira for all the support and guidance, I have also to give a special thanks to Rui Gonçalves who was a great help through this work.
In the end I have to place on record my thanks to everyone I've had the pleasure of studying with during these past five years, I've been happy.

Jorge Silva

"Start where you are. Use what you have. Do what you can."

Arthur Ashe

Contents

List of Figures

LIST OF FIGURES

Abbreviations

ML	Machine Learning
ANN	Artificial Neural Network
MAP	Mean Average Precision

Chapter 1

Introduction

1.1 Context and Framing

E-commerce platforms are growing non-stop nowadays. Lots of scientific research have been employed into the evolution of e-commerce platforms, a simple example is the recommendation engines that are now standard in most e-commerce websites. However, little to no effort has been made to determinate whether or not a given user is more or less prone to buy a product based on his previous actions. The importance of the use of Machine learning techniques in order to analyse user behaviour and support the creation of user models has been rising since the early nineties, with the massification of the internet services usage. However, despite all the interest and demand for this task, there is no major worldwide adoption of such a system. This is due some common issues that need to be overcomed to archive the desired results, such as the need for large and labeled data, concept drift and computational complexity. While the difficulty of these problems should not be underestimated, several approaches have been developed and strong progress has been made.[WPB01]

1.2 Motivation and Goals

The value of digital marketing is directly related to the influence it has in leading the viewer to perform a given action, such as buying a product or registering in a website. These actions are called conversions.

The digital marketers' objective should be to focus publicity to users that, while receptive to such marketing, have not yet made a decision. Given a user navigation path, it should be possible to predict its future intentions, whether or not the user is interested in buying or just scraping prices.

If a trustable automatic approach to understand the profile of an user and predict his actions is successfully implemented, the consequences on the e-commerce environment have the potential to be huge. For a chance, specified publicity or products recommendation can be accurately focused to the right users, this will produce a significant improvement of the platform profit performance as well as the users experience and satisfaction. Other than this, publicity targeted to the most valuable users might be sold at higher values than regular.

The goal of this dissertation is to design and develop an automatic approach for identifying user behaviour, predicting their future actions throughout an website.

In order to archive this goal, machine learning techniques will be applied in order to learn from click-through logs data and infer the users tendency to buy a product or archive a certain goal. A probabilistic model for ranking will be generated from the training data, this model will be capable to predict a given user' future behaviour.

1.3 Report Structure

Besides the introduction, this dissertation has three more chapters.
In chapter 3, the state of the art will be presented, with a broad presentation of the most viable Machine Learning techniques.
In chapter 2 the problem is described in depth.
During chapter 4 the approach to find a solution to the given problem and archive the desired results is explained.
In chapter 5 the resources available will be described, the obtained results will be presented and the document finalized with a conclusion.

Chapter 2

Problem Description

In a e-Commerce environment the knowledge gathered about its userbase might be the key to unveil new potential buyers. This work aims to develop a predictive model capable of classifying a new given user on whether they will be buyers or not based on knowledge learned from the actions of existent users in the system.

Given a log of user actions with several entries describing the action performed by the user and respective timestamp, data will be interpreted and parsed in order to extract and generate relevant variables to understand if an user will archive a conversion.

Experiments will be made using several classifying algorithms as well as machine learning techniques such as cost sensitive learning and automatic feature selection in order to sharpen results. Different exploratory setups will be tested and improvements will be made after relevant insight has been gathered.

By the end of this work conclusions will be made on which variables hold the most predicative potential and which of the explored algorithms are able to archive the best results.

Problem Description

Chapter 3

State-of-the-Art

This dissertation work will be focused on predicting the behaviour and future actions of a given user using data mining techniques to learn from his past actions. These techniques typically work by generating a predictive model given training data, the generated model is then capable of classifying new data.
In this case, the model will be learning from a set of users and their actions/outcomes, being able to classify any user according to how close he is to archive a conversion afterwards.

This chapter will feature the state of the art, exploiting the work existent in this area until this point, a broad explanation of the learning approaches, followed by a specification on how regression and classification algorithms act. By the end of the chapter, instance ranking algorithms will be approached as well as model validation techniques and machine learning tools.

3.0.1 Digital Marketing and e-Commerce

A conversion in the context of online commerce is used to describe the act of a visitor actually spending money on the site, therefore conversion rate is the amount offering buyers among the total of visitors.

The goal of any e-commerce website is always to have a higher conversion rate which means more profit.

$$ConversionRate = \frac{Number\,of\,Goals\,Achieved}{Total\,Visits} \tag{3.1}$$

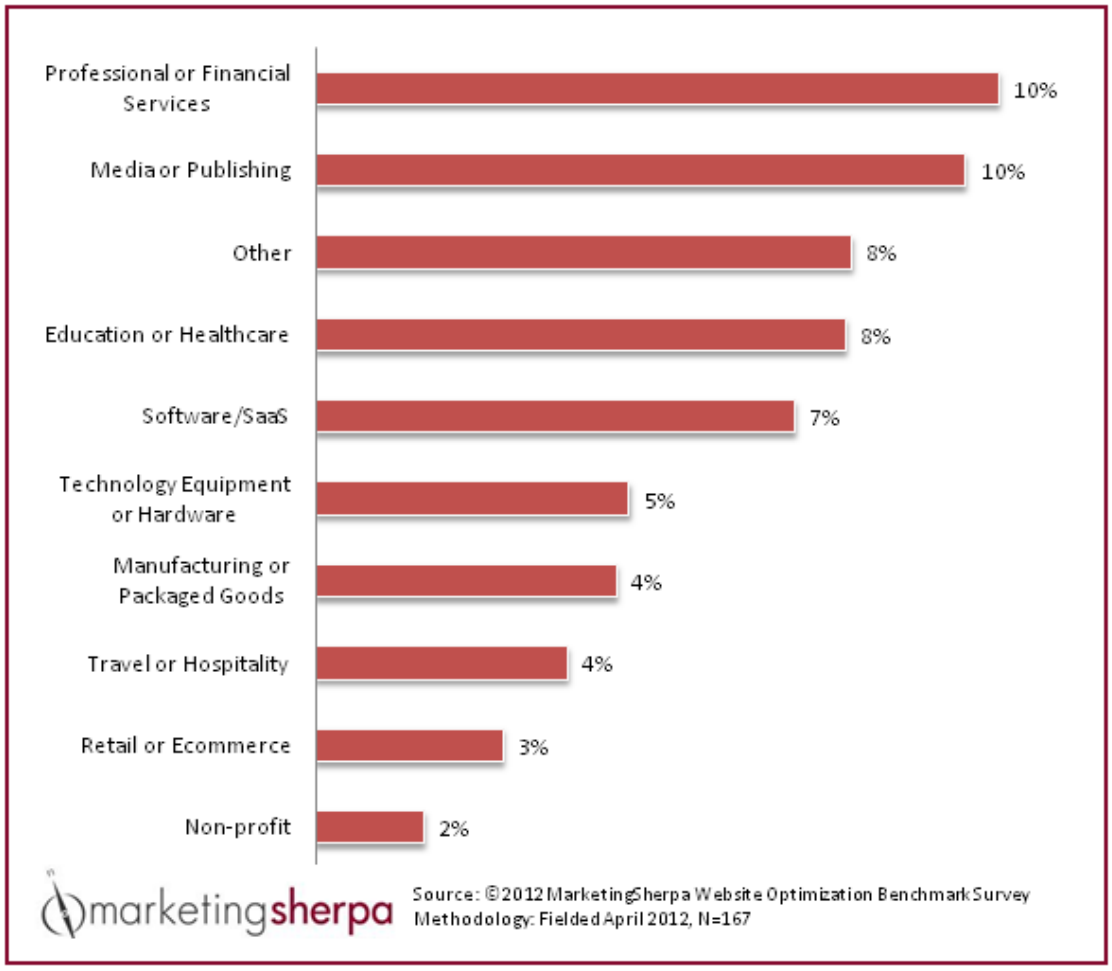

Figure 3.1: Average conversion rates per industry [CRa]

As can be seen from the chart above, the eCommerce sector counts typically with a conversion rate of around 3%. Such a low percentage conversion rate is evidence that the majority of the users don't actually buy anything.

3.1 Related Work

Nowadays with the growing Internet presence and influence as a source of information, it is begging to be the default place for several markets and social interactions. This has sparkled a growing interest in the study of what people actually do online and how their behaviour can be predicted and influenced. A good understanding of users online behavior has become a core need for online businesses striving for survival in the environment in which they compete. [BS09]

Finding or inducing preferences and patterns in user behaviour to carry out a certain task requires a lot of study since each person has their own way to deal with a given situation. The capability of identifying and collecting the information that can characterize a user profile is a crucial step to generate an approach which automatically predicts future user behaviour. [LMJ]

The ability to tell whether a (potential) customer will engage in online-purchasing behaviour during his next visit to the website provides a powerful predictive tool for electronic marketeers that helps them in inferring the goal of their visitors and, consequently, improve their targeting. This is considered to be among the most important steps to improve online conversion rates. [dPB05]

3.1.1 Search Engines

Lots of effort in this area has been targeted to search engines which try to retrieve the most relevant information in the actions of each user. A search engine that finding patterns in user actions and predicts their intentions is very useful so that the user gets exactly what he wants as fast as possible, increasing the search engine accuracy and competitive edge. [AWBG07] [RSD+12]

3.1.2 Behaviour Prediction

Costumer behaviour analysis and prediction might be done in many different ways, having different focus and accounting diverse features. Studies have been conducted evaluating the predictive power of various variables such as:[dPB05]

- Session frequency
- Timing
- Recency
- Time spent
- Number of pages visited
- Viewed content
- Demographics

Some of the variables that have been considered relevant:

- Total number of past visits
- Number of days since last visit
- Total past visit time
- The visit time of the last session
- Total number of clicks in the past
- Average time per click
- Average number of clicks in a session
- Total number of products viewed
- Total number of purchases ever did at the website

3.1.3 Clickstream Data

Clickstream data is the collected digital record of a given platform usage through time .

In the challenge of predicting online purchases most of the information available is clickstream data with the information about what was selected through the navigation on the website and respective timestamps. With this information it should possible to establish a connection between a user purchase and his previous behaviour. Results indicate that the number of visits is not diagnostic of buying propensity and that a site's offering of sophisticated decision aids does not guarantee an increase to the conversion rates. [SB04]

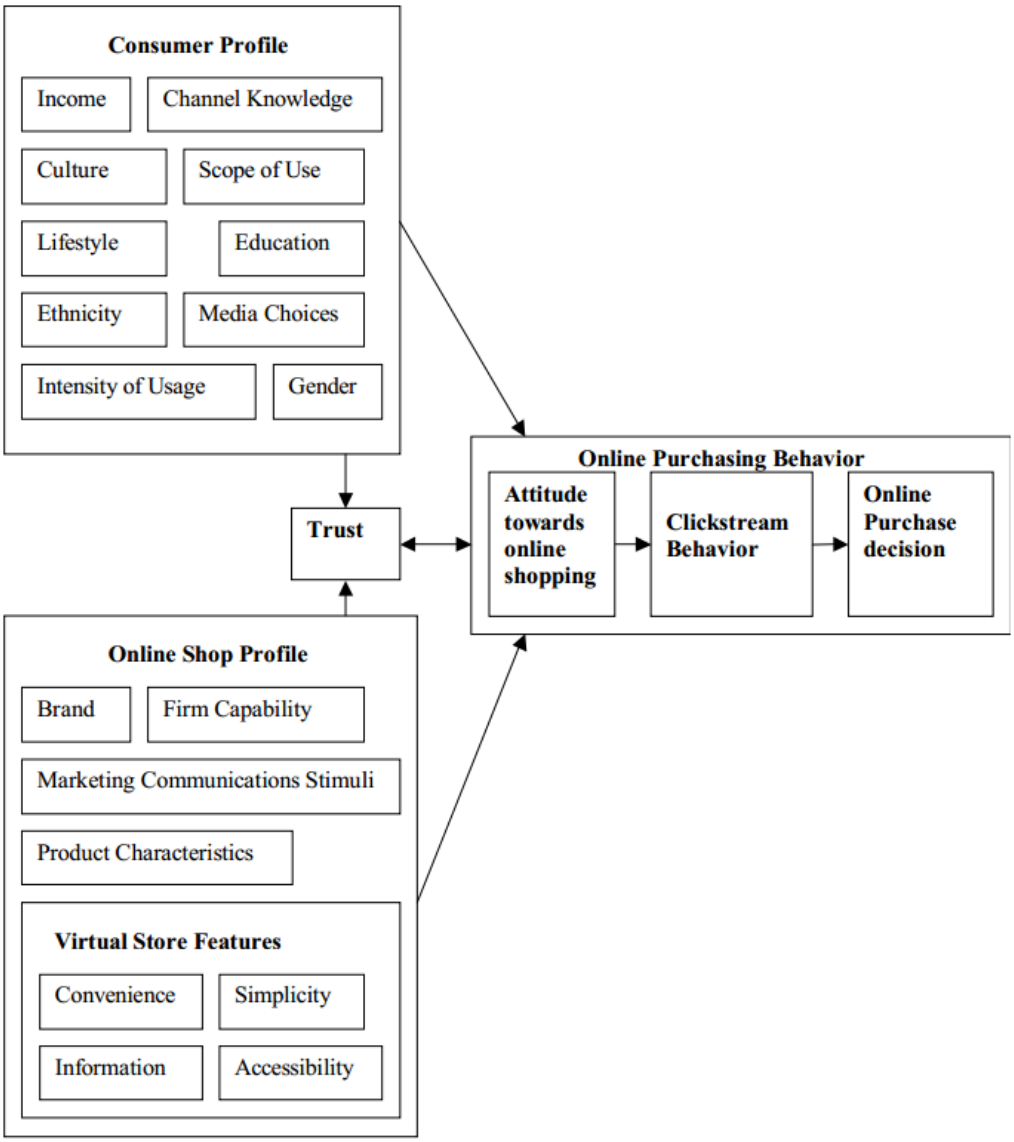

Figure 3.2: Theoretical model of online consumer behaviour [KP06]

3.2 Machine Learning

Machine Learning is an application of artificial intelligence in which large amounts of data are studied by computers in order to learn new information and patterns unfindable by human perception. It relies in several techniques to interpret, manipulate, predict and learn from data.

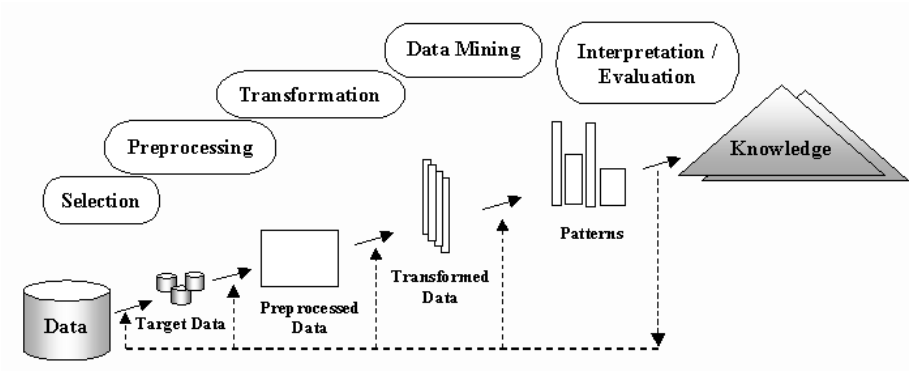

Figure 3.3: Machine Learning Process [DMP]

3.3 Algorithm Types

3.3.1 Unsupervised Learning

Unsupervised learning consists in finding hidden structure in unlabeled data. This kind of learning approach can be used for classifying groups of genes based on some characteristic, separating different types of articles/news on a content aggregator website. In the context of this dissertation, this kind of algorithms could be useful to trim users according to their buying preferences or behaviour, discovering different market segments. Clustering and hidden Markov models are some of the most commonly used Unsupervised Learning methods.[Gha04]

3.3.2 Supervised Learning

Supervised Learning is probably the most common practice in Machine Learning, this kind of algorithm, such as any Unsupervised Learning algorithm, tries to find hidden patterns in the data, the difference here is that this type of approach learns from labeled data in the training set, taking certain features in consideration before predicting. For example if an e-mail spam filter were to be implemented, some importantant features to analyse would probably be things like: "does the the sender of the email appear in the user contacts?", "do words like "viagra", "free", "money", or any common word used in e-mails labeled as spam pop up in the e-mail text?".[V.C]
This kind of problems is typically studied and developed using techniques like regression, classification, Support Vector Machines or Artificial Neural Networks.[Kot07]

3.3.3 Classification

Classification is the problem of identifying which of a set of categories a given observant belongs to. In this case, it might be whether or not a user is a buyer, or if he is interested in a given product.

3.3.3.1 Naive Bayes

Naive Bayes classifiers are simple probabilistic classifiers based on bayes' theorem and some of the most popular and common classifier algorithms.
A naive bayesian classifier basically assumes that the value of any feature is unrelated to the value of any other feature, given the predictor class.
It is a popular baseline method for text categorization or the problem of classifying documents according to its topic. With appropriate preprocessing it is competitive in this domain with more advanced methods like SVMs. [RSTK03]

3.3.3.2 Bayesian Network

A Bayesian network, is a probabilistic statistical model that represents a set of random variables and their conditional dependencies via a directed acyclic graph (DAG).

3.3.3.3 REPTree

Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting). Only sorts values for numeric attributes once. Missing values are dealt with by splitting the corresponding instances into pieces (i.e. as in C4.5).[wek]

3.3.3.4 Random Forest

Random Forests are an ensemble learning method which generates the model by constructing several trees and gathering their individual output. [Bre01]

The training algorithm for random forests applies the technique of bagging to tree learners, from a training set bagging repeatedly selects a bootstrap sample of the training set and fits trees to these samples.

Bagging (Bootstrap Aggregating) is applied to improve stability and accuracy of ML algorithms, it also reduces variance and helps to avoid overfitting. The basic idea of bootstrapping is that inference about a population from sample data can be modelled by re-sampling the sample data and re-performing inference.

Random Forests also use a modified tree learning algorithm that selects, for each candidate split in the learning process, a random subset of the features. The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the predictor variable , these features will be selected in many of the trees, causing them to become correlated.

[JWHT13]

3.3.3.5 Artificial Neural Network

There is no single formal definition of what an artificial neural network is. Commonly, a class of statistical models may be called "neural" if they consist of sets of neurons connected with adaptive weights, tuned by a learning algorithm and are capable of approximating non-linear functions of their inputs. This kind of algorithm can be used either in Supervised or Unsupervised Learning problems.

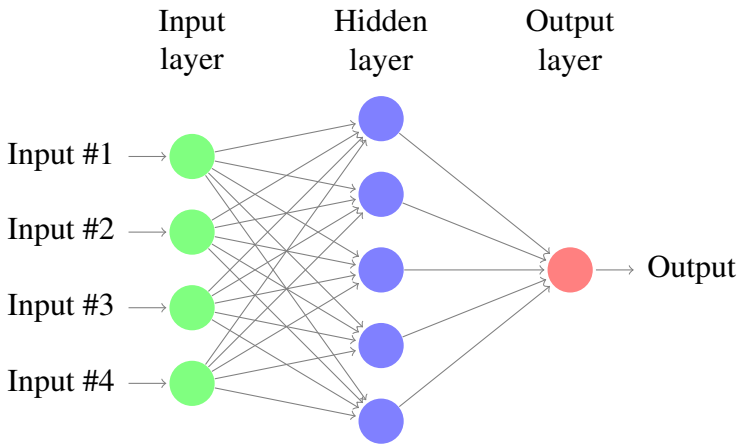

Figure 3.4: Artificial Neural Network example[AAN]

An ANN is typically defined by three types of parameters:

- The interconnection pattern between the different layers
- The learning process for updating the weights of the interconnections
- The activation function that converts a neuron's weighted input to its output activation

The training of a neural network model is an important step in order to have accurate results, there are several algorithms capable of this task, the most common implementations are straightforward applications of a mathematical optimization, which consists of maximizing or minimizing a real function with input values from the data set.

3.3.4 Regression

Regression analysis is a statistical process for estimating relationships between variables, more specifically a regression analysis helps with the study of the behaviour of a dependent variable given the variation of independent ones, therefore aiming to understanding certain patterns and behaviours in the data. The output of a regression analysis is a continuous value, while the output of a classification is a discrete value - a label (class) from a finite set.

3.4 Model Validation

In order to test and compare each model's predictive performance, there is the need to measure the results given by each algorithm.

3.4.1 Measures

3.4.1.1 F-score

The F-score is a measure of accuracy in statistical analysis of binary classification, it considers both the Precision (p) and the Recall (r) in order to compute a score:

$$p = \frac{number\ of\ correct\ results}{number\ of\ all\ returned\ results}$$

$$r = \frac{number\ of\ correct\ results}{the\ number\ of\ results\ that\ should\ have\ been\ returned}$$

$$F\text{-}score = \frac{2 * p * r}{p + r}$$

Precision (p) is the fraction of the results that were actually predicted correctly, this can be useful to verify which algorithm is more "trustable", since higher the precision is, higher the percentage of true positives and lower the amount of false positives.

Recall (r) is the fraction of relevant results that were successfully found. In the context of this dissertation this value might be useful to perceive the probability of a said algorithm to perform right and retrieve the right users.

3.4.1.2 Mean Average Precision (MAP)

By computing a precision and recall at every position in the ranked sequence of results, it is possible to obtain a precision-recall curve. plotting precision $p(r)$ as a function of recall r. Average precision (*AveP*) is the average value of $p(r)$ from $r = 0$ to $r = 1$:

$AveP = \int_0^1 p(r)dr$

Mean average precision is basically the average of the average precisions of all results:

$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$

Where Q is the total number of results and $AveP(q)$ is the average precision of a given result.

3.4.2 Techniques

3.4.2.1 Cross-Validation

Cross-Validation is a model validation technique to study how accurately a predictive model will perform in practice, assessing how the results of a statistical analysis will generalize to an independent data set. One round of cross-validation involves partitioning the dataset into several subsets, then using some subsets to train the model and the remaining ones to test its performance. In order to improve the evaluation of the model, several rounds of cross-validation should be performed, using different partitions of the dataset.[Koh95]

3.4.2.2 K-fold Cross-Validation

In K-fold Cross-Validation, the original dataset is sampled into k equal size subsets, of the k generated sets, one will be chosen for testing and the other $k - 1$ subsets will be used for training the model. This process will be repeated k times in which each subsample will be chosen for testing once.

3.4.2.3 Test File

Testing the a model performance can be done by simply trying to predict information from a new set of instances, this is useful when there is an abundance of data and it is better to use different data instead of testing with selections of the training set.

3.5 Tools and Programming languages

3.5.1 WEKA

Waikato Environment for Knowledge Analysis (WEKA) is a workbench which contains a collection of visualization tools and algorithms for data analysis and predictive modelling. Developed by the University of Waikato, New Zealand, is a free software available under the GNU General Public License.

3.5.2 R

Product of a GNU project, R is a free programming language and software environment for statistical computing and graphics, it's an implementation of the S programming language combined with lexical scoping semantics inspired by Scheme. It is widely used for data mining purposes.

3.5.3 RapidMiner

One of the most popular software for data analysis, RapidMiner provides an integrated environment for machine learning, data mining, text mining, predictive and business analytics. It is used for business and industrial applications as well as for research, education, training, rapid prototyping, and application development and supports all steps of the data mining process including results visualization, validation and optimization.[HK13]

3.6 Conclusion

During this chapter the basics of Machine Learning were explored. Different approaches to learning how to learn from data have been presented, regression and classification have been explained. Numerous algorithms have been listed, also some model validation techniques are described, as well as the tools that will be used thought this dissertation.

After the dataset has been properly analysed, the correct ML algorithms have been applied and the results verified and studied, it should be possible to assert an estimation on a given user's impulse to buy, the output should be a percentage value so that it is easily perceived.

Limitations might come up, being from the lack of data, the performance of the algorithms, or any other issue during the work of this dissertation, these obstacles need to be overcomed and will be part of the study of the subject.

Chapter 4

Approach

4.1 Introduction

In this chapter the approach used to study the problem of learning from user actions and classifying them on whether they will be buyers or not will be presented. There will be a broad overview of the tool used during the development of this work, an explanation of the data selections made from the temporal analysis of the dataset. Feature Selection, Cost Sensitive Learning will be explained, as well as the techniques used to validate the predictive models.

4.2 Tools

The tools used for archiving this work were the WEKA framework which contains implementations of the used classification algorithms (Naive Bayes, BayesNet, Random Forest and REPtree, mentioned on chaper 3). The file parsers created during the work development were implemented in Java and the auxiliary database was created in sqlite3.

4.3 Dataset Parsing

In order to have predictive models trained and tested, files with the relevant data must be created. There is therefore the need to parse the raw dataset in order to easily manipulate the data and construct the desired files.
In the dataset presented some irregularities with the representation of information were found. Sometimes there were multiple actions represented in a single **info={...}**, also usually **""** are the characters used to delimit strings contained in **info** there are however cases in which **”** is used. Other kinds of errors showed up such as this:

```
001634DB14972865 12/08/2013 10:23:56 Demographics  184879931203 info="{'
    productid':'5700381683117','productname':'Mustang 28\" damecykel Model
     Dagmar - creme','categoryname':'ukendt','categoryid':'ukendt','step
    ':'adf.Steps.Purchase'}"
```

where we have ïn the middle of the string with the purpose of meaning "inches", however this character easily causes the JSON parser to choke on errors. in order to solve this, regular expressions were implemented in order to sanitize the input.

After being able to parse the raw dataset, the information was inserted into a SQL database in order to be able to easily retrieve specific queries from the data. The database scheme used:

- **userid** text
- **time** datetime
- **productid** text
- **productname** text
- **categoryid** text
- **categoryname** text
- **step** text
- **hour** integer
- **weekday** integer

4.4 Temporal Sliding Window

In order to emulate a real running environment, the data was separated by sliding windows, each one having a **training period** (past), a **cooldown period** (present) and a **prediction period** (future). The sliding window moves the length of the prediction period every iteration.

The model will be trained with the information gathered in the training period about each user's views, basket actions and categories viewed, having several features extracted from this period, the goal is to predict purchases in the prediction period. The cooldown period represents the "present" period during which the algorithm is running and it might be possible to focus advertisement efforts, being them simply suggestions in the website or a email remainder about some product or promotion of interest in order to persuade the user to buy.

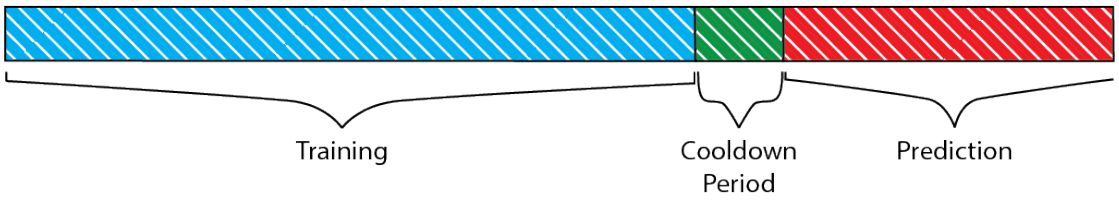

Figure 4.1: Sliding Window Division

4.5 Feature Selection

For the machine learning process to work properly, the most relevant features in the data to archive the desired goal must be acknowledged. In order to do this, after each model construction and testing the results should be studied and interpreted given the data understanding, learned insights must be used to extract new features from the data.
In the early phases of this work, the features selected to train the algorithms were quite simple, the attributes gathered for each given user per sliding window were :

- The amount of views the user made during the training period
- The amount of items added to the basket by the user during the training period
- The amount of categories viewed by the user during the training period
- The total amount of actions during the training period
- Whether or not the user bought anything during the prediction period (1 or 0)

The data was processed by the algorithms and the yielded results studied, given the unsatisfactory results, new features were generated from the data and added to the training data to improve the outcome:

- The amount of views in the last 24 hours before the cooldown period
- The amount of views in the last 48 hours before the cooldown period
- The amount of views in the last 72 hours before the cooldown period
- The amount of items added to the basket by the user 24 hours before the cooldown period
- The amount of items added to the basket by the user 48 hours before the cooldown period
- The amount of items added to the basket by the user 72 hours before the cooldown period

Still not happy with he results, more parameters were created. This time trying to establish a pattern between habits of the user and the rush to buy by analysing the activity of each weekday and each hour of the day:

- The amount of views the user made each weekday during the training period (x7)
- The amount of views the user made each hour of the day during the training period (x24)
- The amount of items added to the basket by the user each weekday during the training period (x7)
- The amount of items added to the basket by the user each hour of the day during the training period (x24)

There are available in WEKA automatic approaches to retrieve a more relevant subset of features to the predictor class. The algorithms of choice were:

CfsSubsetEval Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred.

BestFirst : Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).

4.6 Cost Sensitive Learning

A classifier is trained from a set of training examples with class labels which are discrete and finite, and can be used to predict the class labels from new examples. However, most original classification algorithms pursue to minimize the error rate: the percentage of the incorrect prediction of class labels. They ignore the difference between types of misclassification errors, they implicitly assume that all misclassification errors cost equally. In Cost Sensitive Learning, the misclassification errors costs are taken into consideration, being the goal to minimize the total cost. Therefore, in order to minimize a given error its cost must be increased.[SW10]

Classified as –>	Negative (Non-Buyer)	Positive (Buyer)
Actual Negative (Non-Buyer)	True Negative	False Positive
Actual Positive (Buyer)	False Negative	True Positive

Table 4.1: Confusion Matrix

In the context of this dissertation, we aim to detect the biggest amount of buyers while maintaining consistent predictions, therefore in order to obtain the minimum amount of unidentified buyers (buyers classified as non-buyers), the error that should be minimized is the False Negative error.

By raising the cost of a given error this error will happen less, this however is a trade-off because a unbalanced matrix will make other kinds of error rise, the key is to find an equilibrium in the cost matrix so that the desired results are archived.

4.7 Model Validation

With a large dataset of users and their given actions, a lot of approaches might be used to test the generated model, such as cross validation or percentage split, however these methods will use actions of the same user to train and test, this will stain the results. Therefore the gathered data will be separated having 2/3 of each kind of user (buyer / non buyer) to train the model and the rest for testing.

This ML problem is unbalanced because there is a disproportional amount of non buyers to buyers, therefore some statistics that could be useful to validate a common model such as the percentage of correctly classified instances, might not be that useful for this case. If a model classifies all the users as non buyers it would have a very high percentage of correctly classified instances, however it would not serve the goal of identifying buyers.

Precision (p) is the fraction of the results that were actually predicted correctly, this can be useful to verify which models are more "trustable", since higher the precision is, higher the percentage of true positives and lower the amount of false positives.

Recall (r) is the fraction of relevant results that were successfully found. In the context of this dissertation this value is useful to perceive the probability of a said algorithm to perform accordingly and retrieve most of the actual buyers

Being that the goal is to identify users who are more prone to buy, recall will give us information about of how many of the total buyers were detected. Therefore, since we are aiming for detecting the maximum amount of buyers, results with high recall are good, there is however the need to take care about precision values when using cost sensitive learning, for example if all of the users were classified as buyers recall would have a value of 1 but precision would be something alike 0. We are then trying to detect the maximum amount of the buyers without generating too many false positives, in order to archive good results, there should be an equilibrium between high recall and a good precision.

Approach

Chapter 5

Data and Experimentation

5.1 The Dataset

Given a log of user data containing information about each user action in the website, the goal is to apply machine learning techniques in order to generate a prediction model capable of classifying a new user as buyer or non-buyer based on their previous actions. This log contains several entries with the timestamp of the action, respective user ID and action description. This information after analysed and processed accordingly will be the key to retrieve relevant features to create the predictive model.

5.1.1 Data Analysis

The data of the user actions used in this dissertation is represented in a log file by this kind of syntax:

```
00003870D3D023D7  05/13/2013 14:39:46     info={"productid":"699965011185"
    ,"productname":"Product9","categoryid":"8c944d7c-1999-4f17-9a60-0297327
    c7d95","categoryname":"Brcdristere","step":"adf.Steps.View"}

00003870D3D023D7  05/27/2013 08:25:49     info={"productid":"8710103552277
    ","productname":"Productf","categoryid":"ukendt","categoryname":"ukendt
    ","step":"adf.Steps.Basket"}

0001A1DF3F76ABAB  11/06/2013 14:01:31     info=undefined

0001A1DF3F76ABAB  11/06/2013 14:01:35     info="{'productid
    ':'7340011412243','step':'adf.Steps.Category'},{'productid
    ':'5700380657546','step':'adf.Steps.Category'},{'productid
    ':'3519280012681','step':'adf.Steps.Category'},{'productid
    ':'5700382379880','step':'adf.Steps.Category'},{'productid
    ':'619743724243','step':'adf.Steps.Category'},{'productid
    ':'8710103268192','step':'adf.Steps.Category'}"
```

The first part of each line (e.g. **00003870D3D023D7**) is the ID of the user doing the action, next is the date of the given action (e.g. **05/13/2013 14:39:46**) and following **info={...}** is basicaly a json representation of the information of the action including the product ID, the product name, category ID and the type of action performed:

- "adf.Steps.View" - Represents the action of viewing a certain item
- "adf.Steps.Basket" - Represents the action of adding a certain item to the basket
- "adf.Steps.Category" - Represents the action of viewing a certain category of products
- "adf.Steps.Purchase" - Represents the action of purchasing a certain item

5.1.2 Statistical Analysis

After parsing, the generated database has a total of 20327500 entries, with 4362250 product views, 140586 baskets, 15473906 category views and 23154 purchases.
With a total of 923779 users each user averages 5.8973 product views, 2.2847 different categories seen, 3.3445 items added to the basket and 1.4503 purchases, which make an average of 22.0047 pages seen per user.
Having 15965 buyers from the total of 923779 implies that the current purchase conversion rate of the website is 1.728% . This value is inferior to the 3% average conversion rate of most e-Commerce websites found in the statistical survey from figure 3.0.1 on page 6.

5.2 Experimentation

As said before in this report, from various techniques to test models the elected one was the construction of a auxiliary test file, having 1/3 of the users, the other 2/3 of the users is used for learning. This makes it so that the same user is never used for learning and testing at the same time avoiding stained results.

5.2.1 Tools

All the results present in this chapter were obtained using the WEKA framework and its implementation of some classifier algorithms and other tools:

- Naive Bayes
- BayesNet
- REPTree
- Random Forest
- Cost Sensitive Learning
- Attribute Evaluation and Selection

The machine in which the tests ran has the following configuration:

- Processor: Intel(R) Core(TM) i7-2670QM @ 2.2GHz
- Ram: 6 Gb
- Hard Drive:WD5000BPVT 500GB 5400 RPM 8MB Cache SATA 3.0Gb/s
- Operative System: Windows 8.1 Pro 64-bit

The graphics shown have been generated using R (Weka and R have been mentioned in 3.5).

5.2.2 Baseline

In the early phases simple features were selected, the objective at the time is to have a baseline of results to study and improve. Given this, a simple selection of feature was made, models were trained and tested.

The selected features were:

- number of product views
- number of items added to the basket
- number of different categories seen by the user
- the total amount of actions
- purchases

At this time, the values for the temporal sliding window are fixed with:

- 2 weeks of training period
- 1 hour of cooldown period
- 1 week of prediction period

Variations to the temporal window setup will be made after a selection of relevant features is archived.

5.2.2.1 Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,996	0,941	0,999	0,996	0,997	0,025	0,624	0,999	0
0,059	0,004	0,012	0,059	0,020	0,025	0,624	0,003	1
0,995	0,940	0,998	0,995	0,997	0,025	0,624	0,998	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1236356	5308	a = 0
1016	64	b = 1

5.2.2.2 BayesNet

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	1,000	0,000	0,738	1,000	0
0,000	0,000	0,000	0,000	0,000	0,000	0,738	0,005	1
0,999	0,999	0,998	0,999	0,999	0,000	0,738	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1241664	0	a = 0
1080	0	b = 1

5.2.2.3 Random Forest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	1,000	-0,000	0,697	0,999	0
0,000	0,000	0,000	0,000	0,000	-0,000	0,697	0,004	1
0,999	0,999	0,998	0,999	0,999	-0,000	0,697	0,999	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1241586	78	a = 0
1080	0	b = 1

5.2.2.4 REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	1,000	0,000	0,500	0,999	0
0,000	0,000	0,000	0,000	0,000	0,000	0,500	0,001	1
0,999	0,999	0,998	0,999	0,999	0,000	0,500	0,998	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1241664	0	a = 0
1080	0	b = 1

5.2.2.5 Conclusions

As it can be easily perceived, the algorithm predictions were not very effective at this stage, as almost none retrieved relevant results, the only who identified any buyer was the naive bayes and it only found 64 out of the 1080 buyers. This is ok since it was the first batch of results, now we have a baseline to improve.

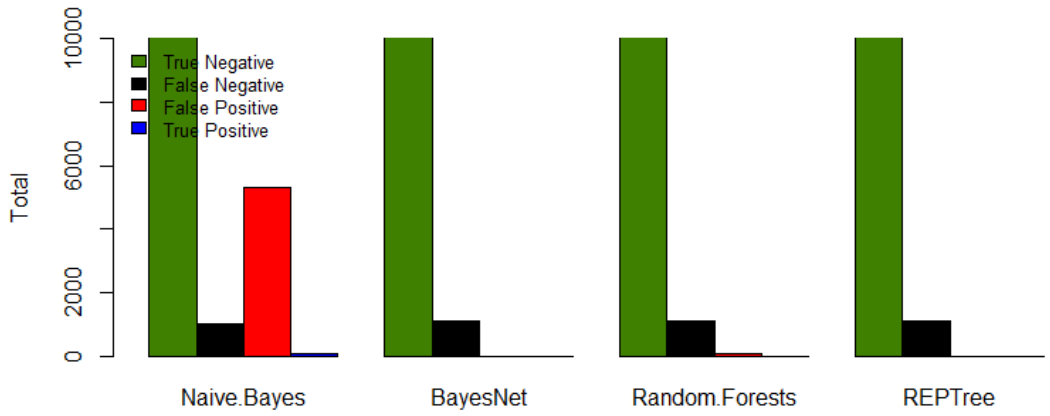

Figure 5.1: Baseline results

5.2.3 Adding new features - recent days analysis

Since the bad results with with first set of features, new features were extracted from the data, in this case there were added variables with information about the most recent days of usage, analysing the last 24, 48 and 72 hours of usage before the cooldown period. This was done because recent actions might be more strongly co-related to the near future.

The same algorithms were executed and the following features were added:

- views24 - amount of product views in the last 24 hours
- views48 - amount of product views in the last 48 hours
- views72 - amount of product views in the last 72 hours
- baskets24 - amount of products added to the cart in the last 24 hours
- baskets48 - amount of products added to the cart in the last 48 hours
- baskets72 - amount of products added to the cart in the last 72 hours

5.2.3.1 Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,994	0,901	0,999	0,994	0,996	0,034	0,638	0,999	0
0,099	0,006	0,013	0,099	0,024	0,034	0,638	0,004	1
0,993	0,900	0,998	0,993	0,996	0,034	0,638	0,999	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1233824	7840	a = 0
973	107	b = 1

5.2.3.2 BayesNet

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,994	0,857	0,999	0,994	0,997	0,051	0,759	1,000	0
0,143	0,006	0,020	0,143	0,035	0,051	0,759	0,006	1
0,993	0,857	0,998	0,993	0,996	0,051	0,759	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1234126	7538	a = 0
926	154	b = 1

5.2.3.3 Random Forest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	0,999	0,999	1,000	1,000	0,002	0,623	0,999	0
0,001	0,000	0,007	0,001	0,002	0,002	0,623	0,004	1
0,999	0,998	0,998	0,999	0,999	0,002	0,623	0,998	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1241514	150	a = 0
1079	1	b = 1

5.2.3.4 REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	1,000	0,000	0,500	0,999	0
0,000	0,000	0,000	0,000	0,000	0,000	0,500	0,001	1
0,999	0,999	0,998	0,999	0,999	0,000	0,500	0,998	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1241664	0	a = 0
1080	0	b = 1

5.2.3.5 Conclusions

In this iteration of results the algorithm which performed the best was the Bayesian Network, and the Naive Bayes results improved, either way results are still not identifying enough buyers. Being that the Tree based algorithms have been failed continuously, still not detecting buyers (Random Forests actually identified one buyer, but it's irrelevant). Overall results have improved, which means recent actions are actually more relevant to predict a near future.

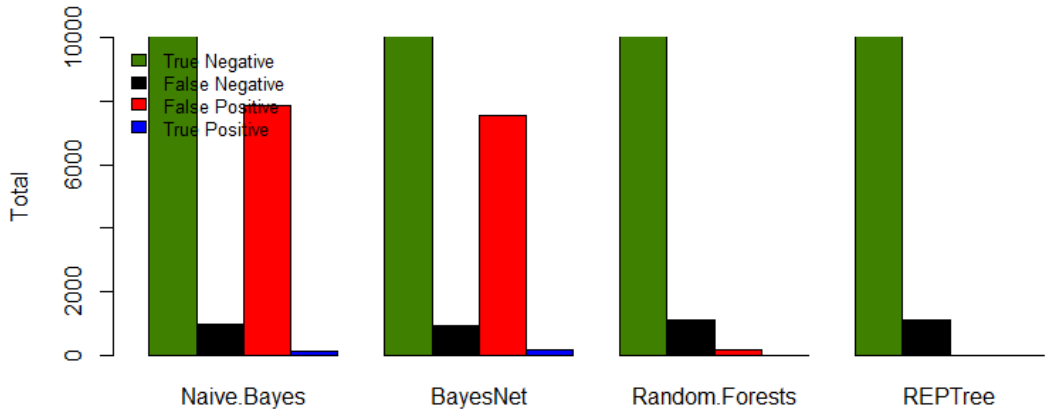

Figure 5.2: Recent days analysis

5.2.4 Adding new features - weekly and hourly analysis

For this round of results there will be added time analysis of the user actions related to their habits, namely the amount of views and baskets actions per hour of each day and day of the week.

The following features were added:

- views(0-23)h - total of product views per hour of the day, 24 total variables (views0h, views1h, ... , views23h)
- baskets(0-23)h - total of products added to the basket per hour of the day, 24 total variables (baskets0h, baskets1h, ... , baskets23h)
- views(0-6)w - total of product views per day of the week, 7 total variables (views0w, views1w, ... , views6w)
- baskets(0-6)w - total of products added to the basket per day of the week, 7 total variables (baskets0w, baskets1w, ... , baskets6w)

5.2.4.1 Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,982	0,820	0,999	0,982	0,991	0,036	0,643	0,999	0
0,180	0,018	0,009	0,180	0,017	0,036	0,643	0,004	1
0,981	0,820	0,998	0,981	0,990	0,036	0,643	0,999	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1219522	22142	a = 0
886	194	b = 1

5.2.4.2 BayesNet

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,974	0,709	0,999	0,974	0,986	0,048	0,774	1,000	0
0,291	0,026	0,009	0,291	0,018	0,048	0,774	0,009	1
0,973	0,709	0,999	0,973	0,985	0,048	0,774	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1208887	32777	a = 0
766	314	b = 1

5.2.4.3 Random Forest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	0,999	-0,000	0,467	0,999	0
0,000	0,000	0,000	0,000	0,000	-0,000	0,467	0,002	1
0,999	0,999	0,998	0,999	0,999	-0,000	0,467	0,998	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1241416	248	a = 0
1080	0	b = 1

5.2.4.4 REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	1,000	-0,000	0,652	0,999	0
0,000	0,000	0,000	0,000	0,000	-0,000	0,652	0,005	1
0,999	0,999	0,998	0,999	0,999	-0,000	0,652	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1241655	9	a = 0
1080	0	b = 1

5.2.4.5 Conclusions

Results have improved somewhat, tree algorithms still unable to predict purchases. One can therefore say that the habits analysis has improved results, however with such a big set of features maybe the most relevant ones are not being taken as such, so we should evaluate which features are the most valuable.

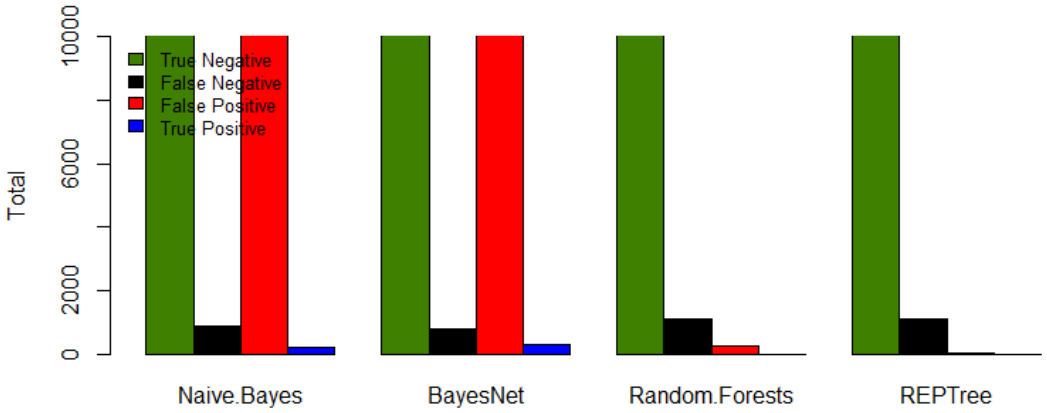

Figure 5.3: Weekly and hourly analysis results

5.2.5 Attribute Evaluation and Selection

In order to optimize results key variables must be found between the lot gathered until now, in order to do this and retrieve the most relevant subset of features to the predictor variable, automatic feature selection algorithms have been used, namely CfsSubsetEval (using BestFirst as the search algorithm) referenced in section 4.5.

Selected attributes (22):

1. baskets
2. baskets24
3. baskets48
4. baskets6h
5. baskets7h
6. baskets8h
7. baskets9h
8. baskets10h
9. baskets11h
10. baskets15h
11. baskets16h
12. baskets17h
13. baskets18h
14. baskets19h
15. baskets20h
16. baskets0w
17. baskets1w
18. baskets2w
19. baskets3w
20. baskets4w
21. baskets5w
22. baskets6w

As it can be easily perceived, the selected attributes were all basket actions, it actually makes sense that basket actions are co-related with purchases. Also one might notice that although statistics from all weekdays was selected some hourly statistics were left out, this might be due the hours with more action on the website being more relevant.

5.2.5.1 Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,989	0,834	0,999	0,989	0,994	0,044	0,629	0,999	0
0,166	0,011	0,013	0,166	0,024	0,044	0,629	0,004	1
0,988	0,834	0,998	0,988	0,993	0,044	0,629	0,998	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1228265	13399	a = 0
901	179	b = 1

5.2.5.2 BayesNet

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,977	0,719	0,999	0,977	0,988	0,051	0,652	0,999	0
0,281	0,023	0,011	0,281	0,020	0,051	0,652	0,007	1
0,977	0,718	0,999	0,977	0,987	0,051	0,652	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1213264	28400	a = 0
776	304	b = 1

5.2.5.3 Random Forest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	0,998	0,999	1,000	1,000	0,006	0,542	0,999	0
0,002	0,000	0,021	0,002	0,003	0,006	0,542	0,004	1
0,999	0,997	0,998	0,999	0,999	0,006	0,542	0,998	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1241569	95	a = 0
1078	2	b = 1

5.2.5.4 REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,999	1,000	1,000	0,000	0,500	0,999	0
0,000	0,000	0,000	0,000	0,000	0,000	0,500	0,001	1
0,999	0,999	0,998	0,999	0,999	0,000	0,500	0,998	<-Weighted Avg.

=== Confusion Matrix ===

a	b	«- classified as
1241664	0	a = 0
1080	0	b = 1

5.2.5.5 Conclusions

Tree algorithms have yet failed to retieve any buyers. Random forest decreased the amount of false positives, however. The results from Naive Bayes and BayesNet did not chance much.

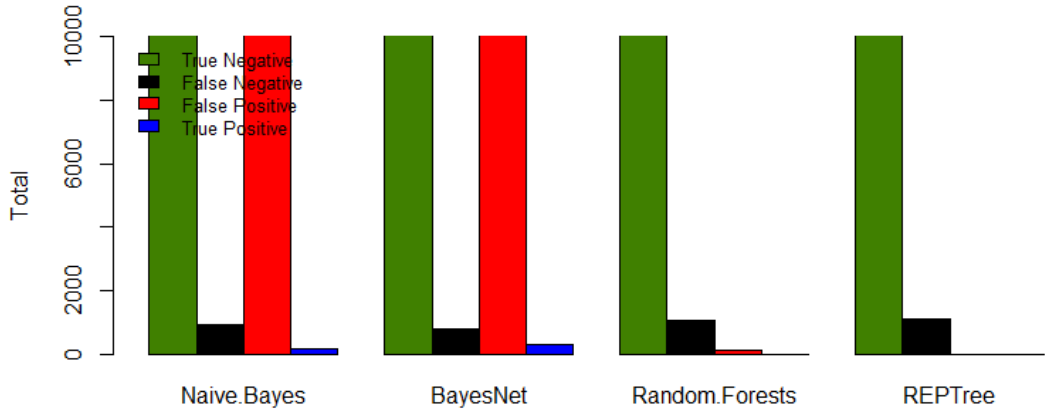

Figure 5.4: Automatic attribute selection results

5.2.6 Changing the time window

In order to improved results, experimentations with the duration of each period of the time window have been done. using the last configuration the algorithms were executed again but with different sliding window configurations.

Original Time Window

- Training Period - 2 Weeks
- Cooldown Period - 1 Hour
- Prediction Period - 1 Week

Time Window 2

- Training Period - 4 Weeks
- Cooldown Period - 2 Hours
- Prediction Period - 2 Weeks

Time Window 3

- Training Period - 8 Weeks
- Cooldown Period - 4 Hours
- Prediction Period - 4 Weeks

5.2.6.1 Time Window 2 - Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,987	0,833	0,998	0,987	0,992	0,068	0,641	0,998	0
0,167	0,013	0,033	0,167	0,055	0,068	0,640	0,012	1
0,985	0,831	0,995	0,985	0,990	0,068	0,641	0,995	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
376282	5066	a = 0
854	171	b = 1

5.2.6.2 Time Window 2 - BayesNet

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,984	0,785	0,998	0,984	0,991	0,081	0,653	0,998	0
0,215	0,016	0,035	0,215	0,060	0,081	0,653	0,022	1
0,982	0,783	0,995	0,982	0,988	0,081	0,653	0,996	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
375287	6061	a = 0
805	220	b = 1

5.2.6.3 Time Window 2 - Random Forest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	0,997	1,000	0,999	-0,000	0,560	0,998	0
0,000	0,000	0,000	0,000	0,000	-0,000	0,560	0,011	1
0,997	0,997	0,995	0,997	0,996	-0,000	0,560	0,995	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
381340	8	a = 0
1025	0	b = 1

5.2.6.4 Time Window 2 - REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	0,997	0,997	1,000	0,999	0,020	0,650	0,998	0
0,003	0,000	0,136	0,003	0,006	0,020	0,650	0,018	1
0,997	0,994	0,995	0,997	0,996	0,020	0,650	0,996	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
381329	19	a = 0
1022	3	b = 1

5.2.6.5 Time Window 3 - Naive Bayes

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,991	0,875	1,000	0,991	0,995	0,022	0,619	1,000	0
0,125	0,009	0,004	0,125	0,009	0,022	0,619	0,001	1
0,990	0,875	0,999	0,990	0,995	0,022	0,619	0,999	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1264964	11944	a = 0
372	53	b = 1

5.2.6.6 Time Window 3 - BayesNet

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,999	0,960	1,000	0,999	0,999	0,021	0,631	1,000	0
0,040	0,001	0,011	0,040	0,018	0,021	0,631	0,002	1
0,999	0,960	0,999	0,999	0,999	0,021	0,631	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1275418	1490	a = 0
408	17	b = 1

5.2.6.7 Time Window 3 - Random Forest

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	0,998	1,000	1,000	1,000	0,013	0,597	1,000	0
0,002	0,000	0,071	0,002	0,005	0,013	0,597	0,002	1
1,000	0,997	0,999	1,000	0,999	0,013	0,597	0,999	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
1276895	13	a = 0
424	1	b = 1

5.2.6.8 Time Window 3 - REPTree

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1,000	1,000	1,000	1,000	1,000	0,000	0,500	1,000	0
0,000	0,000	0,000	0,000	0,000	0,000	0,500	0,000	1
1,000	1,000	0,999	1,000	1,000	0,000	0,500	0,999	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
1276908	0	a = 0
425	0	b = 1

5.2.6.9 Conclusions

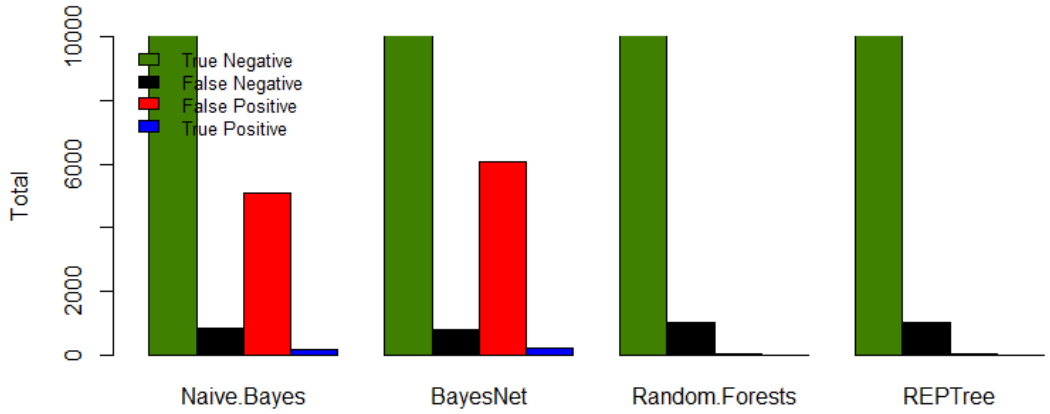

Figure 5.5: Time Window 2 Performance

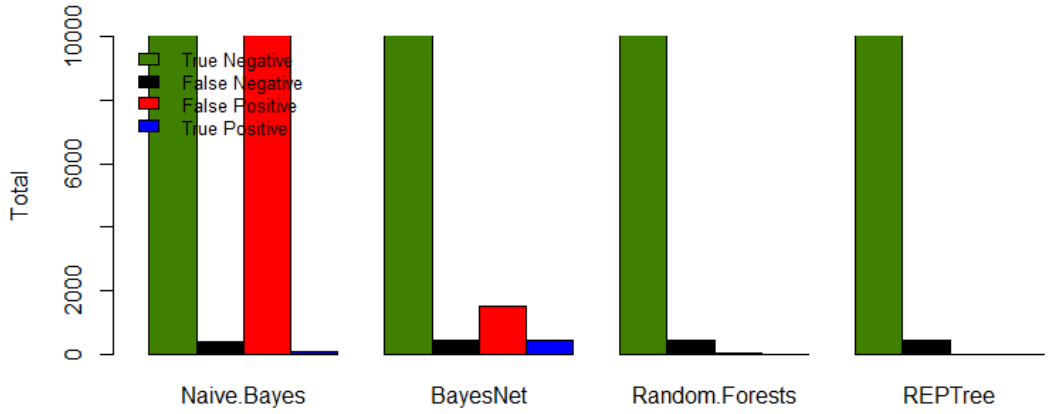

Figure 5.6: Time Window 3 Performance

As it can be seen through precision and recall values for positive values of the predictor variable, the dataset with the Time Window 2 outperformed the other in almost every algorithm. from this one can acknowledge that messing around with the time window values actually takes an impacts on the results.

5.2.7 Cost Sensitive learning

As described in section 4.6, with Cost Sensitive learning we are able to tell the algorithms which errors we want to minimize, in this case the false negative errors should be minimized, this is because the false negative errors correspond to actual buyers classified as non-buyers. We experimented with several values on the cost matrix in order to understand the threshold of values which retrieve good results.

Since the Time Window 2 got the better results until now, its configuration of 4 Weeks for the Training Period, 2 Hours for the Cooldown Period and 2 Weeks of Prediction Period will be used in the upcoming tests.

5.2.7.1 Naive Bayes

Cost Matrix

0	1
1000	0

2

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,982	0,787	0,998	0,982	0,990	0,074	0,641	0,998	0
0,213	0,018	0,030	0,213	0,053	0,074	0,640	0,012	1
0,980	0,785	0,995	0,980	0,987	0,074	0,641	0,995	<-Weighted Avg.

4 === Confusion Matrix ===

a	b	<- classified as
374369	6979	a = 0
807	218	b = 1

Cost Matrix

0 1
10000 0

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,971	0,691	0,998	0,971	0,984	0,085	0,641	0,998	0
0,309	0,029	0,028	0,309	0,051	0,085	0,640	0,012	1
0,969	0,689	0,995	0,969	0,982	0,085	0,641	0,995	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
370184	11164	a = 0
708	317	b = 1

Cost Matrix

```
0       1
100000  0
```

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,971	0,691	0,998	0,971	0,984	0,085	0,641	0,998	0
0,309	0,029	0,028	0,309	0,051	0,085	0,640	0,012	1
0,969	0,689	0,995	0,969	0,982	0,085	0,641	0,995	<-Weighted Avg.

[2] === Confusion Matrix ===

```
     a      b      <- classified as
370173  11175  a = 0
   708    317  b = 1
```

In conclusion Naive Bayes was able to detect a relevant amount of buyers out of the , improving results from the last iterations. Higher cost matrix values seem to not influence much the results of this algorithm, since there is a sightly noticeable difference between its execution with a value for false positives in the cost matrix of 1000 to 10000, and little to no difference from 10000 to 100000.

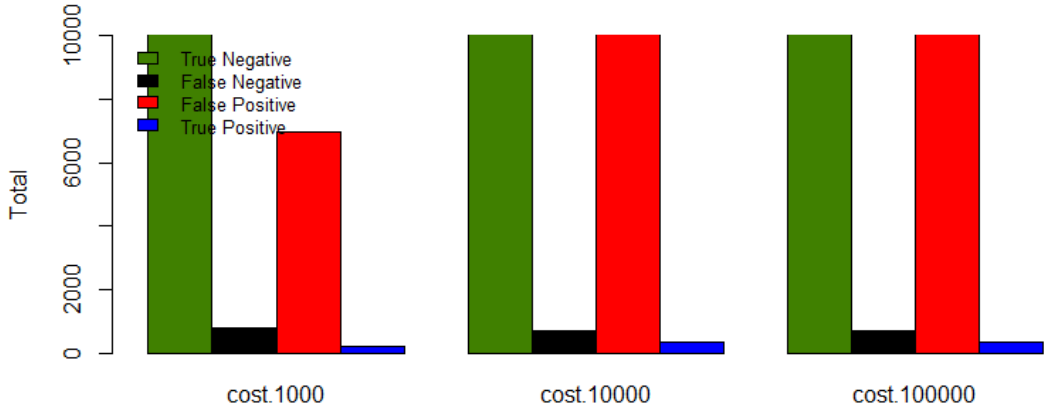

Figure 5.7: Naive Bayes with cost sensitive learning

5.2.7.2 BayesNet

Cost Matrix

0	1
400	0

2

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,968	0,666	0,998	0,968	0,983	0,088	0,651	0,998	0
0,334	0,032	0,027	0,334	0,051	0,088	0,651	0,023	1
0,966	0,665	0,996	0,966	0,980	0,088	0,651	0,996	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
369215	12133	a = 0
683	342	b = 1

Cost Matrix

0 1
750 0

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,968	0,666	0,998	0,968	0,983	0,088	0,651	0,998	0
0,334	0,032	0,027	0,334	0,051	0,088	0,651	0,022	1
0,966	0,665	0,996	0,966	0,980	0,088	0,651	0,996	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
369187	12161	a = 0
683	342	b = 1

Cost Matrix

0	1
800	0

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,000	0,002	0,972	0,000	0,000	-0,007	0,651	0,998	0
0,998	1,000	0,003	0,998	0,005	-0,007	0,651	0,022	1
0,003	0,005	0,969	0,003	0,000	-0,007	0,651	0,996	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
69	381279	a = 0
2	1023	b = 1

In Conclusion results were pretty decent, an recall of 0,334 means that 33.4% of the buyers were found with a precision of 29%As can be easily seen, little to no variation in results takes place while changing the cost matrix from 400 up to 750. However when setting the cost matrix value to 800 or, more results begin to lose quality as the precision drops to 0.3%

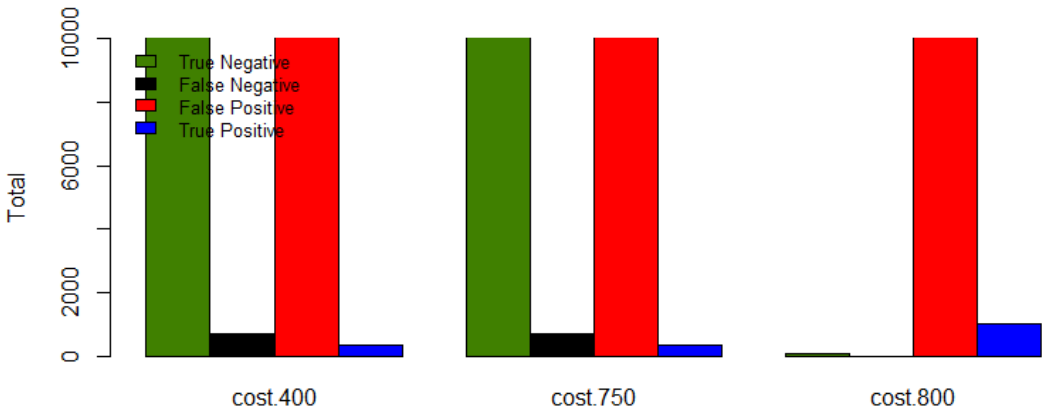

Figure 5.8: BayesNet with cost sensitive learning

5.2.7.3 Random Forest

Cost Matrix

0	1
415	0

[2]

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,978	0,798	0,998	0,978	0,988	0,062	0,530	0,998	0
0,202	0,022	0,024	0,202	0,042	0,062	0,530	0,007	1
0,976	0,796	0,995	0,976	0,985	0,062	0,530	0,995	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
372818	8530	a = 0
818	207	b = 1

Cost Matrix

0	1
420	0

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,010	0,136	0,964	0,010	0,019	-0,065	0,527	0,998	0
0,864	0,990	0,002	0,864	0,005	-0,065	0,527	0,007	1
0,012	0,138	0,961	0,012	0,019	-0,065	0,527	0,995	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
3708	377640	a = 0
139	886	b = 1

In Conclusion Random forests actually delivered decent results, vast improvements from the previous execution which have failed to detect a single buyer. However if the cost matrix value goes higher than 420 bad predictions appear as the prediction drops.

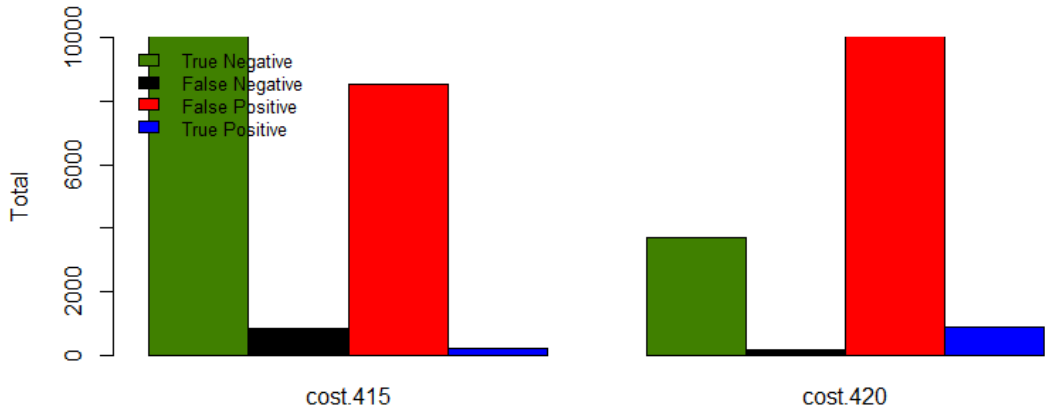

Figure 5.9: Random Forest with Cost Sensitive Learning

5.2.7.4 REPTree

Cost Matrix

0	1
400	0

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,969	0,670	0,998	0,969	0,983	0,087	0,648	0,998	0
0,330	0,031	0,027	0,330	0,051	0,087	0,648	0,015	1
0,967	0,669	0,996	0,967	0,981	0,087	0,648	0,995	<-Weighted Avg.

=== Confusion Matrix ===

a	b	<- classified as
369378	11970	a = 0
687	338	b = 1

Cost Matrix

0	1
410	0

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0,000	0,000	0,000	0,000	0,000	0,000	0,500	0,997	0
1,000	1,000	0,003	1,000	0,005	0,000	0,500	0,003	1
0,003	0,003	0,000	0,003	0,000	0,000	0,500	0,995	<-Weighted Avg.

2 === Confusion Matrix ===

a	b	<- classified as
0	381348	a = 0
0	1025	b = 1

In conclusion REPTree archives good predictions until the cost matrix value for the false positives is higher than 410, after this precision drops hard.

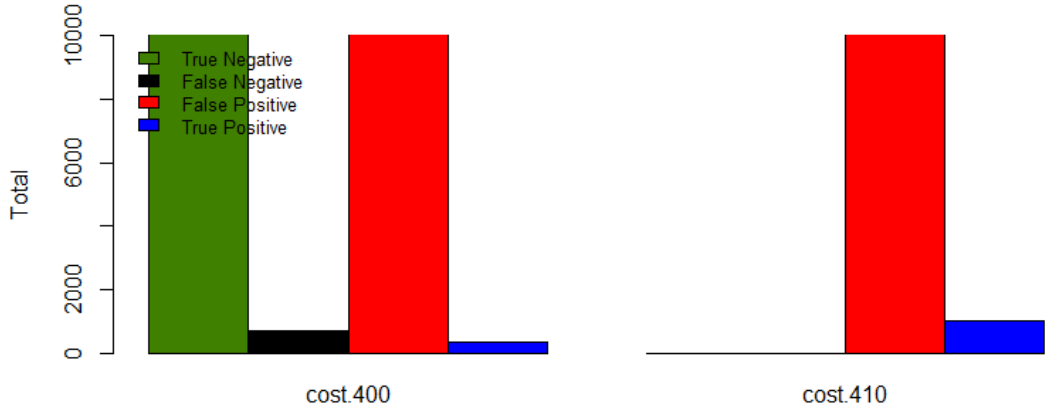

Figure 5.10: REPTree with Cost Sensitive Learning

5.2.7.5 Conclusion

Cost Sensitive Learning improved the outcome of the algorithms by a significant margin, being that the Random Forests Algorithm which had failed to retrieve relevant entries until then actually detected an acceptable amount of buyers. BayesNet and REPTree however were the ones that archived better results having BayesNet obtained a recall for buyers of 33,4% and REPTree of 33%, both with a precision of 2,7%.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Given the problem of classifying an user as buyer or not given data containing information about each user action in the website, the goal of applying machine learning techniques in order to generate a prediction model from this data has been archived.

From the log which contained several entries of user actions we were able to extract crucial data for the prediction task at hand, this data was used to train several models and the important features got refined along a series of iteration, machine learning techniques such as cost sensitive learning have been applied to the models so that the results might improve. One can therefore say the goal has been archived.

6.2 What could improve / Future Work

Although relevant results have been archived, there is always room for improvements. Quality data and key feature extraction are crucial for ensuring a good machine learning process, for this there is the need to have a good understanding knowledge of the avalilable data. This does now happen everytime, for example, during the work there was no idea of how the product categories were actually discriminated, e.g. we did not knew whether a category would describe a "section" of products or a specific brand, maybe if the knowledge about the specificness of categories was available more prediction variables could be extracted from the data about the categories seen by the users.

Still on the feature creation subject, demographic information about the users such as

genre, age and country could have come in handy to perceive buying tendencies in product lines targeted for a specified public.

As implied in 5.1.2 the conversion rate presented in the studied dataset is inferior to the average conversion rate found in the e-Commerce environment. Maybe if the ratio of buyers to non buyers was higher, which would lead to a more well balanced dataset, it could have a positive impact on the predictions.

In order construct new predictive models with higher performance different new features should be extracted from the dataset, as well as more algorithms and experimental setups should be tested and evaluated. The concept of "session" as an interval of time during which the user is using the platform and the buying history of the users are some of the features that were not taken in consideration during this work and might hold predictive capabilities if explored correctly.

References

[AAN] Neural network | tikz example. http://www.texample.net/tikz/examples/neural-network/. Accessed: 2014-02-09.

[AWBG07] Eytan Adar, Daniel Weld, Brian Bershad e Steven Gribble. Why we search: Visualizing and predicting user behavior. 2007.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[BS09] Randolph E. Bucklin e Catarina Sismeiro. Click here for internet insight: Advances in clickstream data analysis in marketing. *Journal of Interactive Marketing*, 23(1):35 – 48, 2009. Anniversary Issue. URL: http://www.sciencedirect.com/science/article/pii/S1094996808000054, doi:http://dx.doi.org/10.1016/j.intmar.2008.10.004.

[CRa] Marketing research chart: Average website conversion rates. http://www.marketingsherpa.com/article/chart/average-website-conversion-rates-industry. Accessed: 2014-05-20.

[DMP] Wooley abstract: Scaling clustering for the data mining. http://www.cs.utexas.edu/users/csed/doc_consortium/DC99/wooley-abstract.html. Accessed: 2014-05-10.

[dPB05] Dirk Van den Poel e Wouter Buckinx. Predicting online-purchasing behaviour. *European Journal of Operational Research*, 166(2):557 – 575, 2005. doi:http://dx.doi.org/10.1016/j.ejor.2004.04.022.

[Gha04] Zoubin Ghahramani. Unsupervised learning. In *Advanced Lectures on Machine Learning*, pages 72–112. Springer-Verlag, 2004.

[HK13] M. Hofmann e R. Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Taylor & Francis, 2013. URL: http://books.google.pt/books?id=5zYTAgAAQBAJ.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie e Robert Tibshirani. *An introduction to statistical learning. With applications in R.* Springer Texts in

Statistics 103. New York, NY: Springer. xiv, 426 p. EUR 64.19; $ 79.99/net , 2013. doi:10.1007/978-1-4614-7138-7.

[Koh95] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[Kot07] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands, 2007. IOS Press. URL: http://dl.acm.org/citation.cfm?id=1566770.1566773.

[KP06] Mehdi Khosrow-Pour, editor. *Emerging trends and challenges in information technology management*, Hershey, Pa. [u.a.], 2006. Idea Group Publ.

[LMJ] Jung Jin Lee, Robert McCartney e Eugene Santos Jr. In Ingrid Russell e John F. Kolen, editors, *FLAIRS Conference*, pages 177–181. AAAI Press.

[RSD+12] Kira Radinsky, Krysta Svore, Susan Dumais, Jaime Teevan, Alex Bocharov e Eric Horvitz. Modeling and predicting behavioral dynamics on the web. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 599–608, New York, NY, USA, 2012. ACM. URL: http://doi.acm.org/10.1145/2187836.2187918, doi:10.1145/2187836.2187918.

[RSTK03] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan e David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *In Proceedings of the Twentieth International Conference on Machine Learning*, pages 616–623, 2003.

[SB04] C. Sismeiro e R. E. Bucklin. Modeling purchase behavior at an e-commerce web site: a task-completion approach. *Journal of Marketing Research*, 41(3):306–323, 2004.

[SW10] C. Sammut e G.I. Webb, editors. *Encyclopedia of Machine Learning*. Springer, Berlin, 2010.

[V.C] G.Suganya V.Christina, S.Karpagavalli. Email spam filtering using supervised machine learning techniques.

[wek] Reptree. http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/REPTree.html. Accessed: 2014-05-20.

[WPB01] Geoffrey I. Webb, Michael J. Pazzani e Daniel Billsus. Machine learning for user modeling. *User Modeling and User-Adapted Interaction*, 11(1-2):19–29, March 2001. URL: http://dx.doi.org/10.1023/A:1011117102175, doi:10.1023/A:1011117102175.