FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

Company Project # 2015-1428

Martin Nicklas Jørgensen tzk173@alumni.ku.dk

User Behavior Analysis Using Decision Trees

Supervisor: Mikkel Rønne Jakobsen mikkelrj@di.ku.dk

January 3, 2016

Abstract

Write something sensible, requirements is on the intranet page.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec a diam lectus. Sed sit amet ipsum mauris. Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit. Donec et mollis dolor. Praesent et diam eget libero egestas mattis sit amet vitae augue. Nam tincidunt congue enim, ut porta lorem lacinia consectetur. Donec ut libero sed arcu vehicula ultricies a non tortor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut gravida lorem. Ut turpis felis, pulvinar a semper sed, adipiscing id dolor. Pellentesque auctor nisi id magna consequat sagittis. Curabitur dapibus enim sit amet elit pharetra tincidunt feugiat nisl imperdiet. Ut convallis libero in urna ultrices accumsan. Donec sed odio eros. Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem facilisis semper ac in est.

Contents

1 Introduction

2 Simplesite ApS

2.1 Product

Simplesite produce and manage an inhouse website Content Management System (CMS) and operate a hosting location where this CMS runs. Customers can register for a free account allowing them to host a website that is edited through the CMS. The free accounts are under restrictins on number fo pages, images and videos that can be added to the site. Customers can then change to a paid subscription which allows them to have more pages, images and videos on their website.

Simplesite also offer additional services to their customers that can be bought for an extra fee if you are already a paying customer, this includes the ability to have a domain attached to your website, a webshop as well as more additional pages, images and videos.

The product is hosted partially on Simplesites own hardware in an offsite location, and using a number of cloud services to provide faster response times for certain data types.

2.2 Departments

- **Administration** - Situated at the ground floor of the Copenhagen office, this department contains the HR functions as well as finance.
- **Sales** - Also on the ground floor of the Copenhagen office, sales consists of full time employeess and student helpers. The primary task is to sell the product, currently through localized ad management.

- **P & C**[1] - The department sits on the upper floor of the Copenhagen office and is responsible for planning new features in cooperation with the developers as well as manage content on Simplesites own websites. P & C also manages communication such as newsletters and localized ad texts.
- **In-House Development** - Sitting next to P & C on the upper floor the developers are responsible for implementing new features as well as maintenence, analytics and bug fixing of the product.
- **Operations** - Daily operations are handled primarily by the company CTO, Thomas, as well as 2 part time students, operations sits on the upper floor in Copenhagen next to the developers.
- **Support** - Since the product is offered in several languages, a supporter is hired per language in a part time basis. All supporters work from home but get together once a month for a status meeting and to make sure new knowledge is shared and that relevant information can be given from the regular departments.
- **Remote Dev 1** - A small number of developers are hired in Bulgaria and have their own office in the Sofia, they are offered machines in the Copenhagen office they can VPN into and use for work. This allows the operations department to work from Copenhagen and still service the remote developers.
- **Remote Dev 2** - A number of developers are also hired from Serbia, like the remote developers in Bulgaria, they also VPN into the Copenhagen office and work on machines maintained by the regular operations employees.
- **Miscellaneous** - Simplesite also occasionally employs external specialists or consultants. Depending on the need, they will either work in the Copenhagen office or from some remote location using VPN.

2.3 Company Structure

Insert finished graphml figure.

3 Problem Description

Simplesite changed their subscription service to a so called freemium model during 2015. This model means that everyone can have a website for free, and it will never be closed. This way of doing subscriptions means that more people sign up, and become potential customers, however a trend is that most customers that sign up either don't become paying customers, or simple make a site and stop using it after a few days.

Simplesite wishes to map the life-cycle of customers in an attempt to find out how good users (users that are *retained*, ie. have logins 3-4 weeks after they are created) use their site, as well as look at the life-cycles of the many free or abandoned customers. They hope to discover what, if any, the significant differences is in the different lifecycles. The goal is to attempt to guide new customers down the paths that are known to be "good" and hopefully detect if customers are stuck or have forgotten about their website.

3.1 Requirements

Simplesite have already created some code in R[2] so any continued work should also be done in the R language. I have also been added to a Github repository with already populated with analytics code and a preferred structure and coding style that should be adhered to.

Two datasets are also available containing different information gathered from the live system. These two datasets should form the basis for all the dataanalysis performed. An additional dataset

[1] Product and Communication.
[2] `https://www.r-project.org/`

is being constructed during the project, but due to the time it takes to populate it with enough observations to be meaningful it might not be finished before the project is over.

3.2 Success Criteria

The project have a number of success criteria that should all be fulfilled to some degree.

1. A model for classifying customers are created.
2. New knowledge about customers lifecycles are acquired from the model. More preferably, do *retained* customers have something in common.
3. A prototype R script that can automatically build/create the model from new customer data should be created.
4. A method for using the model should be designed or reasoned about.

4 Problem Analysis

4.1 Available Datasets

The basis for the analysis is 2 datasets created by Simplesite: `EngagementData` & `CustomerJourney`. Table 1 and Table 2 names the features of each dataset and what they represent. Both datasets contain users created between September 1st 2015 and September 15 2015.

This window should maybe be expanded a little for more data.

Attribute Name	Attribute Data
islogins1	Bool, true if: one or more logins for the user.
islogins2	Bool, true if: two or more logins for the user.
islogins3	Bool, true if: three or more logins for the user.
islogins4	Bool, true if: four or more logins for the user.
isedit30m	Bool, true if: User edited site within 30 minutes of creation.
isedit24h	Bool, true if: User edited site within 24 hours of creation (excluding the first 30 minutes).
isaddpage30m	Bool, true if: User added a new page within 30 minutes of creation.
isaddpage24h	Bool, true if: User added a new page within 24 hours of creation (excluding the first 30 minutes).
isimgupload30m	Bool, true if: User uploaded their own image within 30 minutes of creation.
isimgupload24h	Bool, true if: User uploaded their own image within 24 hours of creation (excluding the first 30 minutes).
iseditdesign30m	Bool, true if: User edited site within 30 minutes of creation.
iseditdesign24h	Bool, true if: User edited site within 24 hours of creation (excluding the first 30 minutes).
customerid	Integer value with the customers unique ID.
marketname	String with the market the user came from (US, TR, DK etc.)
siteverkey	String with what version of the site the user is created in (US, TR, DK etc.)
ispayer	Bool, true if: The customer have a paid subscription.
culturekey	String with language information for the site (en-US, fr-FR etc.)
iso14	Bool used by marketing.

Table 1: Features found in the `EngagementData` dataset.

Attribute Name	Attribute Data
customerid	Integer value with the customers unique ID.
logins14	Integer, number of times the customer logged in the first 14 days (week 1-2 after creation).
logisnw2w4	Integer, number of times the customer logged in in week 3-4 after creation.
edits14	Integer, number of times the customer edited a page within the first 14 days.
iscjtrial	Bool, true if: Always true, everyone starts as a trial.
iscjonboarded	Bool, true if: edits14 ≥ 1.
iscjactivated	Bool, true if: edits14 ≥ 3.
iscjengaged	Bool, true if: edits14 ≥ 6 and logins14 ≥ 2.
iscjinvested	Bool, true if: edits14 ≥ 15 and logins14 ≥ 6.
iscjretained	Bool, true if: logisnw2w4 ≥ 1.
isimgupload1d	Bool, true if: Customer uploaded an image within the first 24 hours of being created.
iseditdesign1d	Bool, true if: Customer edited the design within the first 24 hours of being created.
isaddpage1d	Bool, true if: Customer added a new page within the first 24 hours of being created.
isedit1d	Bool, true if: Customer edited a page within the first 24 hours of being created.

Table 2: Features found in the `CustomerJourney` dataset.

4.2 Dataset Pruning

The initial goal is to find customers who are retained (*iscjretained* = `True`) and see if there is some pattern, that Simplesite can try to guide other customers down in order to increase the number of retained customers. With this in mind there is some attributes of the datasets that will not be helpful, either because they cannot be controlled/changed, or because they do not make sense. The following is a list of attributes removed from the `EngeagementData` dataset during work, along with the reason for the removal.

- *islogins1* : Removed since it is always true for all customers.
- *islogins2* : Removed because the definition of a retained customer requires one or more logins, so this must always be true.
- *islogins3* : Same as *islogins2*.
- *islogins4* : Same as *islogins2*.
- *marketname* : Removed since we are unable to get a customer from a different market, we are insterested in variable we can change for each customer.
- *siteverkey* : Same as *marketname*.
- *ispayer* : Removed because it is an alternative target variable, it does not say anything about how the user behaves, other than they are indeed a good customer.
- *culturekey* : *marketname*.
- *iso14* : Value used by marketing.

is o14 usefull? Clear up with Morten.

The following is a list of attributes removed from the `CustomerJourney` dataset during work, along with the reason for the removal.

- *logisnw2w4* : Removed since this attribute is in the definition of our target variable *iscjretained*.
- *iscjtrial* : Removed since it is always true.
- *iscjonboarded* : Removed since it serves as an alternative target variable, and is set by us, it does not say anything about the user behaviour that is not already present.
- *iscjengaged* : Same as *iscjonboarded*.

- *iscjinvested* : Same as *iscjonboarded*.

In both datasets the *customerid* attribute is kept in each dataset, even though it cannot be used as a feature for analysis since each customer have a unique ID and thus will not yield any patterns, since it can be used to join the two datasets together.

4.3 Tree Type

While researching R, two distinct type of dicision trees came up, an implementation of regular decision trees that closely follow the method described in [1], as well as conditional inference trees[2], which I will shorten to *ctree* in this report.

Both models produce binary trees that can be used to solve classification problems. Each node in the tree represents a variable and each edge out of the node contains a "case" that tells something about the variable (is it true/false, ≥ 5, and so forth). Each leaf of the tree is then a class, there can be several leafs with the same class, they just represent different characteristics of the same class.

The main difference between a regular decision tree and a ctree is how it is created. A regular decision tree will choose to split using information measures such as seen in Quinlan et. al.[3, p. 89], whereas the ctree framework will split based on the relationship between the variables and the covariates.

Write more here. And create code for the experiment.

Write a better explanation without bloating it.(See [2].)

4.3.1 Plotting

Since a big part of the project is for Simplesite to learn about their customers behaviour on the site, as much as letting a computer learn it, the readability of the created models became a criteria as well. In order to guage how easy it was to get information about the models created by the trees I plotted models created on the same data, using different functions. Figure 1 shows the result for the different models and functions.

While which figure is easiest to read can be subjective, I have a few observations about the figures:

1a Displays the most information of the different figures but seems to have issues with being scaled down due to amount of information in display.
1b Shows most of the information of 1a without overflowing.
1c Only shows the edges without showing decisions.
1d Displays much of the same information as 1a and 1b but also seems to have problems with scaling.

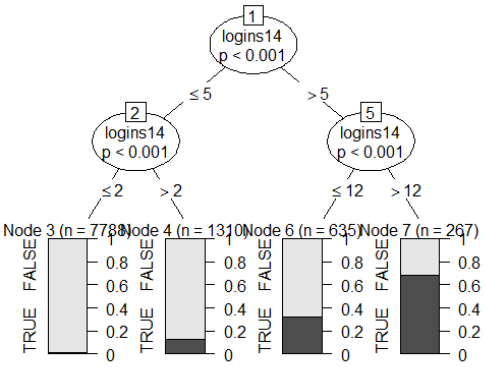
While all the examples in Figure 1 are very simple it does illustrate nicely that some of the plot functions will have trouble later when the tree grows. Based on the above examples I would use the `ctree` model with the plot function from Figure 1b.

4.3.2 Model Precision

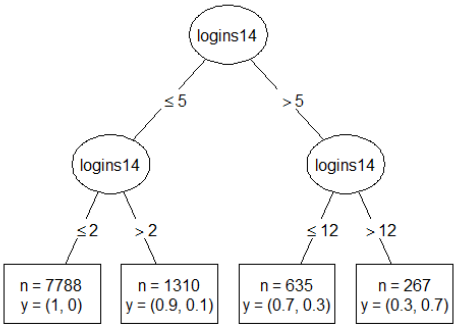
In order to make sure that we do not loose any "precision" by selecting the `ctree` model over the normal decision tree, I performed a number of tests with each of the models in order to determine if this would indeed be the case.

For the tests I performed 5-fold cross validation and different maximum tree depths and recorded the mean precision for each model. The dataset in question was the combined dataset as described in Section 4.2 (excluding the customerid column.) The code for this test can be seen in Appendix **??**.

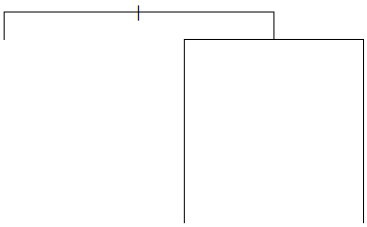
The results shown in Table 3 indicates that for our particular dataset, the accuracy difference for the two models have a very similar performance in terms of accuracy for our dataset. They



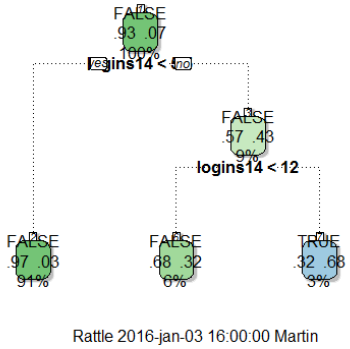
(a) The default plot of a `ctree` model.



(b) Plot of the same model using a customized plot function.



(c) The default plot of a `rpart` model.



(d) Plot of the same model using `fancyRpartPlot`

Figure 1: A side by side view of the different plots produced for the `rpart` and `ctree` models respectively. Code for creating the figures can be found in Appendix **??**.

Max Depth	`rpart` **Accuracy**	`ctree` **Accuracy**
4	0.942799	0.9427990
8	0.942799	0.9431958
12	0.942799	0.9436638

Table 3: The mean accuracy for the different 5-fold cross validation runs.

both predict correctly in around $94,3\%$ of the cases with a difference of less that $0,1\%$ between the best and worst performer.

An interesting observation from the data above is that the `rpart` model have the same accuracy for all three runs indicating that the model created does not change even when it is allowed to grow more complex. After plotting models from each step, I discovered that it did not grow beyond a depth of 2 and based itself solely on the `logins14` variable, similarly to what could be seen in Figure 1d.

4.3.3 Choosing a Model

Based on the two criteria highlighted in in Sections 4.3.1 and 4.3.2, I selected to go with the `ctree` package, because it was easier for me to read the information the model was created on without sacrificing any precision compared to the traditional decision tree implemented in `rapart`.

4.4 Analysis

For this section the two datasets `EngagementData` & `CustomerJourney` have been joined on the *customerid* attribute, into one dataset after which the *customerid* attribute was removed. This is the dataset used for the remainder of this section unless otherwise stated.

5 Results

6 Competencies and Methods

7 Conclusion

References

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. 1984 classification and regression trees. *The Wadsworth Statistics/Probability Series*.

[2] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006.

[3] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.