CM. the heart of mobile
predicting customer loss in high volume text messaging

Klijs, V.J.

Award date:
2015

The heart of mobile

Predicting customer loss in high volume text messaging

Master thesis **Business Information Systems**

Vincent Klijs 0656718

07-08-2015

Supervisors

B.F. van Dongen B.F.v.dongen@tue.nl
Cas Schalkx cs@cm.nl

0 – Abstract

Communication is growing ever more important, and mobile communication is a major contributor in this movement. Smartphones have penetrated all layers of society, and they are used has moved well beyond their original scope. Text messages companies have risen which offer mobile solutions to incorporate text messages within corporate processes.

In this master thesis, we will create a prediction model that will predict if customers of a text message company will stop using their services. The company for which this research will be conducted is CM, located in Breda. CM has been around for over 15 years and offers mobile services that include mobile payments, tele voting and text messaging.

The task of creating a prediction to predict customer loss is quite daunting, and will be split up into three parts. The first part consists of determining when customers are going to buy their next credit. This is done by the use of 2 regression models, and a simple prediction model which is based solely on previous payments information.

The second part consists of determining if customers are going to pay within a particular time limit. This time limit is set to 4 weeks, because 56% of all messages send by customer took place at most 4 weeks prior to a payment. To predict if customers are going to pay within 4 weeks, a decision tree is made which is able to predict with large accuracy.

The third part consists of determining customer loss by analyzing the amount of messages they have sent over a 4 week period. This analysis is done by using decision trees on a data file containing the percentage increase or decrease of the amount of messages sent over a period of 4 weeks. Different periods are examined, among which the 4 weeks prior to the last sent message.

The results found is an indication that it is quite complicated to actually predict customer loss. We are able to create several decision trees to predict either if a payment is due or to predict customer loss. However, either the accuracy is very high but the decision tree is very difficult to interpret, or the accuracy is quite a bit lower and the decision tree is very simple. But, the fact that we have been able to predict if customers are going to pay with a very large accuracy is promising for future research.

Table of Contents

1 – Introduction

Communication is becoming ever more important, and mobile phones play a major role in this worldwide development. As smartphones grow more sophisticated their capabilities grow as well. Smartphone users are not only able to call through the mobile phone networks or send text messages, they are also able to call using their internet connection using VoIP and know where their friends and loved ones are by checking Facebook or other social media platforms. It almost seems that the original functionality of mobile phones, calling and texting, are complete lost and have become redundant, but this is far from the truth. Especially text messaging is proving its worth in these modern times. For example, governments in Europe use them to relay messages to their citizens concerning emergencies through services called NL-Alert (system used in the Netherlands) or Be-Alert (system used in Belgium). Furthermore, more and more companies use text messages as an additional verification method during login procedures. For example, the Dutch bank ING uses text messages to relay a security number which has to be used when sending out payments. Since the early ages of the mobile phone, companies have been around providing exactly these kind of services. Just as the technology grows more sophisticated, so are the services provided by such companies. The latest developments in this sector are mobile payments, and merging web based services such as whatsapp seamlessly with existing text platforms. However, competition also increases, therefor it is vital for companies to attract new customers, and keep existing ones.

1.1 - Company Background

CM [1] is a company which was started 15 years ago, as mobile phones started to gain momentum. At first their services were focused solely on sending text messages. It entailed little more than to notify visitors of nightclubs of future events. This evolved into allowing visitors of night clubs to send messages which would end up on screens at the club. However, as technology evolved, the services provided by CM evolved as well. Currently CM provides a mobile platform for mobile payments, tele voting, text verification, messaging and voice solutions. One of their services is called CM Direct, and this platform will be the focus of this thesis. CM Direct is a platform on which users can register, create contacts, manage groups of contacts, easily buy credit and send text messages through an web-based interface. Figure 2 and Figure 3 are screenshots of CM Direct. Figure 1 shows the contact section in which contacts

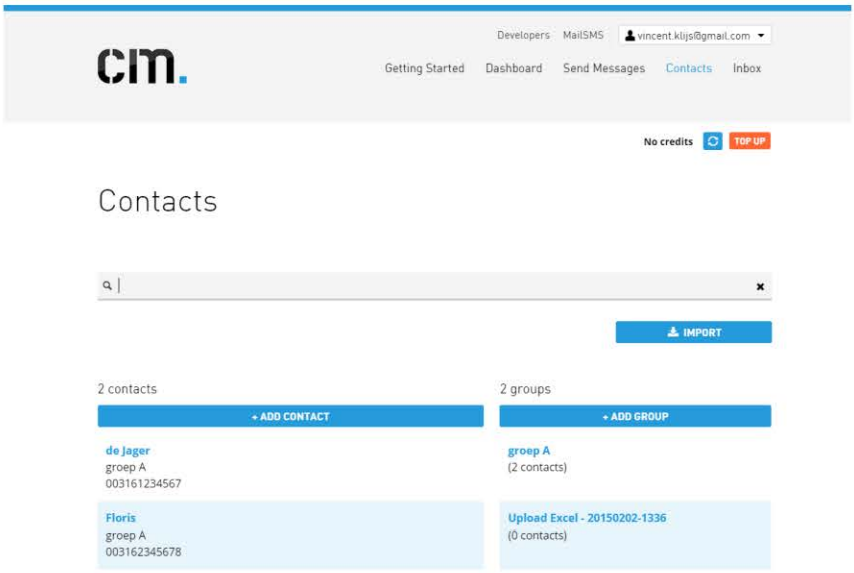


Figure 1 Screenshot of CM Direct

and groups of contact can be managed. Figure 2 shows the part in which messages can be created.

More technical users can also send messages directly to the CM backend using provided API's services. It is important to note that at the time of this project, all customers had to pay for the messages in advance. It was not possible to send messages and pay for them afterwards. This was at the time a limitation of CM Direct. Although this service is very easy to use for customers, there is very little locking so customers can easily stop using CM Direct.

Currently, once a month a balance sheet is created which state how many messages have been sent by which customers. Using this sheet, it is also possible to determine which customers have not sent any messages the past month and which customers have sent the most messages. Customers are considered a loss if they have not sent any messages for an entire month. This is also referred to as churning, e.g. a particular customer has churned. Every month customers churn of whom it was not expected. It is difficult to say if this can be prevented completely as there are factors involved in which CM has no saying. However, CM hopes to gain more insight into when customers shown signs of leaving.

1.2- Research Question

CM hopes to gain insight into when and/or if customers will stop using their service CM Direct. To answer this question, this thesis focuses on three sub questions regarding the customers of CM Direct.

1. Is it possible to predict if a payment is due?
2. Is it possible to predict when the next payment will occur?
3. Is it possible to predict customer loss?

In this thesis, we answer these three questions in the following way. Predictions on whether or not a payment is due are made by using a decision tree. The time to the next payment is estimated by the use of various regression models. Finally, the third question is answered by creating a decision tree based on the amount of messages sent in a particular period. The answers of the first 2 questions can also lead to interesting insights regarding customer loss.

1.3 - Thesis Structure

Chapter 2 discusses the relevant literature and important preliminaries. Among these are papers and publications regarding data mining, process mining and regression models. Data mining is a well-studied discipline of computer science which deals with the retrieval of data from a large body of data. Process mining is like data mining, but is more specific in that it deals with the discovery of process information from event logs. Regression models are known statistical tools which can help discover the relation between a set of variables.

Although CM Direct has not been around since the very beginning of CM, it has been an active part of CM for a couple of years now. As a result, a lot of data regarding the customers and their usage of CM Direct lurks in the vast data landscape of CM. Chapter 3 will discuss what data is extracted from which sources and which steps were necessary to transform the data for further use. The collected data has anomalies which could affect future analysis, and should be removed from the data. Chapter 4 entails what anomalies have been found, and why they are removed.

CM also has an intuition that there are different types of customers regarding CM Direct. In chapter 5, the data will be analyzed to see if there is any kind of evidence to support this intuition.

Chapter 6 entails the explanation of the discovered results regarding the research question. The results of the two used regression models will be presented, as well as the decision tree that we have been able to create to predict if customer are going to pay within a certain time frame. The decision trees predicting customer loss are also discussed here.

Chapter 7 lists the implantation details and chapter 8 will contain the conclusion of this thesis along with recommendations for CM. Further research topics are also covered in this chapter.

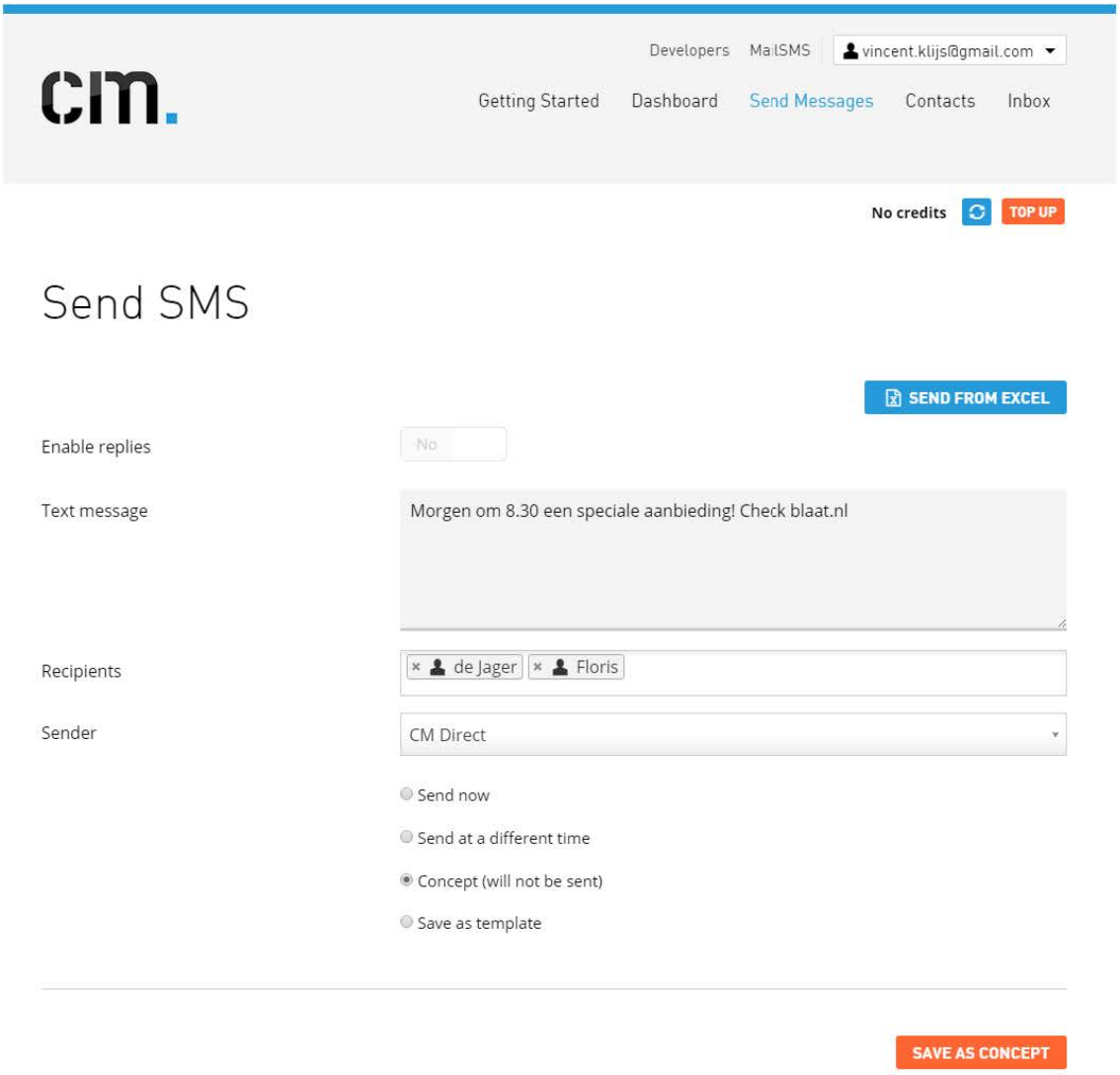


Figure 3 Another screenshot of CM Direct; this entails the message section of CM Direct

2 - Related Work and Preliminaries

The main goal of this thesis is to create a prediction model using the data we can gather from CM Direct. This will inevitably involve various aspect from data mining, which is a very large topic within computer science and some topics need to be explained in some details before going into the actual thesis. This also involves aspect of business intelligence, a rather broad term which is closely related to data mining.

Business Intelligence

As states by [2], the amount of data digitally stored grows at an exponential rate. This data includes personal information such as pictures and digitally kept diaries, but also covers information stored by companies regarding their operations. Over the years, this has led to a growing desire to use this large amount of information and extract business value from it [3].

Business intelligence entails the processes, technologies and tools needed to turn data into information and into knowledge as described above. Although the exact scope may vary depending on the author and companies actively using it [4] , business intelligence generally deals with three different topics.

- Data Acquisition

  Data is collected from internal and external sources. Internal sources can be the operational software, such as accounting programs like Unit4Financials [5]*. External sources can be het Centraal Plan Bureau* [6]*.*
- Data storage

  For smaller datasets a database may suffice. But usually, the data gathered for business intelligence purposes is very large. And other data storage solutions are necessary. This data is usually stored in a data warehouse.
- Data Analysis

  In order to analyze the gathered data, various tools can be used to gain insight. An OLAP database (Online Analytical processing) is a commonly used tool. Usually a OLAP database is filled with aggregated data computed from some sort of transactional database. The OLAP database offers multiple views to allow user to explore the data from various perspectives. Machine Learning is another commonly used practice. Machine Learning is a field of Computer science which encompasses algorithms that use training data to create a model which can be used to predict another set of data records [7].

Business intelligence is a rather broad term that entails much of what is done in this thesis. Especially the Data analysis, and to a minor extend the data acquisition are an active part of this thesis.

Process Mining

A way to analyze data gathered within a company is by use of process mining. Companies use more and more systems in their daily operations. Most of these system create large amounts of log files which entail what has been done at what time by whom. All this information could be combined into a single log file, which lists everything the company has done digitally. The field of process mining deals with the extraction of process information from such Event logs [2]. Process Mining deals with 3 topics.

- Discovery

  Use the event logs to generate a model of the process, or processes within the company.
- Conformance checking

  Compare the generated model against known business models. These models can be known either through experience, or because the processes have at some point been defined.

- Enhancement

  Extend the generated process models by use of additional information such as performance, costs, user feedback etc.

Various techniques exist to create a process model using event data. The α-algorithm is a very basic technique, and can be quite effective but is also very prone to under fitting or overfitting [2]. More advanced techniques, such as Fuzzy Miner allow the user to display only the most frequent paths, and can lead to more less spaghetti like models. Figure 4 shows the area of interest of process mining.

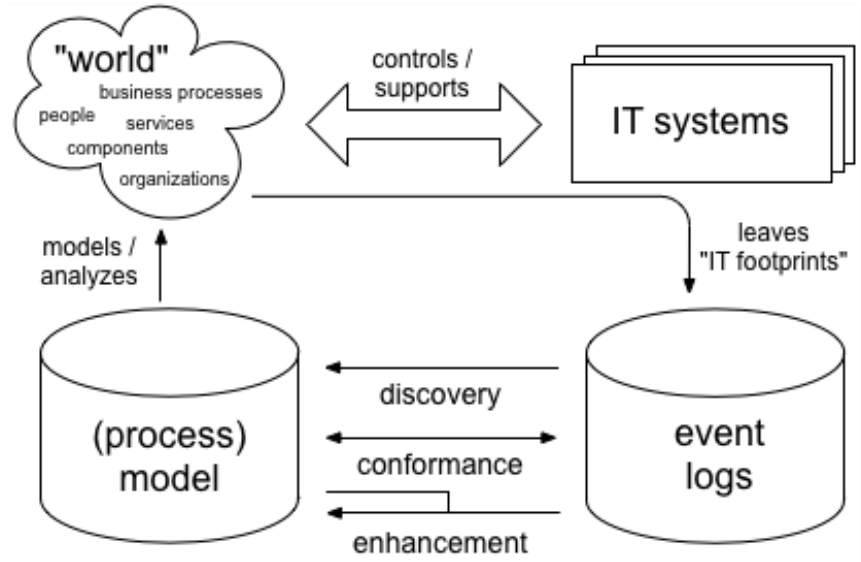


Figure 4 Process mining overview

Machine Learning and Classification

Machine Learning can also be described as the application of induction algorithms as one step in the knowledge discovery process [8].

One type of such an induction algorithm is a classification algorithm. Classification is described by [9] as the task of assigning an object to one of several predefined categories. More formally, a classifier takes a record with attribute set *x* and maps it to a class label *y*. The classes onto which each tuple is mapped are predefined, and can be either defined by a domain expert or as the result of an earlier clustering effort (Figure 5).

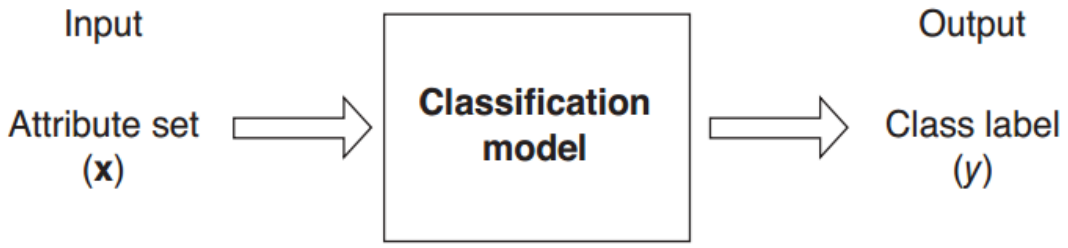


Figure 5 basic classification overview

Decision tree

There are various classification algorithms, each with its own strengths and weaknesses. One such algorithm is the decision tree learner [10]. The decision tree learner predicts the output class label using a set of independent attributes. To build the decision tree, the decision tree learner decides for each node on which attribute it should split the data. Various metrics can be used to decide how to split the data, and the most common metrics focus on the class impurity of the child nodes. A node with evenly distributed classes, an equal amount of records for each class, has a very high impurity. A child node with a very skewed class distribution has a very low impurity. One such impurity measure is GINI, which is computed as the sum of the squares of the class probabilities. Intuitively, GINI represent the probability that a randomly chosen element would be wrongly classified, if randomly labeled.

After building the decision tree learner, the resulting tree can be reduces in size using pruning methods. By pruning the tree, the tree is more generalized. This is a necessary step since decision trees are prone to overfitting [9].

Classification results

The results of a decision tree learner can easily be presented by means of a confusion matrix [8]. A confusion matrix is usually a 2 by 2 matrix, but the dimensions depend on the number of classes. A Confusion matrix shows how many data records have been correctly classified and how many records have been incorrectly classified. Table 1 is an example confusion matrix. In this example, *a* denotes the amount of records who's value should be true and have been classified true. *D* notes the records with value false which have been classified false. *B* is the number of records which have been predicted false, but are in fact true. These records are called False Negatives. *C*, also known as False Positives, are the records who have been predicted true but are in fact false.

Table 1 Example Confusion Matrix

		Predicted Values	
		True	False
Actual Values	True	*a*	*b*
	False	*c*	*d*

Regression models

Another method of analyzing the data is through the use of various regression models. Machine learning algorithms are used mostly to predict nominal values, such as gender. Regression models can be used to predict continues values and determine the relation between a dependent variable and a set of independent variables [11].

For example (example taken from [11]), suppose we encounter a data set with pairs of values (x_i, y_i). Using a scatter plot it is clearly visible that the value of y depends on the value of x. Using a regression analysis it would be possible to approximate the relation between *x* and Predictor *Y* as

$$E(Y|x) = \beta_0 + \beta_1 x + \epsilon$$

Where β_0 and β_1 denotes the regression coefficients, ϵ denotes the random error with mean zero and unknown variance σ^2. This is an example of a linear regression model.

Linear Regression

The example used above is an example of a simple linear regression. The prediction can be altered such that instead of a single predictor, x in the above example, multiple predictors are taken into account. This is called multivariate linear regression. Given a set of n predictors,

$$X = \{x_1, x_2 x_3, \cdots, x_n\}$$

The multiple linear regression model is given by,

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n + \epsilon$$

Where $\beta_0, \beta_1, \ldots, \beta_n$ denote the regression coefficients and ϵ denotes the random error with mean zero and unknown variance δ^2.

Polynomial Regression

Linear regression models can be very powerful to predict relationships that are linear with respect to the unknown parameter β. However, it is not always possible to model the relation between 2 variables in a strictly linear approach. Another regression model which could be used in such a case is a polynomial regression model. If we consider a two variable linear regression model, the model would be given by

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

The second degree polynomial for this equation would be [11]

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 \, x_2 \; + \epsilon$$

When creating a polynomial regression model, is it desired to obtain the lowest-degree model which is still consistent with the data. This is because a lower–degree model will be less complicated to understand and use.

Fitting distributions and Method of Moments

It may occur that a set of data points appears to follow a particular distribution. Knowing which distribution it adheres to, or what parameters are associated with this distribution can be useful.

One method of determining the parameters of a certain distribution is the method of moments. The method of moment uses the first moment and second moment of the data. The first moment equals the mean value of the data. The second moment can be calculated as the average of the squares of the data [11]. Using the method of moments, the parameter λ of the exponential distribution can be estimated by

$$\hat{\lambda} = \frac{1}{\overline{X_n}}$$

Where X_n denotes the first moment. To check whether the estimated fitting is any good, it is good practice to plot the original data against the Empirical distribution function and the probability density function of the predicted distribution. If the data and estimated fitting visually match, additional tests may be used. One such test is the Kolmogorov-Smirnov test [12] The Kolmogorov-Smirnov test is not that difficult to use and interpret. It basically computes the absolute maximum difference between the data and empirical distribution. If the difference exceeds a critical value, the null hypothesis is rejected and it is very unlikely that the data is taken from the tested distribution.

2.1 - Related Work

Data mining and process mining

Penders [13] created a methodology which can be used to predict the information need of an individual visitor of a website. The methodology uses various data mining and process mining techniques, and was assessed through 2 case studies. The first case study involved improving the classification of visitors on a mortgage website. The second case study resulted in the discovery of 2 clusters regarding visitors of the FAQ page.

Non-Parametric regression

Although linear and polynomial regression are not the same, they are very much related. Non parametric regression [14] is quite different from these two, and is lesser known, and more complex. Potentially it is a more powerful type of regression. Non parametric regression bases an estimation on a weighted average of predefined measurements [15]. This basically means that if the relation between the predictors and prediction is more or less known linear and polynomial regression should most likely be used. If the relation is unknown and/or requires a particular adjustment for a single variable, non-parametric regression could give better results.

Crooy [15] used non parametric regression in his thesis to predict the remaining duration of cases handled by an insurance company. Using non-parametric regression, he was able to create a model which was far more precise than when using basic predictors. A basic predictor used was the average case duration minus the care duration so far. Furthermore, the model could be created with no deep knowledge of the process for which it is used.

Time Series Classification

Time series are sequences of data points ordered in time and are thoroughly studied in classification problems. This is because Time series are a typical example of the type of data present in an OLAP database. For example, the amount of messages sent by a particular customer per month represents time series data. The average amount of symbols per message per month can be regarded as time series data. Various studies delve into efficient ways to classify time series data. For instance, the authors of [16] classify various time series by matching subparts of the time series, referred to as shapelets. This technique allows time series to be grouped by means of the matching subsequences. However, sometimes it is desirable to match time series data completely and not partly. [17] show that using Dynamic Time Warping yields very good results when classifying time series. Although in particular cases, simply using the Euclidian distance gives very good results. This is emphasized even more in [18] by showing that choosing the algorithm and distance measure necessary for good results depends on the actual data.

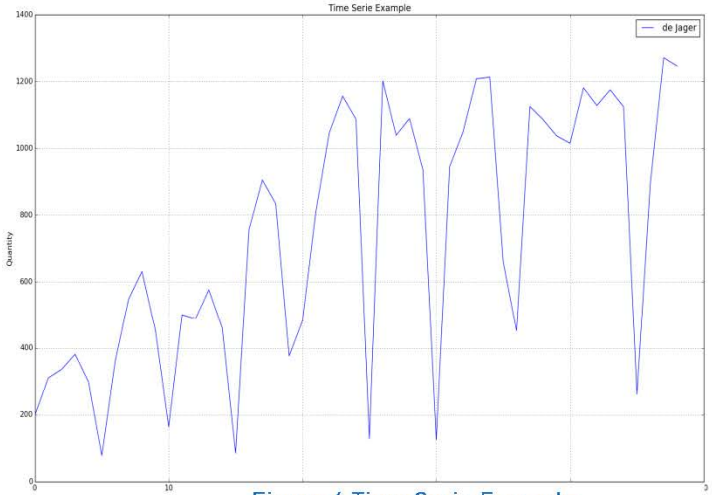


Figure 6 Time Serie Example

3 - Data Gathering & Cleaning

The goal of this project is to determine if we can predict customer loss for CM Direct. To achieve this, it is necessary to obtain data concerning the users and their usage of CM Direct. In terms of Business Intelligence, CM does not have a very sophisticated data warehouse, and although the system usage is logged, the data was not available. They do make use of different systems and combine various information sources into an OLAP server. This chapter will explain which data is present within various sources at CM, and what kind of cleaning was necessary for actual use.

3.1 - Customer process

Prospective customers using the services of CM Direct, all go through a similar process prior to using it. CM does not store process information. However, domain experts at CM are able to sketch the process each customer goes to prior to and while using CM Direct. This process is depicted in Figure 7. This process is also only used to gain insight into where prospective data can be found. It will not be used for validation purposes. Please note, this is a lightly simplified view of the process. For example, the process payment is not as straightforward as depicted here. Also, the outcome of customers support tickets is always conveyed back to the users. Furthermore, a single Support ticket can trigger a chain of emails through the company, depending on the actual problem. There are also multiple payment methods. These include (but are not limited to): Ideal, Paypal and creditcard.

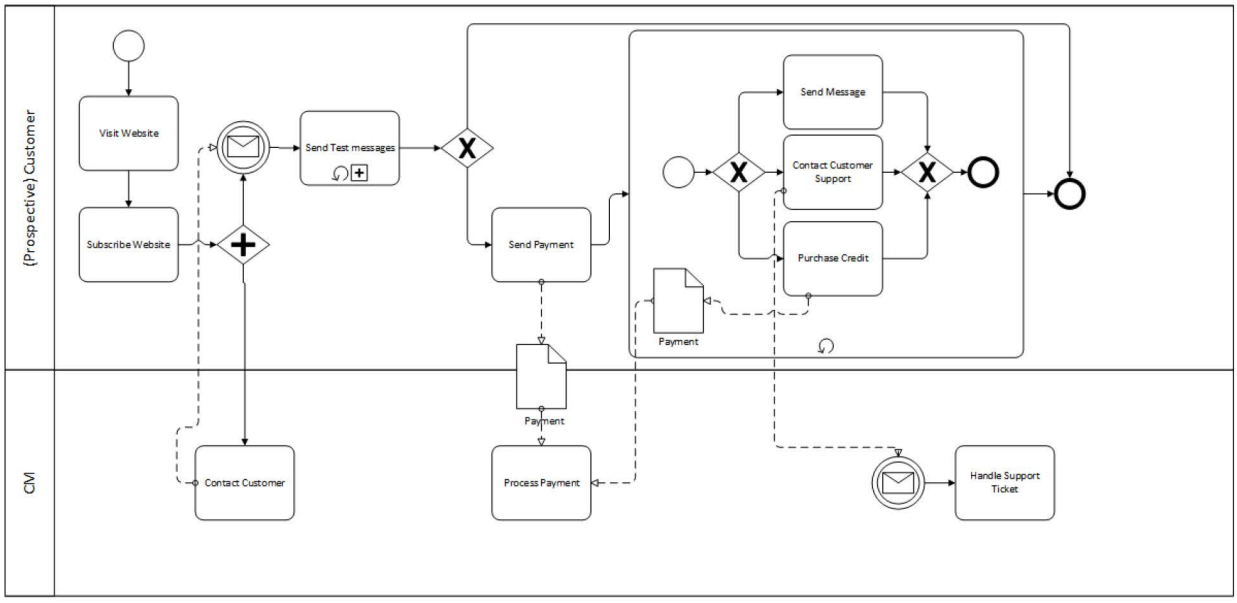


Figure 7 Customer Process

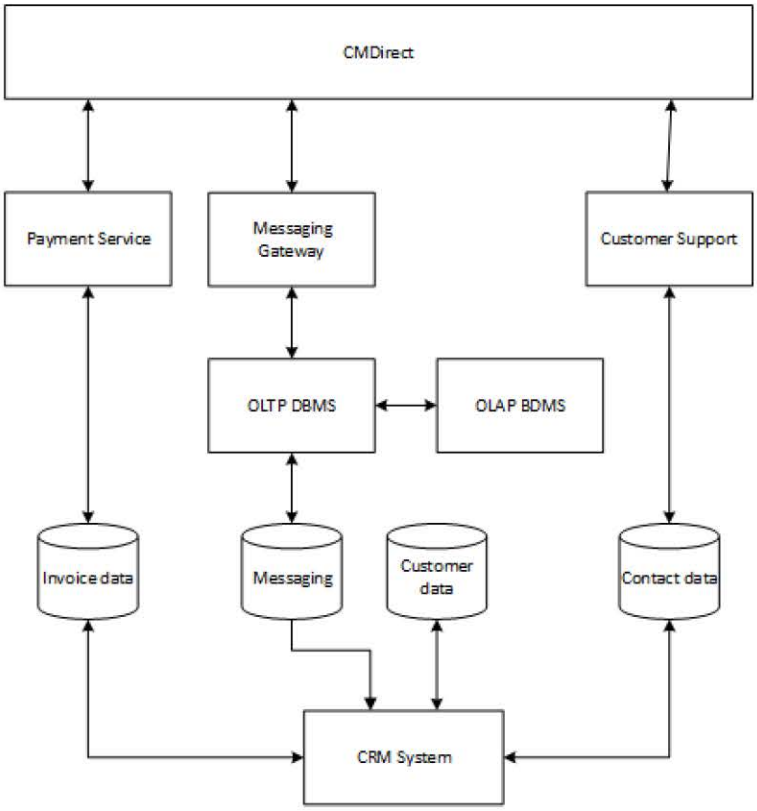
3.2 - Architecture

The customer only has to go through a single application to use the functionality of CM Direct. Some services, such as the payment service, are provided by external providers. But the services are incorporated within CM Direct, so the user experiences it as a single system. As is to be expected, the complete back-end at CM is more complicated. The payments information, messaging information, customer support tickets and account information are all stored in different databases. Furthermore, all data regarding the sent messages is stored in an OLTP database. The data in OLTP is processed and written to the OLAP database every 4 days. Figure 8 shows the party level architecture [19] as described.

Please note that this architecture is a simplified representation. The complete architecture is more complex, but is not necessary for this thesis to detail it further.

Customers can use the payments service to buy more credit, the payment service processes the payment and stores the transaction information in an invoice data database. The messaging gateway handles the outgoing messages sent by customers. Every sent message is also stored in the OLTP database, of which the aggregated data is transferred to the OLAP database every four days. Customer support tickets are stored in the Contact database. Finally, these three database are connected together with the account database at the CRM system, which provides an overview of all customer related information.

Figure 8 Party level architecture for CM Direct



3.3 – Data File

For analysis of the customer behavior, a data file is made based on the customer process. The architecture presented allowed us to retrieve information regarding the payments (invoice data), the sent messages (OLAP DBMS) and the support tickets (Contact data). The process in Figure 7 shows there are numerous steps which are taken by each customer. To represent the customer behavior in the data file, several events are defined. The mapping between the event appearing in the data file and the event stated in the process model are shown in Table 2. Please note that some events could be deduced that are not explicit in the process model. These events are Alter Account and Add User.

Table 2 Mapping of Log Event and Process Event

Event	Event in Process Model
Ingeschreven_Website	Subscribe Process
Wijzig Account	-

Gebruiker_toegevoegd	-
Stuur_Test_SMS_Dashboard	Send Test Message
Stuur_Test_SMS	Send Test Message
Stuur_SMS_Dashboard	Send Message
Stuur_SMS	Send Message
Eerste_Betaling	Send Payment
Betaling	Purchase Credit
Contact	Contact Customer Support

For each event, the following data is collected. Every customer using CM Direct has an account associated with that user. Each Account has a unique identifier, which will be referred to as AccountID

Table 3 Data Field per Event

Event	Data Field
Ingeschreven_Website	AccountID
	Date
	CompanyName
Wijzig_Account	AccountID
	Date
Gebruiker_Toegevoegd	AccountID
	Date
Stuur_Test_SMS_Dashboard	AccountID
	Date
	Status (can be either Accepted, Cancelled, Delivered, Failed, Rejected or Sent)
Stuur_Test_SMS	AccountID
	Date
	Status (can be either Accepted, Cancelled, Delivered, Failed, Rejected or Sent)
Stuur_SMS_Dashboard	AccountID
	Date
	Status (can be either Accepted, Cancelled, Delivered, Failed, Rejected or Sent)
Stuur_SMS	AccountID
	Date
	Status (can be either Accepted, Cancelled, Delivered, Failed, Rejected or Sent)
Eerste_Betaling	AccountID
	Data
Betaling	AccountID
	Date
Contact	AccountID
	Date
	TypeOfContact (Can be Email, Phone, EPIC(internal message board) or note)
	Person Handling the ticket

A part of the resulting event log is shown in Figure 9. Please note, some information may be altered due to privacy and/or non-disclosure agreements.

```
567218  27829,Contact,2014/07/01 00:00:00.000,Note,System
567219  27829,Ingeschreven_Website,2015/02/25 22:35:45.000,
567220  27829,Stuur_Test_SMS_Dashboard,2015/02/25 22:38:01.000,Delivered
567221  27829,Stuur_Test_SMS_Dashboard,2015/02/25 22:38:01.000,Delivered
567222  27829,Contact,2015/02/25 22:45:01.000,Webform, System
567223  27829,Account_Gewijzigd,2015/02/25 22:50:01.000,,
567224  27829,Stuur_Test_SMS_Dashboard,2015/02/25 23:05:03.000,Delivered
567225  27829,Contact,2015/02/26 13:37:34.000, Email, Wendy de Bruin
567226  27829,Eerste_Betaling,2015/02/26 22:07:43.000,
567227  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567228  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567229  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567230  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567231  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567232  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567233  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:05.000,Delivered
567234  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:06.000,Delivered
567235  27829,Stuur_SMS_Dashboard,2015/02/27 14:37:06.000,Delivered
```

Figure 9 Excerpt of the generated log file

3.4 - Enriched Event Log

For classification purposes, the data file has to be enriched with derivative information. This introduces some notions from process mining. The collection of all event regarding a single customer can be seen as a case. This also means that all events can be ordered chronological using the date information. This also allows the extraction of additional information, such as the ordering of events, the time between events and the time since or to a particular event. This kind of data is of no use for process mining, but can give classification algorithms an extra information dimension which could be very usefull. An example of this enriched log can be seen in Figure 10. Again, some information may be altered or removed due to privacy and/or non-disclosure agreements.

The following information was added to each event.

- Customer Status
  Whether or not the Customer is still using the service. This determined as:
  $$Status \equiv Sent\ Last\ message\ within\ the\ last\ 4\ weeks$$
- Event number within case
  All event are ordered chronology starting with the customer creating an account
- Event number within similar events within case
  Just as event are ordered since the creation of the account, the ordering within the collection of similar event is also stored. For example, the first payment, the second payment, the third payment etc.

For the Stuur_Test_SMS, Stuur_Test_SMS_Dashboard, Stuur_Test_SMS and Stuur_SMS_Dashboard event more information could be collected from the database, or deduced from the other events.

- **Mobile Network Operator**
  For example: NL Vodafone, NL T-Mobile or BE Proximus
- **EventDuration**
  Sum of CM Processing Delay, Provider Queuing Delay and provider Processing Delay
- **Event number since last payment**
  Chronological ordering of event since the last payment
- **Time between previously sent message**
  Time in seconds between this event and the previously Stuur_SMS (_Dashboard) event
- **Time since last payment**
  Time in seconds since the last payment
- **Time to next payment**
  Time in seconds to the next payment. This information is not always present since the next payment could be absent either because the user still has credits or has stopped using CM Direct
- **Average intersend times past week**
  The average of the times between all messages, which have been sent in the week prior to this message
- **Average intersend times past 2 weeks**
  The average of the times between all messages, which have been sent in the 2 weeks prior to this message
- **Average intersend times past 3 weeks**
  The average of the times between all messages, which have been sent in the 3 weeks prior to this message

For the events Eerste_Betaling and Betaling the following information fields are added:

- **Height of Payment**
- **PaymentMethod**
  can be Paypal, IDeal, Invoice,Offerte and Eénmalige machtiging

Table 4 shows an overview of which data fields was added to which event. Figure 10 shows an excerpt of the enriched data file.

Table 4 Summary of al added data fields

Event	Data Field	Explanation
Ingeschreven_Website	Customer Status	
	Event Number within Case	
	Event number within similar events within case	
Wijzig_Account	Customer Status	
	Event Number within Case	
	Event number within similar events within case	
Gebruiker_Toegevoegd	Customer Status	
	Event Number within Case	
	Event number within similar events within case	
Stuur_Test_SMS_Dashboard	Customer Status	
	Event Number within Case	
	Event number within similar events within case	
	Mobile Network Operator	
	Event Duration	
	Event Number Since Last Payment	
	Time since last sent message	
	Time since last payment	
	Time to next payment	
	Average Intersendtimes past 2 weeks	
	Average Intersendtimes past 3 weeks	
	Average Intersendtimes past 4 weeks	
Stuur_Test_SMS	Customer Status	
	Event Number within Case	
	Event number within similar events within case	
	Mobile Network Operator	
	Event Duration	
	Event Number Since Last Payment	
	Time since last sent message	
	Time since last payment	
	Time to next payment	
	Average Intersendtimes past 2 weeks	
	Average Intersendtimes past 3 weeks	
	Average Intersendtimes past 4 weeks	
Stuur_SMS_Dashboard	Customer Status	
	Event Number within Case	
	Event number within similar events within case	
	Mobile Network Operator	
	Event Duration	
	Event Number Since Last Payment	
	Time since last sent message	
	Time since last payment	
	Time to next payment	

	Average Intersendtimes past 2 weeks
	Average Intersendtimes past 3 weeks
	Average Intersendtimes past 4 weeks
Stuur_SMS	Customer Status
	Event Number within Case
	Event number within similar events within case
	Mobile Network Operator
	Event Duration
	Event Number Since Last Payment
	Time since last sent message
	Time since last payment
	Time to next payment
	Average Intersendtimes past 2 weeks
	Average Intersendtimes past 3 weeks
	Average Intersendtimes past 4 weeks
Eerste_Betaling	Customer Status
	Event Number within Case
	Event number within similar events within case
	Height of Payment
	Payment Method
Betaling	Customer Status
	Event Number within Case
	Event number within similar events within case
	Height of Payment
	Payment Method
Contact	Customer Status
	Event Number within Case
	Event number within similar events within case

Row ID	Event	Custom...	Custom...	EventDate	Eventn...	Eventn...	EventInfo	EventR...	EventD...	Eventn...	TimeBe...	TimeTill...	TimeTo...	AvgInt...	AvgInt...	AvgInt...
Row5	Stuur_Test_SMS	56	?	06 Oct 2011 14:45...	4	4	Delivered	?	?	?	?	?	?	?	?	?
Row9	Stuur_Test_SMS	56	?	06 Oct 2011 14:46...	8	8	Delivered	?	?	?	?	?	?	?	?	?
Row38	Stuur_SMS	66	FALSE	14 Oct 2011 13:11...	10	7	Delivered	NL - Vodafone	40	7	0	253661	?	36,116.571	36,116.571	36,116.571
Row40	Stuur_SMS	66	FALSE	14 Oct 2011 13:12...	12	9	Delivered	NL - Vodafone	67	9	0	253757	?	28,101.333	28,101.333	28,101.333
Row47	Stuur_Test_SMS	71	FALSE	13 Oct 2011 11:54...	1	1	Delivered	?	?	?	?	?	?	?	?	?
Row66	Contact	75	FALSE	17 Nov 2014 20:39...	10	5	Email	Wendy de Bruin	?	?	?	?	?	?	?	?
Row69	Contact	75	FALSE	18 Nov 2014 15:51...	13	8	Email	Ben de Haas	?	?	?	?	?	?	?	?
Row89	Stuur_SMS	80	FALSE	26 Sep 2012 11:32...	4	3	Delivered	NL - Vodafone	5	3	138	?	?	34,505	34,505	34,505
Row100	Stuur_Test_SMS	81	?	25 Oct 2011 14:51...	9	9	Delivered	?	?	?	?	?	?	?	?	?
Row106	Stuur_Test_SMS	81	?	25 Oct 2011 14:51...	15	15	Delivered	?	?	?	?	?	?	?	?	?
Row125	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	34	34	Delivered	?	?	?	?	?	?	?	?	?
Row144	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	53	53	Failed	?	?	?	?	?	?	?	?	?
Row157	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	66	66	Delivered	?	?	?	?	?	?	?	?	?
Row165	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	74	74	Delivered	?	?	?	?	?	?	?	?	?
Row170	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	79	79	Delivered	?	?	?	?	?	?	?	?	?
Row176	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	85	85	Delivered	?	?	?	?	?	?	?	?	?
Row177	Stuur_Test_SMS	81	?	25 Oct 2011 14:53...	86	86	Failed	?	?	?	?	?	?	?	?	?
Row180	Eerste_Betaling	81	?	25 Oct 2011 15:18...	89	0	15	PayPal	?	?	?	?	?	?	?	?
Row193	Stuur_SMS	81	?	25 Oct 2011 15:33...	102	12	Delivered	NL - T-Mobile	2	12	54	876	?	10.083	10.083	10.083
Row215	Stuur_SMS	193	?	02 Apr 2012 16:22...	20	18	Delivered	NL - Vodafone	1113	18	0	3999165	4313860	1,577.818	1,577.818	1,577.818
Row219	Stuur_SMS	193	?	02 Apr 2012 16:22...	24	22	Failed	NL - Vodafone	14445	22	0	3999168	4313857	1,157.267	1,157.267	1,157.267
Row239	Stuur_SMS_Dashboard	193	?	18 Nov 2014 08:29...	44	39	Delivered	NL - KPN	2	11	7238135	43522382	?	?	?	?
Row264	Stuur_SMS_Dashboard	193	?	18 Dec 2014 10:00...	69	64	Delivered	NL - Vodafone	3	36	0	46119817	?	17,414	17,414	17,414
Row266	Stuur_SMS_Dashboard	193	?	13 Feb 2015 12:46...	71	65	Failed	?	13	37	4934768	51054585	?	?	?	?

Figure 10 Excerpt from classification log file

4 – Cleaning

As stated before, the architecture at CM is very complex. The overview showed in Figure 8 shows the relevant components regarding the customer data, but is a very weak representation of the complete system. It is not relevant to fully disclose the architecture is this thesis, nor is it very feasible to do given the complexity after 15 years of continues development. Considering the complexity of the systems used, it is very likely that not all data gathered is in line with the process described in Figure 7. It is also very likely that there have been migrations in the past which can also have created anomalies in the data. These migrations can be physical, for example from one server to another. But a migration can also involve a shift in supporting systems, such as the accounting software.

The data is analyzed for such anomalies using Disco [20]. Disco is a process mining tool which can be used to generate a process model based on an imported log file. It is also possible to analyze the data from various angles. For example, statistics concerning the activities can be computed, such as frequency of an event in the log. But also information regarding the distribution of the resources can be computed. Using Disco, it was possible to extract every single trace and check for any inconsistencies in the data. From here on, unless stated otherwise, a case is the complete set of events for a single customer. A trace is a single case. The following inconsistencies were found, and traces containing these inconsistencies were removed for further analysis.

Violation and explanation	Amount of cases affected
No Eerste_Betaling event	17% of the cases
These customers should not have been considered as an active user (or inactive user) since apparently they never became a user. However it is also possible that the payments went wrong and have been handled differently than the other payments and do not appear in the invoice database. In either case it contaminates the log and should therefore be excluded	
Initial event is not Ingeschreven_website or Contact	22% of the cases
Some traces started with Stuur_SMS of Stuur_Test_SMS, it appeared that most of these cases took place quite a few years ago and were due to some kind of migration from a predecessor of CM Direct to CM Direct. Is was again unfavorable to keep these traces in the log, as there may have been events preceding these events of which there is no knowledge.	
Stuur_SMS and/or Stuur_SMS_Dashboard is at some point followed by Eerste_Betaling	2% of the cases
Is it difficult to pinpoint why a trace as this should occur, especially since all customers of CMDirect should pay beforehand. However, it does and since this is strange behavior, traces containing this combination of events will be excluded	
All of the violations stated above	39% of the cases
Apparently, some traces contain more than 1 kind of violation, since the percentage of violating traces combining all violations is less than the sum of the percentages of all cases with violation traces (39% versus 41%). However, the difference is only minor.	

4.1 – Customer Process

Even without the violating traces, the resulting data file still contains a large amount of information. The cleaned event log contains 359 different cases and 3578377 events.

Disco has implemented a fuzzy process mining algorithm as described by [2]. Since it can be interesting to know the processes of active and inactive customers, this feature of Disco is used to examine the processes of the both groups of customers.

Figure 11 shows the resulting process for active customers, mined using the fuzzy miner showing 75% of all paths. Figure 12 shows the process of the inactive customers, mined using the fuzzy miner showing 75% of all paths. The color of the events show the absolute frequency of the event in the data file. Even when only 75% of all paths are shown, the model in Figure 11 is pretty chaotic. Stuur_SMS is by far the most executed event, and it is possible for any event to be performed after this (and therefor also before this event). A similar process can be identified when trying to mine the process of the inactive customers. Although in this process (shown in Figure 12) Stuur_SMS_Dashbaord is the most prevailing event instead of Stuur_SMS. The remaining process looks like flower model, a type of model in which the order in which events occur is of no importance and can be done at any time.

In both models, it is possible to see some kind of discriminating behavior between the 2 processes. But as is shown by the colors of the events. Most event are rarely executed in comparison to the Stuur_SMS and Stuur_SMS_Dashboard events and there are but a few traces which are common among all customers, either active or inactive.

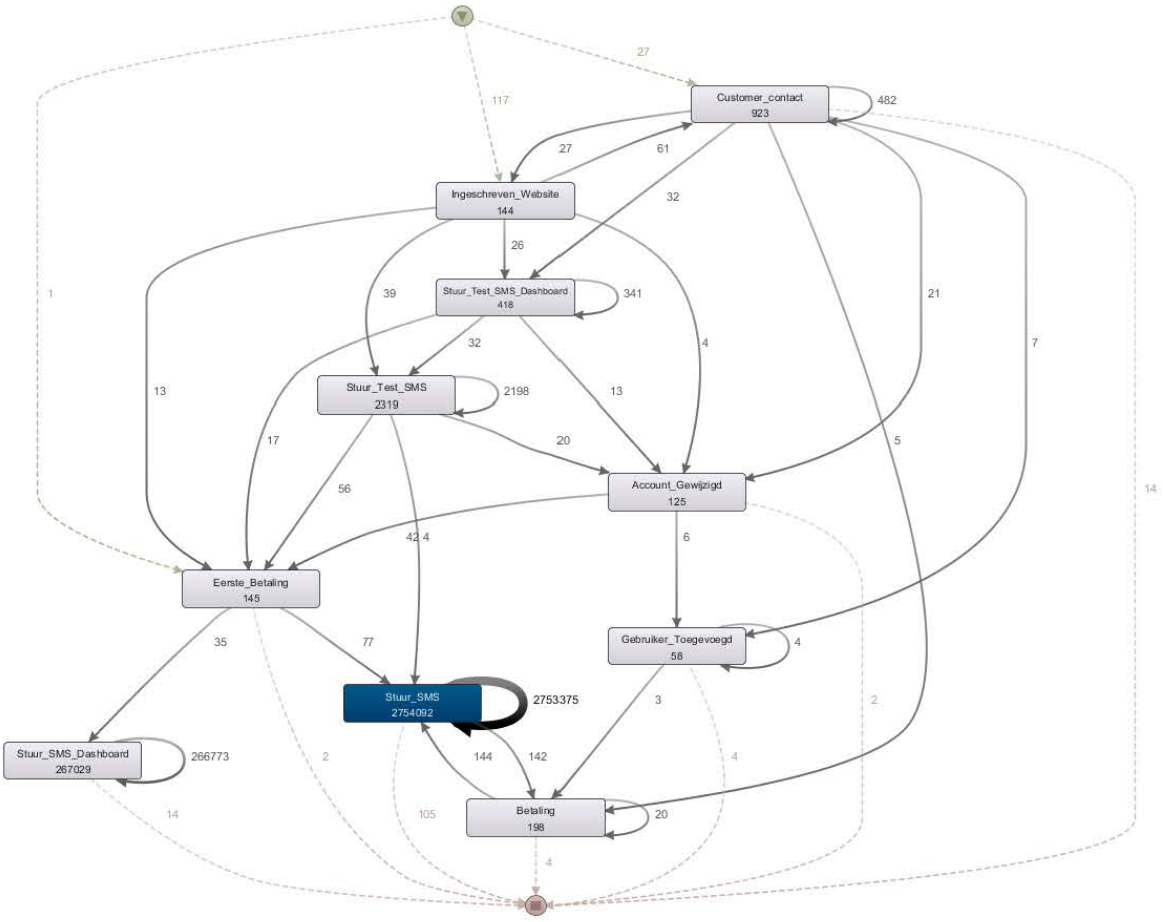


Figure 11 Customer Process

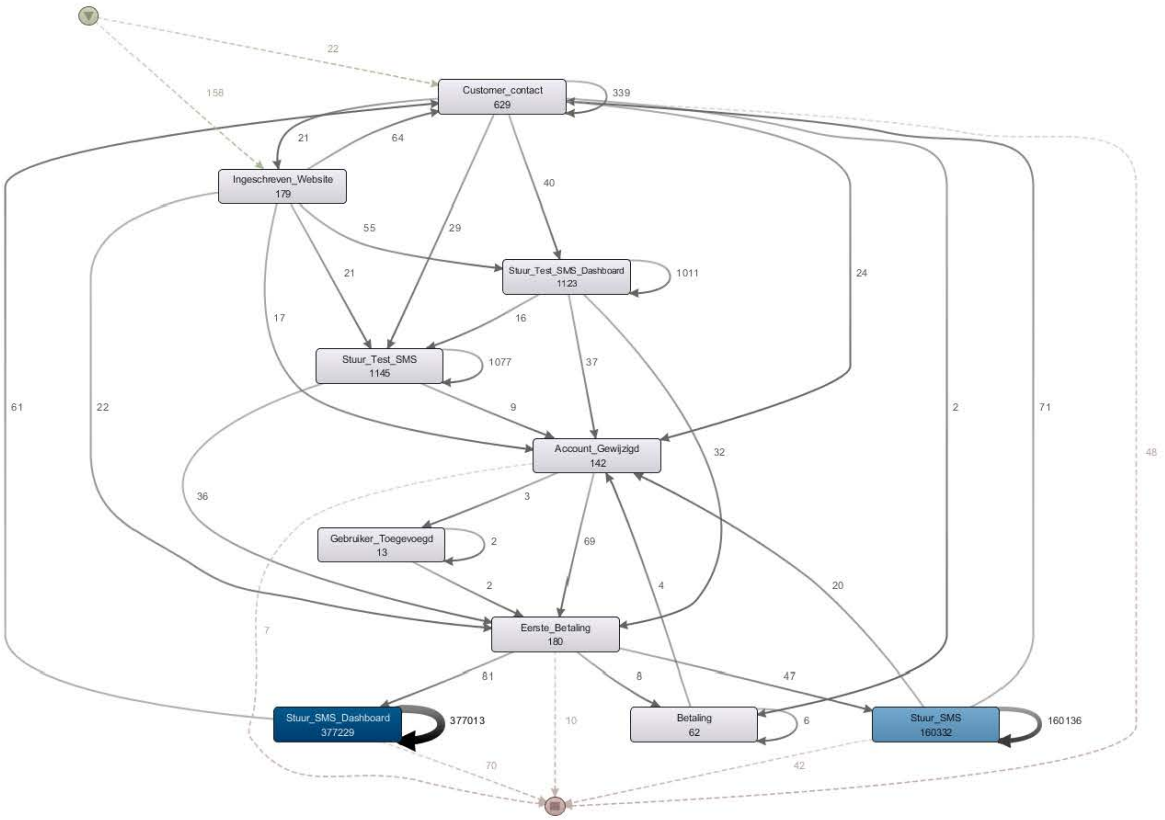


Figure 12 Inactive Customer Process

5 - Customer Characterization

At the beginning of every month, an employee of CM fetches time series data concerning the amount of messages sent every month for each customer using CM Direct from the OLAP database. It is also at this time that it becomes apparent that some customers have stopped using the services provided by CM. Through experience, CM believes there are 3 different types of customers. In this chapter, the data file will be examined to determine if the customer types used by CM can actually be confirmed by the data.

The customer types are:

- Single time users

  There is a group of customers who use CM Direct to send a single message to a group of people, only to never use CM Direct again. This could be for example to promote the opening of a new shopping mall. Another example is to inform customers of a car shop of a sudden call back action.
- Continuous user

  This group of customers actively uses text messages in their company processes and send messages at a constant rate. For example, a taxi company which uses text messages to inform its drivers about new appointments. Such a company is likely to send messages around the clock.

  Using the monthly OLAP update, CM is able to determine the 5 largest customers. Therefore, a fourth group could be identified, Large continuous users. However these customers are in fact the same as continuous users. CM does not see them as any different.
- Periodical Users

  This group of users is like the One day users, only instead of using the service a single time, it will be used more often but not in a continuous nature. For example, consider a supermarket which sends a text message to its regular customers every month. It will use the service exactly once a month, and the text messages are not incorporated in the daily processes.

Using Disco, it was fairly easy to split the data file into 2 parts. One part contains all traces concerning still active customers and the other has all traces concerning inactive customers. The activity statistics [20] show an interesting difference between the 2 groups. It appears that the group of still active customers primarily sends text messages, and barely use the Dashboard. In the log made on April 21th the events *Stuur_SMS* and *Stuur_SMS_Dashboard* took up 91% and 8.9% of all events respectively. The distribution is shown in Table 5. The group of inactive customers shows a significantly different ratio. For this group the *Stuur_SMS* event took up 29.6% whereas the *Stuur_SMS_Dashboard* took up 69.7% of all events. Table 6 numerically shows this distribution. Furthermore, for both groups it holds that these events (*Stuur_SMS* and *Stuur_SMS_Dashboard*) take up 99% of the event log. The process models showed the same behavior, but is now confirmed by the data.

Table 5 Distribution(in numbers) of Events in Customer data file

Activity	Frequency	Relative frequency
Stuur_SMS	2754092	91.03%
Stuur_SMS_Dashboard	267029	8.83%
Stuur_Test_SMS	2319	0.08%
Customer_contact	923	0.03%
Stuur_Test_SMS_Dashboard	418	0.01%
Betaling	198	0.01%
Eerste_Betaling	145	0%
Ingeschreven_Website	144	0%
Account_Gewijzigd	125	0%
Gebruiker_Toegevoegd	58	0%

Table 6 Distribution(in numbers) of Events in inactive Customers data file

Activity	Frequency	Relative frequency
Stuur_SMS_Dashboard	377229	69.72%
Stuur_SMS	160332	29.63%
Stuur_Test_SMS	1145	0.21%
Stuur_Test_SMS_Dashboard	1123	0.21%
Customer_contact	629	0.12%
Eerste_Betaling	180	0.03%
Ingeschreven_Website	179	0.03%
Account_Gewijzigd	142	0.03%
Betaling	62	0.01%
Gebruiker_Toegevoegd	13	0%

As stated before, a case is the complete set of events of a single customers. The time between the first event, and the last event recorded for each customer is the case duration. When examining the case durations, the case durations show a very high initial peak. Figure 13 shows the case distribution for both groups, the red bars denote the inactive customers and the green bars represent the active customers. The longest case of the group of inactive customers takes 1 year and 196 days. For the still active customers this is 3 years and 84 days. It should be noted that these durations are based solely on the *Stuur_SMS* and *Stuur_SMS Dashboard* events since there were some customers with an extremely long case (up to 9 years) because they had an email conversation years before sending actual messages.

The initial peak seems to imply that, at least for the group of inactive customers, customers tend to try CMDirect for some period of time and then decide not to buy any more credits. The group of active customers shops a similar curve, although the graph is less centered around this first peak. This initial peak contains quite a large group of customers who have just started using the service, and of who it will remain unclear whether or not they will continue using the service for some time.

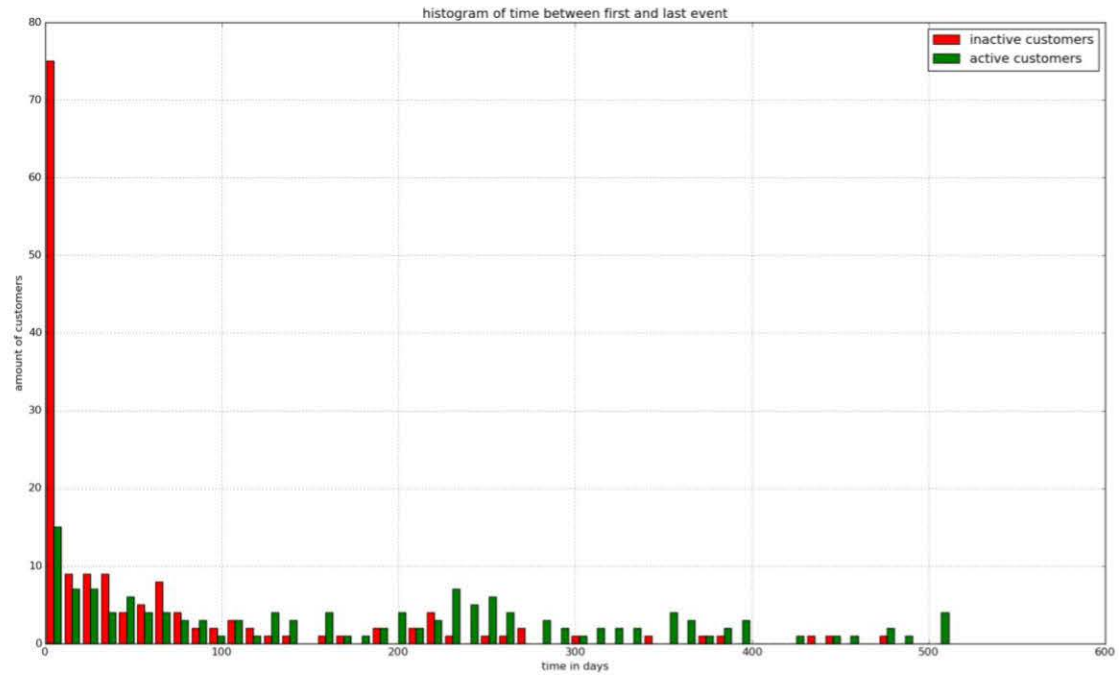


Figure 13 Distribution of the Case duration for the group of inactive and active Customers

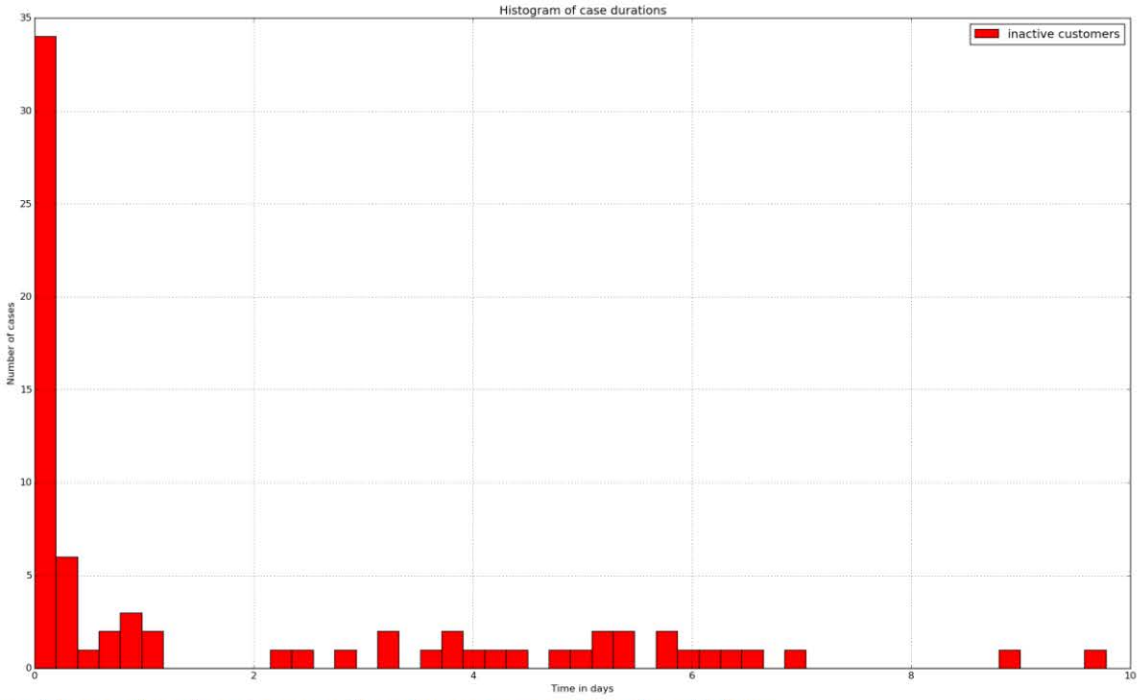


Figure 14 case duration of cases of inactive customers less than 10 days

Furthermore, there appears to be a group of customers who only use the service for a single time. A manual inspecting of the log file showed that this group indeed exist. Figure 14 shows the distribution of case duration of cases shorter than 10 days of the inactive customers. This distribution clearly shows there is quite a large group of customers who have used CM Direct for only a single time.

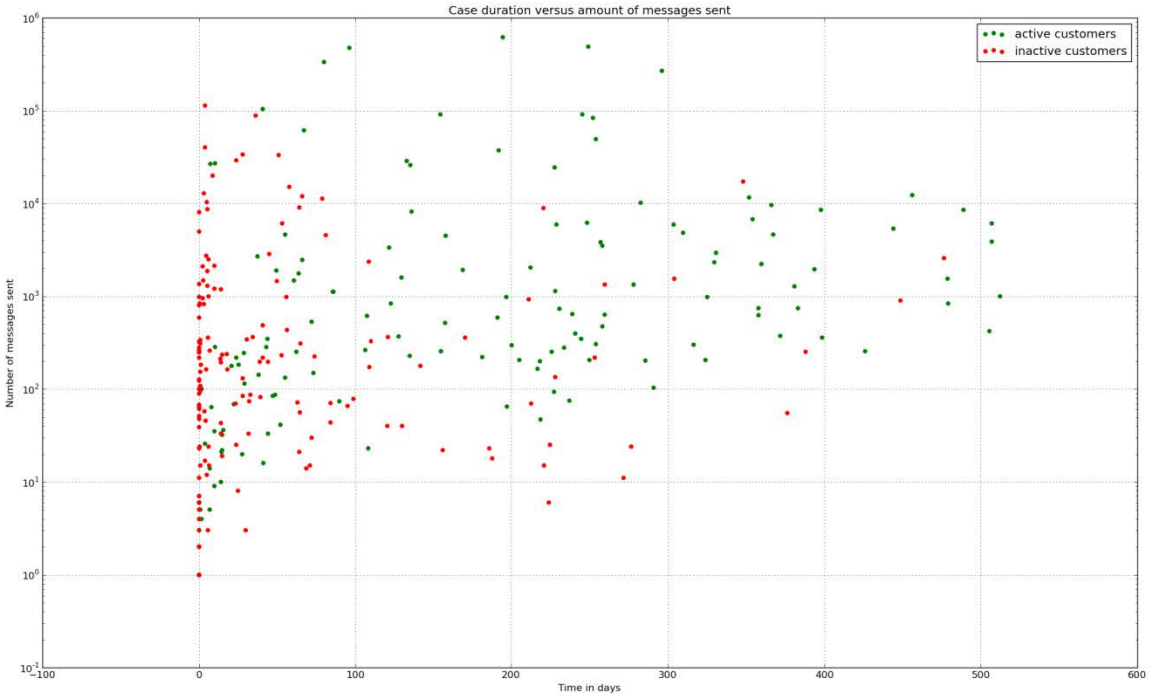


Figure 15 overview of case duration and amount of messages sent

Each month, CM reports the 5 largest customers. Figure 15 shows a scatter plot, with on the X axis the case duration, and on the y axis the amount of messages sent. Each dot represent a customer, and the color depicts whether it is an active or inactive customer. In this plot, 5 customers clearly stand out as being the largest (active) customers. The difference between the smallest biggest customer, and the next largest customer is a big as 15000 messages. These customers clearly stand out.

CM also makes a distinction between continuous senders and periodical senders. To characterize these 2 different groups, the intersend time between messages is examined. As stated before, the intersend time is the time between 2 consecutive messages, and is a very useful metric for this purpose. If a customer is a periodical user, the time between messages is either very short because the message is part of one of the batches that have been sent, or quite large, because it is the first of the batch and the previous batch is some time ago. If a customer is a continuous user, the intersend time probably varies more assuming that this user does not send as many batches of messages as a periodical user. But the intersend time never does get as large as compared to a periodical user. Figure 16 and Figure 17 are added to illustrate this difference. Figure 16 shows the histogram and the empirical cumulative distribution function of what could be considered a typical periodical customer. Is has sent a few batches, most of which we 5 days apart and one just over 25 days.

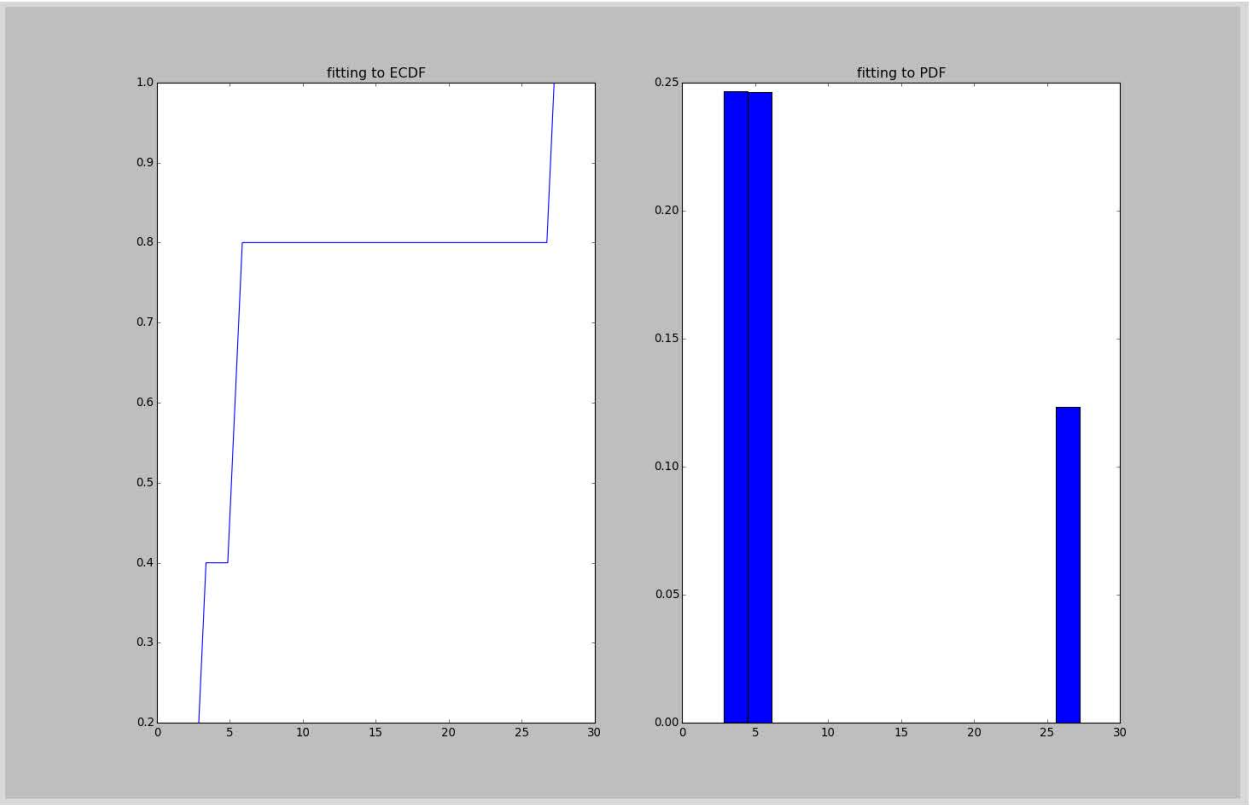


Figure 16 Example periodic customer

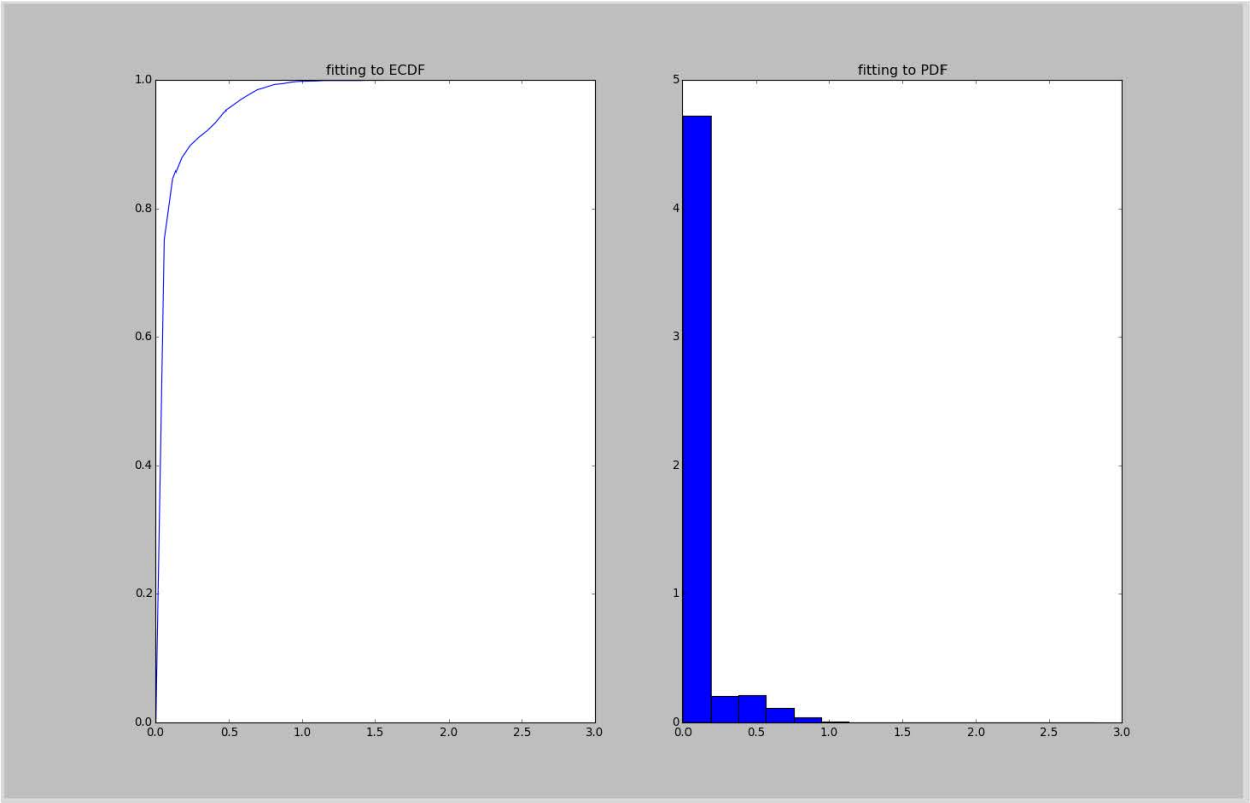


Figure 17 Example continuous customer

Figure 17 shows the histogram and ECDF of a typical continuous customers. Most of the messages are sent very close to each other, and some with more time in between them

The histogram of the intersend times are very different for a periodical customer and a continuous customers. Figure 18 Intersend times histograms of active customers shows the histogram of every active customer in a single plot. The green bar (middle section) is the percentage of intersend times smaller than 24 hours. Intuitively, this represent the amount of messages sent within a day of another message. The left part shows the percentage of intersend times large than 24 hours. It is split into 4 parts. The red part represents the intersend times between 24 hours and 48 hours. The blue part are the intersend times between 48 hours and 72 hours. The part colored yellow, is the percentage of intersend times larger than 72 hours and smaller than 7 days. The last part, colored in magenta, represents the intersend times larger than 8 days. The right section shows how many messages have been sent by this customer in total. The scale of the right section is logarithmic, the other scales are linear.

In Figure 18 it is visible that most of the intersend times are within the 72 hours range, thus a lot a messages are sent within 3 days of each other. It has not been defined by CM how long the intersend times have to be to be considered a periodical sender. However, intuitively 3 days seems a little too short to be deemed periodical. Furthermore, intersend times bigger than 8 days are mostly located with customers who have not sent a lot of messages. But is should be stated that there indeed appear to be continuous and periodical customers, although the continuous customers are the bigger group and the periodical customers are mostly located in the group of customers who have not sent a lot of messages.

Figure 19 shows the same information as Figure 18, but with the data of the inactive customers. A big difference is that the inactive customers have a larger portion of intersend times larger than 8 days, although about 40% of the inactive customers have sent only up to 100 messages. Thus the question remains how relevant the group of periodical customers is.

To conclude, CM stated it has three types of customers. Singe time users, continuous users and periodic user. By analyzing the data, we have found that all three user types actually exist, but the single time users form a relatively small group, and the periodically customers do not appear to be sending a very large amount of messages. The continuous customers form the majority group, and also include most of the customers who have sent over 1000 messages.

Thus customers can be divided into 4 groups. A short summary of the established groups can be seen in Table 7.

Table 7 Customer groups

Group	
Single time users	Time between first and last send message is 1 day or less
Periodical Users	Group of customers who sent messages in batches once a week, or in even larger intervals.
Continuous users	The vast majority of users
Large users	Top 5 users of CM Direct, together they are responsible for over 50% of all sent messages

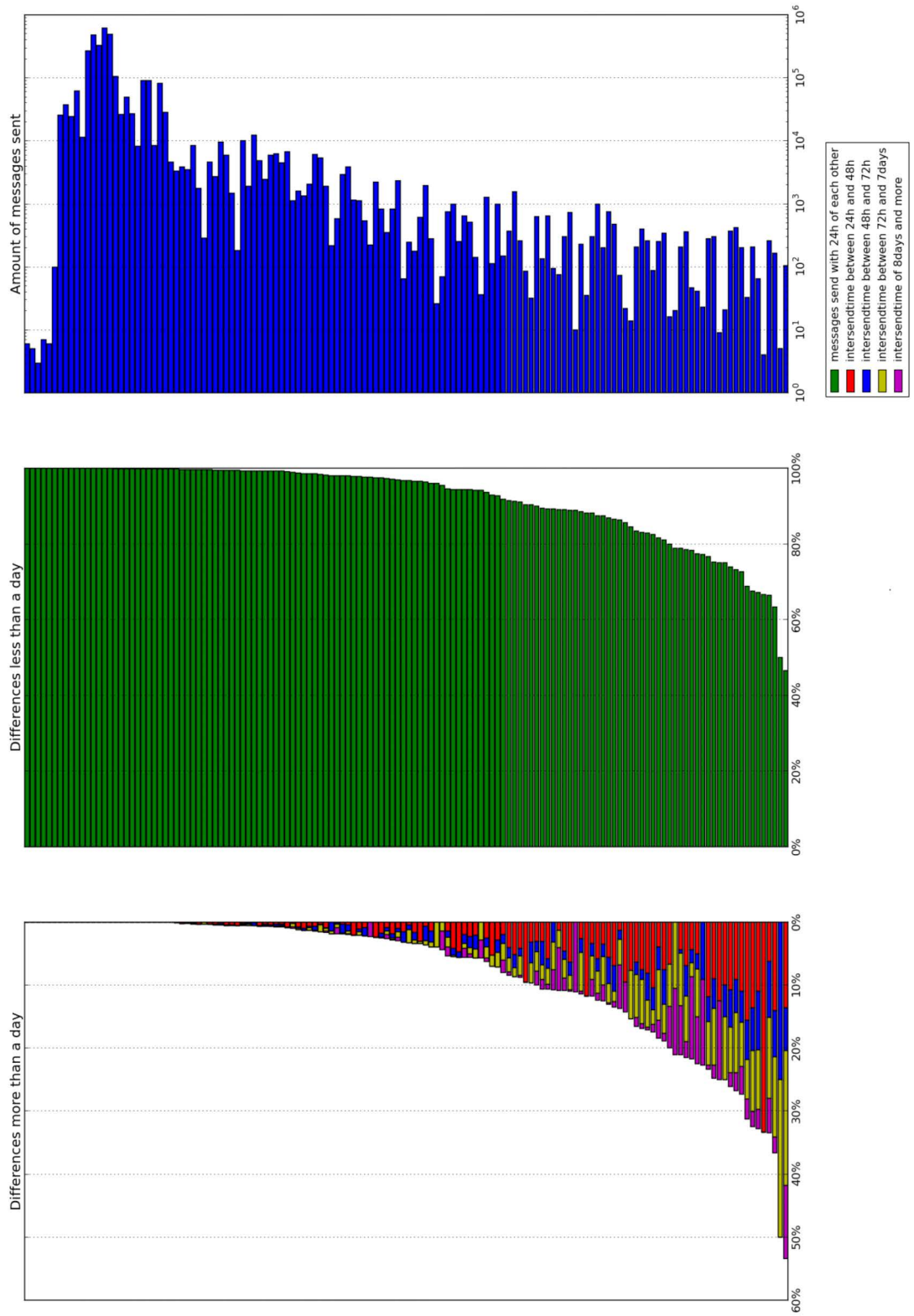


Figure 18 Intersend times histograms of active customers

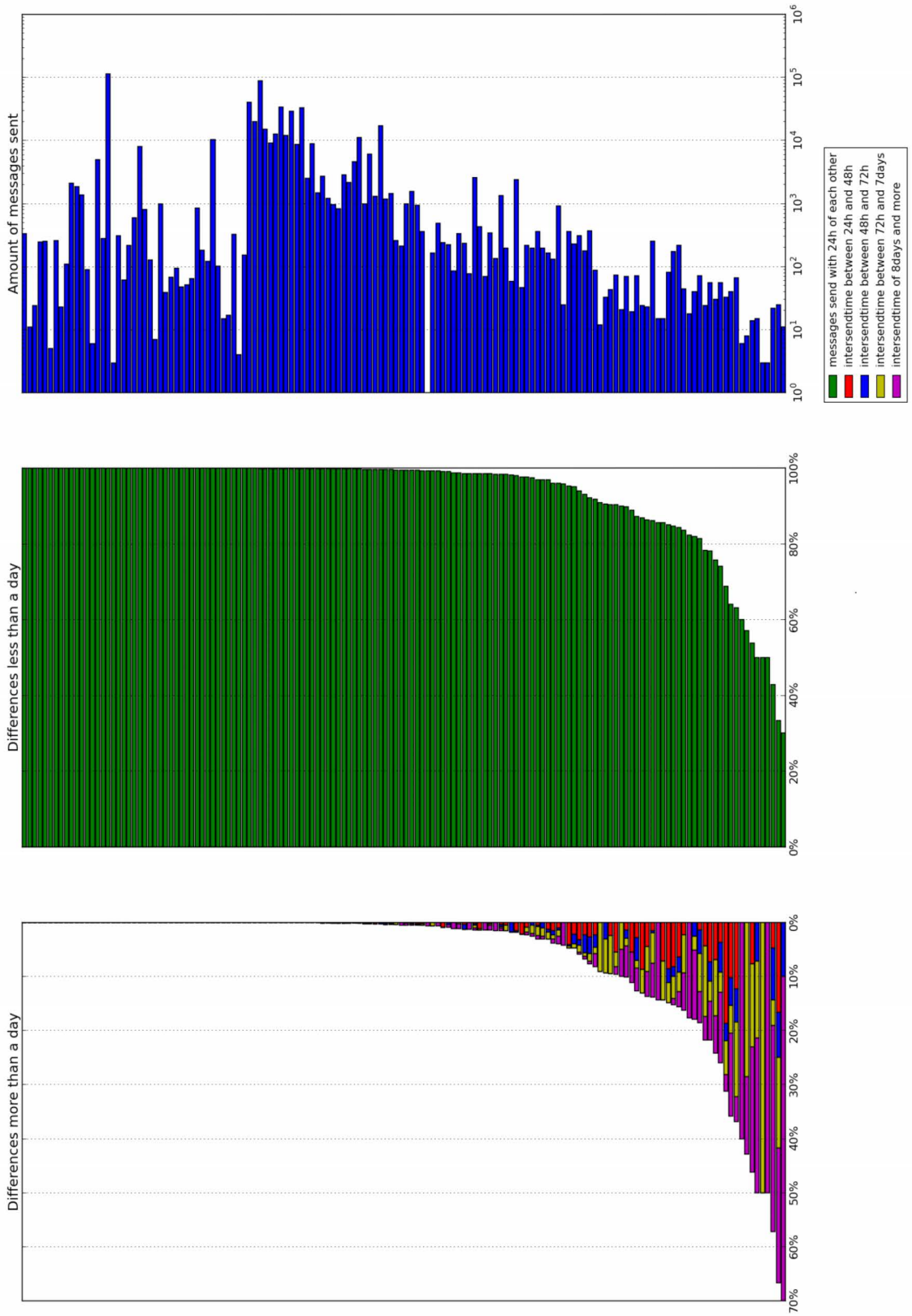


Figure 19 Intersend times histograms of inactive customers

6 - Prediction

The main question of this thesis is if it is possible to predict whether or not users will continue using the services offered by CM Direct. This is answered by examining 3 related questions. Is it possible to predict the Time to Next Payment, is it possible to predict if customers are going to pay and it is possible to predict customer loss.

6.1 - Predicting Time to Next Payment

The enriched log file contains a field stating the time (in seconds) to the next payment. We determine if it is possible to predict this value, by use of the other values in the log. The log is also further enriched by added the customer classification, as discussed in the previous chapter.

In order to predict the time to the next payment, 2 different regression models are used. Both of which are implemented in KNIME [21]. The 2 regression model are multivariate lineair Regression and polynomial regression. Furthermore, a third very simple predictor is created to check if the problem could be solved by some very basic logic.

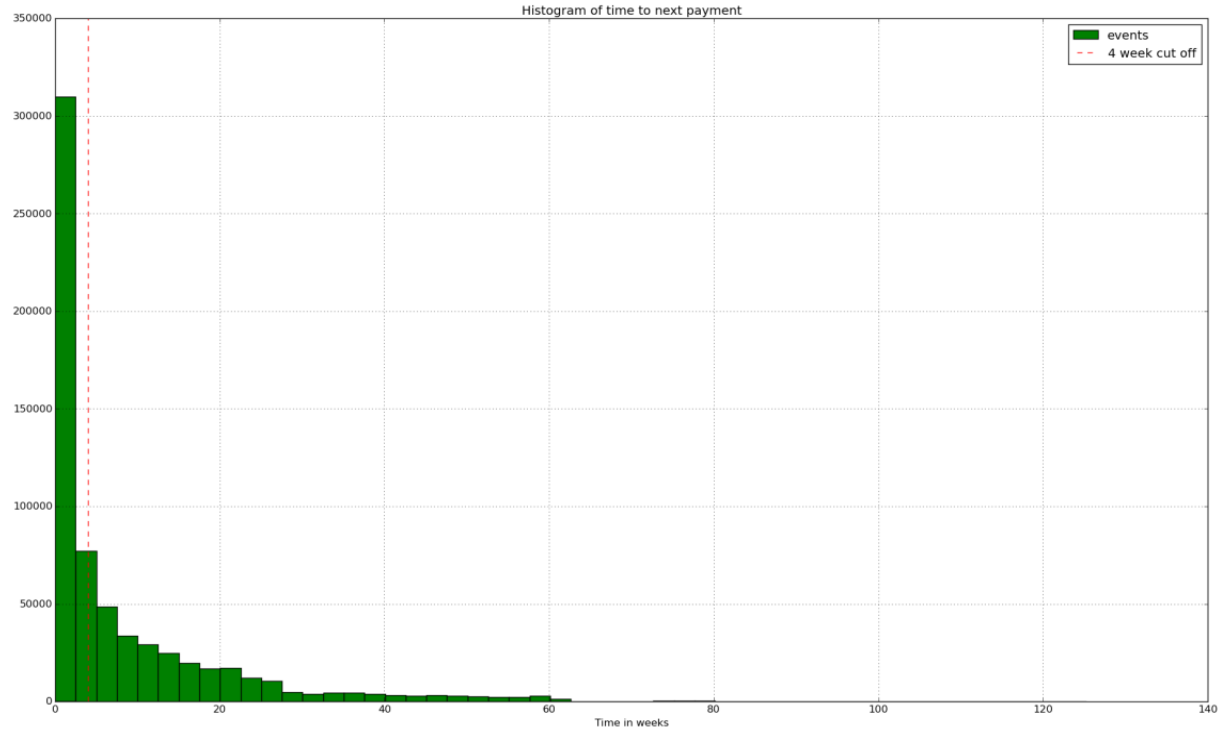


Figure 20 Histogram of Time to Next Payment

Figure 20 show a histogram of the Time to Next Payment. The distribution appears to be exponential. It is good to know if the Time to Next Payments actually follows an exponential distribution, if so it will be quite difficult to make predictions about the Time to Next Payment due to the lack of memory property of the exponential distribution.

Figure 21 shows the results of trying to fit an exponential distribution to the Time to Next Payment. To estimate the parameter λ, the method of moments was used. The fit is not good, although the data does look like an exponential distribution. The result indicate that the initial peak is higher than is to be expected and the remaining data is not as high as the exponential counterpart. A Kolmogorov-Smirnov

test returned a test value of 0.70 and a p-value of 0.0. This is another indication that the Time to Next Payment does not follow an exponential distribution. It furthermore does not appear to follow any distribution, which will make it very hard to predict using the before mentioned regression models.

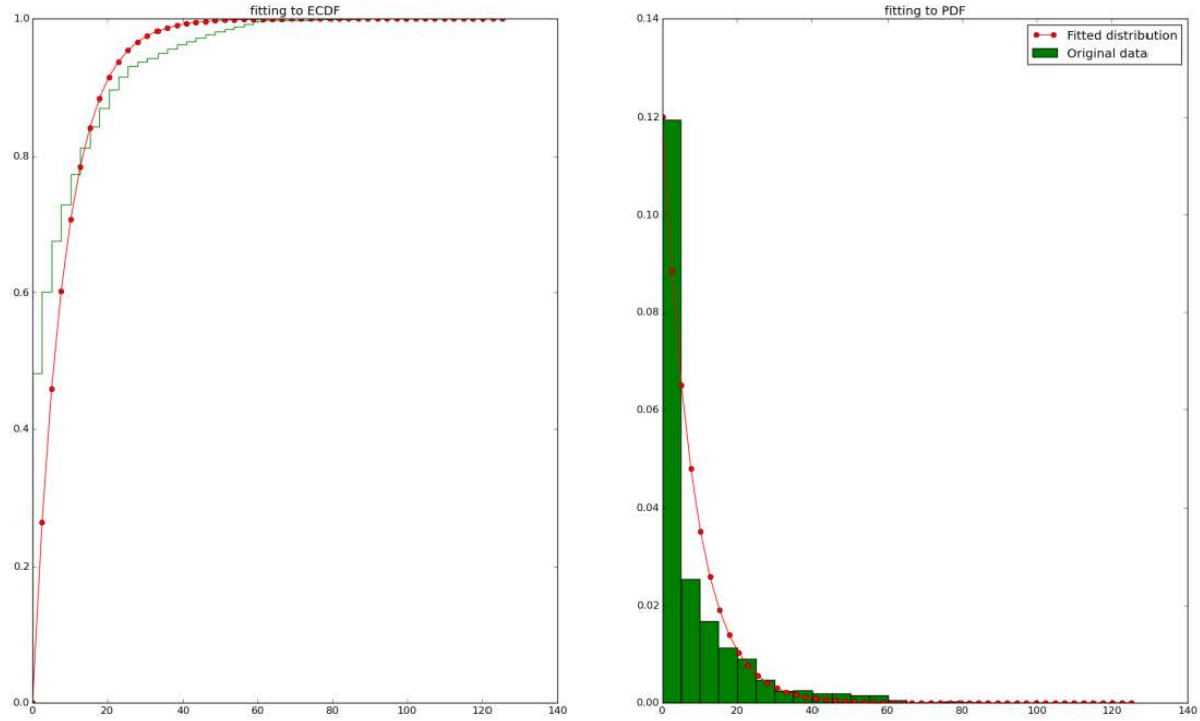


Figure 21 Fitting Time to Next Payment to exponential distribution

Simple Predictor

The simple predictor works by taking all payments per customer and determine the time between each pair of consecutive payments. Next, the time between each consecutive payment is divided by the height of the first of the two payments after which the mean over these ratios is computed. This number represent the average time per paid currency. The prediction for the next payment is then computed by multiplying the earlier computed mean by the height of the latest payment. For each event, the expected time to Next payments simply is the expected time of the next payment minus the time till the last payment. To explain the simple predictor, consider the following example. A customer pays 20 euros every 4 weeks, the customer spends 20/4 euros per week. If the latest payment was again 20 euros, we expect this customer to spend this 20 euros in 4 weeks. The predicted time to next payment for each event is 4 weeks minus the time till the last payment. This predictor should be able to cope very well with very constant customers. However, customers who always buy the same amount of credits but vary in the time they need to use it are likely to perform very bad under this predictor.

Results

Using KNIME, the 2 regression models were trained on 80% of the data, and tested using the remaining 20%. The basic predictor could be implemented and run using the Python programming language.

Figure 22,Figure 23 and Figure 24 show the result of the lineair regression model, polynomial regression model and the basic predictor respectively. Table 8 and Table 9 show the values of the regression coefficients. If all values were correctly predicted, the scatter plots should show a single straight line

(since predicted value equals the actual value). This line is added to all the plot as a red line. Unfortunately, neither model has this. The lineair and polynomial regression mode both show a cloud of point around the line Prediction(Time to Next Payment) = 10000. The basic predictor shows multiple straight lines, most of which indicate a wrong prediction.

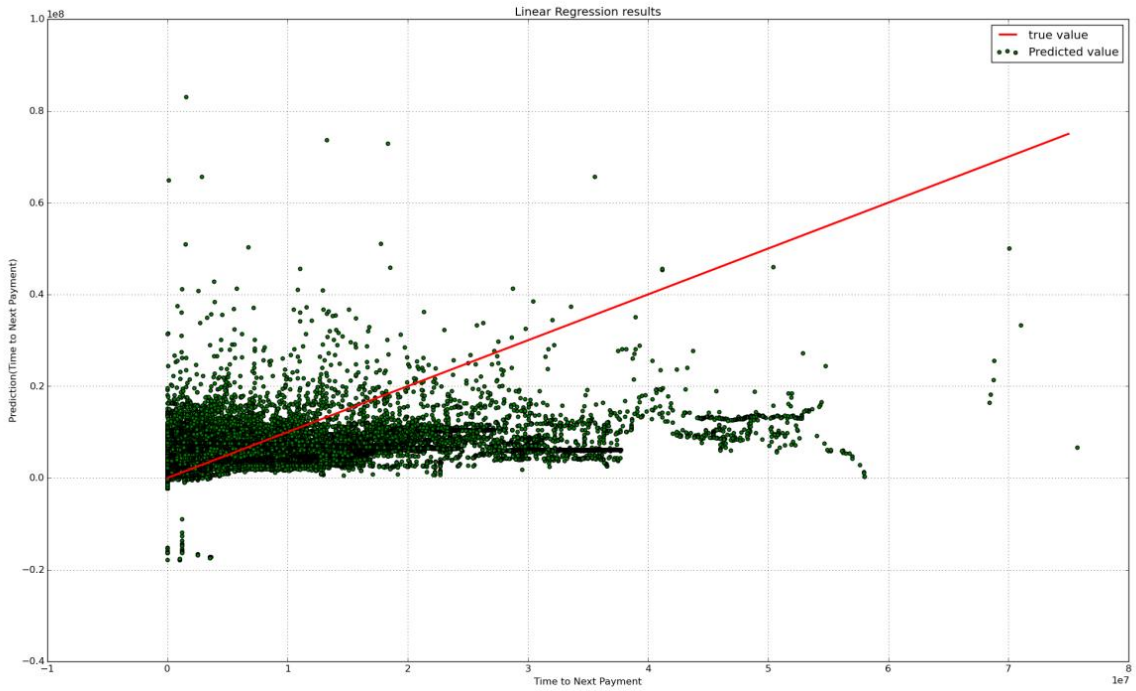


Figure 22 Result of Lineair regression model

Table 8 Linear regression models statistics and coefficient values

Variable	Coefficient	Std. Error	t-value	P>|t|
Event=Stuur_SMS_Dashboard	-1044040	50389.82	-20.7193	0
EventnbWithinCase	-40151.4	382.6296	-104.935	0
EventnbWithinSimilarEvents	40109.52	382.6289	104.8262	0
EventInfo= Cancelled	3042941	748661.6	4.064508	4.81E-05
EventInfo= Delivered	-1363740	747752	-1.82379	0.068185
EventInfo= Failed	-2534485	749210.1	-3.38288	7.17E-04
EventInfo= Rejected	-1006023	776383.3	-1.29578	0.195051
EventInfo= Sent	2050741	1123276	1.825679	0.067899
EventDuration	-7.39747	0.653855	-11.3136	0
EventnbSinceLastPayment	-26.4491	1.426984	-18.535	0
TimeBetweenPreviousEvent	2.307375	0.588162	3.923024	8.75E-05
TimeTillLastPayment	0.091531	0.001374	66.61132	0
AvgIntersendTimesPreviousWeek	25.87968	1.481413	17.46959	0
AvgIntersendTimesPrevious2Weeks	40.28858	1.916519	21.02175	0
AvgIntersendTimesPrevious3Weeks	54.29268	1.540876	35.23495	0

Intercept	7485744	747965.5	10.00814	0

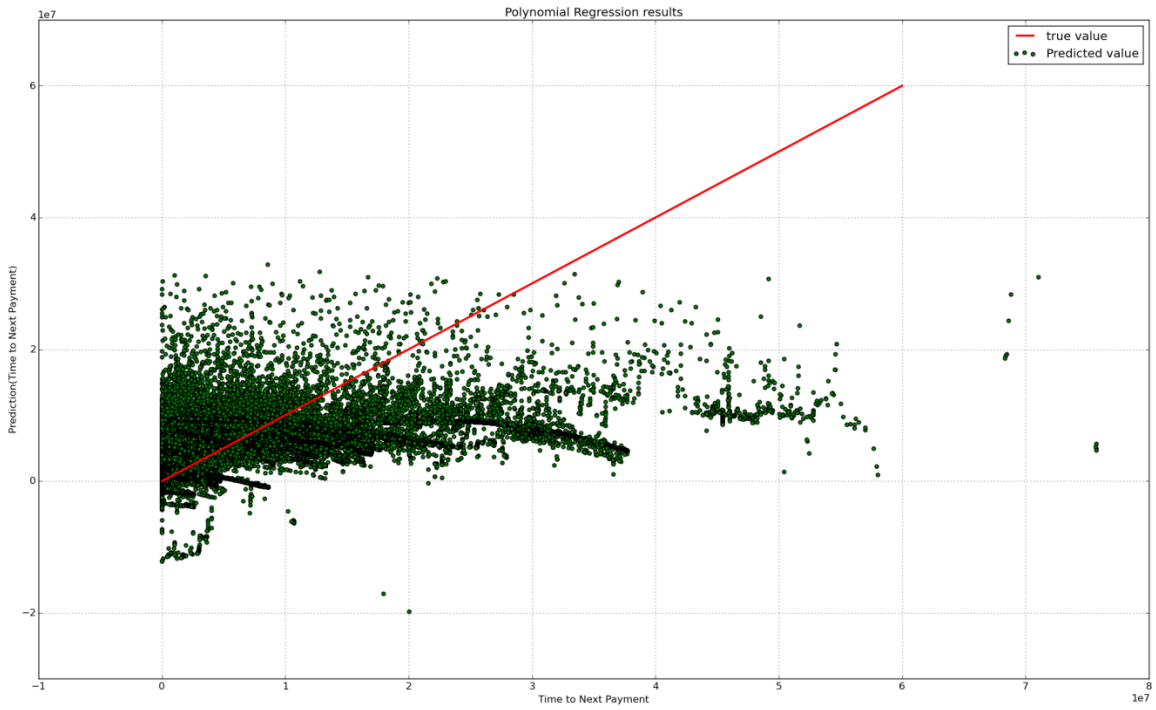


Figure 23 Result of Polynomial Regression Learner

Table 9 Polynomial regression model statistics and coefficients

Variable	Exponent	Coefficient	Std. Error	t-value	P>|t|
EventnbWithinCase	1	-22749.4	440.2158	-51.6777	0
EventnbWithinSimilarEvents	1	22794.31	439.7591	51.83362	0
EventDuration	1	-42.6349	1.678504	-25.4006	0
EventnbSinceLastPayment	1	149.7847	3.452674	43.38223	0
TimeBetweenPreviousEvent	1	17.10358	1.135994	15.05605	0
TimeTillLastPayment	1	0.361823	0.00278	130.1346	0
AvgIntersendTimesPreviousWeek	1	58.81431	2.608123	22.55043	0
AvgIntersendTimesPrevious2Weeks	1	47.48513	3.754252	12.64836	0
AvgIntersendTimesPrevious3Weeks	1	93.41827	3.085099	30.28048	0
EventnbWithinCase	2	-1.16784	0.012342	-94.621	0
EventnbWithinSimilarEvents	2	1.167296	0.012336	94.62218	0
EventDuration	2	2.37E-04	1.62E-05	14.67389	0
EventnbSinceLastPayment	2	-0.00612	9.22E-05	-66.3234	0
TimeBetweenPreviousEvent	2	-4.48E-05	3.15E-06	-14.2267	0
TimeTillLastPayment	2	-8.09E-09	6.81E-11	-118.735	0
AvgIntersendTimesPreviousWeek	2	-1.08E-04	6.15E-06	-17.5535	0

AvgIntersendTimesPrevious2Weeks	2	-1.64E-04	8.87E-06	-18.4964	0
AvgIntersendTimesPrevious3Weeks	2	-1.48E-04	6.07E-06	-24.3398	0
Intercept	0	4271883	25065.34	170.4299	0

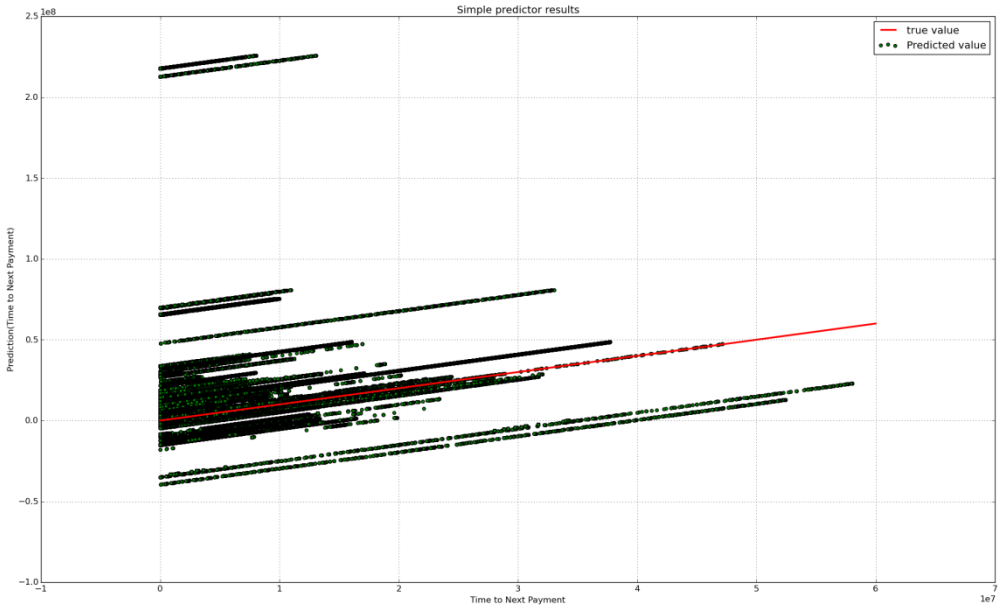


Figure 24 Result of Basic Predictor

There are some interesting observation that can be made from these results. First, the simple predictor manages to predict a negative time to next payments. This happens in some cases because the time between payments is highly irregular. For example, if the time between the first 2 payments was 1 year, and the time between the second and third payment was 1 month the average is around 6 months. However, this means that the predicted time till payment for the messages sent after the first payment will be negative since the predicted end time is actually smaller that the time till last payment at some point. Furthermore, the predictor will be too large for the messages sent after the second payment. This is very difficult to compensate, since quite a few customers only have 2 or 3 payments before they stop using the service.

The regression models created one massive cloud of points. The regression coefficients accordingly show some very large coefficients and standard errors. Some of the larger coefficients indicate that event number, and status of the text message are of importance when examining the linear regression model. It is not that surprising that the status of a text message is of importance. When analyzing the polynomial regression model, the *eventnbwithinsimilarevents* appears to be very important. The *TimeTillLastPayment* has a very low coefficient, which enforces the idea that the time between payments has a very low predictive value.

6.2 - Determining If Payment Is Due

Figure 20 shows the histogram of the TimeToNextPayments for all events predicted by the regression models. The red line indicates the four week line, and 56% of all events have a time to next payment less than 4 weeks. Also considering that CM determines the status of their customer every 4 weeks, we will predict if a payments will be done within the next 4 weeks.

Each line in the log was then extended with a Boolean denoting if a payment is due in the next 4 weeks. This information is off course not always available. The data for which this was available was split into a train and test part, using 80% as training data and 20 as test data. Using KNIME, a decision tree [10] was build predicting the Boolean value. It used the Gini impurity as quality measure for the splitting, and MDL as pruning method. Furthermore, the minimum amount of records per node was set to 100 to avoid overfitting.

Using 10 fold cross validation, the average fault was 2.14%. The results are shown in Table 10. Table 11 shows the confusion matrix of a single trained decision tree. It shows that the number of false predictions is very low, and that the False Positive and False Negatives are pretty balanced.

Table 10 10-fold Cross Validation Results

	Error in %	Size of Fold	Error Count
Fold 0	2.033432	64669	1315
Fold 1	2.21126	64669	1430
Fold 2	2.095285	64669	1355
Fold 3	2.115388	64669	1368
Fold 4	2.19889	64669	1422
Fold 5	2.039617	64669	1319
Fold 6	2.116934	64669	1369
Fold 7	2.265382	64669	1465
Fold 8	2.226724	64669	1440
Fold 9	2.115388	64669	1368

Table 11 Confusion Matrix Decision Tree

		Predicted Values	
		True	False
Actual Values	True	55883	1295
	False	1654	70506

Figure 25 shows the results of a decision trained on 80% of all the data and tested on the remaining 20%. Both the cross validation and the training of the individual tree was done using the same random seed, 2015. Each event is plotted as a single dot, the color depicting whether the value of the Boolean (will pay within 2 weeks) is True of False. Green denotes True and red denotes False. Furthermore, the graph is split into 2 parts, point in the upper part are point for which the Boolean has been predicted True, the lower points are point for which the decision tree predicted False. Thus, if every single point was predicted correctly, no single red point would be in the upper part, and the lower part would be without

any green points. It has already been stated that the amount of False Positives and False Negatives are very low.

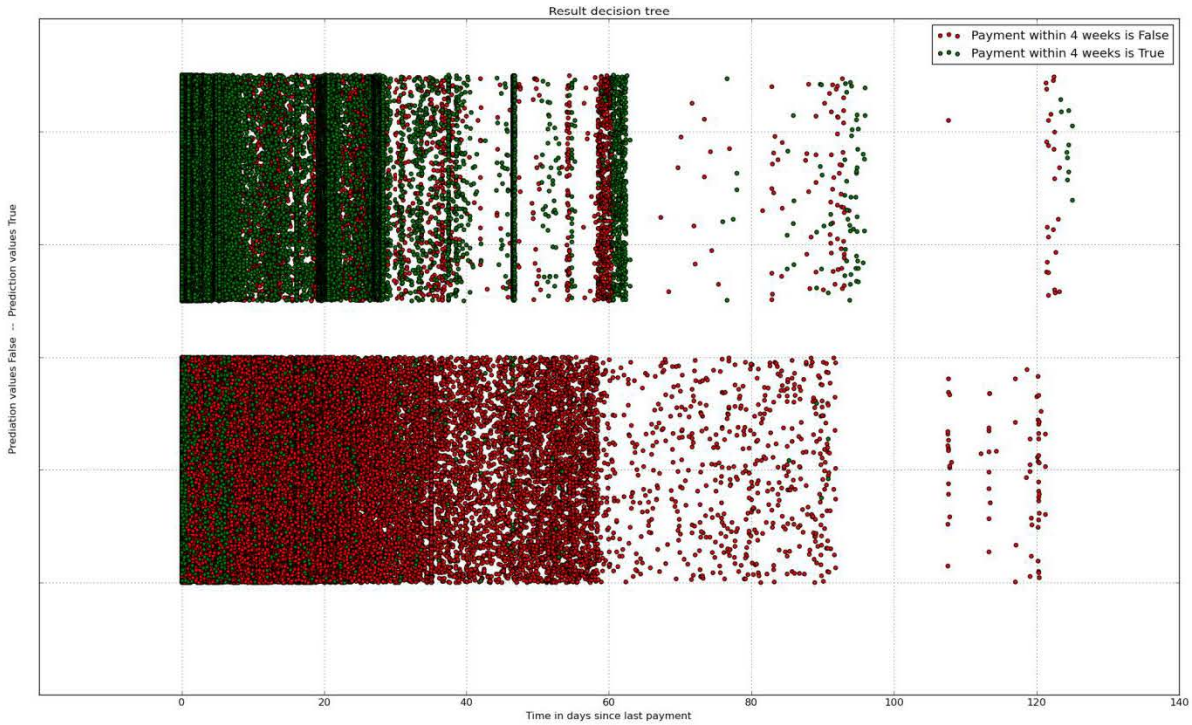


Figure 25 Visual representation of the decision tree results

Figure 26 shows the same data as Figure 25, but the events are grouped per customers and per period payments. This means that every 2 events that have been preceded by the same payment are at the same height. Events that are preceded by a different payment are slightly apart in height. Furthermore, there is a significant gap between each different customer. Also, correctly predicted points are displayed as small dots, with the color corresponding to the original value. Wrongly predicted point as displayed as larger dots.

Both graphs show a similar picture. The wrong predictions are not centered at a particular point, although they do seem to be more frequent at the start and end of payment period. In Figure 25, it looks like there are a lot more False Positives, than there are True Negatives. But the actual numbers were not that much apart(Table 11). Furthermore, in Figure 25 there appear to be certain bars in which the concentration of points is quite high. These bars are around 20 days, 30 days, 40 days and 60 days. There are also relatively few points beyond 60 days. The decision tree itself is very complicated, it is not possible to create a simple overview of the decision tree and an XML representation of the tree takes up to 70 pages.

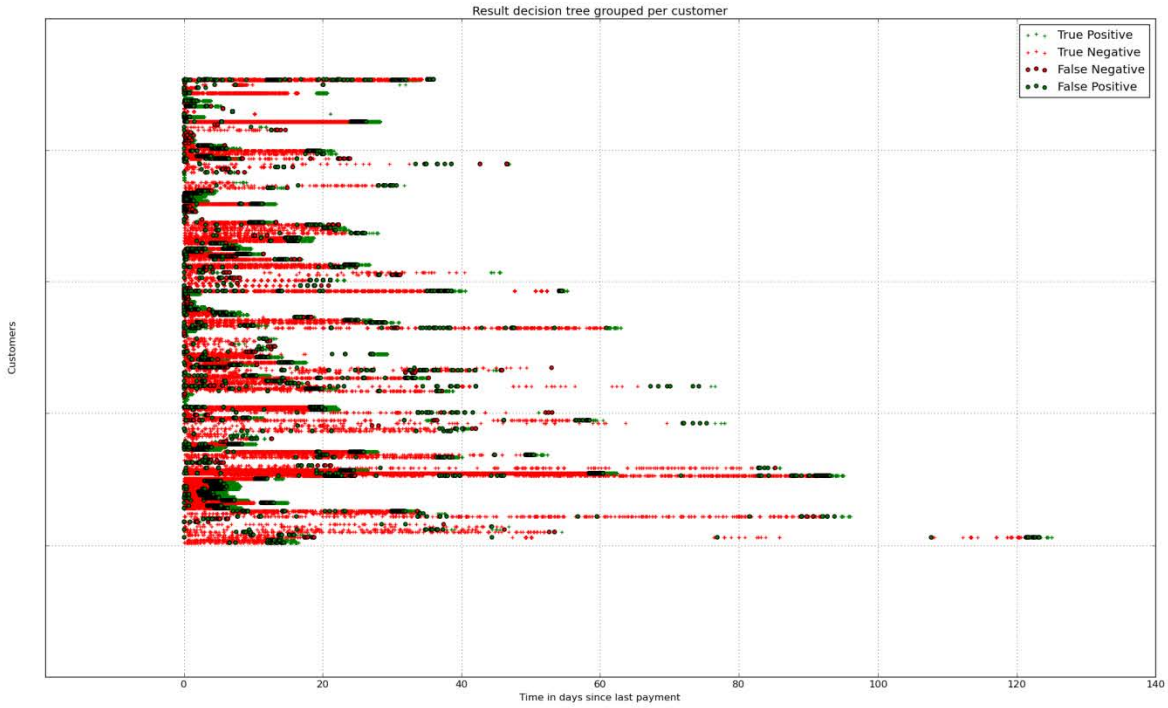


Figure 26 Visual representation of the decision tree results, grouped per customer

6.3 - Predicting Customer Loss

To predict customer loss, a decision tree was trained using 3 different data files. The data files consist of 4 data points. Each data point denotes the percentage increase or decrease in the amount of messages sent compared to the previous week. The 4 data point are the 4 weeks prior to a particular time. One data file contains the 4 weeks prior to the 21st of March. The 21th of march is a significant point, because the customer status is determined by the amount of messages sent after that date. If the customer has not sent any messages after the 21th of March, it is considered an inactive customer.

Another data file contains the percentage increase or decrease in the amount of messages sent in the 4 weeks prior to the last known sent message. A third data file contains data point of the 8 to 4 weeks prior to the last known sent message. The latter two data files are used to determine if customers show a declining sending pattern before stopping using CM Direct altogether.

All three data files consisted of data of continues customers. Single time customers and period customers are left out because they have a significantly different sending pattern which could negatively impact the results. The resulting decision trees are included in the appendix, and have been trained using 80 percent of the data, and testing using the remaining 20 percent. Before splitting the data, the data was sampled such that the amount of active and inactive customers were the same.

Table 12 shows the confusion matrix of the decision trained on the data file with the 4 weeks prior to the 21th of March. The test set is not that large, but the results are promising. The decision tree show that apparently the difference between the 2nd and 3rd week is important. If the percentage increase or decrease is above 0.0094 or below -0.0065 the customer is going the remain active. Otherwise, he will become an inactive customer. Thus, if a customer shows no to little difference in his sending behavior, he becomes inactive. Although this seems quite counter intuitive, it becomes more clear when examining the data. All the inactive customers have send no messages in the 4 weeks prior to the 21th of March. Table

15 shows the result of a 10-fold cross validation using stratified sampling. The results are around the 20% error rate, and we have an average error of 17%.

Table 12 Confusion Matrix of data file 4 weeks before the 21th of March

		Predicted Values	
		True	False
Actual Values	True	13	4
	False	2	21

Table 13 shows the confusion matrix of the decision tree trained on the 4 weeks before the last sent message. First, it should be noted that there have been customers who have not sent for more than 4 weeks. These have not been included. So the data file is not as large as the data file with the data of the 4 weeks prior to the 21th of March. The decision tree results are quite good. The decision tree is bit more complicated. The data is split on the Firstdif and the Thirddif. Firstdif is the percentage of the first week, and Thirddif is the percentage of the third week. If the first difference is greater than 0.0679, or smaller than -0.0089 but greater than -0.3842, the customer will remain active. If the first difference is smaller than -0.3842, the customer will also remain active if the third difference is greater than 0.4167. Otherwise, the customer will become inactive. This decision tree is more complicated, and also less easy to interpret. The difference in the sending behavior between the 2 groups seems minor. Judging by the decision tree, the major difference between the 2 groups is that inactive customer send less messages in the first and third week before stopping completely. Table 15 shows the results of a 10-fold cross validation using stratified sampling, the error rates show more deviation and appear to be higher in general. On average, the error rate is 32%.

Table 13 Confusion Matrix of data file 4 weeks before last sent message

		Predicted Values	
		True	False
Actual Values	True	13	2
	False	4	9

Table 14 shows the confusion matrix of the decision tree trained on the period of 4 weeks, before the 4 weeks prior to the last sent message. Thus 8 to 4 weeks before the last send message. This required the customers to be active for a period of 8 weeks. Not all customers have been active for so long, so there is not as much data available as with the other 2 data files. The results show that the decision tree is not able to distinguish the 2 groups as good the other 2 decision tree. The amount of False Negatives is about the same size as the amount of true negatives and true positives. The decision tree itself only focusses on the third difference. Table 15 shows results of a 10-fold cross validation using stratified sampling, the average error is about 30%. Although the fold size it not that large, it is a strong indication that it is difficult to distinguish the active and inactive customers using this data.

Table 14 Confusion Matrix of data file 8 to 4 weeks before last sent message

		Predicted Values	
		True	False
Actual Values	True	7	0
	False	5	7

Table 15 cross validation results customer loss

	4 weeks before 21th March			4 weeks before last sent message			8 to 4 weeks before last sent message		
	Error in %	Size of Fold	Error Count	Error in %	Size of Fold	Error Count	Error in %	Size of Fold	Error Count
Fold 0	4.76190476	21	1	33.3333333	15	5	50	10	5
Fold 1	14.2857143	21	3	33.3333333	15	5	20	10	2
Fold 2	20	20	4	46.6666667	15	7	20	10	2
Fold 3	40	20	8	20	15	3	10	10	1
Fold 4	15	20	3	35.7142857	14	5	50	10	5
Fold 5	15	20	3	57.1428571	14	8	30	10	3
Fold 6	10	20	2	50	14	7	11.1111111	9	1
Fold 7	20	20	4	35.7142857	14	5	22.2222222	9	2
Fold 8	10	20	2	14.2857143	14	2	66.6666667	9	6
Fold 9	15	20	3	14.2857143	14	2	22.2222222	9	2

6.4 - Final remarks

Although the Time to Next payment is not exponentially distributed, it still is very hard to predict. Efforts made using 2 regression models proved to be unsuccessful as they produced a clutter of points instead of a single straight line. A simple predictor based on the assumption that customer buys credits on a regular interval, also proved ineffective. This is an indication that a lot of customers are not that regular in their behavior. It proved to be more effective to build a decision tree to predict if a customer will pay within 4 weeks. The 4 week interval is relevant since CM checks up on the CM Direct customers once a month and 56% of all event have a Time to Next Payment less or equal to 4 weeks. The decision tree results show very little False Positives and False Negatives, and most errors appear at the start and end of a payment period. However, the decision tree turned out to be very large which is another indication that although we attempted to avoid overfitting as best as possible it is very difficult to predict if a payments is due within the next 4 weeks. We also predicted actual customer loss by using the relative increase or decrease in the amount of messages send over a period of 4 weeks. The 4 weeks prior to the 21th of march has been examined together with the 4 weeks before the last sent message and the period of the 8 to 4 weeks preceding the last sent message. The data file containing the 4 weeks prior to the 21th of march yielded an effective decision tree, but when examining the data it appears that this is mostly due to the fact that a lot of inactive customers are not sending any messages at that time. The other 2 data files negated this effect, as they covered the time in which a customer did sent messages. However, the data files were not as large and the results were mixed. Although they did predict most cases right, the decision trees focused on only 1 data point and multiple runs resulted in different decision trees. It should be noted that those decision trees al have a similar accuracy and structure.

7- Implementation

7.1 - Data Gathering and Cleaning

In order to create the data file and extended log file, a php script was used to connect to the databases and write the records to a text file.

To clean and analyze the data, the Disco tool [20] was used. Disco is a tool developed around the fuzzy process mining algorithm by [2]. Besides process mining, it can also be used to filter the log on activities, recourse and other variables.

Not all analysis proved to be doable for Disco, therefor some of the violation statistics were computed using the Python programming language [22]. During this project, Python 2.6.6 was used. Furthermore, the datetime package was used to properly handle the date objects.

7.2 - Customer Classification

To determine the frequency of the various events in the log, Disco proved to be very easy and powerful to use. Furthermore, several python scripts were used to create the various plots. The following packages were used

Package Name & Version	source
Matplotlib 1.4.3	[23]
Package that can be used to create plot, scatters plots and histograms. The pyplot sub package provides an interface similar to MATLAB	
Numpy 1.9.2	[24]
Powerful package to support scientific computing, primarily used for efficient vector and matrix operations	
Pandas 0.16.0	[25]
Open source package providing powerful data analysis tools and efficient data structures.	

Figure 27 shows 2 screenshots of Disco. As stated before, it is possible to show statistics over the entire log, such as the amount of events over time, events per case and Case Duration. It is also possible to show statistics related to the activities within the log, for example the event distribution.

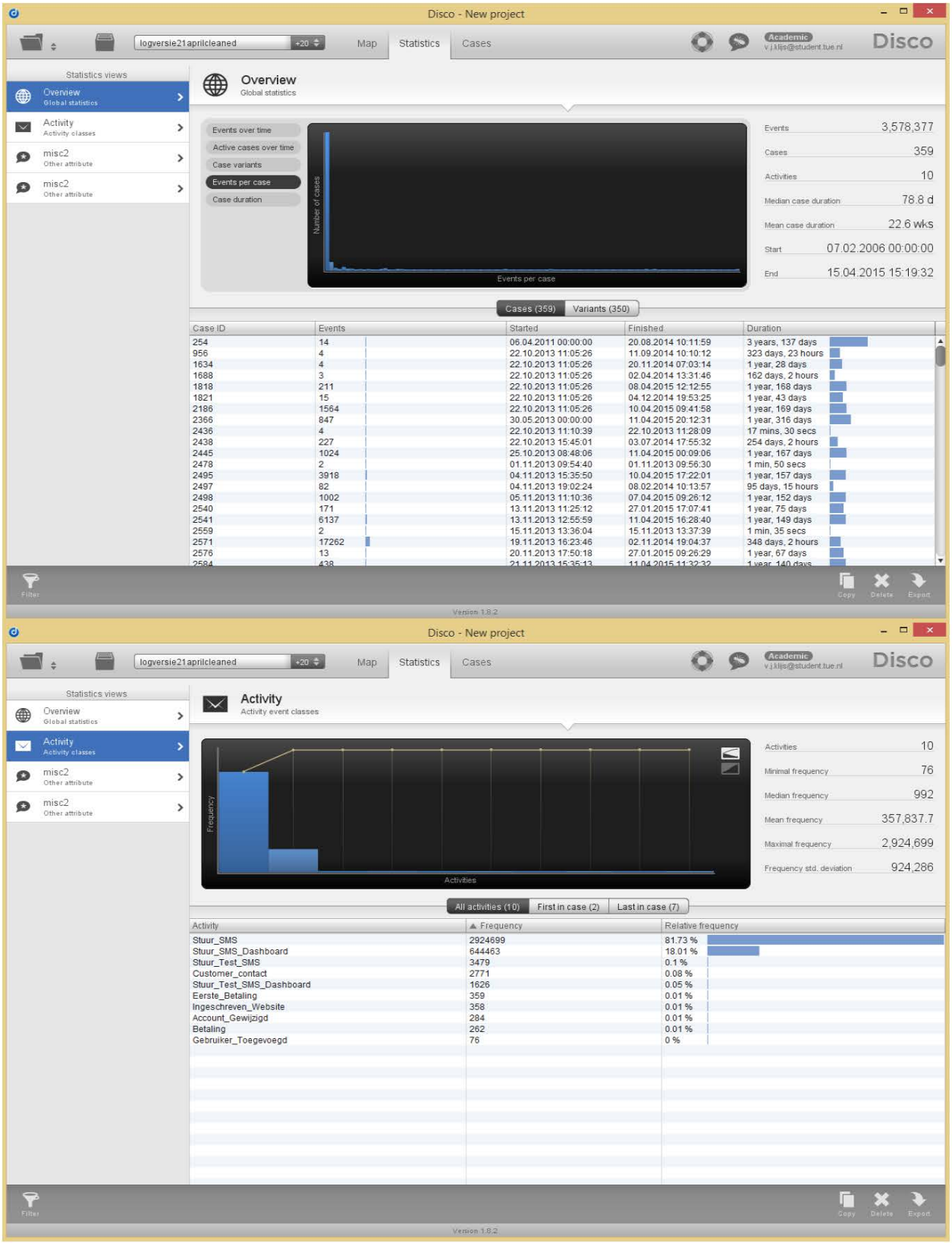


Figure 27 Screenshots of Disco functionality

Furthermore a tool was developed to additionally examine the intersend times of both customer groups. The tool was developed using the following Python packages:

Table 16 Python packages used in intersendtimes tool

Package Name & Version	source
Matplotlib 1.4.3	[23]
Package that can be used to create plot, scatters plots and histograms. The pyplot sub package provides an interface similar to MATLAB	
Numpy 1.9.2	[24]
Powerful package to support scientific computing, primarily used for efficient vector and matrix operations	
Pandas 0.16.0	[25]
Open source package providing powerful data analysis tools and efficient data structures.	
TKinter 8.5	[26]
Cross-platform graphical user interface kit	
Ttk 0.3.2	[27]
Package with additional widgets which can be used in combination with Tkinter	
Math	[28]
Package which is always available in any Python installation and provides some basic mathematics functions.	
Scipy 0.15.1	[29]
Scientific computing package which include the stats package. The stats package could be used to generate numbers according to various distributions.	
Statsmodels 0.6.1	[30]
Package that can be used for various statistical analysis. In this case it was used to estimate the Empirical Density Function of the intersendtimes	

The tool required the intersendtime of the customers to be provided using a text and/or csv file. Using the tool, it is possible to create several different plots, all of which giving insight in the behavior of the selected customer. One of the plots is a sort of ROC curve, which shows how many messages have been sent after how many days. The x axis denotes the time in days, and the y axis show the cumulative sum of the amount of messages sent up to that time. One such plot is shown in Figure 29. Figure 28 shows a different plot of the same data. In this plot, the intersendtimes are plotted against the messages they adhere to. I.e. the time between first and second message is plotted as the first point, the time between the second and third point is plotted as the seconds point and so on. This plot is rather useful to get an insight in the distribution over time of the intersendtimes. Finally, it is also possible to see visually if a set of intersendtimes follows a particular distribution, see Figure 29. The distributions available are; Normal, Exponential, Gamma and Lognormal. The parameters for the various distributions are estimated using the method of moments method.

There are also a few sliders to accommodate various options. One of these options is setting the amount of messages each customer should have sent. If a customer has sent less than the specified amount, the customer will no longer be available in the select customer combobox. It is also possible to set bounds on the time the customer has been active. This can be an upper and lower bound. When trying to fit a distribution to the data, it is also possible to leave out outliers (at most 10) and to transpose the data to zero.

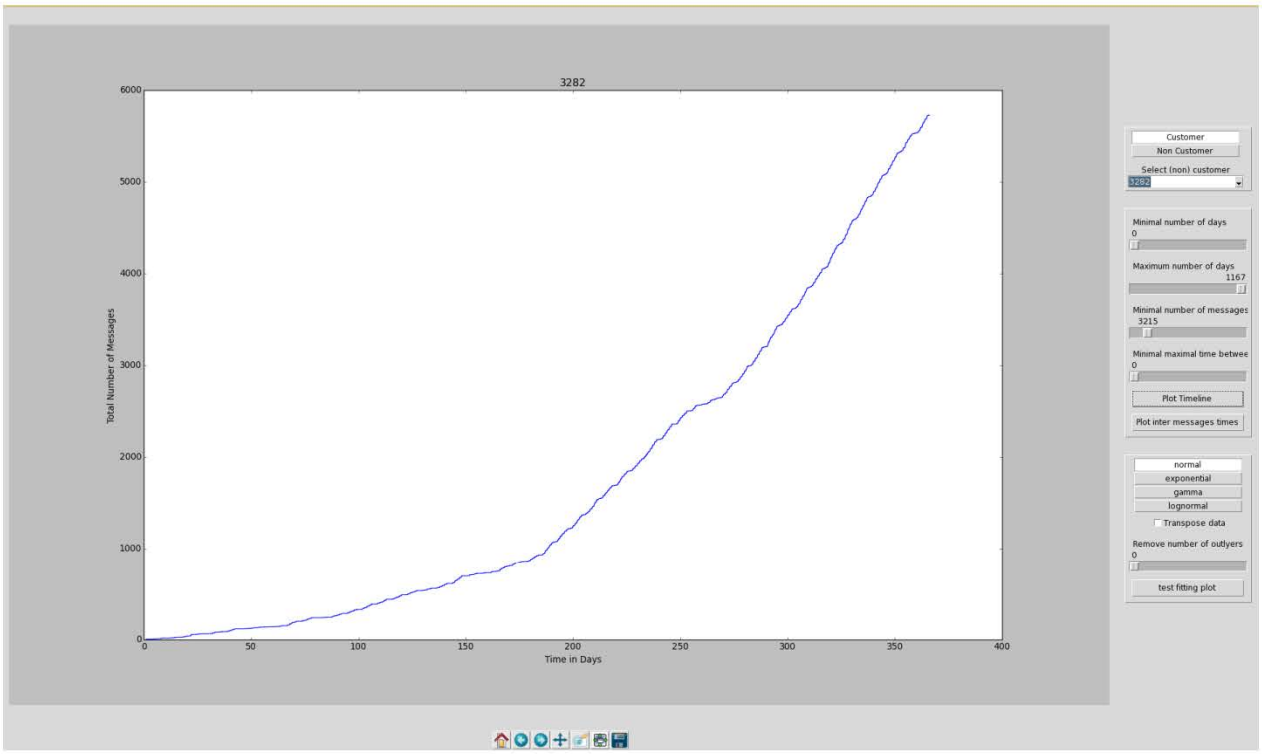


Figure 29 Example plot of intersendtimes tool

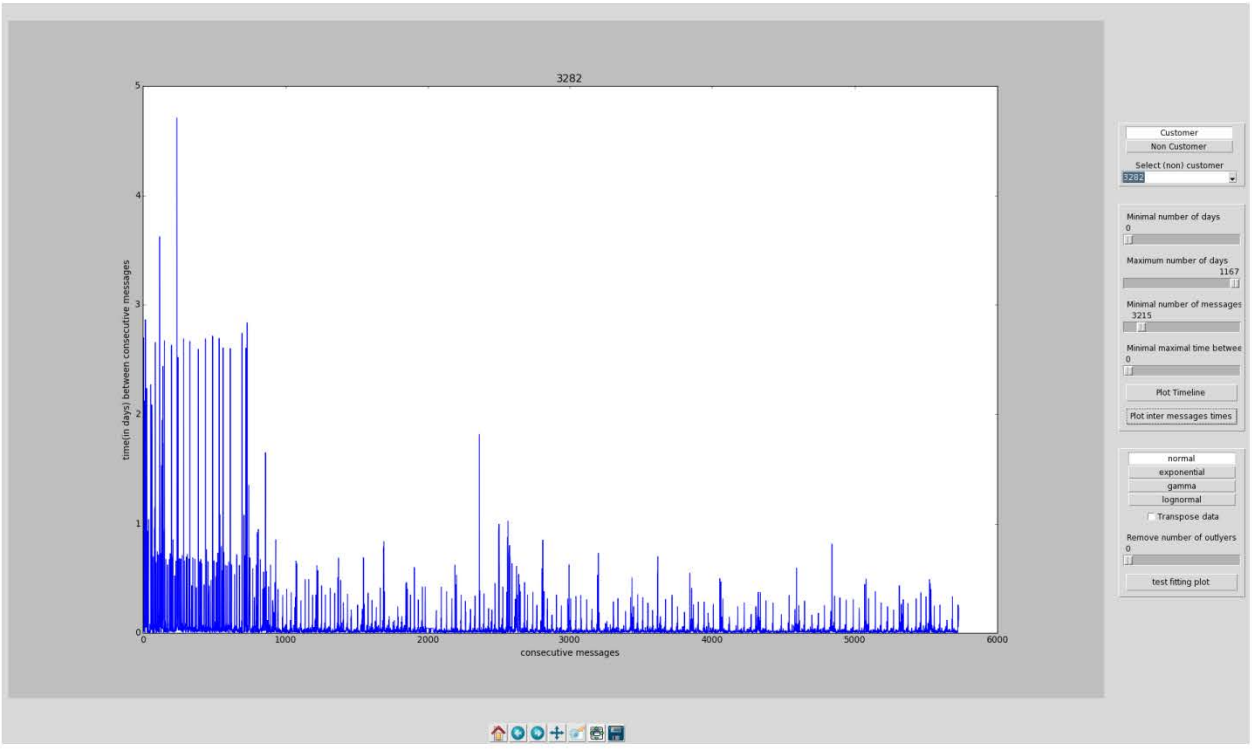


Figure 28 Another Example Plot

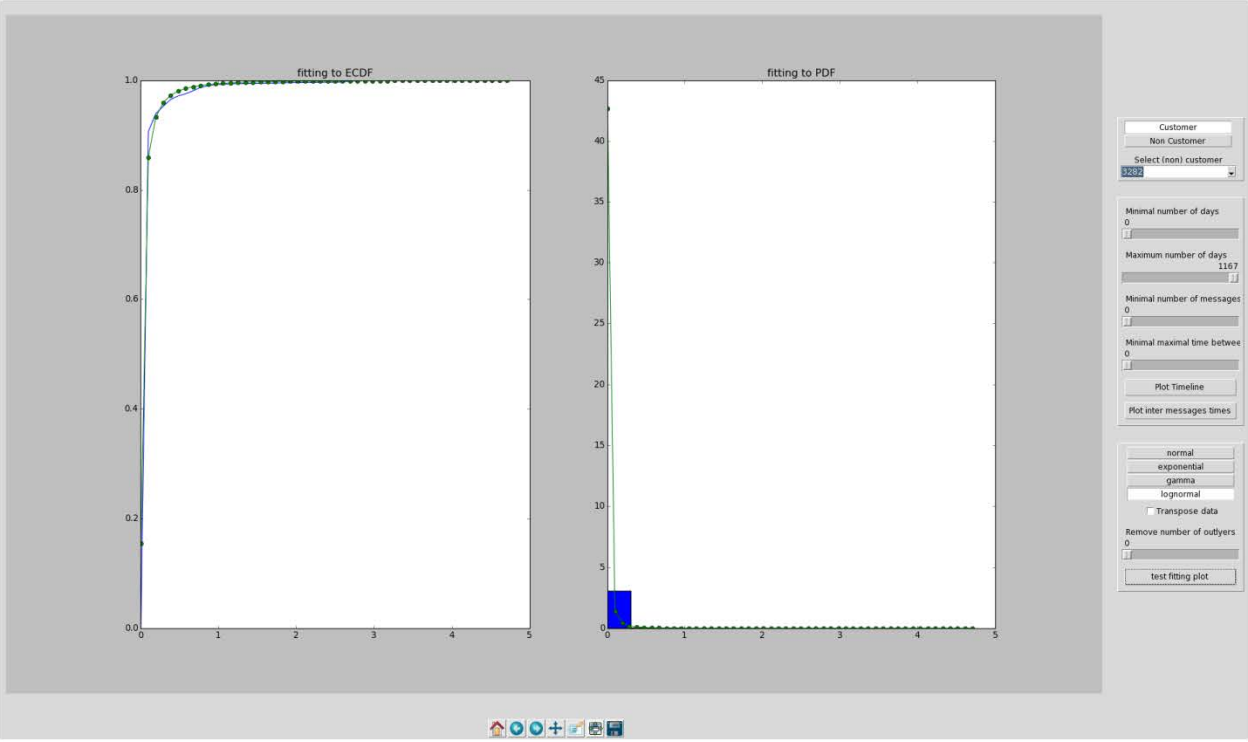


Figure 30 Fitting plot

7.3 - Prediction

KNIME is an open data analytics platform. It is created as an drag and drop environment in which data can be retrieved from various sources, such as an CSV files or by using a Database connection. The data can be manipulated using various column and row filters, splitters and mergers. Finally, a lot of data mining algorithms are implemented. For example, the full Weka data mining package [31] can be used.

Figure 31 shows the workflow used to create the regression models. The data is read using a CSV reader, next a Java snippet is used to only continue the process with the events for which the time to next payment is set. The partitioning node splits the data into a training set (80%) and test set (20%). The results of the Regression predictor are stored using a CSV writer node.

Figure 32 depicts the Decision Tree workflow used in KNIME. The CSV reader is used to read the data that has already been prepared in the regression workflow. Thus, only records with a set Time to Next Payment are used. The column splitter is used to separate the customer data. This did not infer with the working of the regression models, but can cause some severe side effects when this data remains when training a decision tree. This kind of data can lead to some very serious overfitting, as the status of a customer is directly linked to its ID. The workflow shows two distinct paths. The upper path uses an X-partitioner block to perform a 10-fold cross validation. The lower path is used to train and test a decision tree on the complete data set.

Figure 33 shows the Decision Tree workflow used in KNIME when determining customer loss. It is comparable to the Decision Tree workflow from Figure 32 but has an Equal Size Sampling node to ensure that the amount of active and inactive customer are the same.

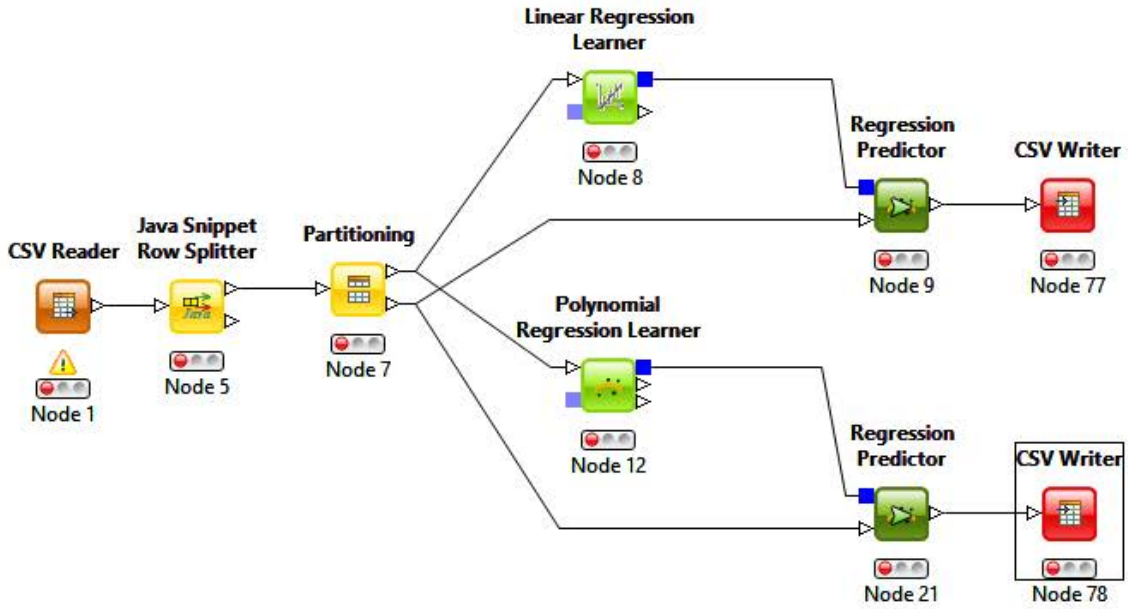


Figure 31 KNIME workflow concerning the regression models

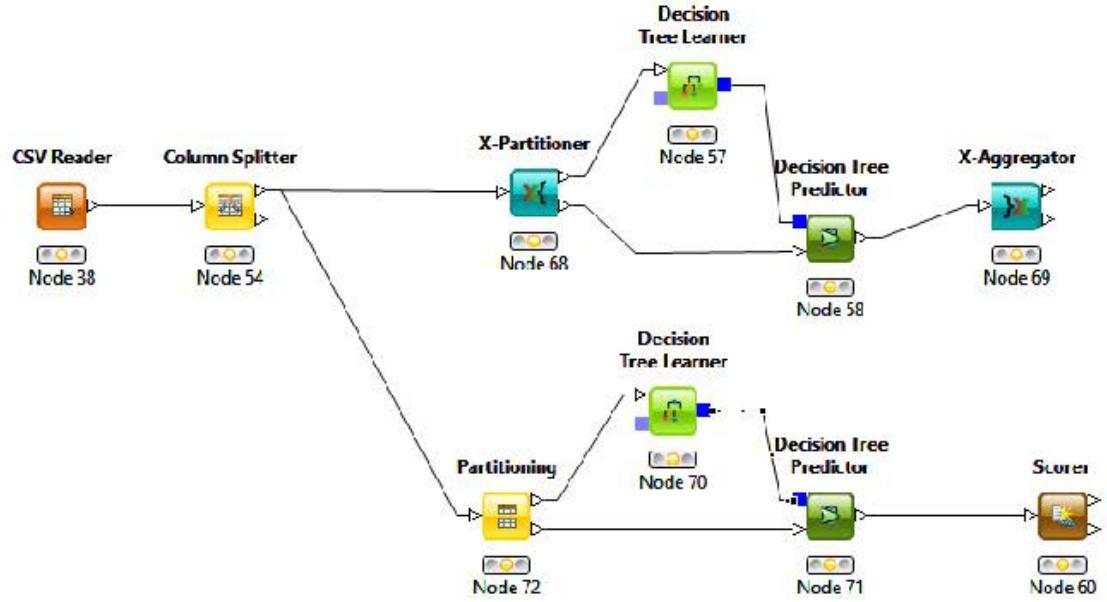


Figure 32 KNIME Decision Tree workflow concerning upcoming payments

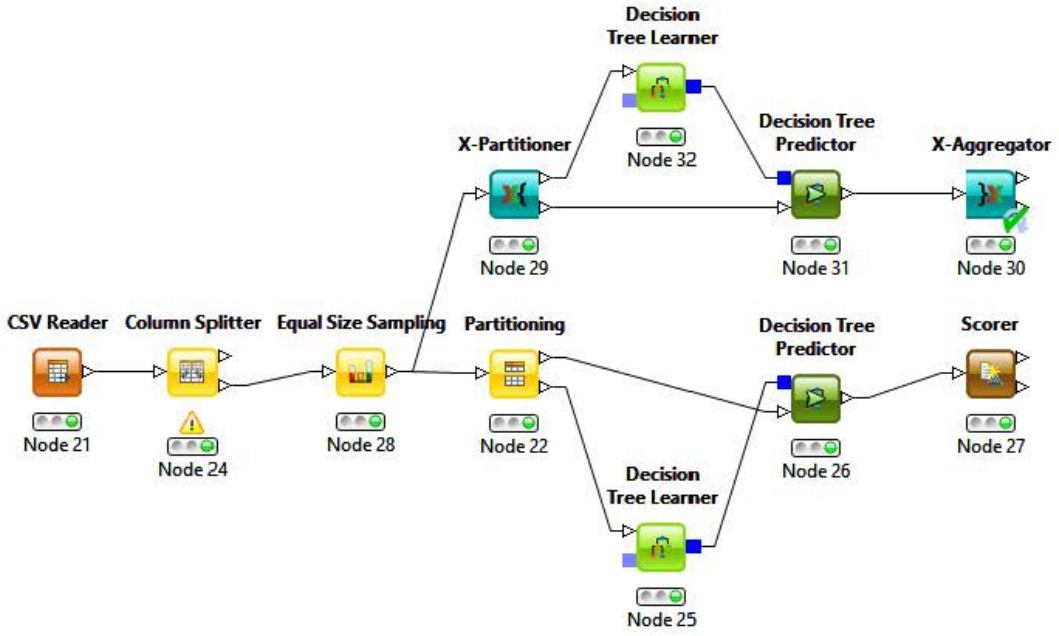


Figure 33 KNIME Decision Tree workflow concerning customer loss

8 - Conclusions and Recommendations

The original research question was consisted of three sub questions, these are:

1. Is it possible to predict if a payment is due?
2. Is it possible to predict when the next payment will occur?
3. Is it possible to predict customer loss?

To answer these questions, we looked into the different customers groups. CM has a gut feeling that they have three types of customers. Single time users are customers who only use the services provided by CM Direct for only a single time. Continuous users are customers who use CM Direct in a continuous manner, constantly sending out messages. The third category consists of periodic users, this type of Customer only uses CM Direct whenever he wishes to send out a particular message to a large group of receivers only once every time period. This period can be a week, a month of even an quartile. It turns out that the Single Day users exist, and take up a small fraction of the entire customer base. There is also an even smaller group of customers who are responsible for 59% of all sent messages. These customers can be considered as absolute outliers in terms of amount of sent messages. The groups of continuous and periodical customers were more difficult to distinguish, but not impossible. By creating a combined histogram of the intersend times of all customers it has become clear that there are indeed customers who only send messages on a periodical basis. However, most customer appear to send messages every 3 days (at most) and customers who use a larger period between batches of messages do not sent that many messages altogether.

We tried to predict when the next payment will occur by using 2 different regression models, and the results are not that promising. An attempt to make a very simple predictor, which only uses the payments information also proved to be insufficient. Examining the intersend times showed that up to 56% of the events were within 4 weeks of an upcoming payment. Therefore, we tried to predict if an payment was to be expected within the next 4 weeks. A decision tree was trained to predict this value, and the result were quite good. On average, it had an error rate of 2.14% but the result did show that most of the wrong values occurred at the beginning and end of an payment period (time between 2 payments), which is where it is most critical to get the prediction right. The decision tree was very large and difficult to interpret. However, there are still some interesting insights that can be distilled from the result. A scatter plot of the predicted data clearly showed that most events take place less than two months before paying. Far less events have a Time to Next payment larger than 60 days. In other words, in very few cases did the time to next payment exceed 60 days. This is indication that if a customer has not been active for over 60 days, it will be very unlikely that he will become active again.

This leaves the third question. We tried to answer this by creating decision trees on data files denoting the percentage increase and/or decrease over a period of 4 weeks. Three different periods were examined. The first period was the 4 weeks before the 21th of march. The second period consisted of the 4 weeks before the last sent message. The last period was the 4 weeks prior to the last 4 weeks before the last sent messages. This period can be also be explained as the 8 to 4 weeks before the last sent message. The decision trees did pretty good, but after examining the data files and decision trees in more detail the results are less good. Especially the decision tress concerning the periods before the last send message appear to indicate that the sending behavior is not that good of an indication for whether a customers is going to continue using CM Direct. The weeks before the 21th of March are a better predictor, but this is mostly due to the fact that a lot of inactive customers have already stopped sending messages at that time. So although it is possible to make some assumptions based on the first 2 question (if the customer is not going to pay, it is not expected of him to continue using CM Direct), we are not able

to truly predict if a customer will stop using CM Direct. We have been unable to find a truly discriminating value, or combination of values to predict if a customer will be lost in the future.

8.1 – Recommendations

Using the decision tree, it is possible to predict if an customer is going to pay within the next 4 weeks. Currently CM determines every month if customers have stopped using CM Direct. If the decision tree is used, they should be able to determine each month which customers should pay the within the upcoming month and are of interest to keep an eye during the coming weeks. This means CM could preemptively react instead of reactively. Furthermore, the results showed that most events were in range 0 to 60 days in terms of time to next payments. Thus if a customer has not been active for over 2 months, the probability that he will start using the services of CM again at some point are very small. Also, the customer classification referred to by CM internally proved to be correct, although the periodical customers and single time customers take up a very small portion of the total amount of customers and sent messages. The focus of CM should therefore be mainly on the continuous users. As they present the largest group of users, and are also responsible for most of the sent messages.

It should also be noted here that although a large amount of data could be collected, some parts of the data were far from complete. The support ticket system which is in place allows for every conversation with customers to be logged for future reference, but currently is not properly used by all employees. Therefore, the customer support information provided was not complete. CM can probably gain a lot of insights into their customers if they were to add more feedback opportunities. It could be beneficial if, for example, customers are able to rate the way they have been helped by customers support. Or to rate how satisfied they are at the moment with CM Direct.

8.2 - Future work

Non-parametric regression

An option which could not be explored due to time limitation, was the use of non-parametric regression to predict an estimate of the time to next payment. As stated before, non-parametric regression is a good direction to follow, if the relation between the data is unknown. This further more allows an emphasis on local extremes. The results of this research at least show that the relation is very complex. A fact which is further emphasized given that the resulting decision tree, which did yield valuable results, is very complex and virtually impossible to interpret. Which is quite an accomplishment given the instinctive nature of decision trees.

Dashboard users

When the event log was first analyzed, a surprising difference revealed itself concerning the active and inactive users. The active users send 9% of their messages through the dashboard. The inactive customers send 70% of the sent messages using the dashboard. This is a significant difference, but it was not within the scope of this thesis to explore it any further. However, some interesting question do arise. For example, is there a causal connection between the dashboard usage, and whether or not a customer remains active or goes inactive. Suppose, there is causal relation between the two, does that mean that the dashboard is ill designed. It is possible that the amount active customers can be increased by improving the dashboard. Suppose the dashboard needs improvement, what kind of improvements are necessary to keep customers, it a merely the GUI that needs improving, or is the functionality not what the customers expect. It is also possible that the dashboard design is of no effect, and that the customers who use the dashboard are very different to the customers that don't use it, and the difference in dashboard use is merely a symptom and not the cause of any problem. If so, is it perhaps possible to determine what sets these customers apart, and would it be possible to classify these user early on in the process.

User characterization

CM has some clear ideas on the type of customers that use CM Direct. During this research project, we have been able to at least confirm the existence of these customer groups. However, it also became clear that although correct, the classification is not very useful as the largest group of customers, who are also responsible for by far the largest part of the total amount of sent messages, all fall in a single group. It would be interesting to see if it is possible to cluster this group even further. Thus, a further analysis of this group, either on the same data but using different techniques such as time series clustering, or on different data could have interesting results.

Bibliography

[1] "CM," 10 7 2015. [Online].

[2] W. M. van der Aalst, Process Mining: Discovery, Conformance and Enhancement of Business Processes, Mairdumont Gmbh & Co. Kg, 2011.

[3] D. Loshin, Business Intelligence, 2013.

[4] R. A. Khan and S. Quadri, "Business Intelligence: An Integrated Approach," *International Journal of Management and Innovation,* pp. 21-31, 2014.

[5] "Unit4 Financials," 8 7 2015. [Online]. Available: http://www.unit4.nl/producten/financieel-management-systemen/coda/informatie.

[6] "CPB," 8 7 2015. [Online]. Available: www.cpb.nl.

[7] F. J. Alexander, "Machine Learning," *Computing in Science & Engineering,* pp. 9-11, 2013.

[8] R. Kohavi and F. Provost, "Glossary of Terms," *Machine Learning,* pp. 271-274, 1998.

[9] P.-N. Tan, M. Steinbach and V. Kumar, Introduction to Data Mining, London: Addison-Wesley, 2006.

[10] J. R. Quinlan, C4.5 Programs for machine learning, San Francisco: Morgan Kaufmann Publishers Inc, 1993.

[11] D. C. Montgomery and G. C. Runger, Applied Statistics and Probability for Engineers, Wiley, 2011.

[12] M. Boon, J. v. Leeuwaarden, B. Mathijsen, J. v. d. Pol and J. Resing, *Stochastic Simulation,* Eindhoven, 2014.

[13] J. Penders, "Behavior-based website redesign : how can data mining and process mining be applied to become more relevant to the end user on websites promoting complex products?," 2015.

[14] W. K. Hardle, Applied nonparametric regression, Cambrigde University Press, 1990.

[15] R. Crooy, "Predictions in Information Systems," 2008.

[16] J. Hills, J. Lines, E. Baranauskas, J. Mapp and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowledge Discovery,* pp. 851-881, 2013.

[17] X. Xi, E. Keogh, C. Shelton, L. Wei and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction," in *ICML '06 Proceedings of the 23rd international conference on Machine Learning*, 2006.

[18] E. Keogh and S. Kasetty, "On the need for time series Data Mining Benchmarks: A Survey and Empirical Demonstration," *Data Mining and Knowlegde Discovery,* pp. 349-371, 2003.

[19] P. Grefen, Mastering E-Business, Routledge, 2010.

[20] "Fluxicon Disco home," 8 7 2015. [Online]. Available: https://fluxicon.com/disco/.

[21] "KNIME analytics platform," 10 7 2015. [Online].

[22] "Python," 10 7 2015. [Online]. Available: www.python.org.

[23] "matplotlib home," 10 7 2015. [Online]. Available: http://matplotlib.org/.

[24] "Numpy Home," 10 7 2015. [Online]. Available: www.numpy.org.

[25] "Pandas Home," 10 7 2015. [Online]. Available: pandas.pydata.org.

[26] "Tkinter package," 20 7 2015. [Online]. Available: https://docs.python.org/2/library/tkinter.html.

[27] "Ttk package," 20 7 2015. [Online]. Available: https://docs.python.org/2/library/ttk.html.

[28] "Math package," 20 7 2015. [Online]. Available: https://docs.python.org/2.6/library/math.html.

[29] "Scipy Home," 20 7 2015. [Online]. Available: www.scipy.org.

[30] "Statsmodels package page," 20 7 2015. [Online]. Available: https://pypi.python.org/pypi/statsmodels.

[31] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations,* 2009.

Appendix – Decision Trees

In this appendix, the three decision trees with respect to prediction customer loss as discussed in 6.3 - Predicting Customer Loss. Figure 34 shows the decision trained on the data file containing the 4 weeks before the 21th of march. Figure 35 shows the decision tree trained on the data file containing the 4 weeks prior to the last sent message. Figure 36 shows the decision tree trained on the data file containing the 8 to 4 weeks before the last sent message.

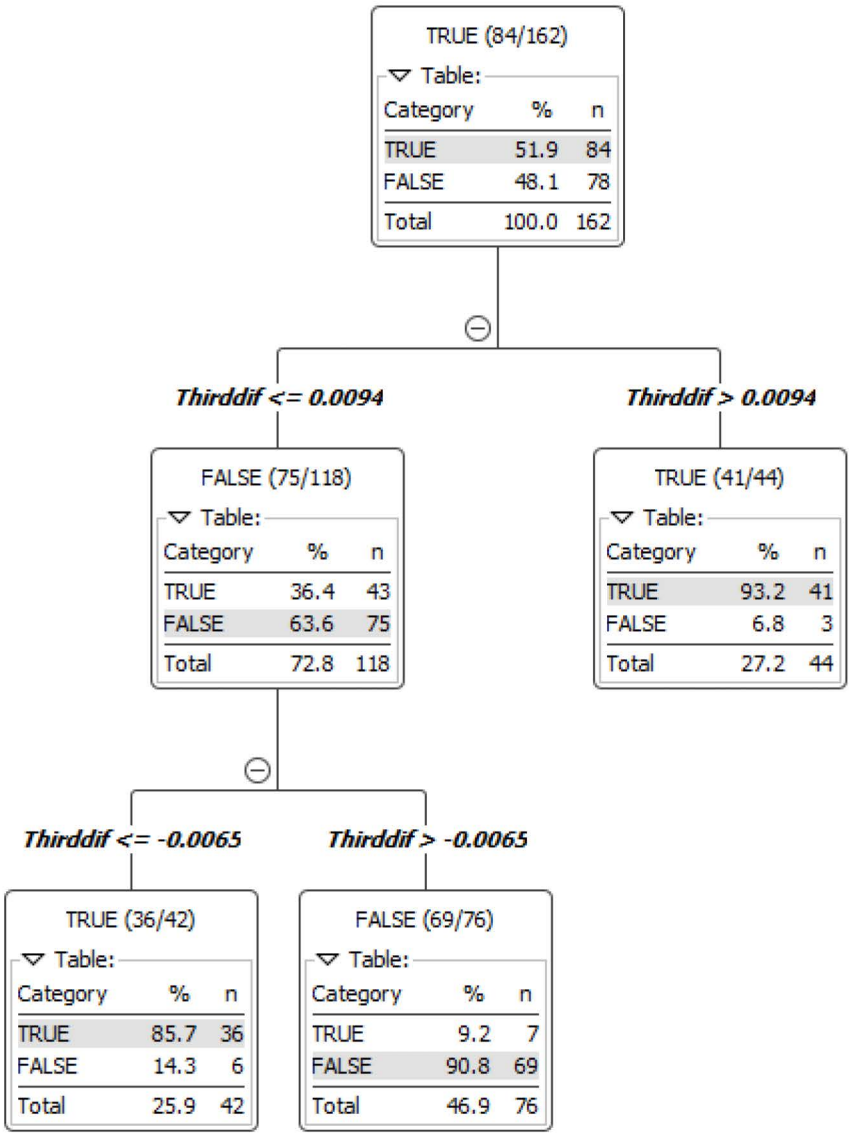


Figure 34 Decision Tree 4 weeks prior to the 21th of March

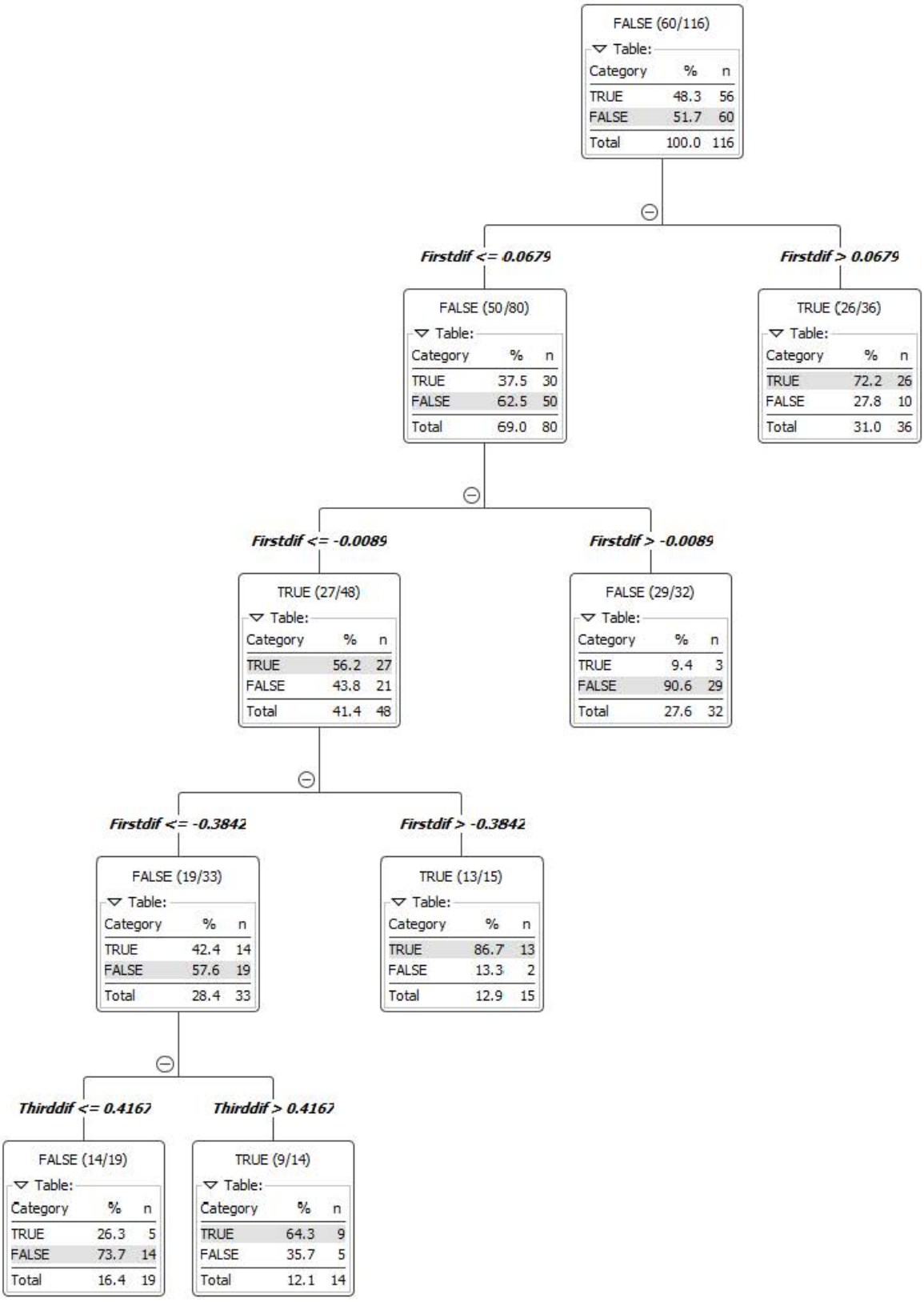


Figure 35 Decision tree 4 weeks prior to last sent message

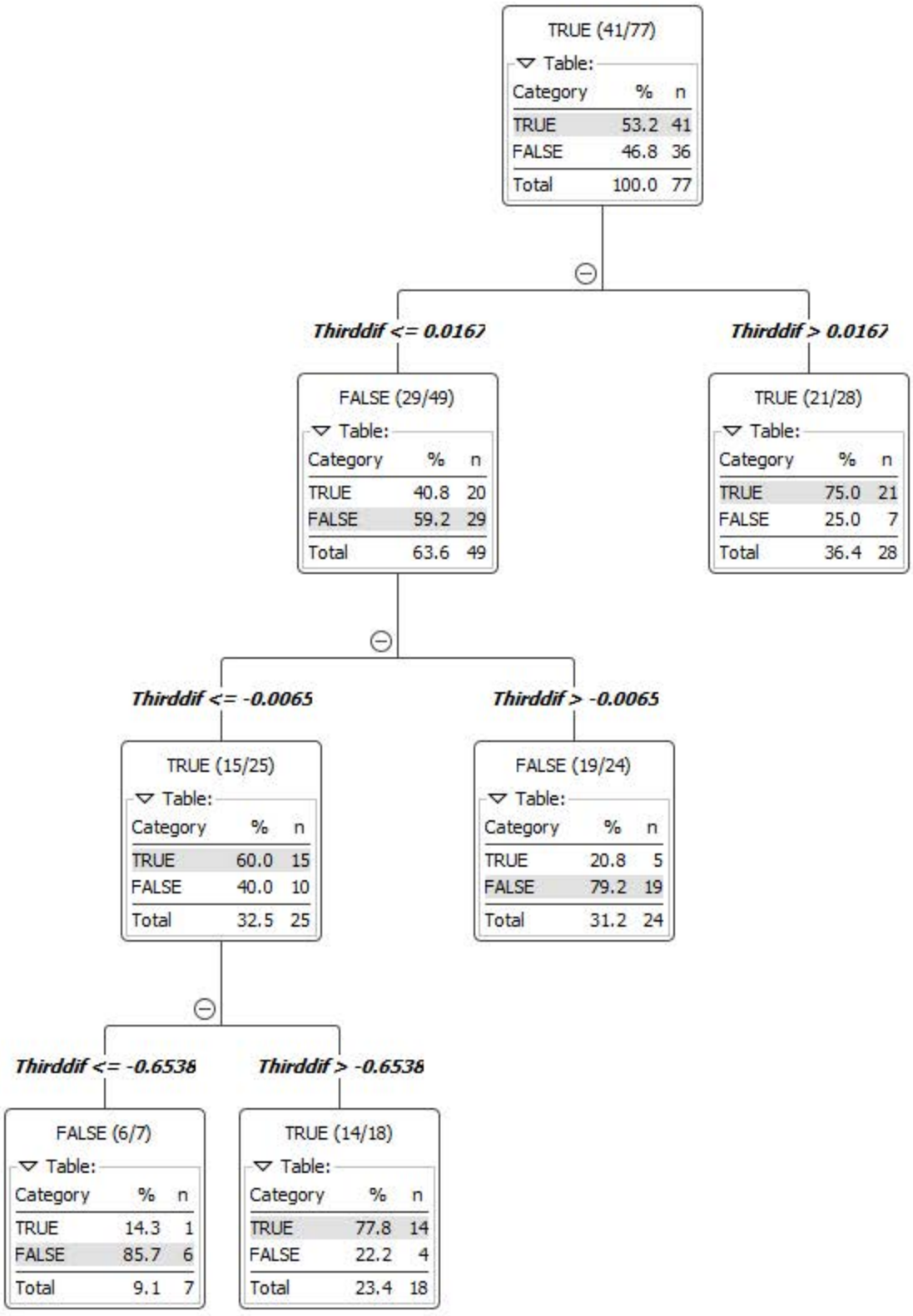


Figure 36 Decision tree 8 to 4 weeks prior to last sent message