

Data Mining

On Rechtwijzer.nl Log data



UNIVERSITY OF TWENTE.

Rechtwijzer

Author: Stefan Hessels
Study: Business Analytics
Date: 03-09-2015

1. Introduction

Nowadays there is a big trend where internet-users look up valuable and high-quality information in order to cope with their problems. A common example is the increased availability on medical information upon the internet. This increased self-efficacy can result in several consequences. For one, it can lower the visits to counsellors, of whom some are publically funded. This can lower the public spending. Another consequence can be that bad websites can provide wrong information. To cope with these consequences for law counselling, Rechtwijzer.nl was developed.

Rechtwijzer.nl is a platform that provides information and tools to support various processes that can involve legal counselling, such as divorce, consumer conflicts, debts and getting fired.

Especially divorce is a complex process where counselling is needed. To support self-efficacy, Rechtwijzer.nl has developed various tools and provides information to guide divorcing couples through each step of this process.

Web usage mining is used to get insight in the behaviour of the website visitors. Known techniques for this are (Fayyad, 1996):

- Classification: Building a model that predicts a certain class
- Regression: Building a model that predicts the value of a certain variable
- Clustering: Providing clusters of users that show similar behaviour
- Association rules: Attempting to identify all important interrelationships between attributes in the data.

Contrary to “traditional” research, while using web site mining no “a priori” hypotheses about relations between variables are formed. Using Data Mining it is tried to identify systematic relations between variables when there are no (or not complete) a priori expectations as to the nature of those relations. In a typical exploratory data analysis process, many variables are taken into account and compared, using a variety of techniques in the search for systematic patterns.

For Rechtwijzer.nl this research tries to identify behaviour that is related to the self-efficacy of divorcing couples. Self-efficacy can be defined as “one's belief in one's ability to succeed in specific situations and to complete tasks and reach goals” (Omrod, 2006):

1. Do users that score high on self-efficacy show different behaviour than users that score low on self-efficacy?
2. The sets of pages that are visited by the same participant are investigated using association rules to answer the question: which pages are visited together often?

This paper is structured as followed: first relevant data mining techniques are discussed. This is followed by every step in the data mining process, starting with the data manipulation, followed by the addition of extra variables and concluded by the results of the data mining. Finally, all results are discussed in the conclusion.

Table of contents

- 1. Introduction..... 2
- 2. Data Mining 4
 - 2.1. Classification..... 4
 - 2.2. Association 5
 - 2.3. Data mining software: WEKA 5
- 3. Research approach..... 6
 - 3.1 Step 1: data processing 6
 - 3.2. Step 2: Adding extra variables..... 11
 - 3.3. Step 3: Data mining 12
- 4. Conclusion 14
- 5. Literature 14
- 6. Appendices 14
 - 6.1. Appendix A: Results Classification..... 14
 - 6.2. Appendix B: Results Association Rule Learning..... 26

2. Data Mining

There are several types of data mining techniques. During this research, three will be used:

- Classification: Predicts the class or the value of a variable
- Association: Which website-pages are visited together?
- Clustering: Can we divide the participants based on unknown patterns?

2.1. Classification

Classification can be done in several ways, in which a clear distinction can be made: most of the classification algorithms, such as naïve Bayes and Decision Trees try to predict a specific class, a nominal group (Fayyad, 1996). An example of a class could be, for example, a type of flower, or brand of cell phone a user uses most, anything where you can put the entire studied population in exactly one group. So, nationality can be troublesome to use as classes, because some may have multiple nationalities. Using these methods we can try to predict, based on actions on the website, whether website users belong to group A, or group B, or any other group. Some other, like linear regression try to predict a specified numerical value (Fayyad, 1996). To classify by a numerical, the entire population should have exactly one value of a certain variable. Examples for this are the length of a person, or the total time spent on electronic devices per day.

Since Self-efficacy is a numerical value, we can do both. We can use self-efficacy as a number, but we can also put them in different classes, like above and under the median value. It has to be mentioned that discretizing numeric attributes to have nominal values results in loss of information and can affect the final result. For example, 4,99 and 5,01 are very close to each other, but if the boundary between to classes is set at 5,00, they are put in different classes. To assess the numerical value, linear regression will be used and for the second, used Naïve Bayes and Decision Trees will be used. These are two of the most used techniques for classification (Fayyad, 1996). Because no specific values of self-efficacy are important, an unsupervised (without human input) technique will be used while discretizing self-efficacy. To generate classes of equal size, equal-frequency binning (Sullivan, 2013) will be used. It is chosen to divide the participants in two, three and four classes.

For the assessment of the linear regression, only one value is important, the correlation the formula derived from the input variables with the variable that is under investigation. A value of 1 or -1 is a perfect fit with the regression model. A value of near 0 is almost no fit with the model.

The quality of the classification can be assessed by looking how much of the values are classified correctly. For this research it is chosen to use tenfold cross-validation. For the assessment of the classification by this validation technique two values are used, % Correct predicted values and relative absolute error:

% Correct predicted values: This value should be better compared to if all values are evenly spread among the classes. Because our classes are roughly the same size, the number to compare to is: 100% divided by number of classes (100%/#Classes).

We provide four example results of a confusion matrix, to give increased insight in this number:

Perfect Fit:

	A	B	C
A	100	0	0
B	0	100	0
C	0	0	100

Worst result:

	A	B	C
A	0	70	30
B	30	0	70
C	70	30	0

Better than random guessing:

	A	B	C
A	40	25	35
B	20	45	35
C	40	30	40

Worse than random guessing:

	A	B	C
A	15	40	35
B	43	25	35
C	37	30	30

Relative absolute error: If this is under 100%, then it is better than random guessing. Best result is 0%.

2.2. Association

Association rule learning is a popular method for discovering interesting relations between variables in large databases (Fayyad, 1996). It is most used for marketing. For example, if $\{\text{onions, potatoes}\} \Rightarrow \{\text{burger}\}$, it states that if a customer has potatoes and onions in his basket, it will most likely also buy burgers. Using this as an analogy, we can look at individual web pages as a “basket-item” and can research, which pages they will most likely also visit.

For Association rule learning the Apriori algorithm will be used (Lee & Kwon, 2008). This is the most popular and researched algorithm. While looking at the results, three outcomes are important:

- X = The frequency an item or a set of items A occurs
- Y = How many of the X items containing A, also contain item B, this is also known as the support
- Confidence = Can be calculated by Y/X

To give an example, this is a possible output from the Apriori-algorithm:

Page 1063=visited Page 1021=visited 81 ==> Page 1001=visited 74 conf:(0.91)

So, in the dataset there are 81 participants that visited pages 1063 and 1021. From these 81 participants, 74 (conf: $74/81=0,91 \rightarrow 91\%$), also visited Page 1001.

2.3. Data mining software: WEKA

Because of previous experience, Weka is the software of choice for the Data Mining. Weka is “a collection of machine learning algorithms for solving real-world data mining problems”. It was developed at the University of Waikato. WEKA enjoys widespread acceptance in both academia and business and has an active community with over 2 million downloads (Hall et al., 2009). It provides tools to execute all the algorithms that are explained above.

3. Research approach

To get valuable insight in Rechtwijzer.nl, three steps have to be performed:

1. The log data has to be processed, so it gives various variables that can be metrics for the data mining and it can be used within Weka and SPSS.
2. Extra variables have to be added to improve the results of the data mining.
3. Data mining has to be performed.

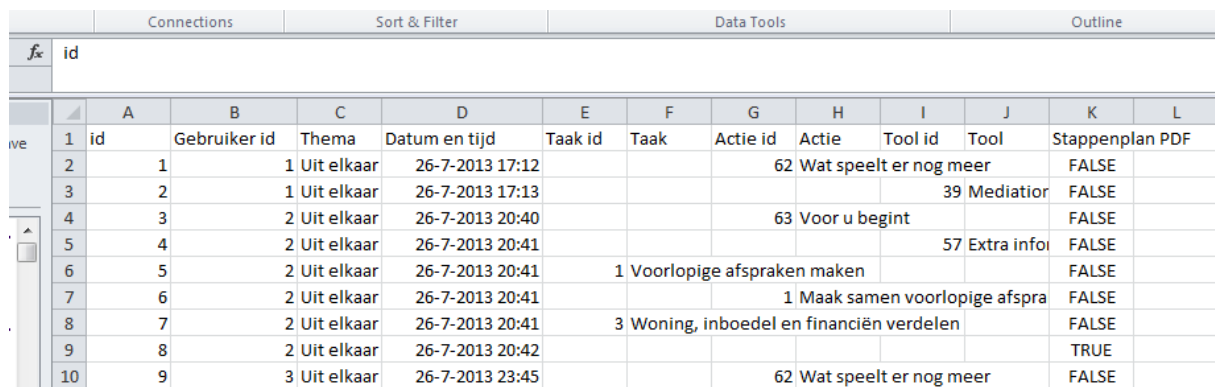
This paper provides a documentation of this process and the final results.

3.1 Step 1: data processing

For the data processing Python was used. Python is the well-used for data processing, because it has valuable extensions that support data processing, such as NumPy and Pandas.

In this chapter every step and the Python-code used for this is discussed.

Starting dataset



	Connections	Sort & Filter	Data Tools	Outline									
<i>f_x</i>	id												
	A	B	C	D	E	F	G	H	I	J	K	L	
ve	1	id	Gebruiker id	Thema	Datum en tijd	Taak id	Taak	Actie id	Actie	Tool id	Tool	Stappenplan PDF	
	2	1	1	Uit elkaar	26-7-2013 17:12			62	Wat speelt er nog meer			FALSE	
	3	2	1	Uit elkaar	26-7-2013 17:13					39	Mediator	FALSE	
	4	3	2	Uit elkaar	26-7-2013 20:40			63	Voor u begint			FALSE	
	5	4	2	Uit elkaar	26-7-2013 20:41					57	Extra info	FALSE	
	6	5	2	Uit elkaar	26-7-2013 20:41		1	Voorlopige afspraken maken				FALSE	
	7	6	2	Uit elkaar	26-7-2013 20:41				1	Maak samen voorlopige afspra		FALSE	
	8	7	2	Uit elkaar	26-7-2013 20:41		3	Woning, inboedel en financiën verdelen				FALSE	
	9	8	2	Uit elkaar	26-7-2013 20:42							TRUE	
	10	9	3	Uit elkaar	26-7-2013 23:45					62	Wat speelt er nog meer	FALSE	

Figure 1: Screenshot of starting dataset

Figure 1 provides a screenshot of the data (opened in Excel). It consists of 5437 rows. The actual file is a CSV-file. As can be seen there are several columns in the dataset:

- Id: the id number of a click
- Gebruikers id: the id number of the user that made that click
- Thema: This is the theme of the page where is clicked on. This is either for Divorce (“Uit Elkaar”) or consumer law (“Consumentenzaken”)
- Datum en tijd: the date and time of the click
- Taak id: the id number of a page that involves a task
- Taak: a short description of the task
- Actie id: the id number of a page that involves an action
- Action: a short description of the action
- Tool id: the id number of a page that involves a tool
- Tool: a short description of the tool
- Stappenplan PDF: Whether there is clicked on a download link to a PDF-file, this is either TRUE or FALSE.

3.1.1. Used programs

For the data processing Python 3.4 is used. This is used within the Text Editor Sublime Text 2. Furthermore Notepad was used to perform simple text mutations.

3.1.2. The code

To handle the data, various functions were written. A function can be recognized by its starting, which is `def`. A function consists of four parts. 1. The name (this is the first that comes after `def`) 2. The input for the function (this comes after the name and can be recognized, because they are within parentheses. 3. The actual code. This starts a line below with a 4-space indentation. 4 the output, this can be recognized by `return`.

```
import csv
import numpy as np
import pandas as pd
from pandas import DataFrame
import datetime
```

The first step is to import Python extensions that are used within the code:

- Csv: is used to handle the importing and exporting to and from CSV-files
- Numpy: a powerful package for scientific computing
- Pandas: provides high-performance, easy-to-use data structures and data analysis tools
- Dataframe: A DataFrame is a 2-dimensional labeled data structure with columns of potentially different types. It is similar to a table or a spreadsheet. In Python, it is the most used structure to handle datasets. Dataframe is used for the manipulation of this dataframe.
- Datetime: is used for transforming and calculating variables that are formatted like dates and/or times.

```
minvisit = 2
maxvisit = 600
```

These are variables that address the minimum and maximum time a visit to a page can last in order to be interpreted as one visit. Currently they are not used in the code, but they are there if this distinction is needed.

```
def csvload(csv):
    #This loads the CSV
    df = pd.read_csv(csv, index_col = 'id')
    df['Datum en tijd'] = pd.read_csv('Book1.csv')
    return df
```

This function loads the CSV and puts it in a Dataframe (df). The “Datum en tijd”-column is imported, because of some problems with the matching the id’s while importing this data.

```

17 def calculatevisit(df):
18     #This calculates the difference between clicktimes and sets the value to "NaN" for each new user
19     df['Datum en tijd'] = pd.to_datetime(df['Datum en tijd'], format= "%d-%m-%Y %H:%M:%S")
20     df['Visittime'] = (df['Datum en tijd']-df['Datum en tijd'].shift(1)).fillna(0).shift(-1)
21     df['Visittimesec'] = df['Visittime'] / np.timedelta64(1, 's')
22     df['New user?'] = df['Gebruiker id'].diff().fillna(0).astype(int).shift(-1)
23     df.loc[df['New user?'] == 1, 'Visittime'] = np.nan
24     df.loc[df['New user?'] == 1, 'Visittimesec'] = np.nan
25     return df
26

```

This function calculates the difference between clicktimes and sets the value to “empty” if it is the last page an user has visited.

Line 19 converts the “Datum en tijd”-column to the Python Datetime-format

Line 20 calculates the time-difference between two clicks

Line 21 divides this time-difference by one second, such that all values are in seconds.

Line 22 gives a value of 1 if the the next click is by another user, else the value is 0.

Line 23 and 24 delete the values of visittimes, if the next click is by another user.

```

27 def splitue(df):
28     #Filters on "Uit Elkaar"
29     dfue = df2[df2.Thema == "Uit elkaar"]
30     dfue = dfue.drop('Thema', 1)
31     return dfue
32
33 def splitcz(df):
34     #Filters on "Consumentenzaken"
35     dfcz = df2[df2.Thema == "Consumentenzaken"]
36     dfcz = dfcz.drop('Thema', 1)
37     return dfcz
38

```

These functions split the dataset in two separate datasets: one for divorce and one for consumer law.

```

def overallstats(df):
    #Groups the clicks per user and generates statistics
    peruser = df.groupby("Gebruiker id")
    #Determines the total visittime, the number of clicks, the first and last click and puts them in a
    totalvisit = peruser["Visittimesec"].sum()
    totalclicks = peruser["Page number"].count().astype(int)
    firstclick = peruser["Page number"].first().astype(int)
    lastclick = peruser["Page number"].last().astype(int)
    click2 = secondclick(df)
    statistics = pd.concat([totalvisit, totalclicks, firstclick, click2, lastclick], axis=1)
    statistics.columns = ["Total visit time", "#Clicks", "First page", "Second page", "Last page"]
    return statistics

```

```

def secondclick(df):
    df['Next Click'] = df["Page number"].shift(-1).astype(float)
    df.loc[df['New user?'] == 1, 'Next Click'] = np.nan

    peruser = df.groupby("Gebruiker id")
    secondclick = peruser["Next Click"].first().astype(float)
    return secondclick

```

This function determines the total visit time of a user, the total number of clicks of a user, the first, second and last click of this user and puts them in a dataframe together. Because determining the second click is more extensive, this is done by a separate function.


```

60 def pagenumber(df2):
61     #Puts Taak id, Actie id en Tool id in 1 column, types differ on number of characters
62     df2['Taak id'] = df2['Taak id'] + 100
63     df2['Actie id'] = df2['Actie id'] + 1000
64     df2['Tool id'] = df2['Tool id'] + 10000
65     df2['Taak id'] = df2['Taak id'].fillna(0)
66     df2['Tool id'] = df2['Tool id'].fillna(0)
67     df2['Actie id'] = df2['Actie id'].fillna(0)
68     df2["Page number"] = df2['Actie id'] + df2['Tool id'] + df2['Taak id']
69     return df2
70

```

Because tasks, actions and tools are actually the same: a certain page, they can be stored in one column, instead of three. To keep the distinction between them they have a different amount of digits (Task: 3, Action: 4, Tool: 5). This function puts them in one column, called “page number”.

```

72 def pagestats(df, col):
73     #This determines the total count of page visits per page per user and the visittime
74     a = df.copy()
75     a.loc[df['Page number'] != col, "Page number"] = np.nan
76     a.loc[df['Page number'] != col, "Visittimesec"] = np.nan
77     #a.loc[a['Visittimesec'] > maxvisit, 'Visittimesec'] = np.nan
78     #a.loc[a['Visittimesec'] < minvisit, 'Visittimesec'] = np.nan
79     col = col.astype(int)
80     peruser = a.groupby("Gebruiker id")
81     totalclicks = peruser["Page number"].count()
82     totalvisit = peruser["Visittimesec"].sum()
83     pagestat = pd.concat([totalclicks, totalvisit], axis = 1)
84     pagestat.columns = [col, col+0.1]
85     return pagestat
86
87 def allpages(df):
88     #finds all unique page numbers
89     b = df["Page number"]
90     a = pd.unique(b.ravel()).astype(int)
91     return a
92
93 def pagestats_all(df, pagenr):
94     stats = pd.DataFrame()
95     for page in pagenr:
96         #global stats
97         a = pagestats(df, page)
98         stats = pd.concat([stats, a], axis = 1)
99     return stats
100

```

The allpages function finds all unique values within the “Page number”-column. This provides a series of alle pages visited. The pagestats_all loops through these pages. It starts with an empty dataframe and adds the pagestats per page. After this function has looped through each page, it returns a dataframe with the pagestats for all pages. The pagestats are determined by the pagestats function. This function deletes all rows that are not the same as the investigated page. The remaining pages are grouped per user. Per user the total visits are counted and the visit time is summed. It returns these values to the pagestats_all-loop.

```

124 def generate_csvue(statue, pagestatsue):
125     final_csv = pd.concat([statue, pagestatsue], axis = 1)
126     final_csv.to_csv("final_uitelkaar.csv")
127     return final_csv
128
129 def generate_csvcz(statcz, pagestatcz):
130     final_csv = pd.concat([statcz, pagestatcz], axis = 1)
131     final_csv.to_csv("final_consumentenzaken.csv")
132

```

Finally, the overall stats and the stats per page are put together in one dataframe. This dataframe is exported to a CSV-file.

3.1.3. Changing to a binary dataset

One can hypothesize that it is not important how many times a page is visited, but solely the fact that the page is visited is enough. For that reason, the dataset was also converted to binary values. In this dataset the visit-time has been removed and the counter of visited pages is only limited to 0 (=0 visits in primary dataset), or 1 (>0 visits in primary dataset)

```
def pagestatsbin(df, col):
    #This determines the total count of page visits per page per user and the visittime
    a = df.copy()
    a.loc[df['Page number'] != col, "Page number"] = np.nan
    a.loc[df['Page number'] != col, "Visittimesec"] = np.nan
    a.loc[a['Visittimesec'] > maxvisit, 'Visittimesec'] = maxvisit
    a.loc[a['Visittimesec'] < minvisit, 'Visittimesec'] = np.nan
    col = col.astype(int)
    peruser = a.groupby("Gebruiker id")
    clicked = peruser["Page number"].count()
    clicked[clicked > 0] = 1
    pagestat = pd.concat([clicked], axis = 1)
    pagestat.columns = [col]
    return pagestat

def pagestats_bin(df, pagenr):
    stats = pd.DataFrame()
    for page in pagenr:
        #global stats
        a = pagestatsbin(df, page)
        stats = pd.concat([stats, a], axis = 1)
    return stats
```

3.1.4. Option for Data Cleaning

Some values can be disruptive. For example, a visit with a duration of <3 seconds can hardly be seen as a visit and for a visit of 15 minutes we can assume that the participant was not actively using that page the whole time. For that reason, such visits could be filtered. In the code the lines start with an hashtag, meaning that they are comments and not part of the code. Removing the hashtag will make these lines part of the code.

```
72 def pagestats(df, col):
73     #This determines the total count of page visits per page per user and the visittime
74     a = df.copy()
75     a.loc[df['Page number'] != col, "Page number"] = np.nan
76     a.loc[df['Page number'] != col, "Visittimesec"] = np.nan
77     #a.loc[a['Visittimesec'] > maxvisit, 'Visittimesec'] = np.nan
78     #a.loc[a['Visittimesec'] < minvisit, 'Visittimesec'] = np.nan
79     col = col.astype(int)
80     peruser = a.groupby("Gebruiker id")
81     totalclicks = peruser["Page number"].count()
82     totalvisit = peruser["Visittimesec"].sum()
83     pagestat = pd.concat([totalclicks, totalvisit], axis = 1)
84     pagestat.columns = [col, col+0.1]
85     return pagestat
```

3.2. Step 2: Adding extra variables

For Data Mining the data has to be transformed in such a way that the results are as good as possible with the given data. This transformation depends on which types of outcomes you want to have and rely on the business environment of the dataset.

In past research on Rechtwijzer.nl a study was done on the self-efficacy of the participants. Unfortunately no change was found in self-efficacy after the usage of Rechtwijzer.nl, comparing the level of self-efficacy before and after the usage of the website. However, we can hypothesize that participants with different self-efficacy show different behaviour while using Rechtwijzer.nl, hence will visit different pages.

A list of pages was provided, which ought to be especially interesting for people with a high self-efficacy, because they provide actions, information and tools that can be done by the participants themselves.

Furthermore a list of pages was provided, which ought to be especially interesting for people with a low self-efficacy because they provide actions, information and tools that can address to help by others.

For both these types of pages, for both the actions as the tools, the total number of visits to such pages and the total time spend on these pages was calculated per participant. Furthermore, the average time spend per page was calculated. The total number of visits was also determined for the binary dataset.

```
def extravariablen(df):
    #Determines the frequency, total time and average time spend on pages that relate to self-efficacy, for acti
    df['CountSEpagesAction'] = df['1001.0'] + df['1005.0'] + df['1006.0'] + df['1008.0'] + df['1009.0'] + df['10
    df['TimeSEpagesAction'] = df['1001.1'].fillna(0) + df['1005.1'].fillna(0) + df['1006.1'].fillna(0) + df['10
    df['TimeSEpagesActionAVG'] = 0
    df.loc[df['TimeSEpagesAction'] != 0, 'TimeSEpagesActionAVG'] = df['TimeSEpagesAction'] / df['CountSEpagesAc

    df['CountSEpagesTool'] = df['10001.0'] + df['10002.0'] + df['10003.0'] + df['10005.0'] + df['10007.0'] + df
    df['TimeSEpagesTool'] = df['10007.1'].fillna(0) + df['10008.1'].fillna(0) + df['10044.1'].fillna(0) + df['1
    df['TimeSEpagesToolAVG'] = 0
    df.loc[df['TimeSEpagesTool'] != 0, 'TimeSEpagesToolAVG'] = df['TimeSEpagesTool'] / df['CountSEpagesTool']

    #Determines the frequency, total time and average time spend on pages that relate to finding help, for acti
    df['CountHelppagesAction'] = df['1002.0'] + df['1007.0'] + df['1011.0'] + df['1018.0'] + df['1031.0'] + df[
    df['TimeHelppagesAction'] = df['1002.1'].fillna(0) + df['1007.1'].fillna(0) + df['1011.1'].fillna(0) + df[
    df['TimeHelppagesActionAVG'] = 0
    df.loc[df['TimeHelppagesAction'] != 0, 'TimeHelppagesActionAVG'] = df['TimeHelppagesAction'] / df['CountHel

    df['CountHelppagesTool'] = df['10006.0'] + df['10010.0'] + df['10012.0'] + df['10015.0'] + df['10035.0'] +
    df['TimeHelppagesTool'] = df['10006.1'].fillna(0) + df['10010.1'].fillna(0) + df['10012.1'].fillna(0) + df[
    df['TimeHelppagesToolAVG'] = 0
    df.loc[df['TimeHelppagesTool'] != 0, 'TimeHelppagesToolAVG'] = df['TimeHelppagesTool'] / df['CountHelppages
    return df
```

Besides these extra variables, a class should be added. Using Excel, the lower and upper bounds for each class was obtained. Using these ranges, the participants were put in 2, 3 and 4 classes.

```

def valtoecat2(df):
    df["Category SE"] = np.nan
    df.loc[df['SelfEfficacy'] < 6.01, "Category SE"] = "Class 2"
    df.loc[df['SelfEfficacy'] < 4.18, "Category SE"] = "Class 1"
    return df

def valtoecat3(df):
    df["Category SE"] = np.nan
    df.loc[df['SelfEfficacy'] < 6.01, "Category SE"] = "Class 3"
    df.loc[df['SelfEfficacy'] < 4.68, "Category SE"] = "Class 2"
    df.loc[df['SelfEfficacy'] < 3.68, "Category SE"] = "Class 1"
    return df

def valtoecat4(df):
    df["Category SE"] = np.nan
    df.loc[df['SelfEfficacy'] < 6.01, "Category SE"] = "Class 4"
    df.loc[df['SelfEfficacy'] < 4.99, "Category SE"] = "Class 3"
    df.loc[df['SelfEfficacy'] < 4.18, "Category SE"] = "Class 2"
    df.loc[df['SelfEfficacy'] < 3.34, "Category SE"] = "Class 1"
    return df

```

Finally, the dataset was prepared to be used for Association Rules Learning. Because the Apriori – Algorithm only can make use of nominal values, the binary dataset was used. In this set, 0 visits-pages were deleted and pages with the value 1 (thus visited) were changed to the value “visited”. This step was mostly done in Excel and Notepad.

3.3. Step 3: Data mining

The final step is the actual Data Mining. For this, the tool Weka was used. In this chapter, only the relevant values are provided. The extended results can be found in the Appendix.

3.3.1. Classification

Dataset		Decision Tree		Naive Bayes	
Type	#Categories	% Correct predicted	Relative Absolute Errors	% Correct predicted	Relative Absolute Errors
Regular	2	54,824%	98,494%	52,706%	94,909%
Filtered	2	49,882%	99,578%	53,177%	94,954%
Regular	3	40,471%	97,458%	40,235%	90,747%
Filtered	3	39,529%	98,370%	38,588%	93,112%
Regular	4	28,235%	99,189%	28,000%	95,513%
Filtered	4	29,647%	99,306%	29,412%	94,556%

Table 1

All the results are only slightly better than 100%/# Classes. Dividing in three classes relatively provides the best results, but they are still poor. By filtering extremely long and short visits, only provides better results in 3 out of 6 results. In conclusion it can be said that, with the techniques used, it is not possible to predict the (classes of) self-efficacy,

3.3.2. Linear Regression

As said before, classification is not meant for numerical values divided in classes. Because of that, linear regression might provide better results.

Dataset	Linear Regression
Type	% Correct predicted
Regular	-0,014
Filtered	-0,014
Bin	0,1984
Bin & filtered	0,1084

Table 2

At first, the regular set provides almost no correlation. Using the binary dataset drastically improves the correlation, but is still very low. Filtered data provides the same or even worse results. So, irrelevant of which type of dataset is used, with the use of linear regression it is not possible to predict the level of self-efficacy.

3.3.3. Association

The real value of a rule found by association, in terms of usefulness and actionability is subjective and depends heavily of the particular domain and business objectives. Because the classification did not give any good results, it can be assumed that the found association rules are not related to self-efficacy.

Furthermore, because the participants are following a “path” on the website, it can be assumed that some pages visited are dependent from each other. For example, if one page contains a link to another page. Some pages can only be accessed from other pages. The link between such pages is obvious and will not give extra insight in the usage of the website. Because such links between pages result in a high confidence, they will be at the top, if we rank the rules on their confidence. Thus, another way of ranking should be used.

Support is also a popular metric, but suffers from the “rare item problem”: rules that give valuable insight, but only occur a few times, are pruned. An alternative is lift: lift measures how many times more often X and Y occur together than expected, if they were statistically independent. It can be found by dividing the confidence by the unconditional probability of Y. In this way, pages that are associated with each other and do not necessarily have 1,0 confidence (and thus are part of the “path”), will be on the top of the list. Unfortunately, Weka does not provide a minimum metric for confidence, while using lift. This should be kept in mind, while assessing the top rules found.

The top 50 rules found are provided in Appendix B.1. For convenience, the output has been translated to the pages they addressed in the original log-output.

These results should be evaluated. Despite the fact that the results should support independency, some results are obvious. Rule #1: “Maak een lijst van de bezittingen en schulden” and “Checklist bezittingen” is a clear example for this. For such kind of associations, for some the confidence rate provides insight in the conversion rate, in above example 81% of the participants whom visited the “Maak een lijst van de bezittingen en schulden” –page, also used the tool. Improving this number could be a goal for the developers of Rechtwijzer. Other results, like the association “Naar de advocaat” and “Scheiding definitief maken” provide some extra insight in the behaviour of the participant on the website.

4. Conclusion

The conclusion of this research is very brief: with the current data, it is not possible to predict self-efficacy by using the surf-behaviour on Rechtwijzer.nl.

Association can be used to evaluate the conversion from task to action and from action to tool by looking at the confidence-rate of the association. This information can be used for improving the website. Furthermore, some associations are less obvious. These associations should be evaluated to gain extra insight in the behaviour of the participants.

5. Literature

Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996). "From Data Mining to Knowledge Discovery in Databases" *AI Magazine* 17(3), 37-54.

Hall, M. et al. (2009). "The WEKA data mining software: an update". *SIGKDD Explor.* 11, 10–18

Lee, H.; Nguyen, N.; Kwon, J. (2008) "Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence"" ; Springer: Berlin/Heidelberg, Germany, 2008; Volume 4682, pp. 952–960.

Ormrod, J. E. (2006). *Educational psychology: Developing learners* (5th ed.). Upper Saddle River, N.J.: Pearson/Merrill Prentice Hall.

Sullivan, David G. (2013) "Data Mining IV: Preparing the Data" Ph.D. Computer Science, Boston University, Spring 2013

6. Appendices

6.1. Appendix A: Results Classification

Two categories Decision Tree

J48 pruned tree

```
1005.0 <= 0
| 1001.0 <= 0
| | 1016.0 <= 0: Class 1 (265.24/113.12)
| | 1016.0 > 0: Class 2 (13.31/5.16)
| 1001.0 > 0
| | 10002.0 <= 1: Class 2 (73.73/24.87)
| | 10002.0 > 1: Class 1 (10.24/2.12)
1005.0 > 0
| 10047.0 <= 0
| | 10006.0 <= 0
| | | 113.0 <= 0: Class 2 (41.99/12.49)
| | | 113.0 > 0: Class 1 (5.12/1.06)
| | 10006.0 > 0: Class 1 (6.14/1.07)
```

| 10047.0 > 0: Class 2 (9.22/1.11)

Number of Leaves : 8

Size of the tree : 15

Time taken to build model: 4.31 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	233	54.8235 %
Incorrectly Classified Instances	192	45.1765 %
Kappa statistic	0.0958	
Mean absolute error	0.4925	
Root mean squared error	0.5037	
Relative absolute error	98.4942 %	
Root relative squared error	100.7377 %	
Total Number of Instances	425	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.396	0.3	0.568	0.396	0.467	0.54	Class 2
	0.7	0.604	0.538	0.7	0.608	0.54	Class 1
Weighted Avg.	0.548	0.452	0.553	0.548	0.538	0.54	

=== Confusion Matrix ===

```
a b <-- classified as
84 128 | a = Class 2
64 149 | b = Class 1
```

Two categories Naïve Bayes

Correctly Classified Instances	224	52.7059 %
Incorrectly Classified Instances	201	47.2941 %
Kappa statistic	0.0525	
Mean absolute error	0.4746	
Root mean squared error	0.6844	
Relative absolute error	94.9093 %	
Root relative squared error	136.8791 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```
a b <-- classified as
37 175 | a = Class 2
26 187 | b = Class 1
```

Two categories Decision Tree with filtered visits

J48 pruned tree

```
10008.0 <= 0
| 1063.0 <= 0: Class 1 (117.87/45.66)
| 1063.0 > 0
| | 10039.0 <= 0
| | | 103.0 <= 0: Class 2 (206.27/84.62)
| | | 103.0 > 0: Class 1 (55.53/24.14)
| | | 10039.0 > 0: Class 1 (22.67/6.28)
| 10039.0 > 0: Class 2 (22.67/8.39)
```

Number of Leaves : 5

Size of the tree : 9

Time taken to build model: 0.49 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	212	49.8824 %
Incorrectly Classified Instances	213	50.1176 %
Kappa statistic	-0.0027	
Mean absolute error	0.4979	
Root mean squared error	0.5057	
Relative absolute error	99.5775 %	
Root relative squared error	101.1369 %	
Total Number of Instances	425	

=== Confusion Matrix ===

a b <-- classified as

92 120 | a = Class 2

93 120 | b = Class 1

Two categories Naïve Bayes with filtered visits

=== Summary ===

Correctly Classified Instances	226	53.1765 %
Incorrectly Classified Instances	199	46.8235 %
Kappa statistic	0.0623	
Mean absolute error	0.4747	
Root mean squared error	0.6508	
Relative absolute error	94.9451 %	
Root relative squared error	130.1599 %	
Total Number of Instances	425	

=== Confusion Matrix ===

a b <-- classified as
55 157 | a = Class 2
42 171 | b = Class 1

Three categories Decision Tree

J48 pruned tree

```
1014.0 <= 0
| 1001.0 <= 0: Class 1 (282.65/153.66)
| 1001.0 > 0
| | 1015.0 <= 0
| | | 10048.0 <= 0
| | | | 1019.0 <= 0
| | | | | 1039.0 <= 0
| | | | | | 10005.0 <= 0
| | | | | | | 10039.0 <= 0: Class 3 (62.47/31.03)
| | | | | | | 10039.0 > 0: Class 2 (12.29/3.17)
| | | | | | | 10005.0 > 0
| | | | | | | TimeSEpagesToolAVG <= 13.428571: Class 3 (12.53/4.37)
| | | | | | | TimeSEpagesToolAVG > 13.428571: Class 1 (10.0/3.0)
| | | | | 1039.0 > 0: Class 2 (6.14/2.09)
| | | | 1019.0 > 0: Class 3 (6.14/1.1)
| | | 10048.0 > 0: Class 2 (5.12/1.07)
| | 1015.0 > 0: Class 2 (12.29/3.17)
1014.0 > 0
| 1014.0 <= 1: Class 3 (11.27/3.19)
| 1014.0 > 1: Class 2 (4.1/1.06)
```

Number of Leaves : 11

Size of the tree : 21

Time taken to build model: 0.37 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	172	40.4706 %
Incorrectly Classified Instances	253	59.5294 %
Kappa statistic	0.0905	
Mean absolute error	0.432	
Root mean squared error	0.48	
Relative absolute error	97.458 %	
Root relative squared error	101.9633 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```

a b c <-- classified as
13 92 39 | a = Class 2
12 126 15 | b = Class 1
25 70 33 | c = Class 3

```

Three categories Naïve Bayes

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	171	40.2353 %
Incorrectly Classified Instances	254	59.7647 %
Kappa statistic	0.0859	
Mean absolute error	0.4023	
Root mean squared error	0.6146	
Relative absolute error	90.7472 %	
Root relative squared error	130.5485 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```

a b c <-- classified as
35 84 25 | a = Class 2
23 111 19 | b = Class 1
28 75 25 | c = Class 3

```

Three categories Decision Tree with filtered visits

J48 pruned tree

```

1001.0 <= 0
| 1012.0 <= 0
| | 0.0 <= 0: Class 1 (221.0/114.56)
| | 0.0 > 0: Class 2 (81.6/44.3)
| 1012.0 > 0: Class 3 (18.13/9.45)
1001.0 > 0
| 10002.0 <= 0: Class 3 (35.13/15.81)
| 10002.0 > 0: Class 2 (69.13/39.18)

```

Number of Leaves : 5

Size of the tree : 9

Time taken to build model: 0.57 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	168	39.5294 %
Incorrectly Classified Instances	257	60.4706 %
Kappa statistic	0.0728	
Mean absolute error	0.4361	

Root mean squared error	0.4739
Relative absolute error	98.3704 %
Root relative squared error	100.6611 %
Total Number of Instances	425

=== Confusion Matrix ===

```

a b c <-- classified as
30 92 22 | a = Class 2
20 117 16 | b = Class 1
31 76 21 | c = Class 3

```

Three categories Naïve Bayes with filtered visits

Correctly Classified Instances	164	38.5882 %
Incorrectly Classified Instances	261	61.4118 %
Kappa statistic	0.0605	
Mean absolute error	0.4128	
Root mean squared error	0.6214	
Relative absolute error	93.1118 %	
Root relative squared error	131.9787 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```

a b c <-- classified as
20 91 33 | a = Class 2
19 117 17 | b = Class 1
22 79 27 | c = Class 3

```

Four categories Decision Tree

J48 pruned tree

```

1037.0 <= 0
| 1014.0 <= 0
| | 106.0 <= 0
| | | 1132.0 <= 0
| | | | 10035.0 <= 0: Class 2 (280.6/202.28)
| | | | 10035.0 > 0: Class 1 (77.83/44.28)
| | | 1132.0 > 0: Class 1 (21.51/10.35)
| | 106.0 > 0: Class 3 (13.31/7.25)
| 1014.0 > 0
| | 1018.0 <= 0: Class 3 (8.19/3.15)
| | 1018.0 > 0: Class 4 (7.17/2.12)
1037.0 > 0
| 1062.0 <= 0: Class 4 (8.19/3.13)
| 1062.0 > 0: Class 2 (8.19/4.15)

```

Number of Leaves : 8

Size of the tree : 15

Time taken to build model: 0.52 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	120	28.2353 %
Incorrectly Classified Instances	305	71.7647 %
Kappa statistic	0.0394	
Mean absolute error	0.372	
Root mean squared error	0.4355	
Relative absolute error	99.1984 %	
Root relative squared error	100.5755 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```
a b c d <-- classified as
11 13 59 22 | a = Class 3
13 10 51 30 | b = Class 2
 9 15 70 15 | c = Class 1
15 14 49 29 | d = Class 4
```

Four categories Naïve Bayes

=== Summary ===

Correctly Classified Instances	119	28	%
Incorrectly Classified Instances	306	72	%
Kappa statistic	0.0361		
Mean absolute error	0.3582		
Root mean squared error	0.5819		
Relative absolute error	95.5125 %		
Root relative squared error	134.384 %		
Total Number of Instances	425		

=== Confusion Matrix ===

```
a b c d <-- classified as
10 17 66 12 | a = Class 3
13 19 57 15 | b = Class 2
 5 18 73 13 | c = Class 1
11 16 63 17 | d = Class 4
```

Four categories Decision Tree with filtered visits

J48 pruned tree

```
114.1 <= 10
| 1001.0 <= 0
| | 0.0 <= 0: Class 1 (109.06/72.67)
| | 0.0 > 0: Class 3 (41.06/27.56)
| 1001.0 > 0: Class 4 (46.03/28.36)
```

```

114.1 > 10
| 10008.0 <= 0
| | 1015.0 <= 0: Class 1 (202.61/145.35)
| | 1015.0 > 0: Class 2 (14.04/8.73)
| 10008.0 > 0: Class 4 (12.21/6.96)

```

Number of Leaves : 6

Size of the tree : 11

Time taken to build model: 0.89 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	126	29.6471 %
Incorrectly Classified Instances	299	70.3529 %
Kappa statistic	0.0579	
Mean absolute error	0.3724	
Root mean squared error	0.4345	
Relative absolute error	99.3055 %	
Root relative squared error	100.3531 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```

a b c d <-- classified as
23 4 59 19 | a = Class 3
28 3 56 17 | b = Class 2
19 5 75 10 | c = Class 1
17 8 57 25 | d = Class 4

```

Four categories Naïve Bayes with filtered visits

=== Summary ===

Correctly Classified Instances	125	29.4118 %
Incorrectly Classified Instances	300	70.5882 %
Kappa statistic	0.055	
Mean absolute error	0.3546	
Root mean squared error	0.5583	
Relative absolute error	94.5562 %	
Root relative squared error	128.9398 %	
Total Number of Instances	425	

=== Confusion Matrix ===

```

a b c d <-- classified as
14 12 64 15 | a = Class 3
12 17 53 22 | b = Class 2
7 12 76 14 | c = Class 1
18 16 55 18 | d = Class 4

```

Linear Regression

Linear Regression Model

SelfEfficacy =

$$\begin{aligned} &0.0886 * \text{\#Clicks} + \\ &0.0048 * 1062.1 + \\ &0.0001 * 10039.1 + \\ &0.0001 * 1063.1 + \\ &0.0001 * 10057.1 + \\ &0.4281 * 1001.0 + \\ &-0.294 * 103.0 + \\ &-0.2197 * 10002.0 + \\ &0.0064 * 1002.1 + \\ &-3.1958 * 10044.0 + \\ &0.4236 * 10041.0 + \\ &0.2713 * 1008.0 + \\ &-0.8242 * 10005.0 + \\ &-0.5241 * 1009.0 + \\ &0.0052 * 1009.1 + \\ &0.0518 * 1024.1 + \\ &0.6424 * 1012.0 + \\ &-0.0733 * 1012.1 + \\ &0.0453 * 1019.1 + \\ &0.0068 * 105.1 + \\ &-0.5168 * 1015.0 + \\ &-0.0036 * 1015.1 + \\ &-0.3696 * 1021.0 + \\ &0.2465 * 10003.0 + \\ &-0.3813 * 102.0 + \\ &0.6936 * 113.0 + \\ &-0.6554 * 1033.0 + \\ &-0.9442 * 1132.0 + \\ &-0.5387 * 120.0 + \\ &0.0468 * 120.1 + \\ &0.2488 * 10035.0 + \\ &0.5615 * 10012.0 + \\ &-0.6761 * 1034.0 + \\ &-1.68 * 114.0 + \\ &0.9876 * 1039.0 + \\ &0.0039 * 10015.1 + \\ &-0.4081 * 10019.0 + \\ &0.3493 * 10008.0 + \\ &-0.002 * 10008.1 + \\ &0.0001 * 10073.1 + \\ &-0.0174 * 10048.1 + \\ &-0.0017 * 1022.1 + \\ &0.0383 * 1007.1 + \\ &0.4526 * 1014.0 + \end{aligned}$$

0.889 * 107.0 +
 1.0344 * 10010.0 +
 0.0001 * 10046.1 +
 -1.5137 * 10061.0 +
 1.6959 * 1131.0 +
 1.2045 * 1026.0 +
 -0.0001 * TimeSEpagesActionAVG +
 -0.0001 * TimeSEpagesTool +
 -0.0044 * TimeHelppagesAction +
 -0.2959 * CountHelppagesTool +
 -0.0001 * TimeHelppagesTool +
 3.0766

Time taken to build model: 2.08 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	-0.0136
Mean absolute error	7.833
Root mean squared error	84.9028
Relative absolute error	937.7206 %
Root relative squared error	8388.9113 %
Total Number of Instances	425
Linear Regression Model	

Linear regression filtered

SelfEfficacy =

0.0886 * #Clicks +
 0.0048 * 1062.1 +
 0.0001 * 10039.1 +
 0.0001 * 1063.1 +
 0.0001 * 10057.1 +
 0.4281 * 1001.0 +
 -0.294 * 103.0 +
 -0.2197 * 10002.0 +
 0.0064 * 1002.1 +
 -3.1958 * 10044.0 +
 0.4236 * 10041.0 +
 0.2713 * 1008.0 +
 -0.8242 * 10005.0 +
 -0.5241 * 1009.0 +
 0.0052 * 1009.1 +
 0.0518 * 1024.1 +
 0.6424 * 1012.0 +
 -0.0733 * 1012.1 +
 0.0453 * 1019.1 +
 0.0068 * 105.1 +
 -0.5168 * 1015.0 +
 -0.0036 * 1015.1 +

-0.3696 * 1021.0 +
 0.2465 * 10003.0 +
 -0.3813 * 102.0 +
 0.6936 * 113.0 +
 -0.6554 * 1033.0 +
 -0.9442 * 1132.0 +
 -0.5387 * 120.0 +
 0.0468 * 120.1 +
 0.2488 * 10035.0 +
 0.5615 * 10012.0 +
 -0.6761 * 1034.0 +
 -1.68 * 114.0 +
 0.9876 * 1039.0 +
 0.0039 * 10015.1 +
 -0.4081 * 10019.0 +
 0.3493 * 10008.0 +
 -0.002 * 10008.1 +
 0.0001 * 10073.1 +
 -0.0174 * 10048.1 +
 -0.0017 * 1022.1 +
 0.0383 * 1007.1 +
 0.4526 * 1014.0 +
 0.889 * 107.0 +
 1.0344 * 10010.0 +
 0.0001 * 10046.1 +
 -1.5137 * 10061.0 +
 1.6959 * 1131.0 +
 1.2045 * 1026.0 +
 -0.0001 * TimeSEpagesActionAVG +
 -0.0001 * TimeSEpagesTool +
 -0.0044 * TimeHelppagesAction +
 -0.2959 * CountHelppagesTool +
 -0.0001 * TimeHelppagesTool +
 3.0766

Time taken to build model: 2.08 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	-0.0136
Mean absolute error	7.833
Root mean squared error	84.9028
Relative absolute error	937.7206 %
Root relative squared error	8388.9113 %
Total Number of Instances	425

Linear Regression Binary

Linear Regression Model

SelfEfficacy =

0 * Total visit time +
 -0.0262 * #Clicks +
 -0.2009 * 1062 +
 -0.4462 * 10039 +
 0.3499 * 1001 +
 -0.208 * 103 +
 0.1824 * 0 +
 -0.3445 * 10005 +
 -0.4833 * 1009 +
 0.3531 * 1010 +
 0.5468 * 1018 +
 0.4227 * 1012 +
 0.2736 * 105 +
 -0.365 * 1015 +
 -0.3246 * 102 +
 -0.6252 * 113 +
 -1.0982 * 1132 +
 0.4389 * 1134 +
 -0.3523 * 10006 +
 -0.9379 * 10040 +
 -0.9436 * 1034 +
 0.7945 * 1039 +
 -0.633 * 10015 +
 0.6862 * 10008 +
 0.7661 * 1014 +
 1.0815 * 107 +
 0.7946 * 109 +
 0.7121 * 1031 +
 0.9178 * 1131 +
 -1.1992 * 10074 +
 0.1377 * CountSEpagesAction +
 0.1029 * CountHelppagesTool +
 4.0579

Time taken to build model: 0.28 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.1984
Mean absolute error	0.8637
Root mean squared error	1.0742
Relative absolute error	103.3977 %
Root relative squared error	106.1413 %
Total Number of Instances	425

Linear Regression Binary & filtered

SelfEfficacy =

0 * Total visit time +

-0.3475 * 1062 +
 0.2028 * 10057 +
 0.3703 * 1001 +
 -0.3289 * 103 +
 0.2556 * 0 +
 -0.3631 * 1002 +
 -0.5738 * 1024 +
 0.692 * 1012 +
 -0.3332 * 102 +
 -0.8755 * 1033 +
 -0.7246 * 1132 +
 0.5528 * 10012 +
 -0.5309 * 1034 +
 0.5098 * 10008 +
 1.0129 * 107 +
 1.1624 * 10010 +
 -1.217 * 10074 +
 0.0638 * CountSEpagesAction +
 0.1551 * CountHelppagesAction +
 -0.1777 * CountHelppagesTool +
 4.0716

Time taken to build model: 0.23 seconds

=== Cross-validation ===

=== Summary ===

Correlation coefficient	0.1014
Mean absolute error	0.8869
Root mean squared error	1.0859
Relative absolute error	106.1695 %
Root relative squared error	107.2958 %
Total Number of Instances	425

6.2. Appendix B: Results Association Rule Learning

Direct results

1. Page 1008=visited 84 ==> Page 10005=visited 68 conf:(0.81) < lift:(5.07)> lev:(0.13) [54] conv:(4.15)
2. Page 10005=visited 68 ==> Page 1008=visited 68 conf:(1) < lift:(5.07)> lev:(0.13) [54] conv:(54.59)
3. Page 104=visited 97 ==> Page 105=visited 69 conf:(0.71) < lift:(3.84)> lev:(0.12) [51] conv:(2.72)
4. Page 105=visited 79 ==> Page 104=visited 69 conf:(0.87) < lift:(3.84)> lev:(0.12) [51] conv:(5.55)
5. Page 101=visited Page 103=visited 97 ==> Page 104=visited 76 conf:(0.78) < lift:(3.44)> lev:(0.13) [53] conv:(3.41)
6. Page 104=visited 97 ==> Page 101=visited Page 103=visited 76 conf:(0.78) < lift:(3.44)> lev:(0.13) [53] conv:(3.41)
7. Page 10007=visited 92 ==> Page 1006=visited 67 conf:(0.73) < lift:(3.37)> lev:(0.11) [47] conv:(2.77)
8. Page 1006=visited 92 ==> Page 10007=visited 67 conf:(0.73) < lift:(3.37)> lev:(0.11) [47] conv:(2.77)

9. Page 101=visited Page 1001=visited 93 ==> Page 10002=visited 69 conf:(0.74) < lift:(3.29)> lev:(0.11) [48] conv:(2.88)

10. Page 10002=visited 96 ==> Page 101=visited Page 1001=visited 69 conf:(0.72) < lift:(3.29)> lev:(0.11) [48] conv:(2.68)

11. Page 103=visited 117 ==> Page 101=visited Page 104=visited 76 conf:(0.65) < lift:(3.22)> lev:(0.12) [52] conv:(2.22)

12. Page 101=visited Page 104=visited 86 ==> Page 103=visited 76 conf:(0.88) < lift:(3.22)> lev:(0.12) [52] conv:(5.67)

13. Page 102=visited 97 ==> Page 1006=visited 67 conf:(0.69) < lift:(3.2)> lev:(0.11) [46] conv:(2.45)

14. Page 1006=visited 92 ==> Page 102=visited 67 conf:(0.73) < lift:(3.2)> lev:(0.11) [46] conv:(2.73)

15. Page 1001=visited 134 ==> Page 10002=visited 96 conf:(0.72) < lift:(3.18)> lev:(0.15) [65] conv:(2.66)

16. Page 10002=visited 96 ==> Page 1001=visited 96 conf:(1) < lift:(3.18)> lev:(0.15) [65] conv:(65.8)

17. Page 1001=visited 134 ==> Page 1063=visited Page 10002=visited 86 conf:(0.64) < lift:(3.18)> lev:(0.14) [58] conv:(2.18)

18. Page 1063=visited Page 10002=visited 86 ==> Page 1001=visited 86 conf:(1) < lift:(3.18)> lev:(0.14) [58] conv:(58.95)

19. Page 1001=visited 134 ==> Page 101=visited Page 10002=visited 69 conf:(0.51) < lift:(3.18)> lev:(0.11) [47] conv:(1.7)

20. Page 101=visited Page 10002=visited 69 ==> Page 1001=visited 69 conf:(1) < lift:(3.18)> lev:(0.11) [47] conv:(47.3)

21. Page 103=visited 117 ==> Page 104=visited 84 conf:(0.72) < lift:(3.15)> lev:(0.13) [57] conv:(2.66)

22. Page 104=visited 97 ==> Page 103=visited 84 conf:(0.87) < lift:(3.15)> lev:(0.13) [57] conv:(5.03)

23. Page 1063=visited Page 1001=visited 123 ==> Page 10002=visited 86 conf:(0.7) < lift:(3.1)> lev:(0.14) [58] conv:(2.51)

24. Page 10002=visited 96 ==> Page 1063=visited Page 1001=visited 86 conf:(0.9) < lift:(3.1)> lev:(0.14) [58] conv:(6.21)

25. Page 103=visited 117 ==> Page 1063=visited Page 104=visited 68 conf:(0.58) < lift:(3.09)> lev:(0.11) [46] conv:(1.9)

26. Page 1063=visited Page 104=visited 80 ==> Page 103=visited 68 conf:(0.85) < lift:(3.09)> lev:(0.11) [46] conv:(4.46)

27. Page 103=visited 117 ==> Page 105=visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)

28. Page 105=visited 79 ==> Page 103=visited 67 conf:(0.85) < lift:(3.09)> lev:(0.11) [45] conv:(4.41)

29. Page 1063=visited Page 103=visited 98 ==> Page 104=visited 68 conf:(0.69) < lift:(3.05)> lev:(0.11) [45] conv:(2.44)

30. Page 104=visited 97 ==> Page 1063=visited Page 103=visited 68 conf:(0.7) < lift:(3.05)> lev:(0.11) [45] conv:(2.49)

31. Page 1001=visited 134 ==> Page 1063=visited Page 1021=visited 74 conf:(0.55) < lift:(2.9)> lev:(0.11) [48] conv:(1.78)

32. Page 1063=visited Page 1021=visited 81 ==> Page 1001=visited 74 conf:(0.91) < lift:(2.9)> lev:(0.11) [48] conv:(6.94)

33. Page 1001=visited 134 ==> Page 1021=visited 81 conf:(0.6) < lift:(2.86)> lev:(0.12) [52] conv:(1.96)

34. Page 1021=visited 90 ==> Page 1001=visited 81 conf:(0.9) < lift:(2.86)> lev:(0.12) [52] conv:(6.17)

35. Page 1063=visited Page 1001=visited 123 ==> Page 1021=visited 74 conf:(0.6) < lift:(2.85)> lev:(0.11) [48] conv:(1.94)

36. Page 1021=visited 90 ==> Page 1063=visited Page 1001=visited 74 conf:(0.82) < lift:(2.85)> lev:(0.11) [48] conv:(3.77)

37. Page 103=visited 117 ==> Page 1008=visited 65 conf:(0.56) < lift:(2.82)> lev:(0.1) [41] conv:(1.77)
38. Page 1008=visited 84 ==> Page 103=visited 65 conf:(0.77) < lift:(2.82)> lev:(0.1) [41] conv:(3.05)
39. Page 1001=visited 134 ==> Page 1063=visited Page 1002=visited 69 conf:(0.51) < lift:(2.74)> lev:(0.1) [43] conv:(1.65)
40. Page 1063=visited Page 1002=visited 80 ==> Page 1001=visited 69 conf:(0.86) < lift:(2.74)> lev:(0.1) [43] conv:(4.57)
41. Page 1001=visited 134 ==> Page 1002=visited 77 conf:(0.57) < lift:(2.69)> lev:(0.11) [48] conv:(1.82)
42. Page 1002=visited 91 ==> Page 1001=visited 77 conf:(0.85) < lift:(2.69)> lev:(0.11) [48] conv:(4.16)
43. Page 1063=visited Page 1001=visited 123 ==> Page 1002=visited 69 conf:(0.56) < lift:(2.63)> lev:(0.1) [42] conv:(1.76)
44. Page 1002=visited 91 ==> Page 1063=visited Page 1001=visited 69 conf:(0.76) < lift:(2.63)> lev:(0.1) [42] conv:(2.81)
45. Page 1063=visited Page 101=visited 127 ==> Page 102=visited 75 conf:(0.59) < lift:(2.59)> lev:(0.11) [46] conv:(1.85)
46. Page 102=visited 97 ==> Page 1063=visited Page 101=visited 75 conf:(0.77) < lift:(2.59)> lev:(0.11) [46] conv:(2.96)
47. Page 101=visited 150 ==> Page 103=visited Page 104=visited 76 conf:(0.51) < lift:(2.57)> lev:(0.11) [46] conv:(1.61)
48. Page 103=visited Page 104=visited 84 ==> Page 101=visited 76 conf:(0.9) < lift:(2.57)> lev:(0.11) [46] conv:(6.05)
49. Page 103=visited 117 ==> Page 102=visited 68 conf:(0.58) < lift:(2.55)> lev:(0.1) [41] conv:(1.81)
50. Page 102=visited 97 ==> Page 103=visited 68 conf:(0.7) < lift:(2.55)> lev:(0.1) [41] conv:(2.35)

Explained

1. Action 8 Maak een lijst van de bezittingen en schulden =visited 84 ==>Tool 5 Checklist bezittingen en schulden=visited 68 conf:(0.81) < lift:(5.07)> lev:(0.13) [54] conv:(4.15)
2. Tool 5 Checklist bezittingen en schulden =visited 68 ==>Action 8 Maak een lijst van de bezittingen en schulden=visited 68 conf:(1) < lift:(5.07)> lev:(0.13) [54] conv:(54.59)
3. Task 4 Naar de advocaat =visited 97 ==>Task 5 Scheiding definitief maken=visited 69 conf:(0.71) < lift:(3.84)> lev:(0.12) [51] conv:(2.72)
4. Task 5 Scheiding definitief maken =visited 79 ==>Task 4 Naar de advocaat=visited 69 conf:(0.87) < lift:(3.84)> lev:(0.12) [51] conv:(5.55)
5. Task 1 Voorlopige afspraken maken =visited &Task 3 Woning, inboedel en financiën verdelen =visited 97 ==>Task 4 Naar de advocaat=visited 76 conf:(0.78) < lift:(3.44)> lev:(0.13) [53] conv:(3.41)
6. Task 4 Naar de advocaat =visited 97 ==>Task 1 Voorlopige afspraken maken=Task 3 Woning, inboedel en financiën verdelen =visited 76 conf:(0.78) < lift:(3.44)> lev:(0.13) [53] conv:(3.41)
7. Tool 7 Echtscheidings- en Ouderschapsplan =visited 92 ==>Action 6 Zelf ouderschapsplan maken=visited 67 conf:(0.73) < lift:(3.37)> lev:(0.11) [47] conv:(2.77)
8. Action 6 Zelf ouderschapsplan maken =visited 92 ==>Tool 7 Echtscheidings- en Ouderschapsplan=visited 67 conf:(0.73) < lift:(3.37)> lev:(0.11) [47] conv:(2.77)
9. Task 1 Voorlopige afspraken maken =visited &Action 1 Maak samen voorlopige afspraken =visited 93 ==>Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën=visited 69 conf:(0.74) < lift:(3.29)> lev:(0.11) [48] conv:(2.88)
10. Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 96 ==>Task 1 Voorlopige afspraken maken=Action 1 Maak samen voorlopige afspraken =visited 69 conf:(0.72) < lift:(3.29)> lev:(0.11) [48] conv:(2.68)

11. Task 3 Woning, inboedel en financiën verdelen =visited 117 ==>Task 1 Voorlopige afspraken maken=Task 4 Naar de advocaat =visited 76 conf:(0.65) < lift:(3.22)> lev:(0.12) [52] conv:(2.22)
12. Task 1 Voorlopige afspraken maken =visited &Task 4 Naar de advocaat =visited 86 ==>Task 3 Woning, inboedel en financiën verdelen=visited 76 conf:(0.88) < lift:(3.22)> lev:(0.12) [52] conv:(5.67)
13. Task 2 Afspraken maken over de kinderen =visited 97 ==>Action 6 Zelf ouderschapsplan maken=visited 67 conf:(0.69) < lift:(3.2)> lev:(0.11) [46] conv:(2.45)
14. Action 6 Zelf ouderschapsplan maken =visited 92 ==>Task 2 Afspraken maken over de kinderen=visited 67 conf:(0.73) < lift:(3.2)> lev:(0.11) [46] conv:(2.73)
15. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën=visited 96 conf:(0.72) < lift:(3.18)> lev:(0.15) [65] conv:(2.66)
16. Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 96 ==>Action 1 Maak samen voorlopige afspraken=visited 96 conf:(1) < lift:(3.18)> lev:(0.15) [65] conv:(65.8)
17. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Action 63 Voor u begint=Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 86 conf:(0.64) < lift:(3.18)> lev:(0.14) [58] conv:(2.18)
18. Action 63 Voor u begint =visited &Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 86 ==>Action 1 Maak samen voorlopige afspraken=visited 86 conf:(1) < lift:(3.18)> lev:(0.14) [58] conv:(58.95)
19. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Task 1 Voorlopige afspraken maken=Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 69 conf:(0.51) < lift:(3.18)> lev:(0.11) [47] conv:(1.7)
20. Task 1 Voorlopige afspraken maken =visited &Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 69 ==>Action 1 Maak samen voorlopige afspraken=visited 69 conf:(1) < lift:(3.18)> lev:(0.11) [47] conv:(47.3)
21. Task 3 Woning, inboedel en financiën verdelen =visited 117 ==>Task 4 Naar de advocaat=visited 84 conf:(0.72) < lift:(3.15)> lev:(0.13) [57] conv:(2.66)
22. Task 4 Naar de advocaat =visited 97 ==>Task 3 Woning, inboedel en financiën verdelen=visited 84 conf:(0.87) < lift:(3.15)> lev:(0.13) [57] conv:(5.03)
23. Action 63 Voor u begint =visited &Action 1 Maak samen voorlopige afspraken =visited 123 ==>Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën=visited 86 conf:(0.7) < lift:(3.1)> lev:(0.14) [58] conv:(2.51)
24. Tool 2 Checklist voorlopige afspraken woning, inboedel en financiën =visited 96 ==>Action 63 Voor u begint=Action 1 Maak samen voorlopige afspraken =visited 86 conf:(0.9) < lift:(3.1)> lev:(0.14) [58] conv:(6.21)
25. Task 3 Woning, inboedel en financiën verdelen =visited 117 ==>Action 63 Voor u begint=Task 4 Naar de advocaat =visited 68 conf:(0.58) < lift:(3.09)> lev:(0.11) [46] conv:(1.9)
26. Action 63 Voor u begint =visited &Task 4 Naar de advocaat =visited 80 ==>Task 3 Woning, inboedel en financiën verdelen=visited 68 conf:(0.85) < lift:(3.09)> lev:(0.11) [46] conv:(4.46)
27. Task 3 Woning, inboedel en financiën verdelen =visited 117 ==>Task 5 Scheiding definitief maken=visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
28. Task 5 Scheiding definitief maken =visited 79 ==>Task 3 Woning, inboedel en financiën verdelen=visited 67 conf:(0.85) < lift:(3.09)> lev:(0.11) [45] conv:(4.41)
29. Action 63 Voor u begint =visited &Task 3 Woning, inboedel en financiën verdelen =visited 98 ==>Task 4 Naar de advocaat=visited 68 conf:(0.69) < lift:(3.05)> lev:(0.11) [45] conv:(2.44)
30. Task 4 Naar de advocaat =visited 97 ==>Action 63 Voor u begint=Task 3 Woning, inboedel en financiën verdelen =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
31. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Action 63 Voor u begint=Action 21 Maak samen voorlopige afspraken over de kinderen =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)

32. Action 63 Voor u begint =visited &Action 21 Maak samen voorlopige afspraken over de kinderen =visited 81 ==>Action 1 Maak samen voorlopige afspraken=visited 74 conf:(0.91) < lift:(2.9)> lev:(0.11) [48] conv:(6.94)
33. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Action 21 Maak samen voorlopige afspraken over de kinderen=visited 81 conf:(0.6) < lift:(2.86)> lev:(0.12) [52] conv:(1.96)
34. Action 21 Maak samen voorlopige afspraken over de kinderen =visited 90 ==>Action 1 Maak samen voorlopige afspraken=visited 81 conf:(0.9) < lift:(2.86)> lev:(0.12) [52] conv:(6.17)
35. Action 63 Voor u begint =visited &Action 1 Maak samen voorlopige afspraken =visited 123 ==>Action 21 Maak samen voorlopige afspraken over de kinderen=visited 74 conf:(0.6) < lift:(2.85)> lev:(0.11) [48] conv:(1.94)
36. Action 21 Maak samen voorlopige afspraken over de kinderen =visited 90 ==>Action 63 Voor u begint=Action 1 Maak samen voorlopige afspraken =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
37. Task 3 Woning, inboedel en financiën verdelen =visited 117 ==>Action 8 Maak een lijst van de bezittingen en schulden=visited 65 conf:(0.56) < lift:(2.82)> lev:(0.1) [41] conv:(1.77)
38. Action 8 Maak een lijst van de bezittingen en schulden =visited 84 ==>Task 3 Woning, inboedel en financiën verdelen=visited 65 conf:(0.77) < lift:(2.82)> lev:(0.1) [41] conv:(3.05)
39. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Action 63 Voor u begint=Action 2 Mediation =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
40. Action 63 Voor u begint =visited &Action 2 Mediation =visited 80 ==>Action 1 Maak samen voorlopige afspraken=visited 69 conf:(0.86) < lift:(2.74)> lev:(0.1) [43] conv:(4.57)
41. Action 1 Maak samen voorlopige afspraken =visited 134 ==>Action 2 Mediation=visited 77 conf:(0.57) < lift:(2.69)> lev:(0.11) [48] conv:(1.82)
42. Action 2 Mediation =visited 91 ==>Action 1 Maak samen voorlopige afspraken=visited 77 conf:(0.85) < lift:(2.69)> lev:(0.11) [48] conv:(4.16)
43. Action 63 Voor u begint =visited &Action 1 Maak samen voorlopige afspraken =visited 123 ==>Action 2 Mediation=visited 69 conf:(0.56) < lift:(2.63)> lev:(0.1) [42] conv:(1.76)
44. Action 2 Mediation =visited 91 ==>Action 63 Voor u begint=Action 1 Maak samen voorlopige afspraken =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
45. Action 63 Voor u begint =visited &Task 1 Voorlopige afspraken maken =visited 127 ==>Task 2 Afspraken maken over de kinderen=visited 75 conf:(0.59) < lift:(2.59)> lev:(0.11) [46] conv:(1.85)
46. Task 2 Afspraken maken over de kinderen =visited 97 ==>Action 63 Voor u begint=Task 1 Voorlopige afspraken maken =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
47. Task 1 Voorlopige afspraken maken =visited 150 ==>Task 3 Woning, inboedel en financiën verdelen=Task 4 Naar de advocaat =visited 67 conf:(0.57) < lift:(3.09)> lev:(0.11) [45] conv:(1.87)
48. Task 3 Woning, inboedel en financiën verdelen =visited &Task 4 Naar de advocaat =visited 84 ==>Task 1 Voorlopige afspraken maken=visited 76 conf:(0.9) < lift:(2.57)> lev:(0.11) [46] conv:(6.05)
49. Task 3 Woning, inboedel en financiën verdelen =visited 117 ==>Task 2 Afspraken maken over de kinderen=visited 68 conf:(0.58) < lift:(2.55)> lev:(0.1) [41] conv:(1.81)
50. Task 2 Afspraken maken over de kinderen =visited 97 ==>Task 3 Woning, inboedel en financiën verdelen=visited 68 conf:(0.7) < lift:(2.55)> lev:(0.1) [41] conv:(2.35)

