NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Application of machine learning techniques for solving real world business problems

The case study – target marketing of insurance policies

Ineta Juozenaite

Project Work presented as partial requirement for obtaining the Master's degree in Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2017

Title: Application of machine learning techniques for solving real world business problems

Subtitle: The case study – target marketing of insurance policies

Student
Ineta Juozenaite

MGI

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

APPLICATION OF MACHINE LEARNING TECHNIQUES FOR SOLVING REAL WORLD BUSINESS PROBLEMS. THE CASE STUDY – TARGET MARKETING OF INSURANCE POLICIES

by

Ineta Juozenaite

Project Work presented as partial requirement for obtaining the Master's degree in Information Management, with a specialization in Knowledge Management and Business Intelligence

Advisor: Mauro Castelli

September 2017

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor Professor Mauro Castelli for his support of my Master project work.

Also, I would like to thank a lot to my best friend and soulmate Gerda Gruzauskaite for her support, patience and motivation during this period.

Moreover, I am thankful a lot to my family, who believed in me and gave me the opportunity to come to Lisbon to study Master of Information Management.

ABSTRACT

The concept of machine learning has been around for decades, but now it is becoming more and more popular not only in the business, but everywhere else as well. It is because of increased amount of data, cheaper data storage, more powerful and affordable computational processing. The complexity of business environment leads companies to use data-driven decision making to work more efficiently. The most common machine learning methods, like Logistic Regression, Decision Tree, Artificial Neural Network and Support Vector Machine, with their applications are reviewed in this work.

Insurance industry has one of the most competitive business environment and as a result, the use of machine learning techniques is growing in this industry. In this work, above mentioned machine learning methods are used to build predictive model for target marketing campaign of caravan insurance policies to achieve greater profitability. Information Gain and Chi-squared metrics, Regression Stepwise, R package “Boruta”, Spearman correlation analysis, distribution graphs by target variable, as well as basic statistics of all variables are used for feature selection. To solve this real-world business problem, the best final chosen predictive model is Multilayer Perceptron with backpropagation learning algorithm with 1 hidden layer and 12 hidden neurons.

KEYWORDS

Machine Learning; Logistic Regression; Decision Tree CART; Artificial Neural Network; Multilayer Perceptron; Backpropagation learning algorithm; Support Vector Machine; Kernel Gaussian Radial Basis Function; Target Marketing; Insurance Policies; Information Gain; Chi-squared metric; Regression Stepwise; Stratified Data Partitioning; Spearman Correlation.

INDEX

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ANN	Artificial Neural Network
CART	Classification and Regression Tree
CHAID	Chi-square Automatic Interaction Detector
DT	Decision Tree
LR	Logistic Regression
ML	Machine Learning
MLP	Multilayer Perceptron
PCA	Principal Component Analysis
RBF	Radial Basis Function Kernel
SVM	Support Vector Machine
AIC	Akaike Information Criterion

1. INTRODUCTION

Machine learning techniques can be defined as automated systems that are able to extract useful information from data or to make predictions based on existing, already collected data (Mohri, Rostamizadeh & Talwalkar, 2012). In 1959, one of machine learning pioneers, Arthur Samuel defined machine learning as “the field of study that gives computers (or machine) that ability to learn without being explicitly programmed” (Samuel, 1959). It means that machine learning algorithms can iteratively learn from data without being programmed to perform specific tasks. Machine learning models are able to adapt independently when new data are presented into the model. These models learn from prior computations to present reliable results and to make right decisions based on the produced results. ("Machine Learning: What it is and why it matters", 2017; Mathivanan & Rajesh, 2016)

Analytics thought leader Thomas H. Davenport (2013) states the importance of machine learning in organizations: “Humans can typically create one or two good models a week, machine learning can create thousands of models a week”. Machine learning goal is to give an increasing level of automation in getting knowledge and taking certain decisions by replacing human activity with automatic systems that can be more accurate and would save huge amount of human time than doing the same things by themselves (Mohri, Rostamizadeh, & Talwalkar, 2012).

Machine learning as a scientific discipline started in the late 1990s. In 1952 Arthur Samuel wrote the first learning computer program, the game of checkers, that was able to improve its performance with each played game. In 1958 Frank Rosenblatt introduced Perceptron, the first neural network algorithm. After this, in the following years, similar and more powerful neural network algorithms were created. The 1980s were the beginning of rule-based systems, also known as expert systems based on rules. In 1990s knowledge-driven logic started to be changing to data-driven logic in machine learning works. It was the beginning of the programs that were able to analyse big amount of data and make decisions based on analysis results. (Carbonell, Michalski & Mitchell, 1983; Parloff, 2016). Looking to the interest of machine learning discipline during the existence of this field, there were ups and downs, but with the enormous growth of computational power and amount of available data, machine learning field has been relaunched (Jordan & Mitchell, 2015; Parloff, 2016).

So even though the concept of machine learning has been around for decades, it is now becoming more and more popular not only in the business, but everywhere else as well. It is because of increased amount of data, cheaper data storage, more powerful and affordable computational processing. With all these things, it is possible automatically to build data-driven models in a short period of time, that will be able to analyse data and provide precise and reliable results. In this way, companies are more capable of identifying profitable opportunities or to avoid risks that would have bad impact on their business. Also, they are able to work more efficiently or even gain an advantage over competitors. ("Machine Learning: What it is and why it matters", 2017; Pyle & Jose, 2015). The complexity of business environment leads companies to use data-driven decision making. According to the research, companies by using data-driven decision techniques perform better than without (Bohanec, Kljajić Borštnar & Robnik-Šikonja, 2017). Also, companies in the top third of their industry, that use machine learning methods to make decisions, are approximately 5 percent more productive and 6 percent more profitable than their competitors that do not use these techniques to make decisions (Bohanec, Kljajić Borštnar & Robnik-Šikonja, 2017; Brynjolfsson, Hitt & Kim, 2011). So, all these improvements and research, that proved the benefits, have increased the potential of machine learning – and the need for business.

Machine learning techniques are used in variety of industries, education, medicine, chemistry, and a lot of other science fields. Machine learning approaches are applied to solve real-world business problems, like found in financial services, transportation, or in marketing and sales perspectives. For the banks and other financial businesses, machine learning is used to identify investment opportunities, clients with high-risk profiles, when to trade or prevent fraud. In transportation industry, like in delivery companies or public transportation, machine learning approaches are applied to find inherent patterns and trends for making routes more efficient and for predicting potential problems on the road. In the marketing and sales perspectives, machine learning methods are used for customer churn prediction, target marketing or it is also used to analyse buying behaviour of customers and make promotions based on analysis. (Linoff & Berry, 2011; "Machine Learning: What it is and why it matters", 2017)

The example of financial services can be insurance industry that has one of the most competitive business environment and because of this, the use of machine learning techniques is growing in this industry (Ansari & Riasi, 2016). Some of the most common examples, where machine learning can be used in insurance companies, are fraud detection, underwriting, claims processing and customer services, including marketing and sales of insurance policies. (Ansari & Riasi, 2016; Delgado-Gómez, Aguado, Lopez-Castroman, Santacruz & Artés-Rodriguez, 2011; Jost, 2016; Salcedo-Sanz, Fernández-Villacañas, Segovia-Vargas & Bousoño-Calzón, 2005; Umamaheswari & Janakiraman, 2014).

One specific example where machine learning approaches can be applied is target marketing of insurance policies (Goonetilleke & Caldera, 2013; Morik & Köpcke, 2004; Rahman, Arefin, Masud, Sultana & Rahman, 2017; Soeini & Rodpysh, 2012). This is real world business problem that is going to be solved in this work. The problem here is to find out to which customers to target. In other words, here machine learning techniques are used to build predictive model to identify customers with the highest probability to be interested in the certain type of insurance policy based on the past behaviour of customers and their characteristics. Having successful target marketing campaign of insurance policies, firstly, higher incomes would be brought to the company. Also, it would improve the relationship between company and its customers by suggesting services that customers are interested in and not bothering them with the offers that customers do not care about at all. Moreover, company would save time and money by contacting customers with high possibility to purchase the insurance policy rather than contacting random customers. Furthermore, one of the most important benefits in competitive insurance industry is to increase possibility for the company to be beyond its competitors, and machine learning would help to achieve it. (Coussement, Harrigan & Benoit, 2015; Kim & Street, 2004; Linoff & Berry, 2011; Perlich, Dalessandro, Raeder, Stitelman & Provost, 2013)

2. STUDY OBJECTIVES

The main objective of this work is to solve a real-world business problem by using machine learning approaches. In this work case, the desirable outcome is to build a model that would be able to predict if customers are interested in purchasing insurance policy or not, based on data about customers like usage product data and sociodemographic data. To build a predictive model for binary classification problem few different machine learning approaches are used. According to the performance results of the models, the best one is chosen for target marketing campaign.

To achieve the main objective, the following objectives are identified.

- To describe the chosen machine learning approaches that will be used in this work.
- To describe and prepare data for the chosen techniques.
- To apply chosen machine learning techniques, combine them, and try to improve the predictive model.
- To evaluate all built predictive models, describe the results, choose the best one.

3. MACHINE LEARNING METHODS

This section presents 4 of the most popular machine learning techniques for solving the real-world business problems. This section reveals specific business problems and approaches that are used to solve each of the problems. Logistic Regression (LR), Decision Tree (DT), Artificial Neural Network (ANN), Support Vector Machine (SVM) are well established, reliable and efficient methods. These 4 machine learning methods have been chosen to solve a real-world business problem of this work. Based on the literature review, these methods show great popularity and good performance in solving difficult business problems that are similar to the main problem of this work. Another reason of choosing above mentioned methods is their implementation in R, that is the software used in the predictive model building.

3.1. LOGISTIC REGRESSION

3.1.1. Overview of the Method

Logistic regression is a method of predictive analysis and it is used to solve classification problems. In logistic regression, the aim is to find a model that fits the regression curve, $y = f(x)$, the best. Here y is dependent categorical variable, while x represents independent variables that can be continuous, categorical or both at the same time. (Shalev-Shwartz & Ben-David, 2014)

3.1.2. Applications of Logistic Regression

Logistic regression is a popular and easily understandable classification technique that is often used for building prediction models to solve various business problems. For example, it can be used for customer churn prediction (Gürsoy, 2010; Neslin, Gupta, Kamakura, Lu & Mason, 2006; Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015), customer segmentation (McCarty & Hastak, 2007) as well as direct marketing (Coussement, Harrigan & Benoit, 2015; Zahavi & Levin, 1997). Gürsoy (2010) in his work shows that after proper data transformation, logistic regression can achieve good results. However, in most of the reviewed literature it did not perform well comparing to more advanced machine learning techniques (Coussement, Harrigan & Benoit, 2015; Delgado-Gómez, Aguado, Lopez-Castroman, Santacruz & Artés-Rodriguez, 2011; Neslin, Gupta, Kamakura, Lu & Mason, 2006; McCarty & Hastak, 2007; Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015). Sometimes real-world business problems can be too complex for logistic regression to solve it. Also, more advanced machine learning techniques are able to improve the performance by learning from data (Mitchell, 1997) and this makes statistical techniques like logistic regression to be less effective. For this reason, there is need to introduce more advanced and complex machine learning techniques.

3.2. DECISION TREE

3.2.1. Overview of the Method

Decision tree method is used for predictive modelling and it can build both classification and regression models. When target variable is continuous, regression trees are built. When target variable is categorical – classification trees. (Shalev-Shwartz & Ben-David, 2014)

The Figure 3.1 illustrates the structure of decision tree that is similar to the structure of the tree. DT method splits the dataset into homogenous sub-sets according to the best splitter of input variables. The splitting process stops when defined stopping criterion is reached. In the case when there is no stopping criterion, dataset is classified perfectly in the complete grown tree. The examples of the stopping criteria can be constraints of the tree size like defining the minimum number of observations in the nodes or maximum depth of the tree. Constraints of the tree size help to avoid over-fitting but this is the greedy approach. Another method to avoid over-fitting is tree pruning with two different techniques: pre-pruning and post-pruning. Pre-pruning stops splitting the tree before the dataset is classified perfectly. Post-pruning removes sub-nodes of perfectly classified dataset that do not bring high classification power. (Breiman, Friedman, Olshen & Stone, 1984; Shalev-Shwartz & Ben-David, 2014)

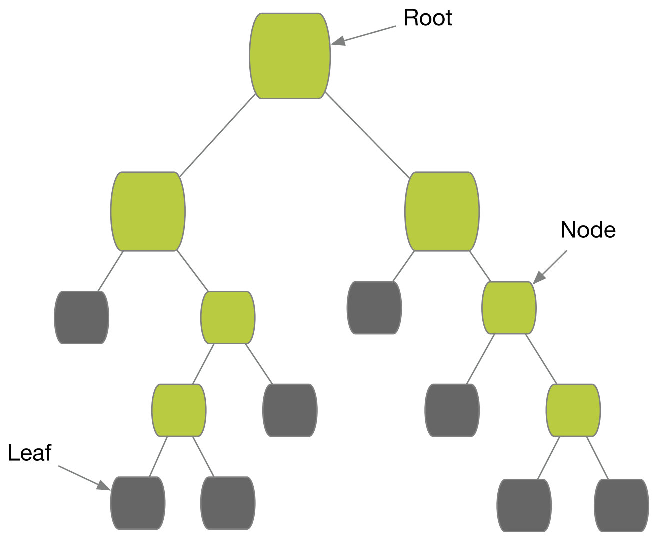


Figure 3.1 – Decision Tree structure

To split dataset and choose the best splitter, decision trees use various algorithms. One of them is CART (Classification and Regression Tree) algorithm that is suitable to use for binary classification problem, the same problem as there is in the empirical studies of this work. CART is designed to use for both classification and regression problems. This algorithm builds binary trees and uses Gini index in order to determine the best next split. CART algorithm uses post-pruning techniques to avoid over-fitting. In this empirical work, the cost-complexity pruning technique is used together with CART algorithm. (Breiman, Friedman, Olshen & Stone, 1984; Rokach & Maimon, 2014)

3.2.2. Applications of Decision Trees

Decision tree is a widely used technique for solving real-world business problems because of the simplicity of its use. Also, the obtained results from decision trees are easy to understand and interpret. The visuality of this method is comprehensible and helpful in understanding decision sequence. If decision trees are well constructed and organized, in other words they do not have too many leaves, then even non-professional users can perceive them easily. (Carbonell, Michalski & Mitchell, 1983; Coussement, Harrigan & Benoit, 2015)

This method can be used in customer segmentation (McCarty & Hastak, 2007). Decision trees are helpful to understand what characteristics customer has in each group, making easy the process of assigning a new customer to a specific group (McCarty & Hastak, 2007). Moreover, Goonetilleke and Caldera (2013), as well as Soeini and Rodpysh (2012) use decision trees for insurance customer attrition. To reduce the rate of attrition is substantial issue in insurance companies. Also, it can be used in prediction of salesman performance as it was done in the work by Delgado-Gómez, Aguado, Lopez-Castroman, Santacruz and Artés-Rodriguez (2011). Another example where decision trees can be applied is target marketing, defining if a customer belongs to the responder or the non-responder group. Coussement, Harrigan and Benoit (2015) in their work use 3 different decision tree algorithms (CHAID, CART, C4.5) to build customer response model for direct mail marketing. Comparing with all used methods in their work, CHAID and CART are in the group of methods that has the best performance, while C4.5 algorithm has quite poor performance. Furthermore, decision trees can be used for customer churn prediction. In the work that was done by Vafeiadis, Diamantaras, Sarigiannidis and Chatzisavvas (2015), decision tree C5.0 algorithm, that is persisted version of C4.5 algorithm, is applied to solve a problem of customer churning by using telecommunication data. As it was written above, C4.5 algorithm has not shown a good performance to solve direct mail targeting problem (Coussement, Harrigan & Benoit, 2015), while its followed-up version C5.0, that is used to solve customer churning problem (Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015), is one of the most effective method among other used techniques, excluding boosted machine learning techniques.

However, decision trees performance depends on the structure of the data. Decision trees algorithms have tendency to perform poorly if there are many complex and non-linear relationships between attributes (Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015). For this reason, the methods that can deal better with complex relationships between attributes are reviewed.

3.3. ARTIFICIAL NEURAL NETWORKS

3.3.1. Overview of the Method

Artificial neural network is computational model based on structure of very simplified human brain model. Like in human brains, neural network uses large number of computational units, also called neurons. Neurons are connected to each other, and also are able to communicate with each other in a complex network where complicated calculations are able to perform. (Shalev-Shwartz & Ben-David, 2014)

As it is shown in the Figure 3.2, artificial neural network is like directed graph, where nodes are neurons and edges are connection links between neurons. There are input neurons, hidden neurons, and output neurons, respectively they are from input layer, hidden layer and output layer. The strength of the connection between neurons is defined by weights. The input to each neuron is a weighted sum of the outputs it receives. The outputs are received by an activation function that transforms given inputs to outputs. The examples of the most popular activation functions are threshold, sigmoid or Gaussian. (Shalev-Shwartz & Ben-David, 2014)

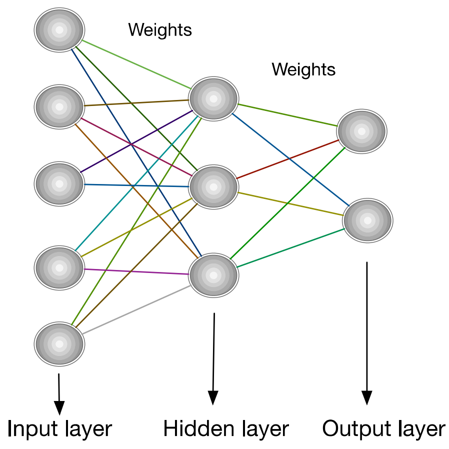


Figure 3.2 – Neural Network structure

In this work, Multilayer Perceptron (MLP) is chosen with backpropagation learning algorithm. One hidden layer is used, and the number of hidden neurons is changed from 1 to 20. The activation function of input and hidden layer is logistic function, as well as for the output layer since our work problem is binary classification. The initial weights of neural network are assigned randomly, and because of the existence of local optimum, MLP converges to the different results each time it is run. It explains some of the peaks in the training and test errors graphs. To be able to produce the same model again, it is important to remember to initialize random generator with the specific seed before building the model. All chosen specifications are based on the research of literature review that was done in this work.

Multilayer Perceptron is a feedforward model with the same structure as a single layer Perceptron, but with addition of one or more hidden layers. Backpropagation is mostly used learning algorithm for multilayer neural networks. Multilayer Perceptron with backpropagation algorithm has two steps: forward and backward. The first step is forward, where outputs are calculated propagating the inputs through activation function from input layer to output layer. In this step, the weights of all connections remain the same. In the second backward step, the weight of each connection is modified. For the output neurons, the weights are changed using Delta Rule, and for the hidden neurons the weights are changed by propagating backward the error of the output neurons to the hidden neurons. The error of the hidden neuron is calculated by summing the errors of all output neurons to which it is connected directly. (Shalev-Shwartz & Ben-David, 2014)

3.3.2. Applications of Artificial Neural Networks

Artificial Neural Network (ANN) shows high performance in prediction, forecasting, also in patterns recognition or identifying trends in the data (Tkáč & Verner, 2016). As a result of this, ANN is one of the most common machine learning approaches that is used on a wide range of real-world business problems (Coussement, Harrigan & Benoit, 2015; Tkáč & Verner, 2016). Tkáč and Verner (2016) in their work show artificial neural networks accomplishment in the business by identifying 412 articles in which ANNs were applied in various business disciplines areas. ANNs can deal with complex problems. They can use different topologies, that refer to the way the neurons can be connected. It brings an essential part in ANN functioning and learning (Miikkulainen, 2011; Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015). Moreover, the ability to use wide range of learning algorithms also has significant influence on ANN functioning and it makes ANN powerful machine learning approach (Miikkulainen, 2011; Tkáč & Verner, 2016).

One example of the complex problems for which ANN can be used is churn prediction. To build classifier for insurance policy holders who are about to terminate their policy, Goonetilleke and Caldera (2013) use neural network with Multilayer Perceptron structure and backpropagation learning algorithm, that performs better than another used machine learning method – Decision Tree C4.5. In addition, Vafeiadis, Diamantaras, Sarigiannidis and Chatzisavvas (2015) in their work use ANN approaches to build model for customer churning in telecommunication industry. These authors train Multilayer Perceptron model using backpropagation with one hidden layer and a varying number of hidden neurons. ANN with 15 hidden units shows the best performance going side by side with another machine learning method - Decision Tree 5.0 (Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015). However, corresponding to reviewed articles (Mozer, Wolniewicz, Grimes, Johnson & Kaushansky, 2000; Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015; Wai-Ho Au, Chan & Xin Yao, 2003) for customer churn prediction problem ANNs have better performance than logistic regression and decision trees.

Another example of ANNs in insurance companies is presented by Rahman, Arefin, Masud, Sultana and Rahman (2017). They use neural network with Multilayer Perceptron structure and backpropagation learning algorithm to predict the behaviour of future insurance policy owners by classifying them as regular or irregular premium payers. Insurance company, that is able to identify regular future payers, increases its profit significantly (Rahman, Arefin, Masud, Sultana & Rahman, 2017).

Furthermore, Coussement, Harrigan, and Benoit (2015) use ANNs for direct marketing to build response model and this is another complex business problem where ANN can be applied. In their work (Coussement, Harrigan & Benoit, 2015), Multilayer Perceptron model with one hidden layer is one of the best performing machine learning approaches but at the same time competing with decision trees CHAID and CART. For both described problems ANN with one hidden layer is chosen, even though as many hidden layers as necessary can be used in building ANN model. Most of the problems can be solved using one or two layers, even though the problem is quite complex, one-two hidden layers are powerful enough to approximate any function (Bishop, 1995; Coussement, Harrigan & Benoit, 2015).

In addition, another example of ANN application is from transportation industry. Most of the transportation companies want to increase the number of customers that use their services. To achieve this goal travel agencies need to improve reliability of bus travel times. To predict bus travel speeds and give updated information to clients using real time information is a quite complex business problem. Julio, Giesen and Lizana (2016) use Bayesian Regulation backpropagation neural network with two hidden layers to predict bus travel speed and travel time. The same authors also show that one-two hidden layers are enough to represent non-linearity of the variables. Julio, Giesen, and Lizana (2016) in their work mention the problem of ANN algorithms, that they tend to fall into local optimum instead of global one while training. Because of this, these authors introduce other machine learning approach with stronger mathematical background - support vector machine, the one that always finds global optimum while training. Even though it does not mean that support vector machine will outperform artificial neural network as it happens in the work of Julio, Giesen, and Lizana (2016), where artificial neural network outperforms support vector machine and gives the best results. As a result of this, it is still necessary to review support vector machine methods and to see their applications.

3.4. Support Vector Machines

3.4.1. Overview of the Method

Support Vector Machine (SVM) is more refined and more recent method of machine learning. As it was written before, this method achieves a global optimum in the training phase and it is the advantage of SVM against ANN (Julio, Giesen & Lizana, 2016). However, it does not mean that SVM will give the better results than ANN (Julio, Giesen & Lizana, 2016).

Support vector machine can be used for both classification and regression problems. The main specification of SVM is the usage of Kernel function, and another one is that local optimum does not become global optimum in SVM algorithms (Shalev-Shwartz & Ben-David, 2014). In empirical studies of this work, SVM, that is suitable for classification problem, is used and described below.

To separate two classes SVM finds the optimal hyperplane between them by maximizing the margin between closest points of classes, that are also called support vectors. The optimal hyperplane is the line in the middle of the margin (Shalev-Shwartz & Ben-David, 2014). The Figure 3.3 provides the visual explanation of SVM. Having hyperplane that separates data records into non-overlapping classes can cause over-fitting. To avoid this, SVM finds the optimal hyperplane not only by maximizing the margin, but also by minimizing the misclassification. (Shalev-Shwartz & Ben-David, 2014)

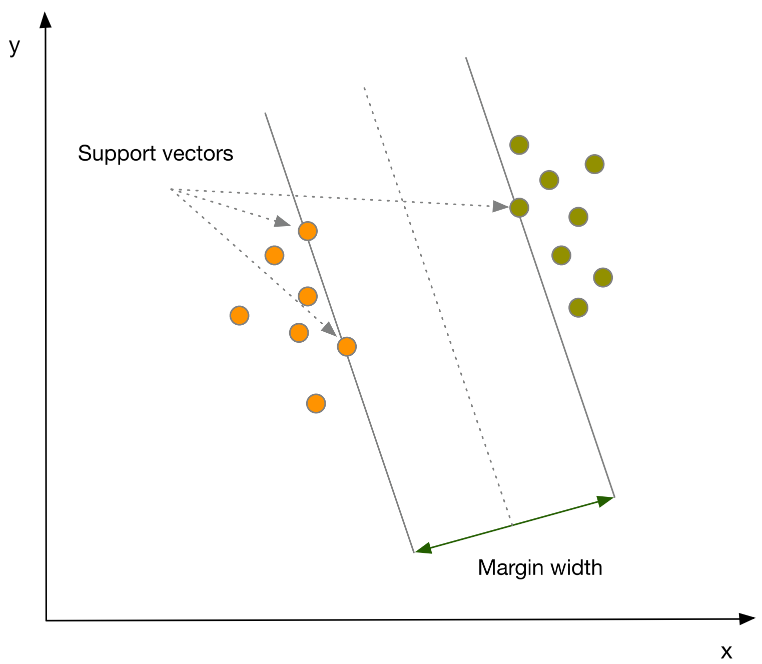


Figure 3.3 – Visual explanation of Support Vector Machine

For difficult real-world business problems data is not linearly separable. SVM manages this by projecting data points into other dimensional space, usually higher, where the same data points would be linearly separable. To map data in different space, SVM uses non-linear functions. In this work, Kernel Gaussian Radial Basis (Kernel RBF) function is chosen based on the literature review, and because it is general function for classification problems with a good performance. Another good factor, there are only two parameters that must be defined, C and γ. C is the cost of misclassification. For SVM a high value of parameter C lets to misclassify as less training cases as possible and in this way prediction function becomes complex, while for the low value of parameter C is opposite. Parameter γ defines which training examples are considered as support vectors by the model. If value of the parameter γ is high, only the closest training examples, and vice versa for low value γ. (Karatzoglou, Meyer & Hornik, 2006; Shalev-Shwartz & Ben-David, 2014)

3.4.2. Applications of Support Vector Machines

SVM can be used in customer churn prediction. Vafeiadis, Diamantaras, Sarigiannidis and Chatzisavvas (2015) that use logistic regression, decision trees, artificial neural networks, also use support vector machines for customer churning prediction in telecommunication industry. Gaussian radial basis Kernel function and the polynomial Kernel function are chosen in this work. None of them are in the best performing machine learning approaches that are used, but close enough to the decision tree C5.0 (Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015). For customer churn prediction problem SVM usually gives the better results than DT, sometimes it performs better than ANN. Results mainly depend on the type of data and relationship between variables (Kerdprasop, Kongchai & Kerdprasop, 2013; Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015). Another example of SVM application can be performance of the salesman prediction. It can give huge advantage to the company by selecting the right candidates in the recruitment process and reduce the expenses of the companies. The example of this problem can be the work by authors Delgado-Gómez, Aguado, Lopez-Castroman, Santacruz and Artés-Rodriguez (2011). In their work, SVM are the main approach to solve the business problem. SVM is used with Gaussian Kernel function and have better performance compared to other used methods like decision trees or discriminant analysis.

4. EVALUATION TECHNIQUES

To evaluate models and to select the one with the best performance, evaluation techniques of models like accuracy and F-measure are used in empirical studies of this work. Both measures are calculated based on confusion matrix (see the Table 4.1), which shows all true positive (TP), false negative (FN), false positive (FP) and true negative (TN) values of classifier. (Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015)

	Predicted: NO	**Predicted: YES**
Actual: NO	True Negative (TN)	False Positive (FP)
Actual: YES	False Negative (FN)	True Positive (TP)

Table 4.1 – Confusion Matrix

In the above table, the confusion matrix for binary classifications (no and yes) is presented. True Positives (TP) identify the cases where events (yes cases) are classified correctly, while True Negatives (TN) - correctly predicted non-events (no cases). False Positives (FP) show wrongly classified events, it means the events are predicted as no cases, but actually they are yes cases. False Negatives (FN) show wrongly identified non-events (no cases), more clearly, no cases are identified as yes cases.

Accuracy is the proportion of the total number of predictions that were correct. The formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

F-measure is a combination of sensitivity and positive predictive value, that also can be called precision. The formula:

$$F\text{-}measure = \frac{2 \times Precision \times Sensitivity}{Precision + Sensitivity}$$

The sensitivity measures the proportion of actual positives that are correctly identified. The formula:

$$Sensitivity = \frac{TP}{TP + FN}$$

The precision is the proportion of positives that were correctly identified. The formula:

$$Precision = \frac{TP}{TP + FP}$$

These two measures, the sensitivity and the precision, are not able to evaluate the performance of the model alone and because of this, the combination of these two measures, F-measure, is used. When the value of F-measure is closer to 1, the better the classifier is. The F-measure is getting closer to 1 when the Precision and Sensitivity are increasing and getting closer to 1 as well. (Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015)

5. PREDICTIVE MODELLING FOR DIRECT MARKETING IN INSURANCE SECTOR

All data analysis and model building is done with R programming language. R is a free software that compiles on a different UNIX platforms and systems. R has a wide range of various statistical and graphical techniques, and it is able to be extended to the other much difficult level like creating your own functions or procedures ("R: The R Project for Statistical Computing", n.d.). The R code of this work data analysis can be found in Appendix 8.8.

5.1. DATA SOURCE

The data source that is used in this work for empirical studies was provided by the Dutch data mining company Sentient Machine Research and is based on a real-world business problem (“Sentient Machine Research”, 2000). This data source contains information about customers of insurance company, and also the results of already performed marketing campaign “Caravan Insurance Policy” that tells if customers were interested in this insurance policy or not.

Dataset to train and test predictive models consists of 5822 customers. The customers are described by 86 variables, including sociodemographic data and product ownership data. Also, including target variable that gives information if customer purchased caravan insurance policy or not. The sociodemographic data is obtained from customers’ zip codes. It means that customers with the same zip code are characterized with the same sociodemographic features. The validation dataset, that helps to build better predictive model, is not used because of the small size of sample (Maimon & Rokach, 2005). Dataset for predictions consist of 4000 customers with the same attributes excluding target variable which is supposed to be returned by chosen model.

In the Appendix 8.1, there are two tables with all variables and their explanations.

5.2. DATA EXPLORATION AND PRE-PROCESSING

Before applying techniques that are going to be used for the empirical studies, data has to be explored and pre-processed. It means that data should be analysed in order to get familiar with it and also to recognise possible data pre-processing techniques. Pre-processing techniques are cleaning techniques that detect and remove errors and inconsistencies from data, like outliers that could cause misinterpretation of analysis (Han, Kamber, & Pei, 2012).

Exploring caravan policies data, any data inconsistency has not been detected. There are no any missing or strange values. All variables contain values that are grouped (see the Appendix 8.1). For example, variable *MGEMLEEF* presents average age groups, variable *PPLEZIER* presents contribution to boat policies where amount of contribution is presented in the groups; if the value of this variable is 4, it means customer contributed between 200 and 499 money. The results of the data in this format leads to clean and consistent data that does not have strange values or significant outliers that should be treated.

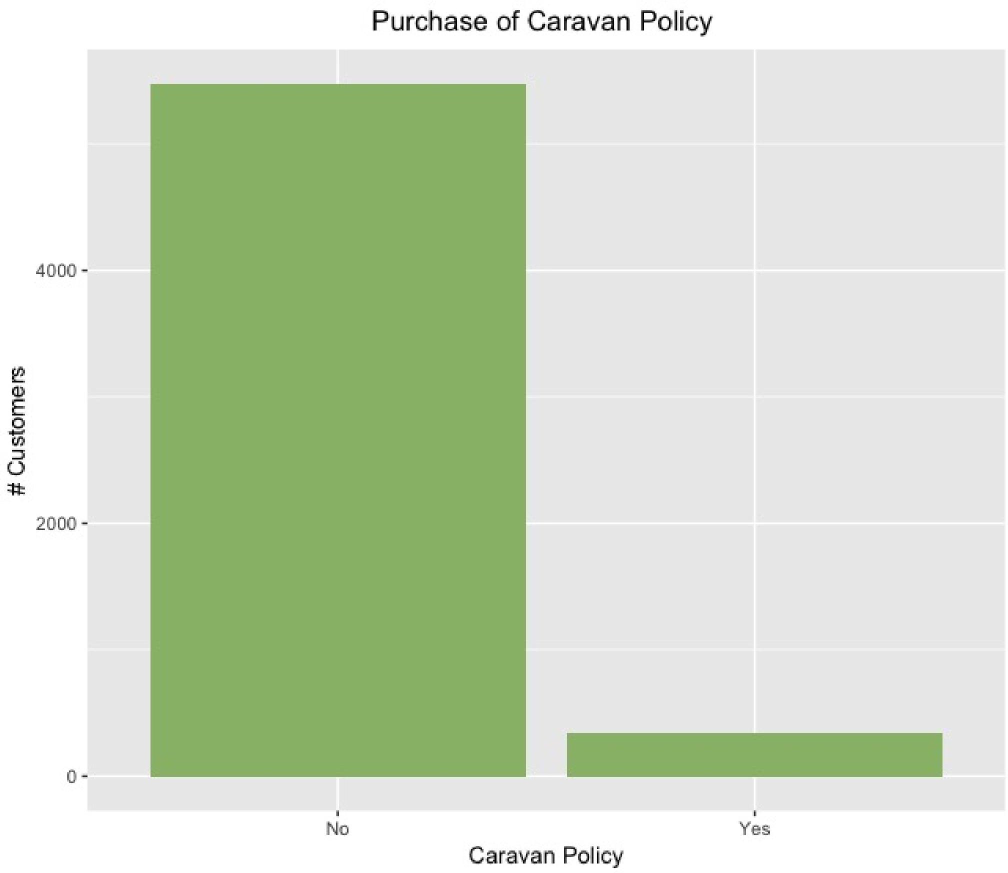


Figure 5.1 – Number of caravan insurance holders

The above Figure 5.1 shows distribution of target variable - caravan policy owners. It reveals that the data is very imbalanced. Only 6% of customers purchased caravan policy. It is not surprising that data is imbalanced because usually solving direct marketing problems there are more non-responders than responders (Ling & Li, 1998; Pan & Tang, 2014).

Other very important step is to analyse variables from the data source, because probably not all of them are useful, or some of them presents the same information, or even some of them might not bring any value to the predictive model. For this purpose, the table that helps to analyse variables by each level of each independent variable is created (see the Appendix 8.2). This table shows the number of customers that belongs to each group (level of each variable). Also, the percentage and standard deviation of customers that purchased caravan insurance policy in that group. The table reveals that the customers with the main type “Driven Growers” have 13.15 % of caravan insurance holders. It is a higher percentage than average percentage of customers that purchased caravan insurance policy. Moreover, the variable *MBERHOOG* reveals that the customers who live in the neighbourhood, where more than 37% - 49% people are with a high status, have a higher than average percentage of caravan insurance holders as well. Furthermore, variables *MAUT1, MAUT2, MAUT0* show the tendency of customers, who have at least one car, to purchase the caravan insurance policy. In addition, variables *MHHUUR* and *MHKOOP* reveals a small raise of percentage of caravan insurance holders when the number of home owners in customer’s neighbourhood is increasing. This table reveals that kind of information about every variable and it will be useful in the variables selection stage.

To present information visually, the histograms of each variable by dependent variable are created as well. The histograms of all variables can be found in the Appendix 8.3. However, since the percentage of customers with caravan insurance policy is very low, it is hard to see from the histograms which variables give clearly visible distribution between purchasing insurance policy and which ones not. Having a quick look to histograms, it has not escaped the notice that there are not many customers in the 20 - 30 age range, while customers in the 40 - 50 age group tend to purchase caravan insurance policy. Also, the histogram of variable *PPERSAUT* shows that customers who spent averagely 1000 – 4999 money for car policies have a tendency to hold a caravan insurance policy. In addition, customers who have one car insurance policy are more likely to get caravan insurance policy (variable *APERSAUT*). The histograms of these 3 variables are shown in the Figure 5.2.

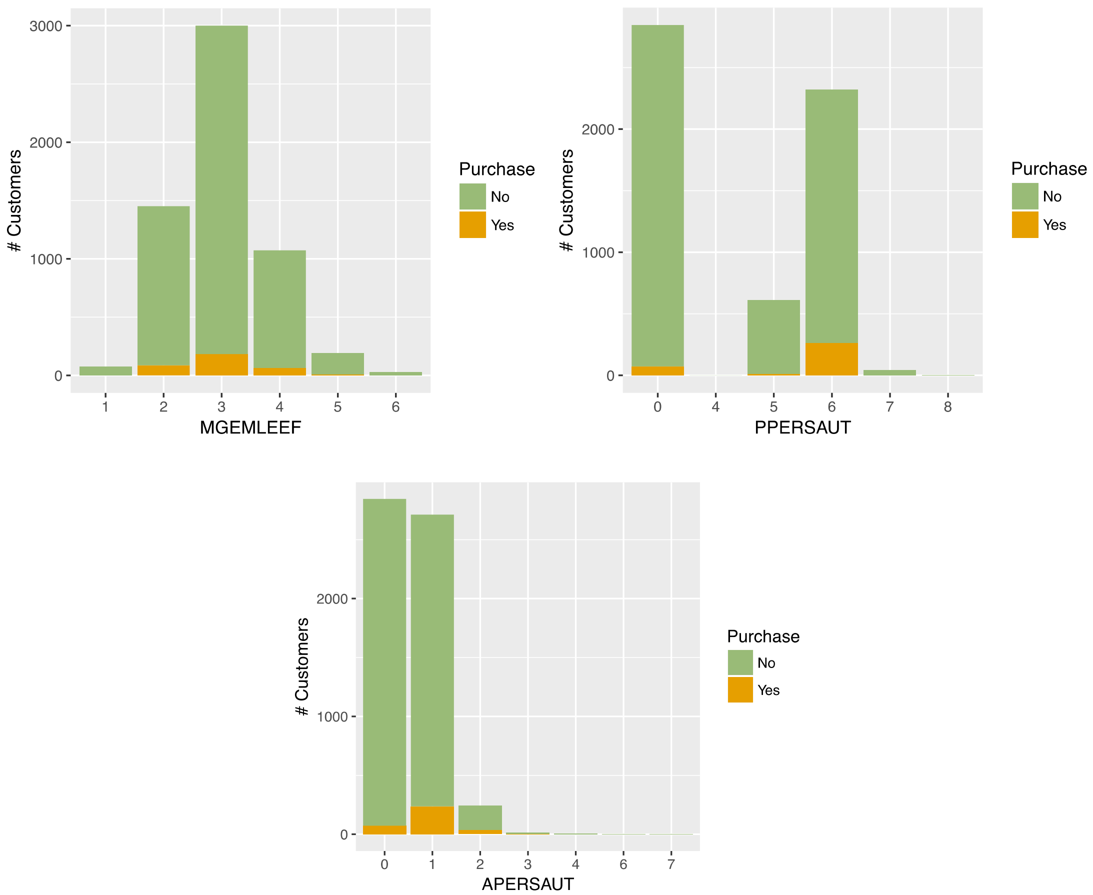


Figure 5.2 - Histograms of age, contribution to car policies and number of car policies variables by dependent variable

In exploration phase, Spearman correlation analysis between variables has been performed. Spearman correlation method is appropriate to use when the variables are not normally distributed or the relationship between the variables is not linear. These reasons determine to use Spearman rank correlation method.

In the Appendix 8.4, there is a table with Spearman correlations between dependent variable and all independent variables. From this table, it is clear that there is any significantly correlated variable with the purchase of caravan insurance policy. The variable *PPERSAUT*, that shows contribution of car policies, is correlated to caravan insurance holders the most (0,164), and the least correlated variables with target variable are the contribution of third party insurance firms (*PWABEDR*) and the purchased number of the same insurance (*AWABEDR*). The correlation is less than 0,001.

Also, the Spearman correlation matrix between all independent variables has been calculated and can be found in the Appendix 8.5. Since there are 85 variables, the Spearman correlation matrix of independent variables is visualized in the Figure 5.3 to have a quick view in it, too. In this correlation matrix plot, it is easy to see which variables are highly correlated and which ones not. In this graphical presentation of Spearman correlation matrix, the intensity of correlation is presented by the colour and the ellipse form. The stronger the correlation, the more narrowed ellipse. If the ellipse is shown from the left to the right, the correlation is positive, and vice versa. Also, the colour identifies the sign of the correlation. The blue shows positive and the red – negative. It is useful to look at correlation matrix for variables transformations, because the highly correlated variables might be beneficial to merge into one variable. Sometimes data source might contain similar variables that give the same information, so before eliminating any variable it is good to see correlations between them. Having highly correlated variables in the predictive model might make the predictive model unstable.

There are 4 variables about marital status: *MRELGE* (Married), *MRELSA* (Living together), *MRELOV* (Other relation), *MFALLEN* (Singles). These variables are quite correlated among each other, so it has been decided to leave only one variable that shows the percentage of single people in the customer's neighbourhood. In this way, the percentage of single or not single people is clear. *MFGEKIND* (Household without children) and *MFWEKIND* (Household with children) variables show the same information, how many households there are with children and without, and also these variables are quite correlated. Household with children variable has higher correlation with target variable, so it has not been removed from the variable set. Also, dataset contains two variables *MHHUUR* (rented house) and *MHKOOP* (home owners) that present the same information and have a high correlation between each other (0,999), so *MHKOOP* variable has been eliminated from the dataset. There is the same situation with the variables *MZFONDS* (national health service) and *MZPART* (private health insurance), that also have a correlation close to 1 between each other, so only one variable is left, *MZFONDS*. Moreover, there are 3 variables that shows information about the number of having cars. Since these variables are also quite correlated and *MAUT2* does not have a high correlation with target variable, it is enough to know if there is at least one car or not, so it is decided to leave only one variable *MAUT0* to have that information.

Furthermore, there are several groups of variables in which any highly correlated variables have not been detected so that it would be possible to eliminate some of them. One of that groups of variables shows the customer's status (high status, entrepreneur, farmer, middle management, skilled labours, unskilled labours), the second group of variables presents social class (social class A, B1, B2, C, D), and the third one shows the average income. In addition, it is worth to mention that the dataset contains some kind of similar variables with a high correlation between each other that have not been eliminated. At this stage, it is still not clear which variables having in the variables combination would bring a higher value to the predictive model. The example of these variables could be the main type of customer and the subtype of customer.

Moreover, looking to the correlation matrix graph (see the Figure 5.3), it is clearly visible that two types of product ownership variables, the contribution and the number of purchased insurance policies, are very highly correlated. For example, the variable that shows contribution of car policies and the variable that shows the number of car policies have correlation equal to 1, other correlations of these variables combinations are also very close to 1. This reason motivated to create another variable, that would include these two types of variables. The most reasonable combination of these two types of variables is multiplication that would show the interaction of the variables. So, the primary product ownership variables have been eliminated and the new combinations of these variables are going to be used in the predictive model. Since all variables contain grouped values, it is hard to think about more variables transformations that would make sense for the predictive modelling.

To sum up, after eliminating and creating new variables, the final dataset contains 46 variables, from which 31 variables presents sociodemographic data, 14 variables presents product ownership data and also the target variable is included in the final dataset.

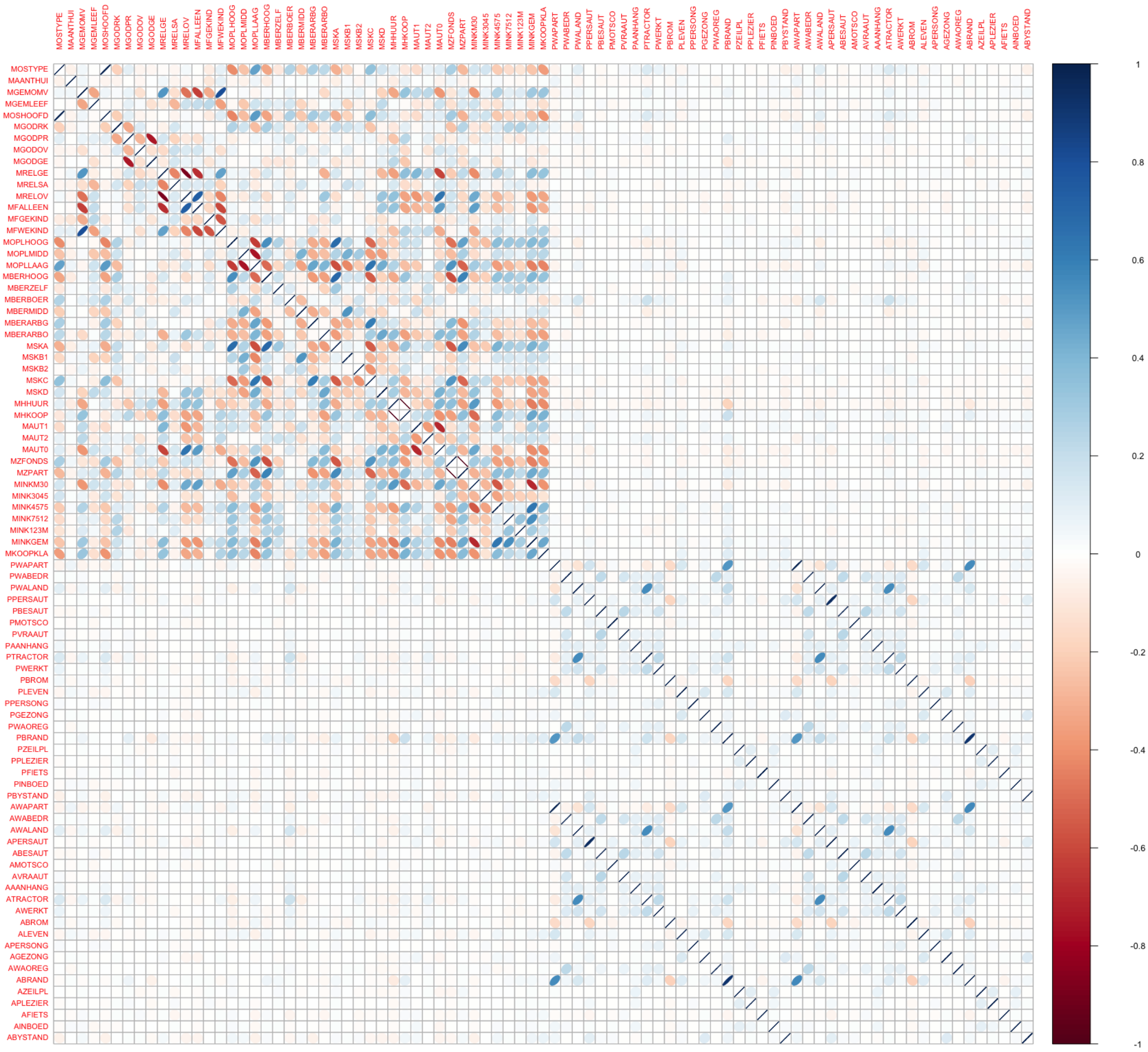


Figure 5.3 - Graphical presentation of Spearman correlation matrix

5.3. Variable Selection

Based on variable selection techniques and also already gained knowledge from data exploration phase, the variables subsets are created, so that they could be tested to detect which of them brings the highest value to the predictive model.

5.3.1. Information Gain and Chi-squared

First of all, Information Gain and Chi-squared metrics are calculated to see the feature importance for each variable. The Figure 5.4 illustrates these metrics in the bar charts. The graphs show only 14 variables because for other variables the metrics are equal to zero. As it was clarified earlier in this work, these graphs also display that there is any variable that would explain well caravan insurance policy owners. These graphs give the idea of importance of each separate variable to prediction. However, building the prediction model the combination of the variables, that gives a high value to predictive model, is necessary. On the other hand, these metrics reveal the variables that does not have any worth to prediction and the variables that might be eliminated from further analysis. As it can be seen from Information Gain and Chi-squared bar charts, the transformed variable (*IBRAND*) from the number of fire policies and contribution to fire policies has the highest importance to prediction. Based on Information Gain, the second variable with the highest importance is the average income (*MINKGEM*) while looking to Chi-squared metric it is the interaction variable of boat policies. Both of these variables look more meaningful than the first one – the fire policies variable. It is quite hard to explain why the fire policies has the highest impact to prediction, but it is essential to emphasize that the worth to prediction of each variable is very low.

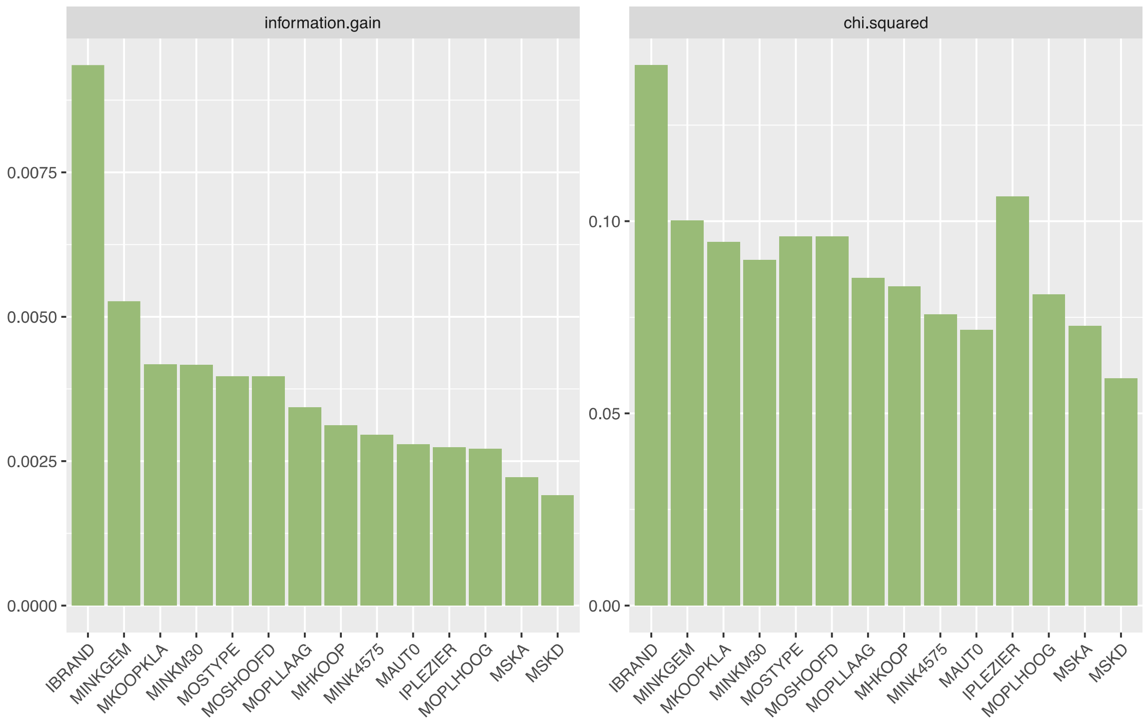


Figure 5.4 – Information Gain and Chi-squared bar charts

5.3.2. Stepwise Logistic Regression

The second method, that is used for variable selection, is stepwise logistic regression. This method selects the model by AIC criterion using forward stepwise regression, backward stepwise regression or combination of both, subtraction and addition of variables to the model. The lowest AIC value gives the model with the best variables combination.

This method showed that the best variable combination is from 17 variables: *MGEMLEEF* (Average age), *MOPLMIDD* (Medium level education), *MOPLLAAG* (Lower level education), *MBERBOER* (Farmer), *MBERMIDD* (Middle management), *MSKC* (Social class C), *MHKOOP* (home owners), *MAUT0* (No car), *MINK123M* (Income >123.000), *MINKGEM* (Average income), *IWERKT* (Interaction of agricultural machines policies), *IBROM* (Interaction of moped policies), *IWAOREG* (Interaction of disability insurance policies), *IBRAND* (Interaction of fire policies), *IPLEZIER* (Interaction of boat policies), *IFIETS* (Interaction of bicycle policies), *IBYSTAND* (Interaction of social security insurance policies).

5.3.3. R Package "Boruta"

The third method to select variables is used from R package "Boruta". This variable selection algorithm is based on Random Forest. It eliminates variables that do not perform well recursively at each iteration. At the end, Boruta algorithm presents all variables, either they are relevant to the model or not, by showing if the variable has been rejected or not. So, Boruta shows even the variables that give small relevance to the prediction. ("Wrapper Algorithm for All Relevant Feature Selection [R package Boruta version 5.2.0]", 2017)

The Boruta algorithm confirmed 34 variables. The boxplots graph that shows which variables gained the highest importance, based on Boruta algorithm, is presented in the Figure 5.5. Red boxplots display *Z* score of rejected variables, yellow – tentative variables (it means that decision to reject or confirm the variable still has to be made), and green – confirmed variables. The 3 top most important variables are interaction variable of boat policies (*IPLEZIER*), customer subtype (*MOSTYPE*) and lower level education (*MOPLLAAG*). Also, even though Boruta algorithm rejected many product ownership variables, it is important to note that only these variables present exact information about customers, because sociodemographic variables provide information from the neighbourhood the customer is living in. Due to this fact, creating the variables subsets, the product ownership variables should not be eliminated drastically.

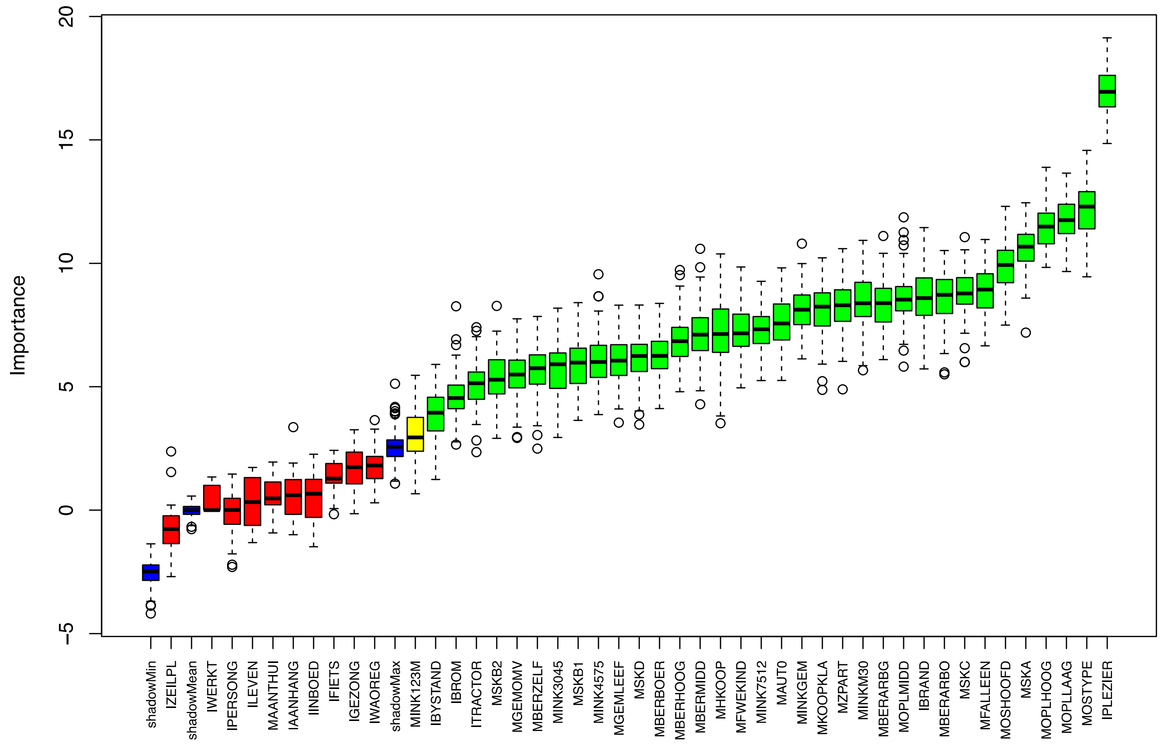


Figure 5.5 – Feature selection by “Boruta” algorithm

5.3.4. Variables Combinations

Based on the results obtained from above described methods, also gained insights from Spearman correlation analysis and distribution graphs, 7 variables subsets have been created to test on all machine learning models in order to choose the best variables subset:

Variables sets:

1. All sociodemographic variables that have not been eliminated in exploring variables stage as well as all transformed product ownership variables: *MOSTYPE, MAANTHUI, MGEMOMV, MGEMLEEF, MOSHOOFD, MFALLEEN, MFWEKIND, MOPLHOOG, MOPLMIDD, MOPLLAAG, MBERHOOG, MBERZELF, MBERBOER, MBERMIDD, MBERARBG, MBERARBO, MSKA, MSKB1, MSKB2, MSKC, MSKD, MHKOOP, MAUT0, MZPART, MINKM30, MINK3045, MINK4575, MINK7512, MINK123M, MINKGEM, MKOOPKLA, IAANHANG, ITRACTOR, IWERKT, IBROM, ILEVEN, IPERSONG, IGEZONG, IWAOREG, IBRAND, IZEILPL, IPLEZIER, IFIETS, IINBOED, IBYSTAND*. Total of 46 variables.
2. Variables, that did not have zero importance to prediction, obtained from Information Gain and Chi-squared metrics: *IBRAND, MINKGEM, MKOOPKLA,*

MINKM30, MOSTYPE, MOSHOOFD, MOPLLAAG, MHKOOP, MINK4575, MAUT0, IPLEZIER, MOPLHOOG, MSKA, MSKD. Total of 14 variables.

3. Variables selected from stepwise regression method: *MGEMLEEF, MOPLMIDD, MOPLLAAG, MBERBOER, MBERMIDD, MSKC, MHKOOP, MAUT0, MINK123M, MINKGEM, IWERKT, IBROM, IWAOREG, IBRAND, IPLEZIER, IFIETS, IBYSTAND*. Total of 17 variables.
4. Variables that have been confirmed by Boruta algorithm from R package: *IPLEZIER, MOSTYPE, MOPLLAAG, MOPLHOOG, MSKA, MOSHOOFD, MFALLEEN, MSKC, IBRAND, MBERARBO, MOPLMIDD, MINKM30, MBERARBG, MZPART, MINKGEM, MKOOPKLA, MAUT0, MINK7512, MFWEKIND, MHKOOP, MBERMIDD, MBERHOOG, MBERBOER, MSKD, MGEMLEEF, MINK4575, MSKB1, MINK3045, MBERZELF, MGEMOMV, MSKB2, ITRACTOR, IBROM, IBYSTAND*. Total of 34 variables.
5. The total variables mixture based on all performed variable selection methods: *IPLEZIER, MOSTYPE, MOPLLAAG, IBRAND, MBERARBO, MZPART, MINKGEM, MKOOPKLA, MAUT0, MHKOOP, MBERMIDD, MGEMLEEF, ITRACTOR, IBROM, IBYSTAND, MINK123M, IZEILPL*. Total of 17 variables.
6. Variables mixture that includes all transformed product ownership variables and mixture of sociodemographic variables, that have been selected based on all performed variable selection methods: *MOSTYPE, MOPLLAAG, MBERARBO, MZPART, MINKGEM, MKOOPKLA, MAUT0, MHKOOP, MBERMIDD, MGEMLEEF, MINK123M, IPLEZIER, IBRAND, ITRACTOR, IBROM, IBYSTAND, IWAOREG, IGEZONG, IFIETS, IAANHANG, IINBOED, ILEVEN, IWERKT, IPERSONG, IZEILPL*. Total of 25 variables.
7. Variables mixture that has been built by testing various combinations of the variables. It includes all transformed product ownership variables, and also mixture of variables that showed the highest importance to prediction in variable selection methods: *MKOOPKLA, MINKGEM, MOPLLAAG, MHKOOP, MAUT0, IPLEZIER, IBRAND, ITRACTOR, IBROM, IBYSTAND, IWAOREG, IGEZONG, IFIETS, IAANHANG, IINBOED, ILEVEN, IWERKT, IPERSONG, IZEILPL*. Total of 19 variables.

5.4. DATA PARTITIONING

The objective of data partitioning is to avoid under-fitting or over-fitting of the model. By avoiding this, the computational models have a high accuracy in prediction and are well generalized (Mitchell, 1997). Usually, dataset is divided into three parts: training, validation, and testing. Training dataset is used to estimate model parameters. The validation dataset is used to validate the performance of the model built by using training dataset. The third test dataset is used to estimate the accuracy of the model on unseen data that gives realistic performance of the model. (Reitermanov, 2010)

Following practical recommendations, in this work data is going to be divided into 2 datasets – 70% training and 30% test. There is not going to be the validation dataset because of the small size of the sample. (Maimon & Rokach, 2005)

The chosen partition method is stratified sampling. The method explores the structure of the dataset and divides it into homogenous sets. Stratified sampling assures that all three datasets, training, validation and test, are well balanced (Reitermanov, 2010). Usually solving direct marketing problems, the data is imbalanced because there are more non-responders than responders. This is the main reason why stratified method is chosen. (Ling & Li, 1998; Pan & Tang, 2014)

5.5. BUILDING PREDICTIVE MODEL

Before starting to build model, the number of customers that will be selected in the test and predictive datasets should be defined. The boundary probability that would help to select potential customers should be defined. The customers with a higher probability of purchasing caravan insurance policy than the boundary probability would be selected. The boundary probability depends on the costs of contacting potential caravan insurance customers and the profit of selling insurance policies. Without knowing it, it has been decided to select 20% of the customers with the highest probability to purchase the caravan insurance policy.

After splitting dataset to training and test datasets, the test dataset has 1746 customers with 104 caravan insurance policy owners. The numbers of customers of each dataset are displayed in the Figure 5.1.

	# Holders of Caravan Policy	# Non-Holders of Caravan Policy	# All Customers
Training Dataset	244	3832	4076
Test Dataset	104	1642	1746

Table 5.1 – Training and Test datasets

By selecting 20% customers of the test dataset, 349 customers with the highest probability of having insurance policy are going to be selected to test all models. Selecting 349 customers randomly from the test dataset, approximately 21 caravan insurance policy holders should appear there. This number is going to be as the base line for the obtained results from the predictive models. If the model gives the number of correctly identified policy holders close to this number, it means the model performance is really poor and more analysis should be done to get better results.

5.5.1. Choosing the Final Variables Combination

The first step is to run all variables subsets on all chosen machine learning techniques and based on obtained results and already having insights of data, choose the final variables subset that will be used for predictive modelling. In the Appendix 8.6, there is a table with the evaluation metrics (Identified True Positives, Sensitivity, Precision and F-measure), that are described in the 4th section, for all possible combinations of chosen variables subsets and machine learning techniques. The Table 5.2 presents the average of identified true positives of all used models for each variables subset, and also the maximum of identified true positives. In this table, it is encouraging to see that for all variables subsets the average of identified true positives is higher than random guess. Also, keeping in mind that some of the used machine learning methods did not perform very well decreased the average. The detailed table is in the Appendix 8.6.

Variable Subset	**Average of Identified True Positives**	**Max of Identified True Positives**
1 - All socio-demographic & policies interactions	39,33	56
2 - Information Gain & Chi-Squared	40,63	56
3 - Stepwise regression	37,60	53
4 - Boruta	36,43	49
5 - Total Mix	38,45	54
6 - All policies interactions & Socio-demographic mix	37,28	52
7 - All policies interactions & Socio-demographic mix	38,23	54

Table 5.2 – Results of each variables subset

The variables subset that has been created based on Information Gain and Chi-squared metrics has the highest average of correctly identified caravan policy holders. The variables subset of all not removed variables has very similar results, as well as the fifth and the seventh variables subsets. In the first and the second variables subsets there are quite many variables that are highly correlated with each other and it might make the predictive model unstable. While the fifth subset contains also some highly correlated variables but less than in the seventh subset, and also considering that the seventh subset has less variables than the fifth, it is decided to choose the seventh variables subset as the final variables combination for predictive model building.

5.5.2. Final Predictive Model

The final chosen variables that are included in the predictive model are the variables from the seventh variables subset. It includes all transformed product ownership variables (*IPLEZIER, IBRAND, ITRACTOR, IBROM, IBYSTAND, IWAOREG, IGEZONG, IFIETS, IAANHANG, IINBOED, ILEVEN, IWERKT, IPERSONG, IZEILPL*) and also mixture of variables that showed the highest importance to prediction in variable selection methods (*MKOOPKLA* – Purchasing power class, *MINKGEM* – Average income, *MOPLLAAG* – Lower level education, *MHKOOP* – Home owners, *MAUT0* – No car). The highest number of correctly identified caravan policy holders with these variables is 54. It is higher than the number 21, that would be obtained by selecting customers randomly.

To be sure, that the results are obtained not only by model accidently fitting the dataset well, the different dataset partitions are tested on models. It might happen that the model fits the specific dataset really well, while on another dataset the same model performs quite poorly. So, 10 different seeds in data partition are used to verify the obtained results. Also, running models on different datasets helps to choose the final model as well as to test how much models are stable. The detailed results of all 10 different partitioned datasets are presented in the Appendix 8.7.

To choose the best model, the average of correctly identified caravan insurance owners and the highest number of identified true positives are calculated for each model (see the Table 5.3).

Model	Average of Identified True Positives	Max of Identified True Positives	Model	Average of Identified True Positives	Max of Identified True Positives
Logistic Regression	43,6	48	MLP - 19 neurons	42,2	50
CART	42,6	50	MLP - 20 neurons	42,1	52
MLP - 1 neuron	28,4	49	SVM gamma = 0.001 C = 10	26,8	34
MLP - 2 neurons	42	47	SVM gamma = 0.001 C = 20	28,9	41
MLP - 3 neurons	43,5	50	SVM gamma = 0.001 C = 30	28,2	36
MLP - 4 neurons	43,4	48	SVM gamma = 0.001 C = 40	26,5	33
MLP - 5 neurons	43,1	50	SVM gamma = 0.001 C = 50	28,3	35
MLP - 6 neurons	43,2	51	SVM gamma = 0.001 C = 100	29,3	38
MLP - 7 neurons	43	54	SVM gamma = 0.01 C = 10	25,8	34
MLP - 8 neurons	41,7	49	SVM gamma = 0.01 C = 20	28,7	36
MLP - 9 neurons	42,1	49	SVM gamma = 0.01 C = 30	27,7	40
MLP - 10 neurons	42,8	50	SVM gamma = 0.01 C = 40	28,5	40
MLP - 11 neurons	43,2	52	SVM gamma = 0.01 C = 50	24,3	33
MLP - 12 neurons	44	55	SVM gamma = 0.01 C = 100	26,5	39
MLP - 13 neurons	43	50	SVM gamma = 0.1 C = 10	25,1	34
MLP - 14 neurons	41,9	53	SVM gamma = 0.1 C = 20	26,8	37
MLP - 15 neurons	42	52	SVM gamma = 0.1 C = 30	27,3	39
MLP - 16 neurons	42,4	51	SVM gamma = 0.1 C = 40	27,6	39
MLP - 17 neurons	42,6	52	SVM gamma = 0.1 C = 50	27,8	38
MLP - 18 neurons	42,8	54	SVM gamma = 0.1 C = 100	27,5	40

Table 5.3 – The results obtained from 10 different partitioned datasets

Based on the produced results of the Table 5.3, Multilayer Perceptron with 12 hidden neurons is chosen as the best model. This model has the highest correctly identified caravan insurance holders in average, and also with this model the highest number of correctly identified customers is obtained. As well as Sensitivity, Precision and F-measure of this model are the highest compared to other models (see in the Appendix 8.7).

It should also be pointed out that this model might not be very stable, because of the wide range between the lowest (36) and the highest (55) numbers of correctly identified caravan insurance holders. However, having 44 correctly identified insurance owners as the average is satisfying enough. The result is much better than randomly selecting customers and getting 21 correctly identified caravan insurance holders. This model might not be appropriate in some other areas like medicine, where accuracy of the model is significantly important aspect. However, in this case the main focus is on the number of correctly identified caravan insurance owners.

5.5.3. The Graphs of the Final Model

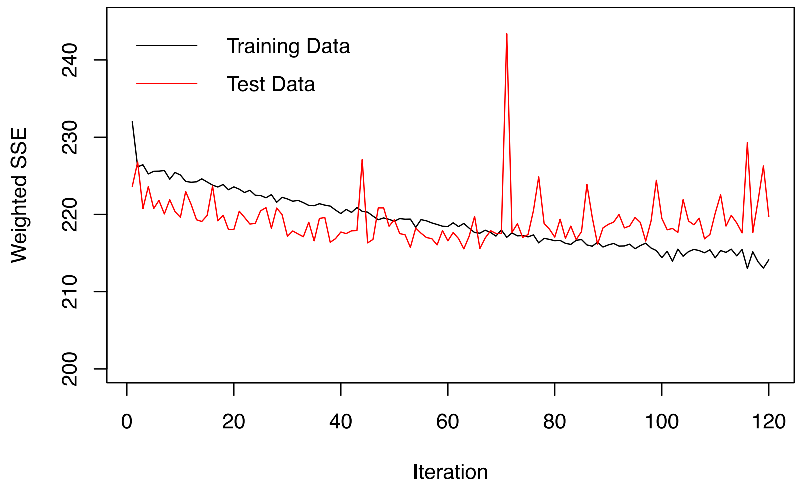


Figure 5.6 – Weighted sum square error graph of the final model

The above Figure 5.6 shows the Multilayer-Perceptron with 12 hidden neurons weighted sum square error graph by each iteration for training and test datasets. This graph can help to determine when the model training should be stopped. It is the point where the training and testing data cross each other. In this case, training should be stopped at around 70 iterations as it is done in the model building of this work. It is worth to mention that around 70 there is a peak very well seen in the graph. Even though the error difference between the peak and the error line is not that significant, still it has been tested that the peak does not appear at 70 iterations.

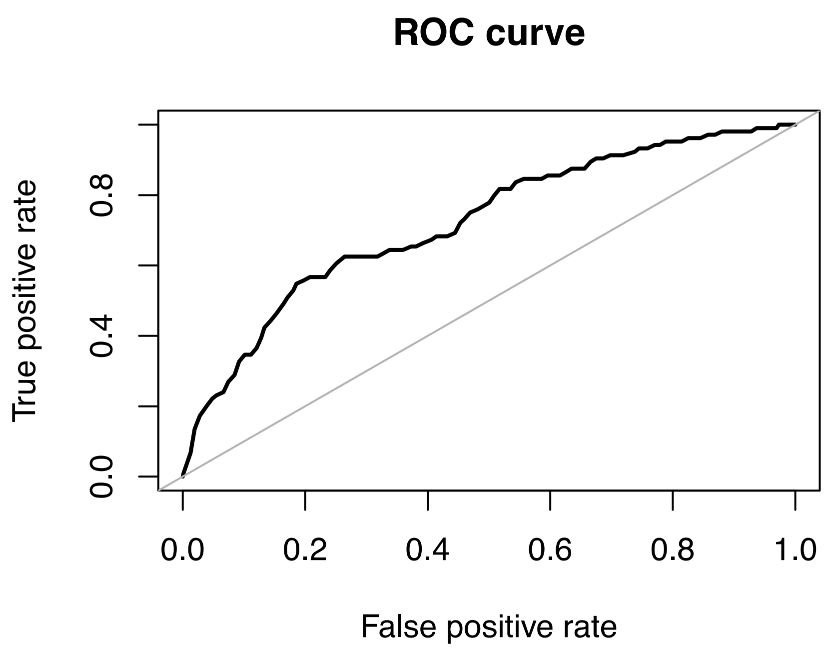


Figure 5.7 – ROC curve of the final model

The figure 5.7 displays ROC curve of the final model. ROC curve shows the relationship between true positive rate and false positive rate for all possible cut-offs. The closer curve is to the diagonal, the lower accuracy of the model is.

5.6. Predictions – Scoring Unseen Data

From prediction dataset, with the final predictive model 800 potential customers, that have the highest possibility to purchase caravan insurance policy, are identified. Having targets for prediction dataset, that are provided by the company Sentient Machine Research, it is possible to know that 104 potential caravan insurance holders have been identified correctly by the final chosen predictive model. The prediction dataset has 238 caravan insurance holders, so the obtained result is quite good. Randomly selecting customers from prediction dataset, it is possible to identify around 48 caravan insurance holders, so the performance of selected predictive model gives more than 50% better results. For more detailed results, see the Table 5.4.

Model	Seed	True Positives in the Test dataset	Identified True Positives	Random Guess	Lift	Sensitivity	Precision	F-measure
MLP - 12 neurons	888	238	104	48	56	0,437	0,130	0,200

Table 5.4 – The final predictive model performance

6. CONCLUSION

Using the data about insurance customers from the company Sentient Machine Research, data analysis has been made and useful information has been extracted. This allowed to identify the customers with the highest probability to purchase insurance caravan policy in order to contact them for direct marketing campaign of caravan insurance policies.

The main aim of the project is achieved, the predictive model to solve a real-world business problem is built. The final model correctly predicted 104 customers from 800 selected customers from prediction dataset. It is a half more correctly identified potential customers than it would have been achieved by random selection. The accuracy of the final model is not high but it is important to emphasize that dataset contains only 6% caravan insurance policy holders. This made some difficulties in variable selection and model building phases.

From 86 variables none of them with the significant importance to the prediction has been found. Any variable is not able to separate clearly enough caravan insurance holders and non-holders. The subset of the variables that would be important for prediction had to be found. In this work phase, Information Gain and Chi-squared metrics, Regression Stepwise, R package “Boruta”, Spearman correlation analysis, distribution graphs by target variable, as well as basic statistics of all variables are used to select variables combinations. For the future works, it is recommended to do better analysis for possible techniques to select variables for predictive model. One of them could be Principal Component Analysis (PCA). Also, it is advisable to do more variable transformations, for example, variable that represents the sub-type of customer could be split to binary variables.

Moreover, it should be remarked that sociodemographic data presents information not about exact customer but about the area in which customer is living in. To increase the accuracy of the predictive model it would be useful to have this information by each customer. However, to gain this data is quite difficult for insurance company because of the customers’ privacy.

Furthermore, looking at all obtained results from used machine learning methods, Multilayer Perceptron with backpropagation learning algorithm with 1 hidden layer and 12 hidden neurons shows the best results. Logistic regression and Decision Tree CART are not too far behind of the selected model. However, Support Vector Machine performance is quite poor. As a consequence, for the future works it is recommended to use the tuning algorithm for SVM parameters. The good performance of Logistic Regression shows that to solve a real-world business problem it might be enough to use a simpler method. In this way the black box of unnecessary complicated methods would be avoided. Anyway, it is advisable for the future works to apply boosting algorithm for all used methods. The performance of boosted machine learning methods has been improved in the reviewed articles (du Jardin, 2017; Vafeiadis, Diamantaras, Sarigiannidis & Chatzisavvas, 2015).

In addition, it is worth to emphasize that in the predictive model building phase, the models have been run several times with different data partitions so that randomness of the obtained results and model overfitting for the particular dataset would be avoided. So, it means that the choice of the final model is validated by testing models on different dataset partitions. The highest number of correctly identified caravan insurance holders made the highest impact for choosing the final predictive model.

7. BIBLIOGRAPHY

Author, A. A., Author, B. B., & Author, C. C. (Year). Title of article. *Title of Periodical, volume number* (issue number), pages.

Ansari, A., & Riasi, A. (2016). Modelling and evaluating customer loyalty using neural networks: Evidence from startup insurance companies. *Future Business Journal*, *2*(1), 15-30.

Benoit, D., & Van den Poel, D. (2012). Improving customer retention in financial services using kinship network information. *Expert Systems With Applications*, *39*(13), 11435-11442.

Bishop, C. (1995). *Neural networks for pattern recognition*. Oxford: Oxford University Press.

Bohanec, M., Kljajić Borštnar, M., & Robnik-Šikonja, M. (2017). Explaining machine learning models in sales predictions. *Expert Systems With Applications*, *71*, 416-428.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Wadsworth & Brooks. Monterey, CA.

Brynjolfsson, E., Hitt, L., & Kim, H. (2011). Strength in Numbers: How Does Data-Driven Decisionmaking Affect Firm Performance?. *SSRN Electronic Journal*.

Carbonell, J., Michalski, R., & Mitchell, T. (1983). *Machine learning*. Pittsburgh, Pa.: Dept. of Computer Science, Carnegie-Mellon University.

Coussement, K., Harrigan, P., & Benoit, D. (2015). Improving direct mail targeting through customer response modeling. *Expert Systems With Applications*, *42*(22), 8403-8412.

Davenport, T. (2013). *Industrial-Strength Analytics with Machine Learning*. *WSJ*. Retrieved 13 June 2017, from https://blogs.wsj.com/cio/2013/09/11/industrial-strength-analytics-with-machine-learning/?mod=wsj_streaming_latest-headlines

Delgado-Gómez, D., Aguado, D., Lopez-Castroman, J., Santacruz, C., & Artés-Rodriguez, A. (2011). Improving sale performance prediction using support vector machines. *Expert Systems With Applications*, *38*(5), 5129-5132.

du Jardin, P. (2017). Dynamics of firm financial evolution and bankruptcy prediction. *Expert Systems With Applications*, *75*, 25-43.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Transactions On Systems, Man, And Cybernetics, Part C (Applications And Reviews)*, *42*(4), 463-484.

Goonetilleke, T. O., & Caldera, H. A. (2013). Mining life insurance data for customer attrition analysis. *Journal of Industrial and Intelligent Information*, 1(1).

Gürsoy, U. T. Ş. (2010). Customer churn analysis in telecommunication sector. *Istanbul University Journal of the School of Business*, *39*(1), 35-49

Han, J., Kamber, M., & Pei, J. (2012). *Data mining*. Amsterdam: Elsevier/Morgan Kaufmann.

Jordan, M., & Mitchell, T. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Jost, P. (2016). Competitive insurance pricing with complete information, loss-averse utility and finitely many policies. *Insurance: Mathematics And Economics*, *66*, 11-21.

Julio, N., Giesen, R., & Lizana, P. (2016). Real-time prediction of bus travel speeds using traffic shockwaves and machine learning algorithms. *Research In Transportation Economics*, *59*, 250-257.

Karatzoglou, A., Meyer, D., & Hornik, K. (2006). Support Vector Machines in R. *Journal Of Statistical Software*, *15*(9), 1-28.

Kerdprasop, N., Kongchai, P., & Kerdprasop, K. (2013). Constraint mining in business intelligence: A case study of customer churn prediction. *International Journal of Multimedia and Ubiquitous Engineering, 8*(3), 11-20.

Kim, Y., & Street, W. (2004). An intelligent system for customer targeting: a data mining approach. *Decision Support Systems*, *37*(2), 215-228.

Ling, C. X., & Li, C. (1998). Data mining for direct marketing: Problems and solutions. *KDD*, *98*, 73-79.

Linoff, G. S., & Berry, M. J. (2011). *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons.

Machine Learning: What it is and why it matters. (2017). *Sas.com*. Retrieved 12 June 2017, from https://www.sas.com/en_us/insights/analytics/machine-learning.html

Maimon, O., & Rokach, L. (2005). *Data mining and knowledge discovery handbook*. New York: Springer.

Mathivanan, B., & Rajesh, R. (2016). *Communication and Power Engineering*. Walter de Gruyter GmbH & Co KG.

McCarty, J., & Hastak, M. (2007). Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression. *Journal Of Business Research*, *60*(6), 656-662.

Miikkulainen, R. (2011). *Topology of a Neural Network. In Encyclopedia of Machine Learning*. Springer US.

Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. Cambridge, MA: The MIT Press.

Morik, K., & Köpcke, H. (2004). Analysing customer churn in insurance data–a case study. *Knowledge Discovery in Databases: PKDD 2004*, 325-336.

Mozer, M., Wolniewicz, R., Grimes, D., Johnson, E., & Kaushansky, H. (2000). Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry. *IEEE Transactions On Neural Networks*, *11*(3), 690-696.

Neslin, S., Gupta, S., Kamakura, W., Lu, J., & Mason, C. (2006). Defection Detection: Measuring and Understanding the Predictive Accuracy of Customer Churn Models. *Journal Of Marketing Research*, *43*(2), 204-211.

Pan, Y., & Tang, Z. (2014). Ensemble methods in bank direct marketing. *Service Systems and Service Management (ICSSSM),* 1-5.

Parloff, R. (2016). *Why Deep Learning Is Suddenly Changing Your Life*. *Fortune*. Retrieved 13 June 2017, from http://fortune.com/ai-artificial-intelligence-deep-machine-learning/

Perlich, C., Dalessandro, B., Raeder, T., Stitelman, O., & Provost, F. (2013). Machine learning for targeted display advertising: transfer learning in action. *Machine Learning*, *95*(1), 103-127.

Pyle, D., & Jose, C. (2015). *An executive's guide to machine learning*. *McKinsey & Company*. Retrieved 13 June 2017, from http://www.mckinsey.com/industries/high-tech/our-insights/an-executives-guide-to-machine-learning

Rahman, M. S., Arefin, K. Z., Masud, S., Sultana, S., & Rahman, R. M. (2017). Analyzing Life Insurance Data with Different Classification Techniques for Customers' Behavior Analysis. *In Advanced Topics in Intelligent Information and Database Systems* (15-25). Springer International Publishing.

Reitermanov, Z. (2010). Data splitting. *WDS*, *10*, 31-36.

Rokach, L., & Maimon, O. (2014*). Data mining with decision trees: theory and applications*. World scientific.

Salcedo-Sanz, S., Fernández-Villacañas, J., Segovia-Vargas, M., & Bousoño-Calzón, C. (2005). Genetic programming for the prediction of insolvency in non-life insurance companies. *Computers & Operations Research*, *32*(4), 749-765.

Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal Of Research And Development*, *3*(3), 210-229.

Sentient Machine Research. (2000). The Insurance Company (TIC), Amsterdam, Amsterdam: Author.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning*. New York: Cambrige University Press.

Soeini, R. A., & Rodpysh, K. V. (2012). Applying data mining to insurance customer churn management. *Int. Proc. Comput. Sci. Inf. Technol*, 30, 82-92.

Tkáč, M., & Verner, R. (2016). Artificial neural networks in business: Two decades of research. *Applied Soft Computing*, *38*, 788-804.

Umamaheswari, K., & Janakiraman, S. (2014). Role of Data mining in Insurance Industry. *An international journal of advanced computer technology*, 3(6), 961-966.

Vafeiadis, T., Diamantaras, K., Sarigiannidis, G., & Chatzisavvas, K. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice And Theory*, *55*, 1-9.

Wai-Ho Au, Chan, K., & Xin Yao. (2003). A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Transactions On Evolutionary Computation*, *7*(6), 532-545.

*Wrapper Algorithm for All Relevant Feature Selection [R package Boruta version 5.2.0]. (*2017). *Cran.r-project.org*. Retrieved 26 September 2017, from https://cran.r-project.org/web/packages/Boruta/index.html

Zahavi, J., & Levin, N. (1997). Applying neural computing to target marketing. *Journal of Interactive Marketing*, 11(4), 76-93.

Zhang, C., & Ma, Y. (2012). *Ensemble Machine Learning*. Dordrecht: Springer.

Zięba, M., Tomczak, S., & Tomczak, J. (2016). Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems With Applications*, *58*, 93-101.

8. APPENDIX

8.1. VARIABLES LIST

In the below table, there are presented all variables with the variable number in the data source, name, description and containing values and their meaning.

Number	Name	Description
1	MOSTYPE	Customer Subtype, see L0
2	MAANTHUI	Number of houses, 1-10
3	MGEMOMV	Average size household, 1 – 6
4	MGEMLEEF	Average age, see L1
5	MOSHOOFD	Customer main type, see L2
6	MGODRK	Roman catholic, see L3
7	MGODPR	Protestant, see L3
8	MGODOV	Other religion, see L3
9	MGODGE	No religion, see L3
10	MRELGE	Married, see L3
11	MRELSA	Living together, see L3
12	MRELOV	Other relation, see L3
13	MFALLEEN	Singles, see L3
14	MFGEKIND	Household without children, see L3
15	MFWEKIND	Household with children, see L3
16	MOPLHOOG	High level education, see L3
17	MOPLMIDD	Medium level education, see L3
18	MOPLLAAG	Lower level education, see L3
19	MBERHOOG	High status, see L3
20	MBERZELF	Entrepreneur, see L3
21	MBERBOER	Farmer, see L3
22	MBERMIDD	Middle management, see L3
23	MBERARBG	Skilled labourers, see L3
24	MBERARBO	Unskilled labourers, see L3
25	MSKA	Social class A, see L3
26	MSKB1	Social class B1, see L3
27	MSKB2	Social class B2, see L3
28	MSKC	Social class C, see L3
29	MSKD	Social class D, see L3
30	MHHUUR	Rented house, see L3
31	MHKOOP	Home owners, see L3
32	MAUT1	1 car, see L3
33	MAUT2	2 cars, see L3
34	MAUT0	No car, see L3
35	MZFONDS	National Health Service, see L3
36	MZPART	Private health insurance, see L3

37	MINKM30	Income < 30.000, see L3
38	MINK3045	Income 30-45.000, see L3
39	MINK4575	Income 45-75.000, see L3
40	MINK7512	Income 75-122.000, see L3
41	MINK123M	Income >123.000, see L3
42	MINKGEM	Average income, see L3
43	MKOOPKLA	Purchasing power class, see L3
44	PWAPART	Contribution private third party insurance, see L4
45	PWABEDR	Contribution third party insurance (firms), see L4
46	PWALAND	Contribution third party insurance (agriculture), see L4
47	PPERSAUT	Contribution car policies, see L4
48	PBESAUT	Contribution delivery van policies, see L4
49	PMOTSCO	Contribution motorcycle/scooter policies, see L4
50	PVRAAUT	Contribution lorry policies, see L4
51	PAANHANG	Contribution trailer policies, see L4
52	PTRACTOR	Contribution tractor policies, see L4
53	PWERKT	Contribution agricultural machines policies, see L4
54	PBROM	Contribution moped policies, see L4
55	PLEVEN	Contribution life insurances, see L4
56	PPERSONG	Contribution private accident insurance policies, see L4
57	PGEZONG	Contribution family accidents insurance policies, see L4
58	PWAOREG	Contribution disability insurance policies, see L4
59	PBRAND	Contribution fire policies, see L4
60	PZEILPL	Contribution surfboard policies, see L4
61	PPLEZIER	Contribution boat policies, see L4
62	PFIETS	Contribution bicycle policies, see L4
63	PINBOED	Contribution property insurance policies, see L4
64	PBYSTAND	Contribution social security insurance policies, see L4
65	AWAPART	Number of private third-party insurance, 1 - 12
66	AWABEDR	Number of third party insurance (firms), 1 - 12
67	AWALAND	Number of third party insurance (agriculture), 1 - 12
68	APERSAUT	Number of car policies, 1 - 12
69	ABESAUT	Number of delivery van policies, 1 - 12
70	AMOTSCO	Number of motorcycle/scooter policies, 1 - 12
71	AVRAAUT	Number of lorry policies, 1 - 12
72	AAANHANG	Number of trailer policies, 1 - 12
73	ATRACTOR	Number of tractor policies, 1 - 12
74	AWERKT	Number of agricultural machines policies, 1 - 12
75	ABROM	Number of moped policies, 1 - 12
76	ALEVEN	Number of life insurances, 1 - 12
77	APERSONG	Number of private accident insurance policies, 1 - 12
78	AGEZONG	Number of family accidents insurance policies, 1 - 12
79	AWAOREG	Number of disability insurance policies, 1 - 12

80	ABRAND	Number of fire policies, 1 - 12
81	AZEILPL	Number of surfboard policies, 1 - 12
82	APLEZIER	Number of boat policies, 1 - 12
83	AFIETS	Number of bicycle policies, 1 - 12
84	AINBOED	Number of property insurance policies, 1 - 12
85	ABYSTAND	Number of social security insurance policies, 1 - 12
86	Purchase	Number of mobile home policies, 0 - 1

Table 8.1 – Variables List

Below there are tables that describe containing values in variables.

L0:	**Value**	**Description**
	1	High Income, expensive child
	2	Very Important Provincials
	3	High status seniors
	4	Affluent senior apartments
	5	Mixed seniors
	6	Career and childcare
	7	Dinki's (double income no kids)
	8	Middle class families
	9	Modern, complete families
	10	Stable family
	11	Family starters
	12	Affluent young families
	13	Young all American family
	14	Junior cosmopolitan
	15	Senior cosmopolitans
	16	Students in apartments
	17	Fresh masters in the city
	18	Single youth
	19	Suburban youth
	20	Ethnically diverse
	21	Young urban have-nots
	22	Mixed apartment dwellers
	23	Young and rising
	24	Young, low educated
	25	Young seniors in the city
	26	Own home elderly
	27	Seniors in apartments
	28	Residential elderly
	29	Porchless seniors: no front yard
	30	Religious elderly singles
	31	Low income catholics
	32	Mixed seniors

	33	Lower class large families
	34	Large family, employed child
	35	Village families
	36	Couples with teens 'Married with children'
	37	Mixed small town dwellers
	38	Traditional families
	39	Large religious families
	40	Large family farms
	41	Mixed rurals

L1:	**Value**	**Description**
	1	20-30 years
	2	30-40 years
	3	40-50 years
	4	50-60 years
	5	60-70 years
	6	70-80 years
L2:	**Value**	**Description**
	1	Successful hedonists
	2	Driven Growers
	3	Average Family
	4	Career Loners
	5	Living well
	6	Cruising Seniors
	7	Retired and Religious
	8	Family with grown ups
	9	Conservative families
	10	Farmers
L3:	**Value**	**Description**
	0	0%
	1	1 - 10%
	2	11 - 23%
	3	24 - 36%
	4	37 - 49%
	5	50 - 62%
	6	63 - 75%
	7	76 - 88%
	8	89 - 99%
	9	100%
L4:	**Value**	**Description**
	0	0
	1	1 – 49

2	50 – 99
3	100 – 199
4	200 – 499
5	500 – 999
6	1000 – 4999
7	5000 – 9999
8	10.000 - 19.999
9	20.000 - ?

Table 8.2- Description of variables values

8.2. Basic Analysis by Each Level of Each Predictor Variable

This table shows the number of customers that belongs to each group (level of each variable). Also, the percentage and standard deviation of customers that purchased caravan insurance policy in that group.

Category	Variable	# Customers in this group	% Policy holders in the group	sd Policy holders in the group
Affluent senior apartments	STYPE	52	3,85	0,1942
Affluent young families	STYPE	111	14,41	0,3528
Career and childcare	STYPE	119	10,08	0,3024
Couples with teens 'Married with children'	STYPE	225	7,11	0,2576
Dinki's (double income no kids)	STYPE	44	6,82	0,255
Etnically diverse	STYPE	25	8	0,2769
Family starters	STYPE	153	5,88	0,2361
Fresh masters in the city	STYPE	9	0	0
High Income, expensive child	STYPE	124	10,48	0,3076
High status seniors	STYPE	249	10,04	0,3011
Large family farms	STYPE	71	0	0
Large family, employed child	STYPE	182	4,95	0,2174
Large religous families	STYPE	328	5,79	0,234
Low income catholics	STYPE	205	2,93	0,169
Lower class large families	STYPE	810	5,68	0,2316
Middle class families	STYPE	339	15,04	0,358
Mixed apartment dwellers	STYPE	98	4,08	0,1989
Mixed rurals	STYPE	205	2,44	0,1546
Mixed seniors	STYPE	186	5,38	0,2262
Mixed small town dwellers	STYPE	132	7,58	0,2656
Modern, complete families	STYPE	278	4,32	0,2036
Own home elderly	STYPE	48	2,08	0,1443

Porchless seniors: no front yard	STYPE	86	2,33	0,1516
Religious elderly singles	STYPE	118	3,39	0,1817
Residential elderly	STYPE	25	0	0
Senior cosmopolitans	STYPE	5	0	0
Seniors in apartments	STYPE	50	2	0,1414
Single youth	STYPE	19	0	0
Stable family	STYPE	165	5,45	0,2278
Students in apartments	STYPE	16	0	0
Suburban youth	STYPE	3	0	0
Traditional families	STYPE	339	6,78	0,2519
Very Important Provincials	STYPE	82	7,32	0,262
Village families	STYPE	214	3,74	0,1901
Young all American family	STYPE	179	7,26	0,2602
Young and rising	STYPE	251	1,59	0,1255
Young seniors in the city	STYPE	82	2,44	0,1552
Young urban have-nots	STYPE	15	0	0
Young, low educated	STYPE	180	2,78	0,1648
1	MAANTHUI	5267	5,98	0,2372
2	MAANTHUI	505	6,53	0,2474
3	MAANTHUI	39	0	0
4	MAANTHUI	2	0	0
5	MAANTHUI	1	0	0
6	MAANTHUI	1	0	0
7	MAANTHUI	5	0	0
8	MAANTHUI	1	0	0
10	MAANTHUI	1	0	0
1	MGEMOMV	284	2,82	0,1657
2	MGEMOMV	2131	5,4	0,226
3	MGEMOMV	2646	6,46	0,2459
4	MGEMOMV	693	7,22	0,2589
5	MGEMOMV	68	5,88	0,237
20-30	MGEMLEEF	74	1,35	0,1162
30-40	MGEMLEEF	1452	5,99	0,2374
40-50	MGEMLEEF	3000	6,1	0,2394
50-60	MGEMLEEF	1073	5,96	0,2369
60-70	MGEMLEEF	193	6,22	0,2421
70-80	MGEMLEEF	30	3,33	0,1826
Average Family	MOSHOOFD	886	6,66	0,2495
Career Loners	MOSHOOFD	52	0	0
Conservative families	MOSHOOFD	667	6,3	0,2431
Cruising Seniors	MOSHOOFD	205	1,95	0,1387
Driven Growers	MOSHOOFD	502	13,15	0,3383
Family with grown ups	MOSHOOFD	1563	5,69	0,2318

Farmers	MOSHOOFD	276	1,81	0,1336
Living well	MOSHOOFD	569	2,64	0,1604
Retired and Religious	MOSHOOFD	550	3,64	0,1874
Successful hedonists	MOSHOOFD	552	8,7	0,282
0%	MGODRK	3228	5,48	0,2277
1 - 10%	MGODRK	1599	6,69	0,25
11 - 23%	MGODRK	733	7,37	0,2614
24 - 36%	MGODRK	152	4,61	0,2103
37 - 49%	MGODRK	66	1,52	0,1231
50 - 62%	MGODRK	18	5,56	0,2357
63 - 75%	MGODRK	13	7,69	0,2774
76 - 88%	MGODRK	6	0	0
89 - 99%	MGODRK	3	0	0
100%	MGODRK	4	0	0
0%	MGODPR	78	1,28	0,1132
1 - 10%	MGODPR	134	3,73	0,1902
11 - 23%	MGODPR	396	6,06	0,2389
24 - 36%	MGODPR	590	4,92	0,2164
37 - 49%	MGODPR	1607	5,41	0,2264
50 - 62%	MGODPR	1501	6,46	0,2459
63 - 75%	MGODPR	714	5,46	0,2274
76 - 88%	MGODPR	564	9,75	0,2969
89 - 99%	MGODPR	65	4,62	0,2115
100%	MGODPR	173	4,62	0,2106
0%	MGODOV	2003	6,49	0,2464
1 - 10%	MGODOV	2014	4,62	0,2099
11 - 23%	MGODOV	1388	7,13	0,2575
24 - 36%	MGODOV	257	7	0,2557
37 - 49%	MGODOV	132	5,3	0,2249
50 - 62%	MGODOV	28	3,57	0,189
63 - 75%	MGODOV	0	0	0
76 - 88%	MGODOV	0	0	0
89 - 99%	MGODOV	0	0	0
100%	MGODOV	0	0	0
0%	MGODGE	456	8,11	0,2734
1 - 10%	MGODGE	230	9,13	0,2887
11 - 23%	MGODGE	1055	6,54	0,2474
24 - 36%	MGODGE	1453	5,99	0,2373
37 - 49%	MGODGE	1334	5,25	0,2231
50 - 62%	MGODGE	963	5,3	0,2241
63 - 75%	MGODGE	217	1,84	0,1348
76 - 88%	MGODGE	101	8,91	0,2863
89 - 99%	MGODGE	5	0	0

100%	MGODGE	8	0	0
0%	MRELGE	64	3,12	0,1754
1 - 10%	MRELGE	75	1,33	0,1155
11 - 23%	MRELGE	157	1,91	0,1373
24 - 36%	MRELGE	246	2,44	0,1546
37 - 49%	MRELGE	324	3,09	0,1732
50 - 62%	MRELGE	946	5,07	0,2196
63 - 75%	MRELGE	1172	6,06	0,2387
76 - 88%	MRELGE	1683	6,89	0,2534
89 - 99%	MRELGE	361	6,93	0,2542
100%	MRELGE	794	8,31	0,2762
0%	MRELSA	2448	6,58	0,2479
1 - 10%	MRELSA	2030	5,91	0,2359
11 - 23%	MRELSA	1075	5,49	0,2279
24 - 36%	MRELSA	159	4,4	0,2058
37 - 49%	MRELSA	78	1,28	0,1132
50 - 62%	MRELSA	18	0	0
63 - 75%	MRELSA	13	0	0
76 - 88%	MRELSA	1	0	0
89 - 99%	MRELSA	0	0	0
100%	MRELSA	0	0	0
0%	MRELOV	1173	7,93	0,2703
1 - 10%	MRELOV	539	6,12	0,24
11 - 23%	MRELOV	1756	6,26	0,2424
24 - 36%	MRELOV	1152	6,42	0,2453
37 - 49%	MRELOV	648	4,01	0,1964
50 - 62%	MRELOV	266	2,26	0,1488
63 - 75%	MRELOV	179	2,23	0,1482
76 - 88%	MRELOV	64	1,56	0,125
89 - 99%	MRELOV	21	0	0
100%	MRELOV	24	4,17	0,2041
0%	MFALLEEN	1757	7,29	0,26
1 - 10%	MFALLEEN	951	6,73	0,2507
11 - 23%	MFALLEEN	1247	6,01	0,2378
24 - 36%	MFALLEEN	848	4,83	0,2146
37 - 49%	MFALLEEN	519	4,05	0,1972
50 - 62%	MFALLEEN	259	5,02	0,2188
63 - 75%	MFALLEEN	127	3,15	0,1753
76 - 88%	MFALLEEN	67	1,49	0,1222
89 - 99%	MFALLEEN	24	4,17	0,2041
100%	MFALLEEN	23	0	0
0%	MFGEKIND	371	6,2	0,2415
1 - 10%	MFGEKIND	372	6,45	0,246

11 - 23%	MFGEKIND	1060	5,38	0,2257
24 - 36%	MFGEKIND	1498	5,94	0,2365
37 - 49%	MFGEKIND	1455	6,05	0,2385
50 - 62%	MFGEKIND	606	4,95	0,2171
63 - 75%	MFGEKIND	321	8,41	0,278
76 - 88%	MFGEKIND	96	8,33	0,2778
89 - 99%	MFGEKIND	14	7,14	0,2673
100%	MFGEKIND	29	3,45	0,1857
0%	MFWEKIND	153	3,27	0,1784
1 - 10%	MFWEKIND	292	3,42	0,1822
11 - 23%	MFWEKIND	635	6,14	0,2403
24 - 36%	MFWEKIND	973	6,06	0,2388
37 - 49%	MFWEKIND	1137	5,8	0,2339
50 - 62%	MFWEKIND	1106	5,24	0,223
63 - 75%	MFWEKIND	783	7,15	0,2579
76 - 88%	MFWEKIND	351	6,55	0,2478
89 - 99%	MFWEKIND	206	8,74	0,2831
100%	MFWEKIND	186	7,53	0,2645
0%	MOPLHOOG	2147	4,24	0,2015
1 - 10%	MOPLHOOG	1322	5,52	0,2285
11 - 23%	MOPLHOOG	1144	6,12	0,2398
24 - 36%	MOPLHOOG	547	7,13	0,2576
37 - 49%	MOPLHOOG	326	11,35	0,3177
50 - 62%	MOPLHOOG	187	11,23	0,3166
63 - 75%	MOPLHOOG	67	11,94	0,3267
76 - 88%	MOPLHOOG	51	11,76	0,3254
89 - 99%	MOPLHOOG	22	9,09	0,2942
100%	MOPLHOOG	9	11,11	0,3333
0%	MOPLMIDD	423	3,31	0,1791
1 - 10%	MOPLMIDD	383	5,74	0,233
11 - 23%	MOPLMIDD	937	5,12	0,2206
24 - 36%	MOPLMIDD	1330	6,02	0,2379
37 - 49%	MOPLMIDD	1426	6,03	0,2381
50 - 62%	MOPLMIDD	738	7,32	0,2606
63 - 75%	MOPLMIDD	348	5,46	0,2275
76 - 88%	MOPLMIDD	157	11,46	0,3196
89 - 99%	MOPLMIDD	37	8,11	0,2767
100%	MOPLMIDD	43	9,3	0,2939
0%	MOPLLAAG	299	9,7	0,2964
1 - 10%	MOPLLAAG	243	11,11	0,3149
11 - 23%	MOPLLAAG	667	9,75	0,2968
24 - 36%	MOPLLAAG	680	6,91	0,2538
37 - 49%	MOPLLAAG	851	5,99	0,2375

50 - 62%	MOPLLAAG	1009	4,86	0,2151
63 - 75%	MOPLLAAG	856	3,74	0,1898
76 - 88%	MOPLLAAG	640	4,22	0,2012
89 - 99%	MOPLLAAG	254	5,12	0,2208
100%	MOPLLAAG	323	2,48	0,1557
0%	MBERHOOG	1524	4,79	0,2136
1 - 10%	MBERHOOG	1245	4,9	0,2159
11 - 23%	MBERHOOG	1364	5,72	0,2323
24 - 36%	MBERHOOG	756	7,01	0,2555
37 - 49%	MBERHOOG	397	8,31	0,2764
50 - 62%	MBERHOOG	249	5,22	0,2229
63 - 75%	MBERHOOG	138	13,04	0,338
76 - 88%	MBERHOOG	92	15,22	0,3612
89 - 99%	MBERHOOG	26	11,54	0,3258
100%	MBERHOOG	31	6,45	0,2497
0%	MBERZELF	4171	5,59	0,2297
1 - 10%	MBERZELF	1202	6,82	0,2522
11 - 23%	MBERZELF	348	7,47	0,2633
24 - 36%	MBERZELF	37	8,11	0,2767
37 - 49%	MBERZELF	12	8,33	0,2887
50 - 62%	MBERZELF	52	5,77	0,2354
63 - 75%	MBERZELF	0	0	0
76 - 88%	MBERZELF	0	0	0
89 - 99%	MBERZELF	0	0	0
100%	MBERZELF	0	0	0
0%	MBERBOER	4176	6,8	0,2518
1 - 10%	MBERBOER	854	4,22	0,2011
11 - 23%	MBERBOER	487	4,11	0,1987
24 - 36%	MBERBOER	143	4,2	0,2012
37 - 49%	MBERBOER	77	1,3	0,114
50 - 62%	MBERBOER	59	1,69	0,1302
63 - 75%	MBERBOER	14	0	0
76 - 88%	MBERBOER	3	0	0
89 - 99%	MBERBOER	5	0	0
100%	MBERBOER	4	0	0
0%	MBERMIDD	667	5,25	0,2231
1 - 10%	MBERMIDD	403	4,71	0,2122
11 - 23%	MBERMIDD	1491	5,7	0,2319
24 - 36%	MBERMIDD	1394	4,88	0,2155
37 - 49%	MBERMIDD	953	6,4	0,2449
50 - 62%	MBERMIDD	431	7,66	0,2662
63 - 75%	MBERMIDD	211	8,06	0,2728
76 - 88%	MBERMIDD	178	12,36	0,33

89 - 99%	MBERMIDD	14	0	0
100%	MBERMIDD	80	10	0,3019
0%	MBERARBG	1167	6,6	0,2484
1 - 10%	MBERARBG	921	9,01	0,2865
11 - 23%	MBERARBG	1382	6,22	0,2417
24 - 36%	MBERARBG	1167	4,2	0,2006
37 - 49%	MBERARBG	604	4,14	0,1994
50 - 62%	MBERARBG	310	4,19	0,2008
63 - 75%	MBERARBG	169	3,55	0,1856
76 - 88%	MBERARBG	68	7,35	0,2629
89 - 99%	MBERARBG	24	12,5	0,3378
100%	MBERARBG	10	10	0,3162
0%	MBERARBO	968	8,06	0,2723
1 - 10%	MBERARBO	980	7,76	0,2676
11 - 23%	MBERARBO	1439	5,7	0,2319
24 - 36%	MBERARBO	1109	4,51	0,2076
37 - 49%	MBERARBO	772	4,92	0,2165
50 - 62%	MBERARBO	331	4,53	0,2083
63 - 75%	MBERARBO	122	4,92	0,2171
76 - 88%	MBERARBO	66	3,03	0,1727
89 - 99%	MBERARBO	9	11,11	0,3333
100%	MBERARBO	26	0	0
0%	MSKA	1738	4,83	0,2145
1 - 10%	MSKA	1569	5,1	0,22
11 - 23%	MSKA	1198	5,59	0,2299
24 - 36%	MSKA	685	7,01	0,2555
37 - 49%	MSKA	261	11,11	0,3149
50 - 62%	MSKA	127	11,02	0,3144
63 - 75%	MSKA	96	12,5	0,3325
76 - 88%	MSKA	79	16,46	0,3731
89 - 99%	MSKA	13	7,69	0,2774
100%	MSKA	56	0	0
0%	MSKB1	1353	5,32	0,2245
1 - 10%	MSKB1	1480	5,47	0,2275
11 - 23%	MSKB1	1783	5,66	0,2312
24 - 36%	MSKB1	775	8,39	0,2774
37 - 49%	MSKB1	298	6,38	0,2447
50 - 62%	MSKB1	78	5,13	0,222
63 - 75%	MSKB1	25	20	0,4082
76 - 88%	MSKB1	5	0	0
89 - 99%	MSKB1	8	12,5	0,3536
100%	MSKB1	17	0	0
0%	MSKB2	990	5,86	0,235

1 - 10%	MSKB2	861	5,46	0,2273
11 - 23%	MSKB2	1676	5,97	0,2369
24 - 36%	MSKB2	1175	6,21	0,2415
37 - 49%	MSKB2	652	7,21	0,2588
50 - 62%	MSKB2	357	4,76	0,2133
63 - 75%	MSKB2	96	6,25	0,2433
76 - 88%	MSKB2	6	0	0
89 - 99%	MSKB2	7	0	0
100%	MSKB2	2	0	0
0%	MSKC	364	6,87	0,2533
1 - 10%	MSKC	272	11,03	0,3138
11 - 23%	MSKC	870	8,16	0,2739
24 - 36%	MSKC	1090	5,6	0,23
37 - 49%	MSKC	1159	5,35	0,2251
50 - 62%	MSKC	1168	3,94	0,1946
63 - 75%	MSKC	487	5,95	0,2369
76 - 88%	MSKC	217	4,61	0,2101
89 - 99%	MSKC	71	11,27	0,3184
100%	MSKC	124	4,84	0,2155
0%	MSKD	2607	7,21	0,2587
1 - 10%	MSKD	1563	6,27	0,2425
11 - 23%	MSKD	852	4,69	0,2117
24 - 36%	MSKD	441	3,17	0,1755
37 - 49%	MSKD	223	2,24	0,1484
50 - 62%	MSKD	100	1	0,1
63 - 75%	MSKD	22	4,55	0,2132
76 - 88%	MSKD	13	7,69	0,2774
89 - 99%	MSKD	0	0	0
100%	MSKD	1	0	0
0%	MHHUUR	949	9,91	0,2989
1 - 10%	MHHUUR	428	8,64	0,2814
11 - 23%	MHHUUR	717	5,3	0,2242
24 - 36%	MHHUUR	593	6,58	0,2481
37 - 49%	MHHUUR	517	5,03	0,2188
50 - 62%	MHHUUR	519	4,82	0,2143
63 - 75%	MHHUUR	382	6,02	0,2382
76 - 88%	MHHUUR	425	4,47	0,2069
89 - 99%	MHHUUR	532	3,01	0,171
100%	MHHUUR	760	4,08	0,1979
0%	MHKOOP	760	4,08	0,1979
1 - 10%	MHKOOP	530	3,02	0,1713
11 - 23%	MHKOOP	426	4,46	0,2067
24 - 36%	MHKOOP	382	6,02	0,2382

37 - 49%	MHKOOP	499	5,01	0,2184
50 - 62%	MHKOOP	520	5	0,2182
63 - 75%	MHKOOP	604	6,46	0,246
76 - 88%	MHKOOP	724	5,25	0,2232
89 - 99%	MHKOOP	428	8,64	0,2814
100%	MHKOOP	949	9,91	0,2989
0%	MAUT1	19	0	0
1 - 10%	MAUT1	14	0	0
11 - 23%	MAUT1	58	1,72	0,1313
24 - 36%	MAUT1	231	3,03	0,1718
37 - 49%	MAUT1	448	2,9	0,168
50 - 62%	MAUT1	1210	4,88	0,2155
63 - 75%	MAUT1	1663	5,47	0,2275
76 - 88%	MAUT1	1413	8,42	0,2778
89 - 99%	MAUT1	261	7,28	0,2603
100%	MAUT1	505	7,72	0,2672
0%	MAUT2	1854	5,77	0,2333
1 - 10%	MAUT2	1468	5,72	0,2323
11 - 23%	MAUT2	1748	6,41	0,245
24 - 36%	MAUT2	385	5,71	0,2324
37 - 49%	MAUT2	301	6,31	0,2436
50 - 62%	MAUT2	56	5,36	0,2272
63 - 75%	MAUT2	9	11,11	0,3333
76 - 88%	MAUT2	1	0	0
89 - 99%	MAUT2	0	0	0
100%	MAUT2	0	0	0
0%	MAUT0	1450	8,34	0,2767
1 - 10%	MAUT0	776	6,19	0,241
11 - 23%	MAUT0	1625	6,65	0,2492
24 - 36%	MAUT0	1066	4,6	0,2095
37 - 49%	MAUT0	587	2,21	0,1473
50 - 62%	MAUT0	174	2,3	0,1503
63 - 75%	MAUT0	89	5,62	0,2316
76 - 88%	MAUT0	25	0	0
89 - 99%	MAUT0	13	0	0
100%	MAUT0	17	0	0
0%	MZFONDS	55	12,73	0,3363
1 - 10%	MZFONDS	15	0	0
11 - 23%	MZFONDS	307	8,79	0,2837
24 - 36%	MZFONDS	177	8,47	0,2793
37 - 49%	MZFONDS	357	7,84	0,2692
50 - 62%	MZFONDS	974	7,39	0,2618
63 - 75%	MZFONDS	875	5,14	0,221

76 - 88%	MZFONDS	1511	5,76	0,233
89 - 99%	MZFONDS	699	4,01	0,1962
100%	MZFONDS	852	4,58	0,2091
0%	MZPART	852	4,58	0,2091
1 - 10%	MZPART	699	4,01	0,1962
11 - 23%	MZPART	1511	5,76	0,233
24 - 36%	MZPART	849	5,3	0,2242
37 - 49%	MZPART	992	7,26	0,2596
50 - 62%	MZPART	364	7,69	0,2668
63 - 75%	MZPART	178	8,43	0,2786
76 - 88%	MZPART	307	8,79	0,2837
89 - 99%	MZPART	15	0	0
100%	MZPART	55	12,73	0,3363
0%	MINKM30	1304	7,52	0,2637
1 - 10%	MINKM30	630	8,57	0,2802
11 - 23%	MINKM30	1094	8,32	0,2763
24 - 36%	MINKM30	1079	4,63	0,2103
37 - 49%	MINKM30	599	3,51	0,1841
50 - 62%	MINKM30	568	2,99	0,1705
63 - 75%	MINKM30	293	3,07	0,1728
76 - 88%	MINKM30	156	4,49	0,2077
89 - 99%	MINKM30	48	2,08	0,1443
100%	MINKM30	51	0	0
0%	MINK3045	465	5,59	0,23
1 - 10%	MINK3045	268	6,72	0,2508
11 - 23%	MINK3045	919	5,98	0,2373
24 - 36%	MINK3045	1147	6,45	0,2458
37 - 49%	MINK3045	1356	6,19	0,2411
50 - 62%	MINK3045	931	4,83	0,2146
63 - 75%	MINK3045	406	6,16	0,2407
76 - 88%	MINK3045	205	4,88	0,2159
89 - 99%	MINK3045	35	5,71	0,2355
100%	MINK3045	90	10	0,3017
0%	MINK4575	891	4,26	0,2022
1 - 10%	MINK4575	657	3,65	0,1877
11 - 23%	MINK4575	1165	4,12	0,1988
24 - 36%	MINK4575	1215	6,91	0,2538
37 - 49%	MINK4575	1034	8,51	0,2792
50 - 62%	MINK4575	498	8,23	0,2751
63 - 75%	MINK4575	125	8,8	0,2844
76 - 88%	MINK4575	93	7,53	0,2653
89 - 99%	MINK4575	53	5,66	0,2333
100%	MINK4575	91	4,4	0,2061

0%	MINK7512	3246	4,68	0,2113
1 - 10%	MINK7512	1359	7,14	0,2575
11 - 23%	MINK7512	736	7,88	0,2696
24 - 36%	MINK7512	246	7,72	0,2675
37 - 49%	MINK7512	147	9,52	0,2945
50 - 62%	MINK7512	71	9,86	0,3002
63 - 75%	MINK7512	8	0	0
76 - 88%	MINK7512	1	0	0
89 - 99%	MINK7512	4	0	0
100%	MINK7512	4	25	0,5
0%	MINK123M	4900	5,9	0,2356
1 - 10%	MINK123M	763	6,55	0,2476
11 - 23%	MINK123M	96	8,33	0,2778
24 - 36%	MINK123M	36	2,78	0,1667
37 - 49%	MINK123M	24	0	0
50 - 62%	MINK123M	1	0	0
63 - 75%	MINK123M	0	0	0
76 - 88%	MINK123M	1	0	0
89 - 99%	MINK123M	0	0	0
100%	MINK123M	1	0	0
0%	MINKGEM	25	0	0
1 - 10%	MINKGEM	49	2,04	0,1429
11 - 23%	MINKGEM	651	3,07	0,1727
24 - 36%	MINKGEM	1932	3,57	0,1856
37 - 49%	MINKGEM	1854	7,5	0,2634
50 - 62%	MINKGEM	733	9,55	0,2941
63 - 75%	MINKGEM	355	6,76	0,2514
76 - 88%	MINKGEM	131	12,98	0,3373
89 - 99%	MINKGEM	70	11,43	0,3205
100%	MINKGEM	22	0	0
0%	MKOOPKLA	0	0	0
1 - 10%	MKOOPKLA	587	3,07	0,1726
11 - 23%	MKOOPKLA	425	3,53	0,1847
24 - 36%	MKOOPKLA	1524	4,66	0,2108
37 - 49%	MKOOPKLA	902	5,1	0,2201
50 - 62%	MKOOPKLA	583	5,15	0,2211
63 - 75%	MKOOPKLA	901	7,33	0,2607
76 - 88%	MKOOPKLA	474	14,14	0,3488
89 - 99%	MKOOPKLA	426	8,22	0,2749
100%	MKOOPKLA	0	0	0
f 0	PWAPART	3482	4,22	0,2011
f 1-49	PWAPART	201	3,98	0,196
f 50-99	PWAPART	2128	8,98	0,2859

f 100-199	PWAPART	11	18,18	0,4045
f 200-499	PWAPART	0	0	0
f 500-999	PWAPART	0	0	0
f 1000-4999	PWAPART	0	0	0
f 5000-9999	PWAPART	0	0	0
f 10000-19999	PWAPART	0	0	0
f 20000-?	PWAPART	0	0	0
f 0	PWABEDR	5740	5,98	0,2371
f 1-49	PWABEDR	7	0	0
f 50-99	PWABEDR	30	6,67	0,2537
f 100-199	PWABEDR	23	13,04	0,3444
f 200-499	PWABEDR	17	0	0
f 500-999	PWABEDR	1	0	0
f 1000-4999	PWABEDR	4	0	0
f 5000-9999	PWABEDR	0	0	0
f 10000-19999	PWABEDR	0	0	0
f 20000-?	PWABEDR	0	0	0
f 0	PWALAND	5702	6,05	0,2384
f 1-49	PWALAND	0	0	0
f 50-99	PWALAND	3	0	0
f 100-199	PWALAND	57	3,51	0,1856
f 200-499	PWALAND	60	1,67	0,1291
f 500-999	PWALAND	0	0	0
f 1000-4999	PWALAND	0	0	0
f 5000-9999	PWALAND	0	0	0
f 10000-19999	PWALAND	0	0	0
f 20000-?	PWALAND	0	0	0
f 0	PPERSAUT	2845	2,53	0,1571
f 1-49	PPERSAUT	0	0	0
f 50-99	PPERSAUT	0	0	0
f 100-199	PPERSAUT	0	0	0
f 200-499	PPERSAUT	1	0	0
f 500-999	PPERSAUT	613	2,28	0,1495
f 1000-4999	PPERSAUT	2319	11,3	0,3166
f 5000-9999	PPERSAUT	41	0	0
f 10000-19999	PPERSAUT	3	0	0
f 20000-?	PPERSAUT	0	0	0
f 0	PBESAUT	5774	5,99	0,2374
f 1-49	PBESAUT	0	0	0
f 50-99	PBESAUT	0	0	0
f 100-199	PBESAUT	0	0	0
f 200-499	PBESAUT	0	0	0
f 500-999	PBESAUT	10	0	0

f 1000-4999	PBESAUT	35	5,71	0,2355
f 5000-9999	PBESAUT	3	0	0
f 10000-19999	PBESAUT	0	0	0
f 20000-?	PBESAUT	0	0	0
f 0	PMOTSCO	5600	5,93	0,2362
f 1-49	PMOTSCO	0	0	0
f 50-99	PMOTSCO	0	0	0
f 100-199	PMOTSCO	3	66,67	0,5774
f 200-499	PMOTSCO	136	6,62	0,2495
f 500-999	PMOTSCO	32	12,5	0,336
f 1000-4999	PMOTSCO	49	2,04	0,1429
f 5000-9999	PMOTSCO	2	0	0
f 10000-19999	PMOTSCO	0	0	0
f 20000-?	PMOTSCO	0	0	0
f 0	AAUT	5813	5,99	0,2373
f 1-49	AAUT	0	0	0
f 50-99	AAUT	0	0	0
f 100-199	AAUT	0	0	0
f 200-499	AAUT	1	0	0
f 500-999	AAUT	0	0	0
f 1000-4999	AAUT	7	0	0
f 5000-9999	AAUT	0	0	0
f 10000-19999	AAUT	0	0	0
f 20000-?	AAUT	1	0	0
f 0	PAANHANG	5757	5,94	0,2364
f 1-49	PAANHANG	19	5,26	0,2294
f 50-99	PAANHANG	38	13,16	0,3426
f 100-199	PAANHANG	6	0	0
f 200-499	PAANHANG	1	0	0
f 500-999	PAANHANG	1	0	0
f 1000-4999	PAANHANG	0	0	0
f 5000-9999	PAANHANG	0	0	0
f 10000-19999	PAANHANG	0	0	0
f 20000-?	PAANHANG	0	0	0
f 0	PTRACTOR	5679	6,04	0,2382
f 1-49	PTRACTOR	0	0	0
f 50-99	PTRACTOR	0	0	0
f 100-199	PTRACTOR	79	2,53	0,1581
f 200-499	PTRACTOR	27	0	0
f 500-999	PTRACTOR	28	7,14	0,2623
f 1000-4999	PTRACTOR	9	11,11	0,3333
f 5000-9999	PTRACTOR	0	0	0
f 10000-19999	PTRACTOR	0	0	0

f 20000-?	PTRACTOR	0	0	0
f 0	PWERKT	5801	6	0,2375
f 1-49	PWERKT	0	0	0
f 50-99	PWERKT	4	0	0
f 100-199	PWERKT	6	0	0
f 200-499	PWERKT	8	0	0
f 500-999	PWERKT	0	0	0
f 1000-4999	PWERKT	3	0	0
f 5000-9999	PWERKT	0	0	0
f 10000-19999	PWERKT	0	0	0
f 20000-?	PWERKT	0	0	0
f 0	PBROM	5426	6,27	0,2424
f 1-49	PBROM	0	0	0
f 50-99	PBROM	34	2,94	0,1715
f 100-199	PBROM	282	2,13	0,1446
f 200-499	PBROM	63	0	0
f 500-999	PBROM	16	6,25	0,25
f 1000-4999	PBROM	1	0	0
f 5000-9999	PBROM	0	0	0
f 10000-19999	PBROM	0	0	0
f 20000-?	PBROM	0	0	0
f 0	PLEVEN	5529	5,88	0,2352
f 1-49	PLEVEN	9	0	0
f 50-99	PLEVEN	28	0	0
f 100-199	PLEVEN	84	7,14	0,2591
f 200-499	PLEVEN	94	11,7	0,3232
f 500-999	PLEVEN	35	11,43	0,3228
f 1000-4999	PLEVEN	38	5,26	0,2263
f 5000-9999	PLEVEN	3	0	0
f 10000-19999	PLEVEN	1	0	0
f 20000-?	PLEVEN	1	0	0
f 0	PPERSONG	5791	5,99	0,2374
f 1-49	PPERSONG	3	0	0
f 50-99	PPERSONG	18	5,56	0,2357
f 100-199	PPERSONG	4	0	0
f 200-499	PPERSONG	3	0	0
f 500-999	PPERSONG	1	0	0
f 1000-4999	PPERSONG	2	0	0
f 5000-9999	PPERSONG	0	0	0
f 10000-19999	PPERSONG	0	0	0
f 20000-?	PPERSONG	0	0	0
f 0	PGEZONG	5784	5,91	0,2359
f 1-49	PGEZONG	0	0	0

f 50-99	PGEZONG	25	8	0,2769
f 100-199	PGEZONG	13	30,77	0,4804
f 200-499	PGEZONG	0	0	0
f 500-999	PGEZONG	0	0	0
f 1000-4999	PGEZONG	0	0	0
f 5000-9999	PGEZONG	0	0	0
f 10000-19999	PGEZONG	0	0	0
f 20000-?	PGEZONG	0	0	0
f 0	PWAOREG	5799	5,93	0,2362
f 1-49	PWAOREG	0	0	0
f 50-99	PWAOREG	0	0	0
f 100-199	PWAOREG	0	0	0
f 200-499	PWAOREG	1	0	0
f 500-999	PWAOREG	1	0	0
f 1000-4999	PWAOREG	19	21,05	0,4189
f 5000-9999	PWAOREG	2	0	0
f 10000-19999	PWAOREG	0	0	0
f 20000-?	PWAOREG	0	0	0
f 0	PBRAND	2666	4,09	0,1981
f 1-49	PBRAND	161	1,86	0,1356
f 50-99	PBRAND	535	1,12	0,1054
f 100-199	PBRAND	920	7,39	0,2618
f 200-499	PBRAND	1226	12,32	0,3288
f 500-999	PBRAND	149	5,37	0,2262
f 1000-4999	PBRAND	155	1,94	0,1382
f 5000-9999	PBRAND	9	0	0
f 10000-19999	PBRAND	1	0	0
f 20000-?	PBRAND	0	0	0
f 0	PZEILPL	5819	5,96	0,2368
f 1-49	PZEILPL	2	50	0,7071
f 50-99	PZEILPL	0	0	0
f 100-199	PZEILPL	1	0	0
f 200-499	PZEILPL	0	0	0
f 500-999	PZEILPL	0	0	0
f 1000-4999	PZEILPL	0	0	0
f 5000-9999	PZEILPL	0	0	0
f 10000-19999	PZEILPL	0	0	0
f 20000-?	PZEILPL	0	0	0
f 0	PPLEZIER	5789	5,79	0,2335
f 1-49	PPLEZIER	5	60	0,5477
f 50-99	PPLEZIER	5	40	0,5477
f 100-199	PPLEZIER	5	40	0,5477
f 200-499	PPLEZIER	13	30,77	0,4804

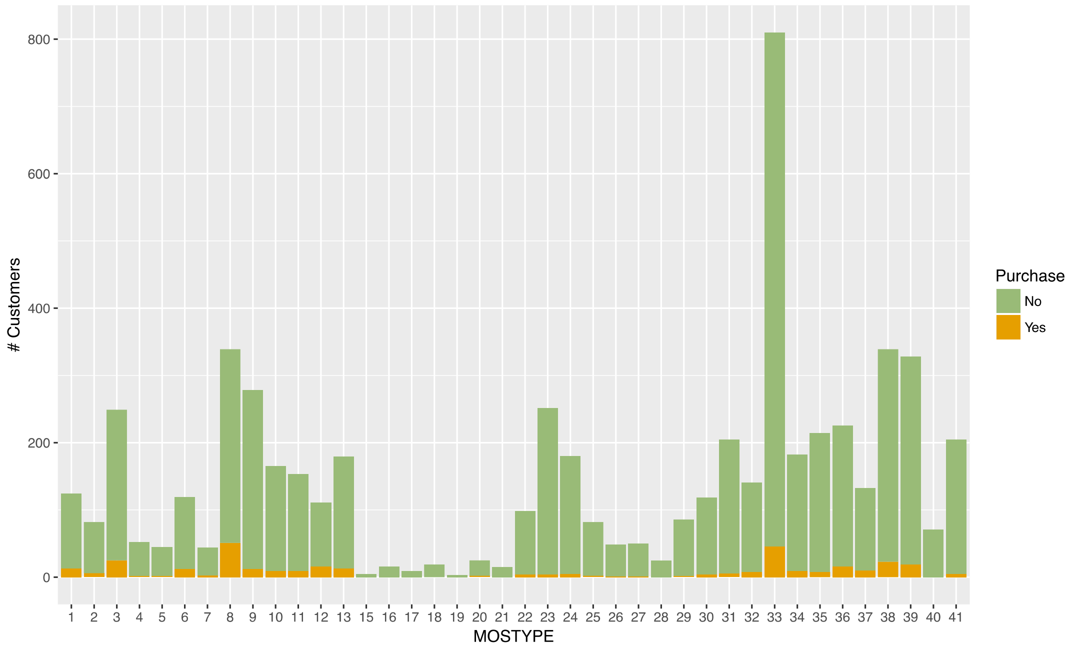
f 500-999	PPLEZIER	2	0	0
f 1000-4999	PPLEZIER	3	66,67	0,5774
f 5000-9999	PPLEZIER	0	0	0
f 10000-19999	PPLEZIER	0	0	0
f 20000-?	PPLEZIER	0	0	0
f 0	PFIETS	5675	5,87	0,235
f 1-49	PFIETS	147	10,2	0,3037
f 50-99	PFIETS	0	0	0
f 100-199	PFIETS	0	0	0
f 200-499	PFIETS	0	0	0
f 500-999	PFIETS	0	0	0
f 1000-4999	PFIETS	0	0	0
f 5000-9999	PFIETS	0	0	0
f 10000-19999	PFIETS	0	0	0
f 20000-?	PFIETS	0	0	0
f 0	PINBOED	5777	5,94	0,2363
f 1-49	PINBOED	18	16,67	0,3835
f 50-99	PINBOED	16	12,5	0,3416
f 100-199	PINBOED	6	0	0
f 200-499	PINBOED	3	0	0
f 500-999	PINBOED	1	0	0
f 1000-4999	PINBOED	1	0	0
f 5000-9999	PINBOED	0	0	0
f 10000-19999	PINBOED	0	0	0
f 20000-?	PINBOED	0	0	0
f 0	PBYSTAND	5740	5,78	0,2335
f 1-49	PBYSTAND	0	0	0
f 50-99	PBYSTAND	15	26,67	0,4577
f 100-199	PBYSTAND	22	18,18	0,3948
f 200-499	PBYSTAND	44	18,18	0,3902
f 500-999	PBYSTAND	1	0	0
f 1000-4999	PBYSTAND	0	0	0
f 5000-9999	PBYSTAND	0	0	0
f 10000-19999	PBYSTAND	0	0	0
f 20000-?	PBYSTAND	0	0	0
0	AWAPART	3482	4,22	0,2011
1	AWAPART	2334	8,61	0,2806
2	AWAPART	6	0	0
0	AWABEDR	5740	5,98	0,2371
1	AWABEDR	81	6,17	0,2422
5	AWABEDR	1	0	0
0	AWALAND	5702	6,05	0,2384
1	AWALAND	120	2,5	0,1568

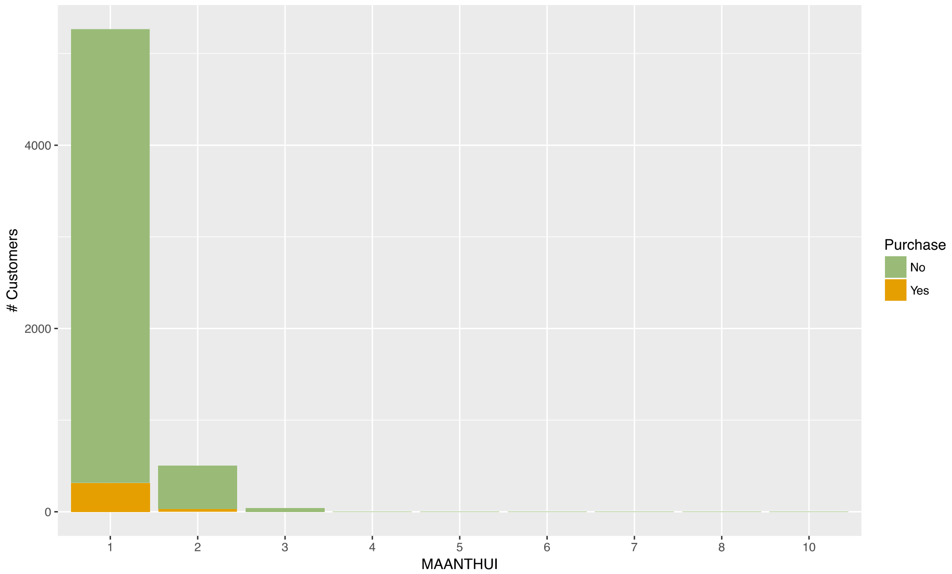
0	APERSAUT	2845	2,53	0,1571
1	APERSAUT	2712	8,74	0,2825
2	APERSAUT	246	15,45	0,3621
3	APERSAUT	12	8,33	0,2887
4	APERSAUT	5	0	0
6	APERSAUT	1	0	0
7	APERSAUT	1	0	0
0	ABESAUT	5774	5,99	0,2374
1	ABESAUT	40	5	0,2207
2	ABESAUT	4	0	0
3	ABESAUT	3	0	0
4	ABESAUT	1	0	0
0	AMOTSCO	5600	5,93	0,2362
1	AMOTSCO	211	7,11	0,2576
2	AMOTSCO	10	10	0,3162
8	AMOTSCO	1	0	0
0	AVRAAUT	5813	5,99	0,2373
1	AVRAAUT	6	0	0
2	AVRAAUT	2	0	0
3	AVRAAUT	1	0	0
0	AAANHANG	5757	5,94	0,2364
1	AAANHANG	59	10,17	0,3048
2	AAANHANG	4	0	0
3	AAANHANG	2	0	0
0	ATRACTOR	5679	6,04	0,2382
1	ATRACTOR	105	3,81	0,1923
2	ATRACTOR	29	3,45	0,1857
3	ATRACTOR	3	0	0
4	ATRACTOR	6	0	0
0	AWERKT	5801	6	0,2375
1	AWERKT	12	0	0
2	AWERKT	6	0	0
3	AWERKT	2	0	0
6	AWERKT	1	0	0
0	ABROM	5426	6,27	0,2424
1	ABROM	382	2,09	0,1434
2	ABROM	14	0	0
0	ALEVEN	5529	5,88	0,2352
1	ALEVEN	173	4,62	0,2106
2	ALEVEN	100	10	0,3015
3	ALEVEN	11	18,18	0,4045
4	ALEVEN	8	37,5	0,5175
8	ALEVEN	1	0	0

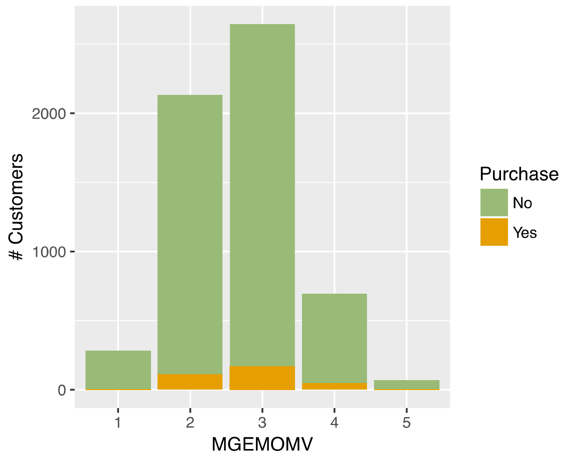
0	APERSONG	5791	5,99	0,2374
1	APERSONG	31	3,23	0,1796
0	AGEZONG	5784	5,91	0,2359
1	AGEZONG	38	15,79	0,3695
0	AWAOREG	5799	5,93	0,2362
1	AWAOREG	19	21,05	0,4189
2	AWAOREG	4	0	0
0	ABRAND	2666	4,09	0,1981
1	ABRAND	3017	7,69	0,2665
2	ABRAND	126	5,56	0,23
3	ABRAND	7	0	0
4	ABRAND	3	0	0
5	ABRAND	2	0	0
7	ABRAND	1	0	0
0	AZEILPL	5819	5,96	0,2368
1	AZEILPL	3	33,33	0,5774
0	APLEZIER	5789	5,79	0,2335
1	APLEZIER	31	38,71	0,4951
2	APLEZIER	2	50	0,7071
0	AFIETS	5675	5,87	0,235
1	AFIETS	111	9,01	0,2876
2	AFIETS	34	11,76	0,327
3	AFIETS	2	50	0,7071
0	AINBOED	5777	5,94	0,2363
1	AINBOED	44	11,36	0,321
2	AINBOED	1	0	0
0	ABYSTAND	5740	5,78	0,2335
1	ABYSTAND	81	19,75	0,4006
2	ABYSTAND	1	0	0

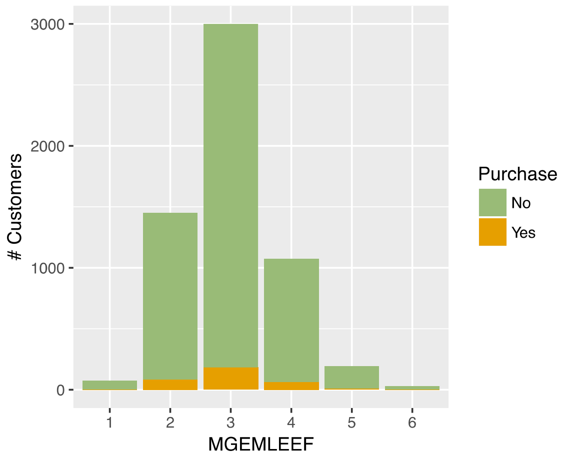
Table 8.3 – Analysis of the independent variables

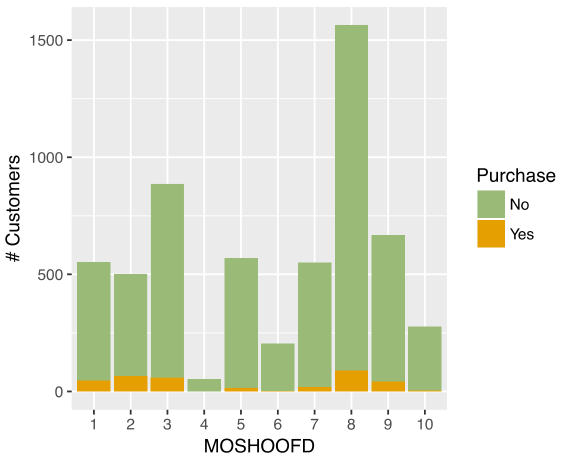
8.3. Histograms by Target Variable



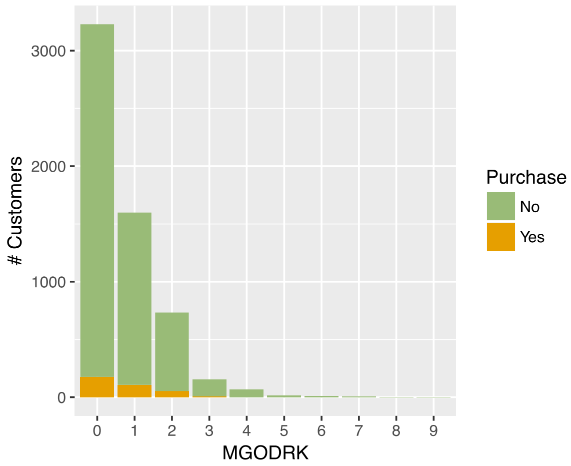




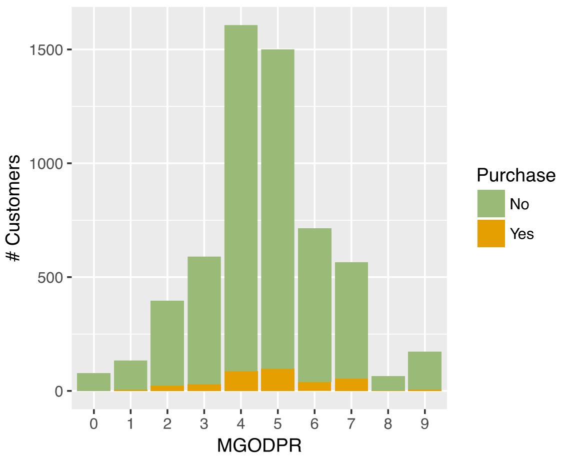




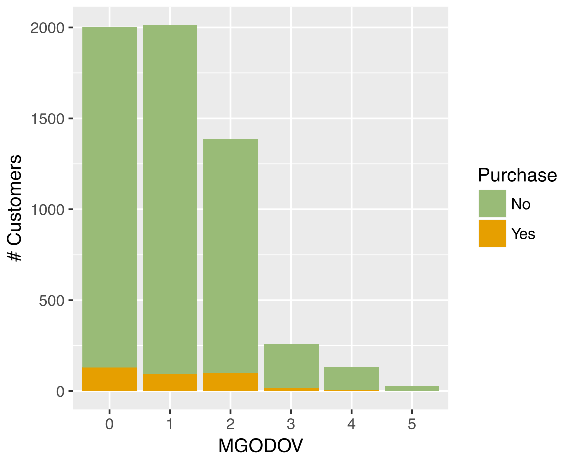
Customers
MOSHOOFD
Purchase
No
Yes



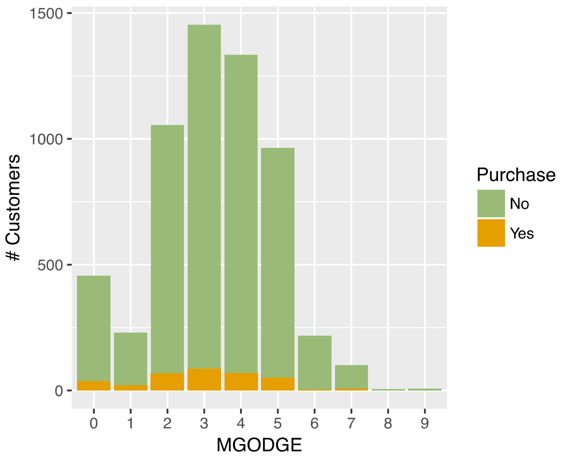
Customers
MGODRK
Purchase
No
Yes



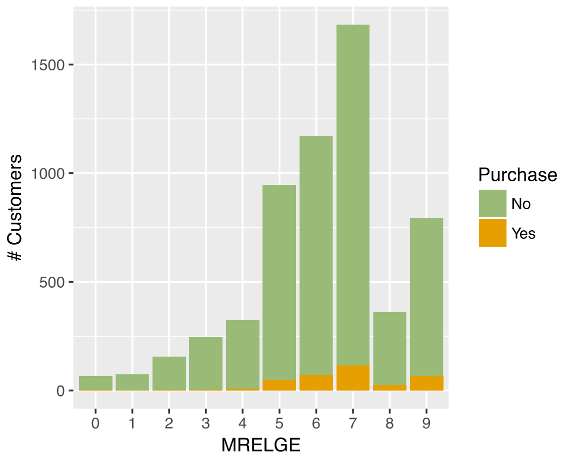
Customers
MGODPR
Purchase
No
Yes



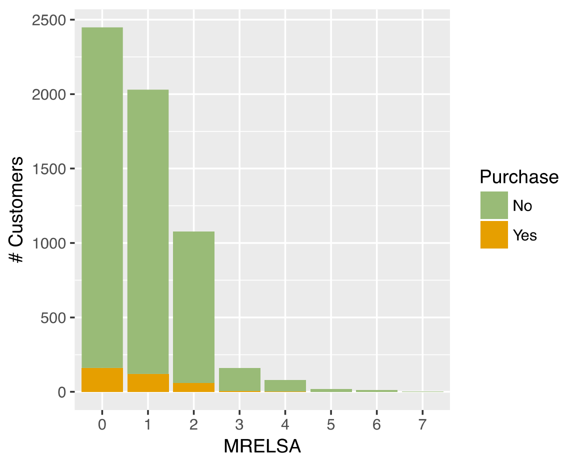
Customers
MGODOV
Purchase
No
Yes

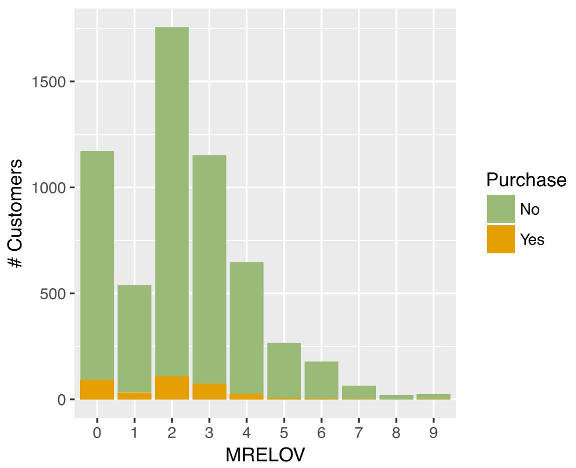


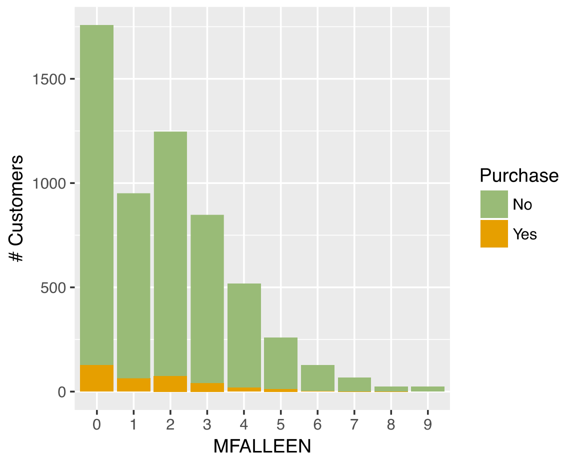
Customers
MGODGE
Purchase
No
Yes

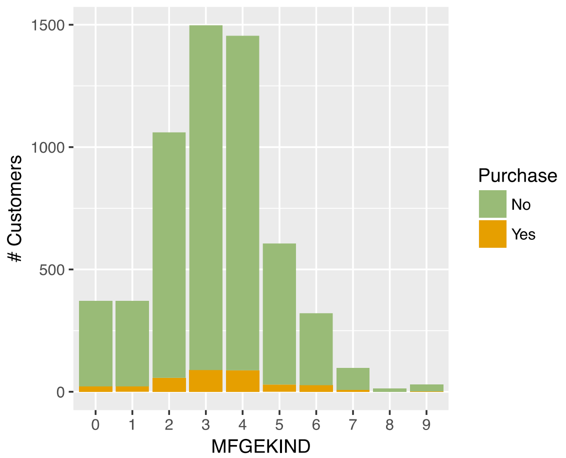


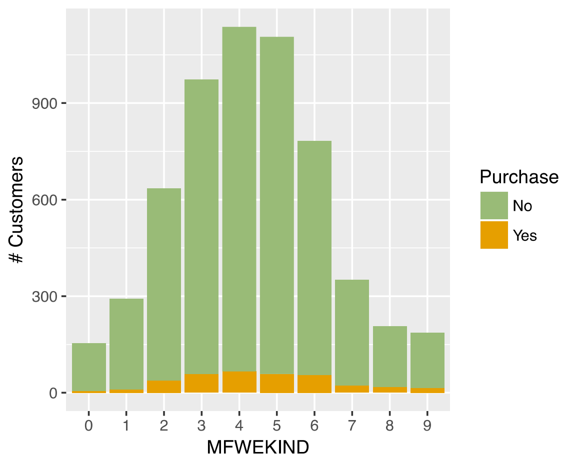
Customers
MRELGE
Purchase
No
Yes

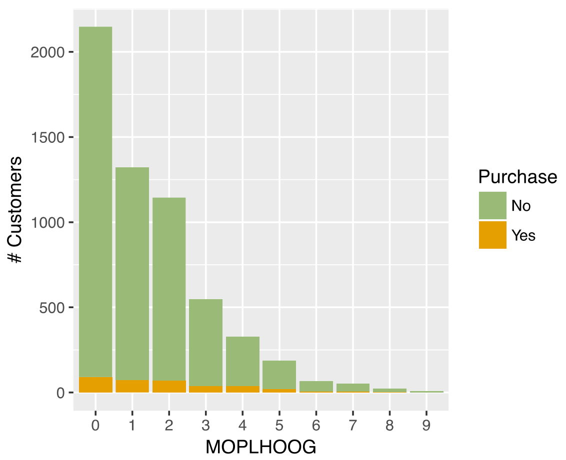

Customers
2500
2000
1500
1000
500
0
0 1 2 3 4 5 6 7
MRELSA
Purchase
No
Yes

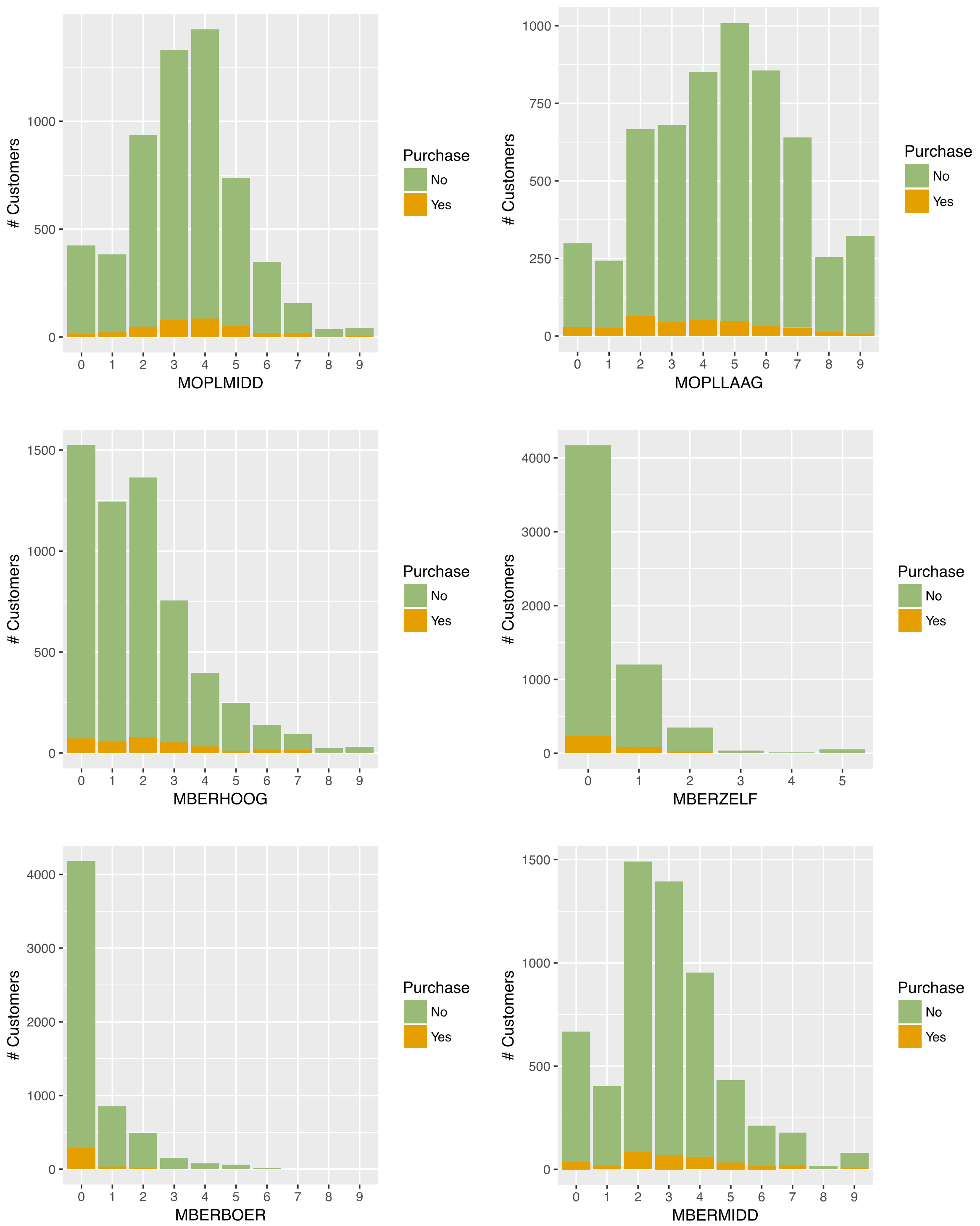

Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MRELOV
Purchase
No
Yes


Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MFALLEEN
Purchase
No
Yes

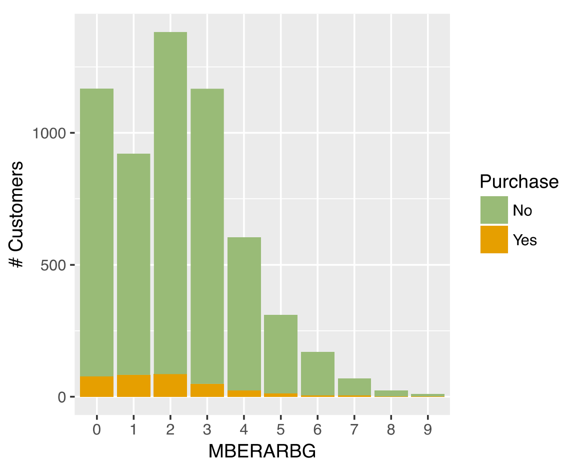

Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MFGEKIND
Purchase
No
Yes


Customers
900
600
300
0
0 1 2 3 4 5 6 7 8 9
MFWEKIND
Purchase
No
Yes

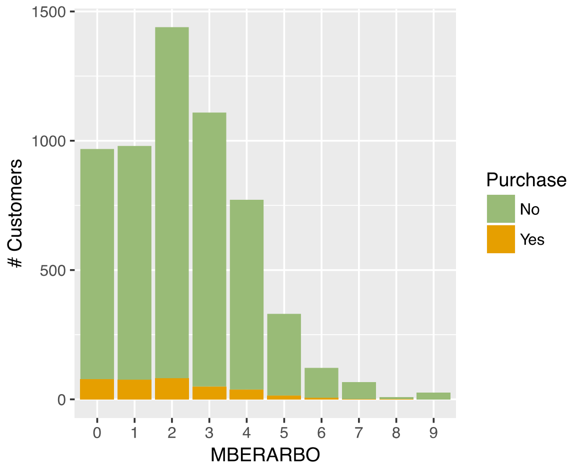

Customers
2000
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MOPLHOOG
Purchase
No
Yes



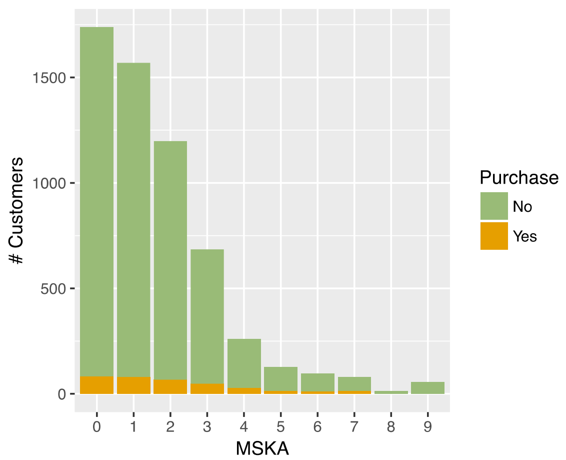
Customers
MOPLMIDD
Purchase
No
Yes
Customers
MOPLLAAG
Purchase
No
Yes
Customers
MBERHOOG
Purchase
No
Yes
Customers
MBERZELF
Purchase
No
Yes
Customers
MBERBOER
Purchase
No
Yes
Customers
MBERMIDD
Purchase
No
Yes



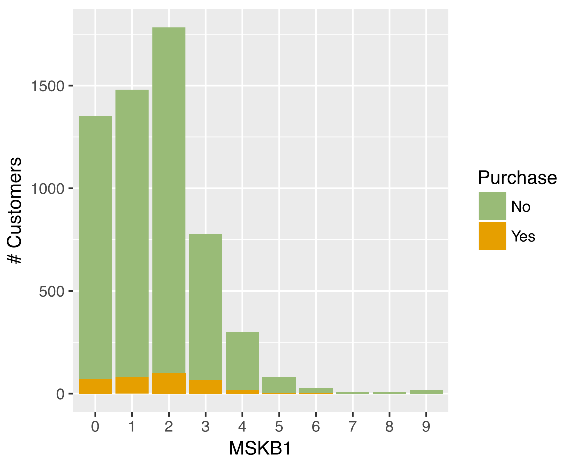
Customers
0
500
1000
0 1 2 3 4 5 6 7 8 9
MBERARBG
Purchase
No
Yes



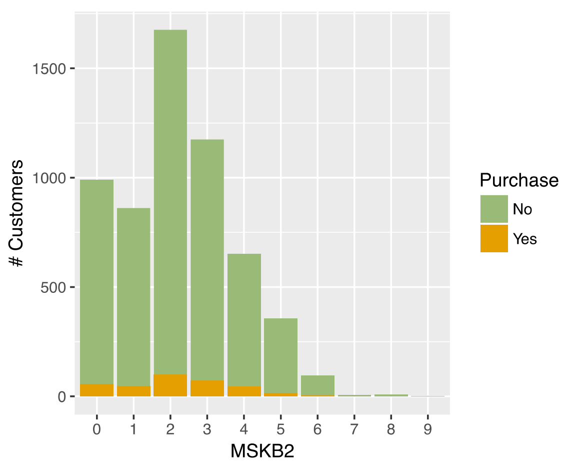
Customers
0
500
1000
1500
0 1 2 3 4 5 6 7 8 9
MBERARBO
Purchase
No
Yes



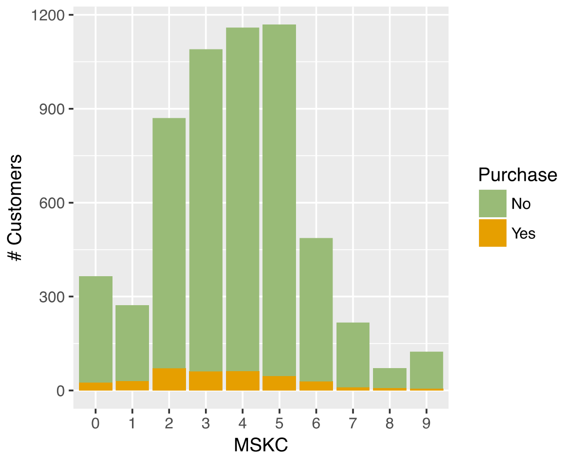
Customers
0
500
1000
1500
0 1 2 3 4 5 6 7 8 9
MSKA
Purchase
No
Yes



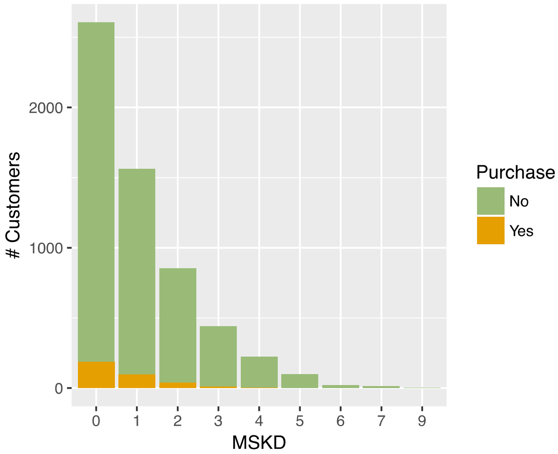
Customers
0
500
1000
1500
0 1 2 3 4 5 6 7 8 9
MSKB1
Purchase
No
Yes



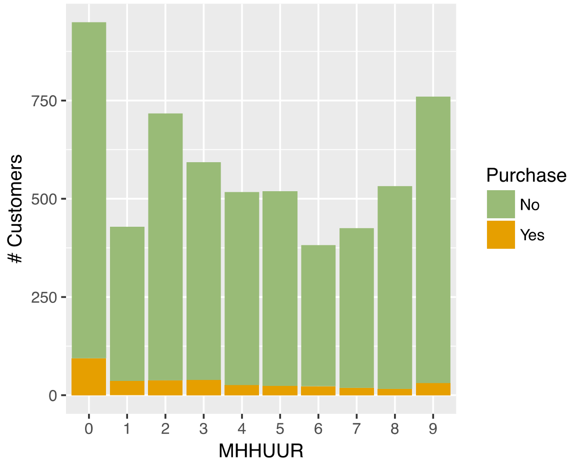
Customers
0
500
1000
1500
0 1 2 3 4 5 6 7 8 9
MSKB2
Purchase
No
Yes



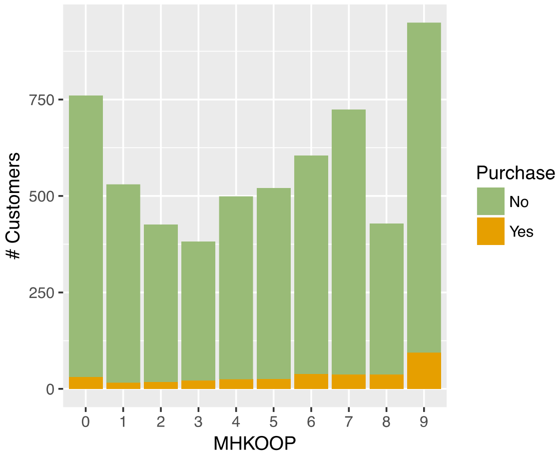
Customers
0
300
600
900
1200
0 1 2 3 4 5 6 7 8 9
MSKC
Purchase
No
Yes



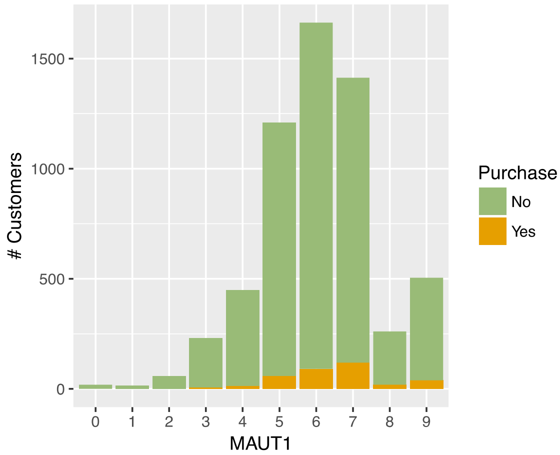
Customers
2000
1000
0
0 1 2 3 4 5 6 7 9
MSKD
Purchase
No
Yes



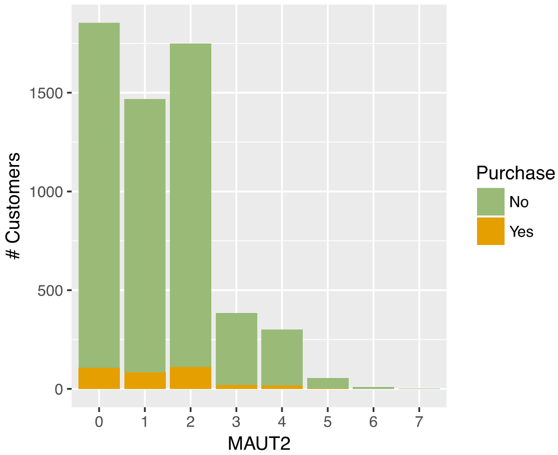
Customers
750
500
250
0
0 1 2 3 4 5 6 7 8 9
MHHUUR
Purchase
No
Yes



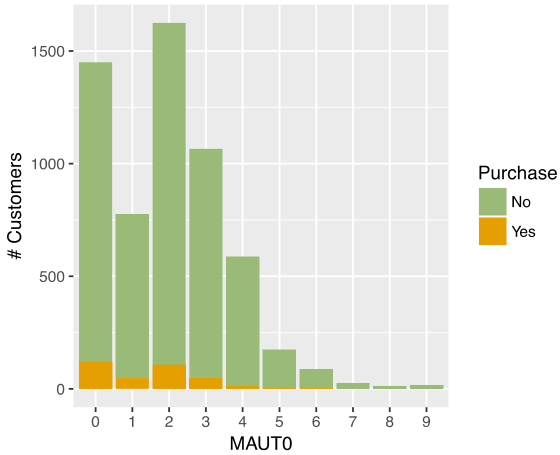
Customers
750
500
250
0
0 1 2 3 4 5 6 7 8 9
MHKOOP
Purchase
No
Yes



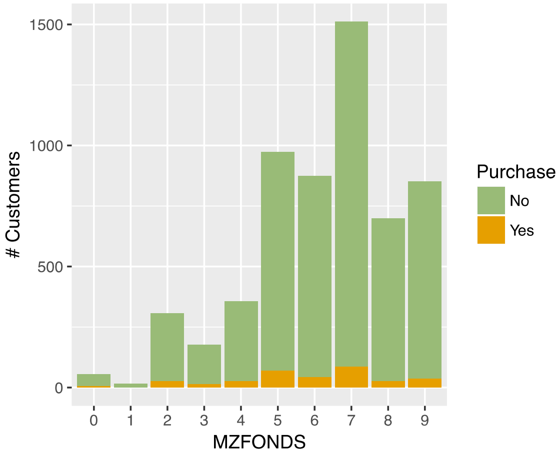
Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MAUT1
Purchase
No
Yes



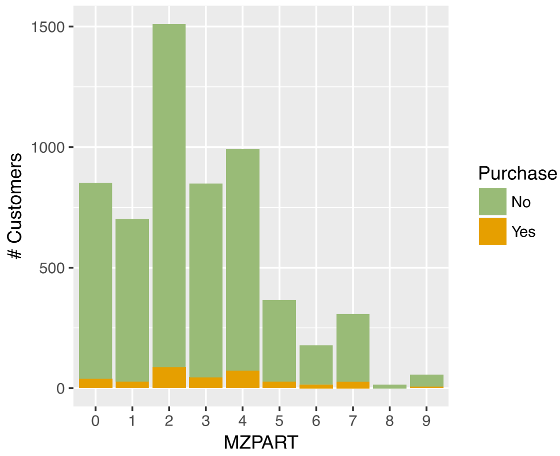
Customers
1500
1000
500
0
0 1 2 3 4 5 6 7
MAUT2
Purchase
No
Yes



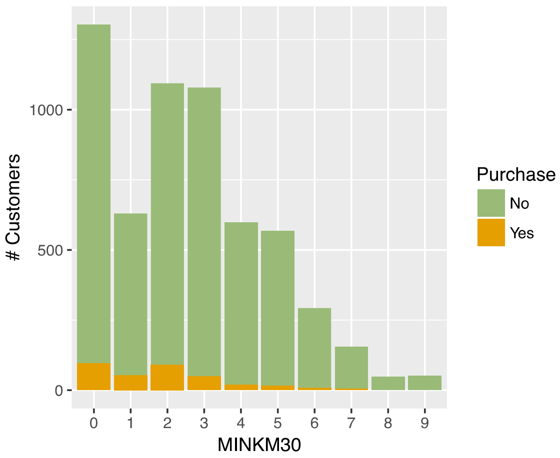
Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MAUT0
Purchase
No
Yes



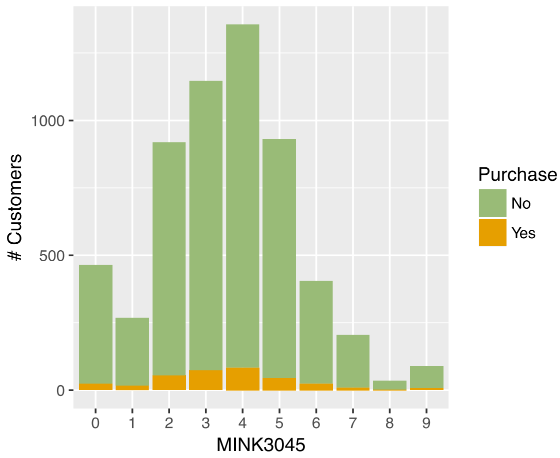
Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MZFONDS
Purchase
No
Yes



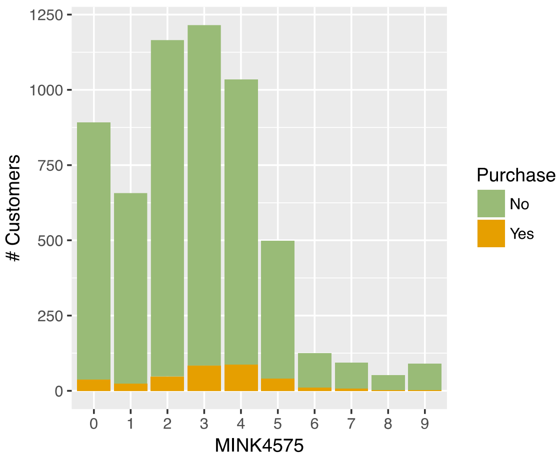
Customers
1500
1000
500
0
0 1 2 3 4 5 6 7 8 9
MZPART
Purchase
No
Yes



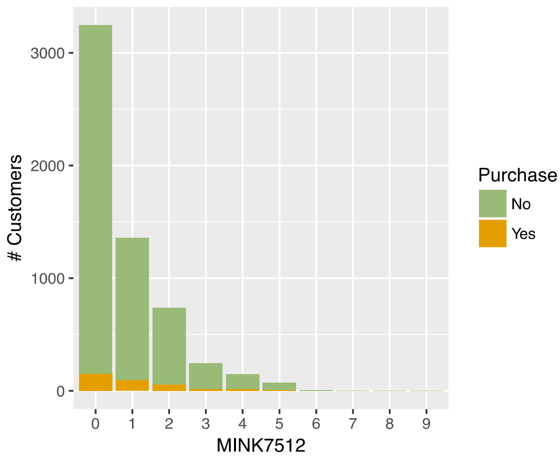
Customers
1000
500
0
0 1 2 3 4 5 6 7 8 9
MINKM30
Purchase
No
Yes



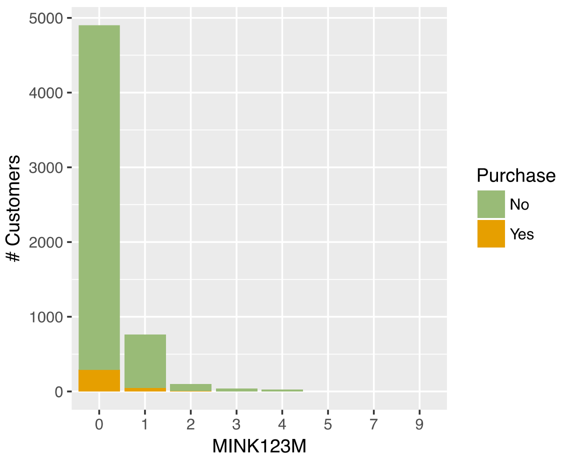
Customers
1000
500
0
0 1 2 3 4 5 6 7 8 9
MINK3045
Purchase
No
Yes



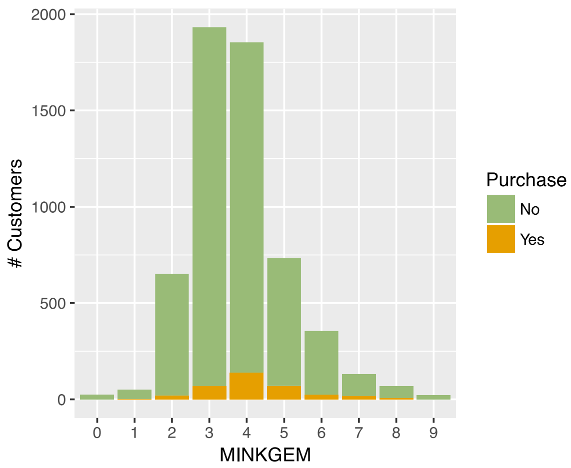
Customers
1250
1000
750
500
250
0
0 1 2 3 4 5 6 7 8 9
MINK4575
Purchase
No
Yes



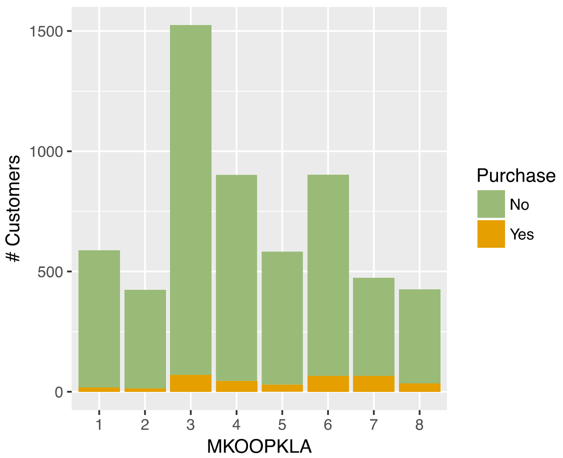
Customers
3000
2000
1000
0
0 1 2 3 4 5 6 7 8 9
MINK7512
Purchase
No
Yes



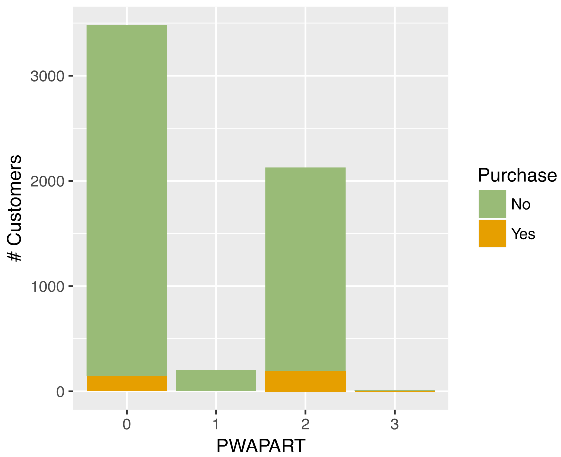
Customers
MINK123M
Purchase
No
Yes



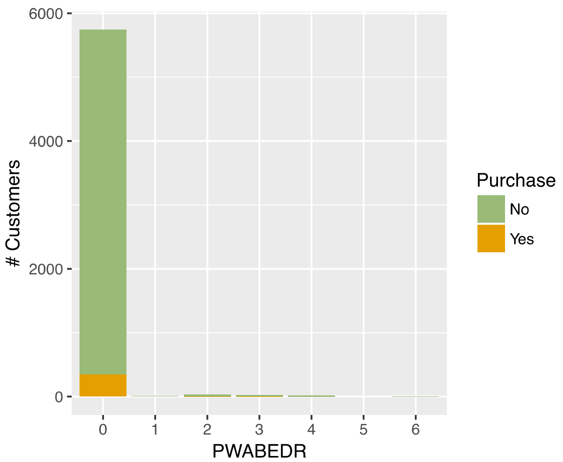
Customers
MINKGEM
Purchase
No
Yes



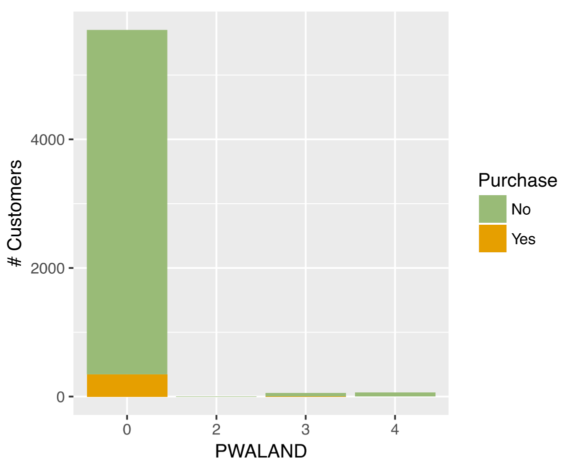
Customers
MKOOPKLA
Purchase
No
Yes



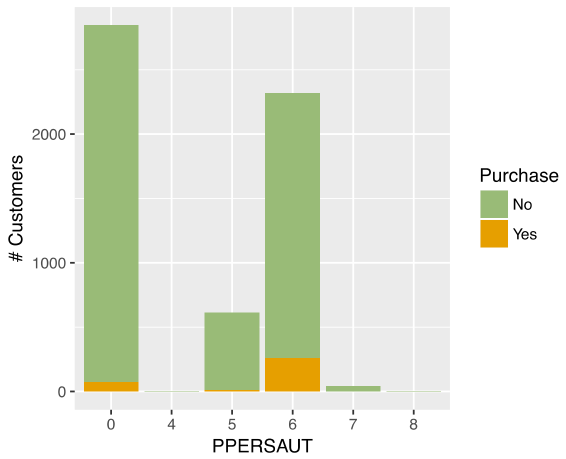
Customers
PWAPART
Purchase
No
Yes



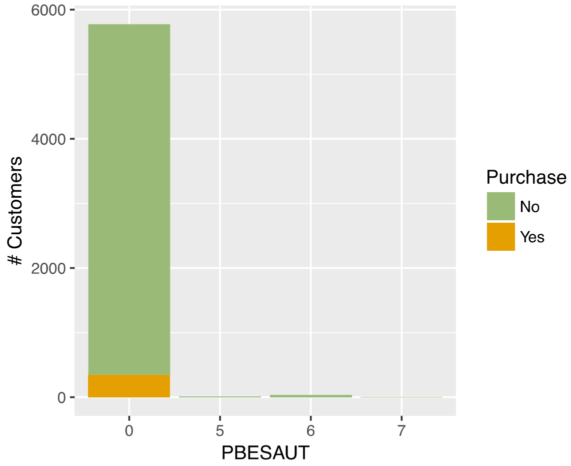
Customers
PWABEDR
Purchase
No
Yes



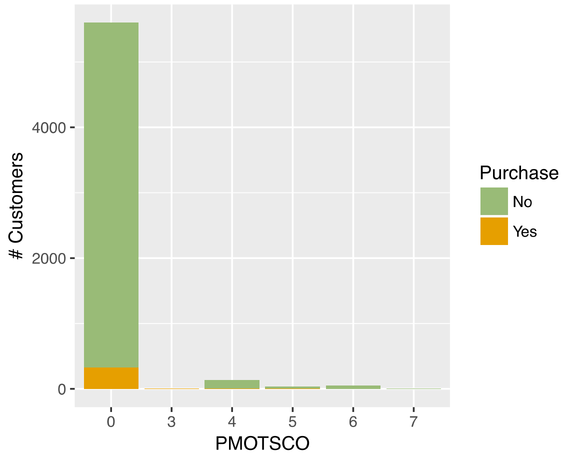
Customers
PWALAND
Purchase
No
Yes



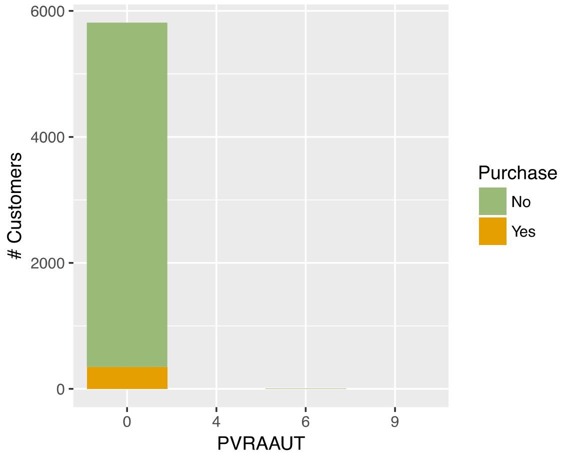
2000
1000
0
Customers
0
4
5
6
7
8
PPERSAUT
Purchase
No
Yes



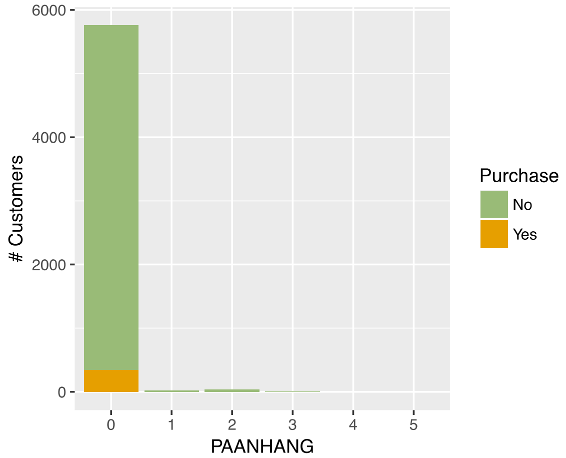
6000
4000
2000
0
Customers
0
5
6
7
PBESAUT
Purchase
No
Yes



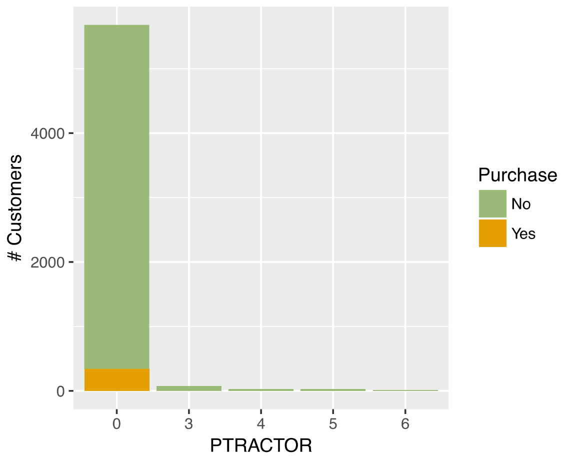
4000
2000
0
Customers
0
3
4
5
6
7
PMOTSCO
Purchase
No
Yes



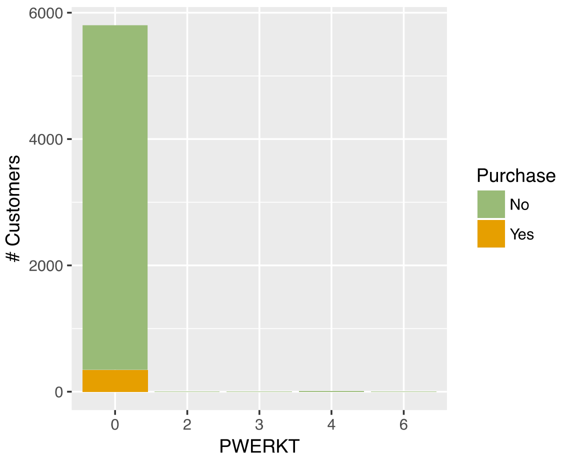
6000
4000
2000
0
Customers
0
4
6
9
PVRAAUT
Purchase
No
Yes



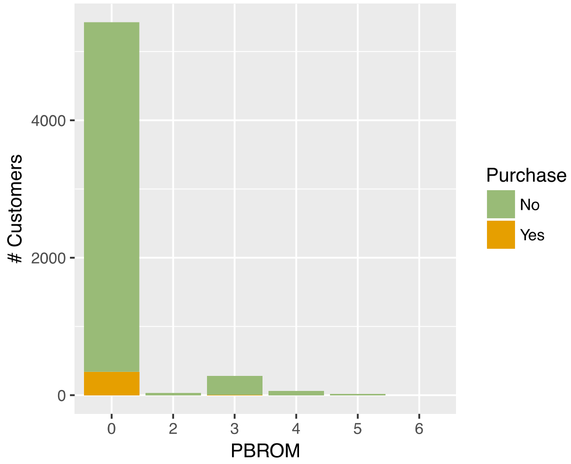
6000
4000
2000
0
Customers
0
1
2
3
4
5
PAANHANG
Purchase
No
Yes



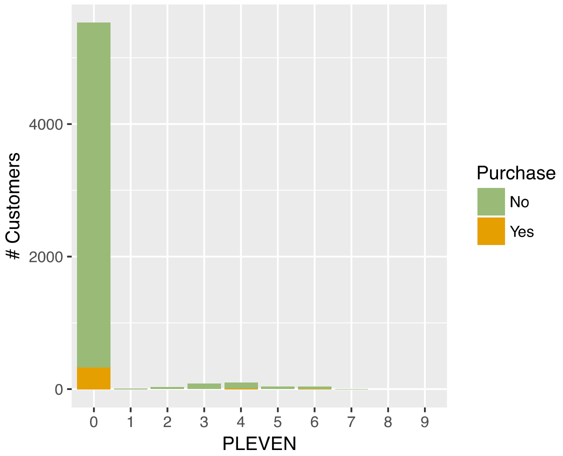
4000
2000
0
Customers
0
3
4
5
6
PTRACTOR
Purchase
No
Yes



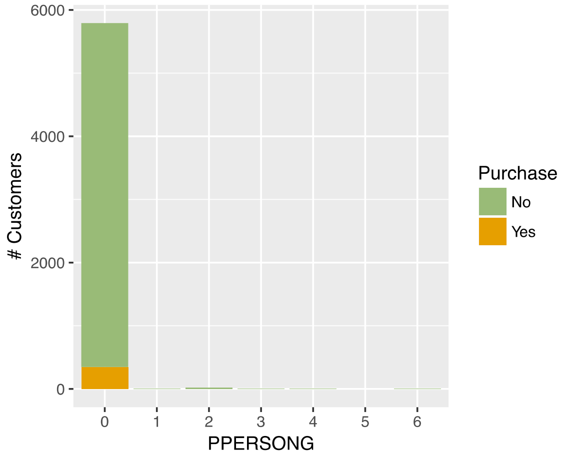
6000
4000
2000
0
Customers
0
2
3
4
6
PWERKT
Purchase
No
Yes



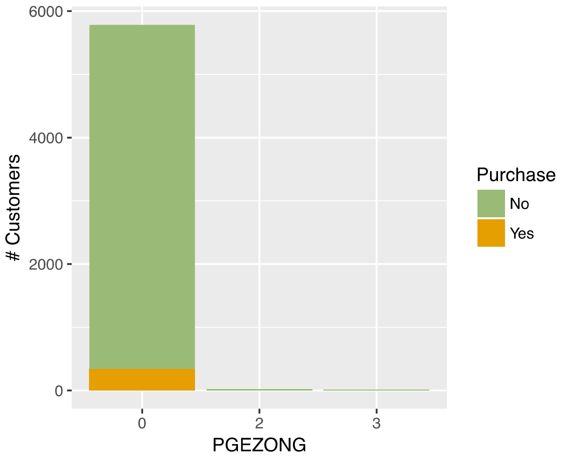
4000
2000
0
Customers
0
2
3
4
5
6
PBROM
Purchase
No
Yes



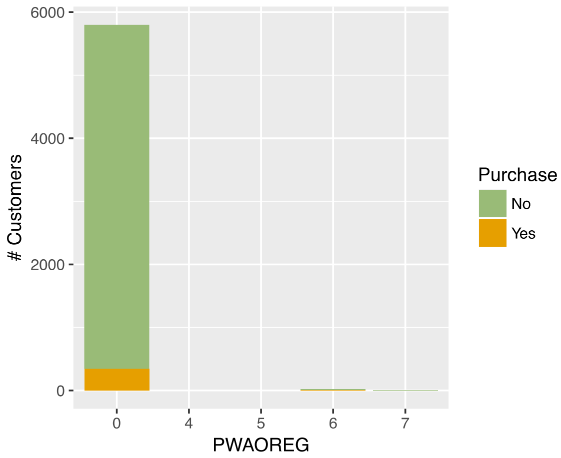
4000
2000
0
Customers
0
1
2
3
4
5
6
7
8
9
PLEVEN
Purchase
No
Yes



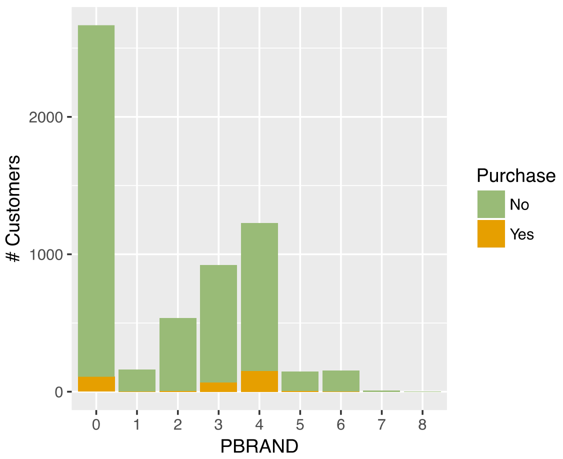
6000
4000
2000
0
Customers
0
1
2
3
4
5
6
PPERSONG
Purchase
No
Yes



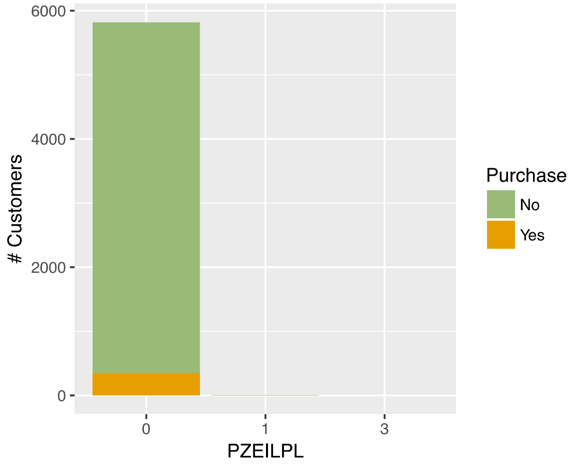
6000
4000
2000
0
Customers
0
2
3
PGEZONG
Purchase
No
Yes



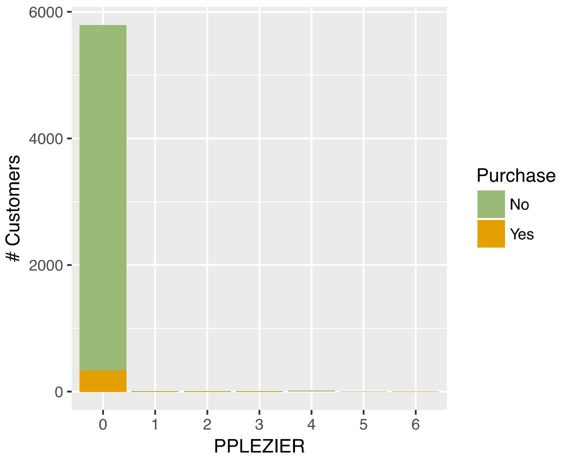
6000
4000
2000
0
Customers
0
4
5
6
7
PWAOREG
Purchase
No
Yes



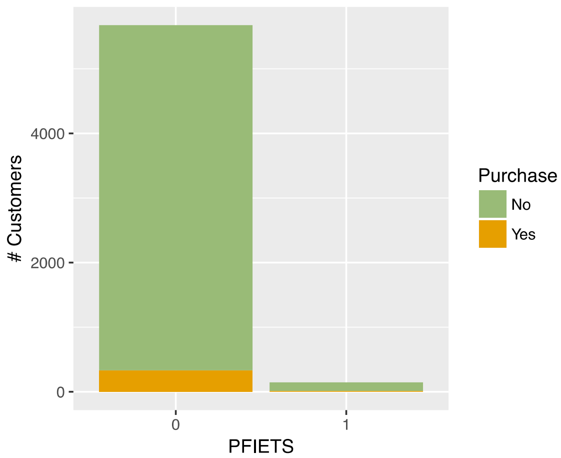
Customers
2000
1000
0
0 1 2 3 4 5 6 7 8
PBRAND
Purchase
No
Yes



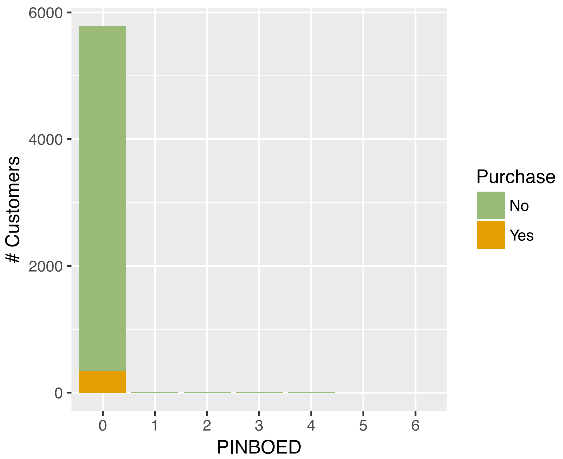
Customers
6000
4000
2000
0
0 1 3
PZEILPL
Purchase
No
Yes



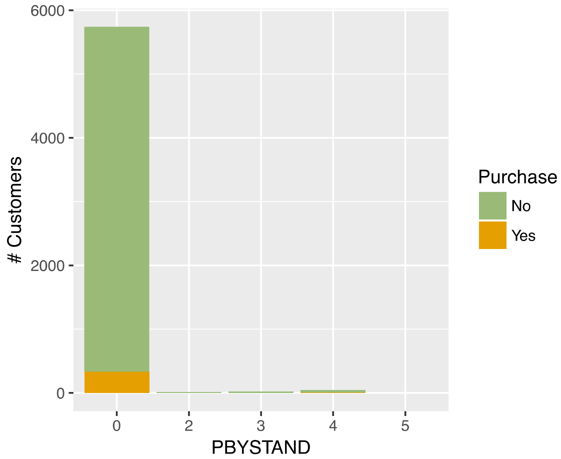
Customers
6000
4000
2000
0
0 1 2 3 4 5 6
PPLEZIER
Purchase
No
Yes



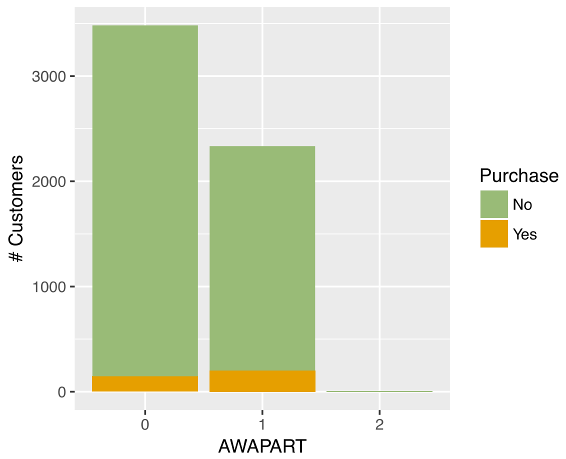
Customers
4000
2000
0
0 1
PFIETS
Purchase
No
Yes



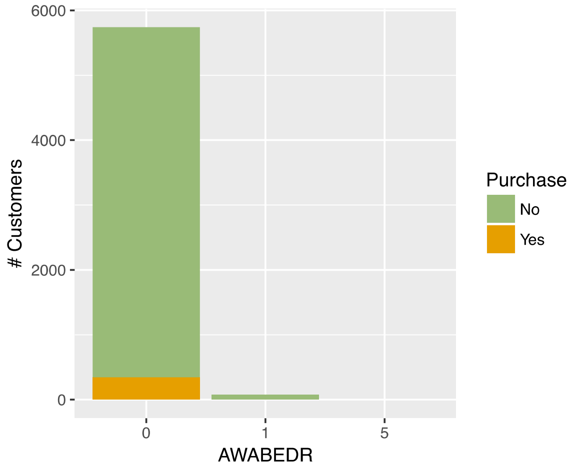
Customers
6000
4000
2000
0
0 1 2 3 4 5 6
PINBOED
Purchase
No
Yes



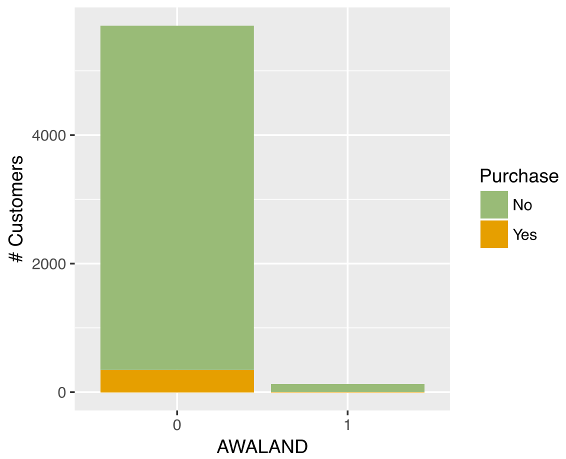
Customers
6000
4000
2000
0
0 2 3 4 5
PBYSTAND
Purchase
No
Yes



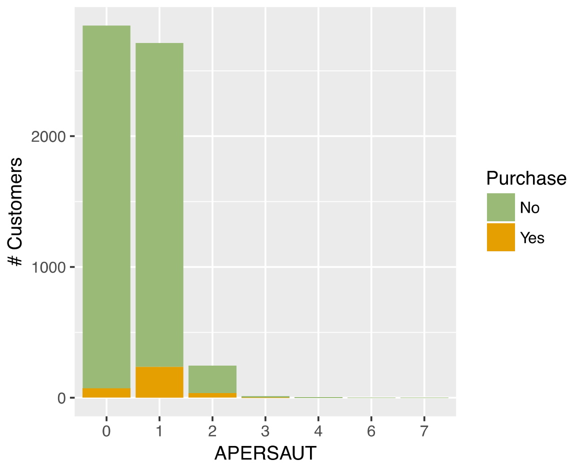
Customers
3000
2000
1000
0
0
1
2
AWAPART
Purchase
No
Yes



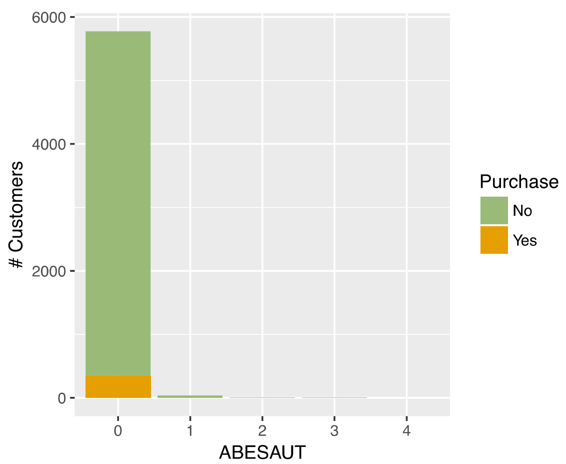
Customers
6000
4000
2000
0
0
1
5
AWABEDR
Purchase
No
Yes



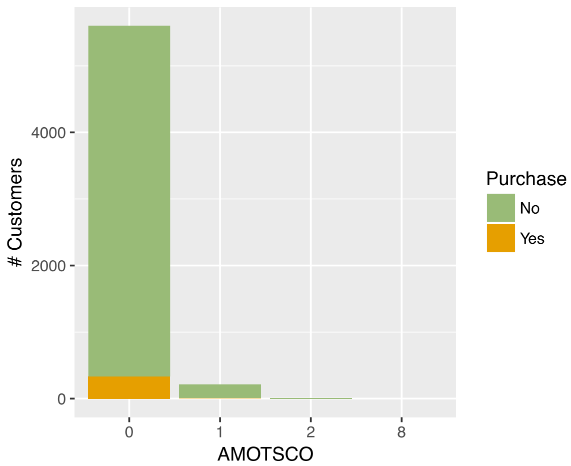
Customers
4000
2000
0
0
1
AWALAND
Purchase
No
Yes



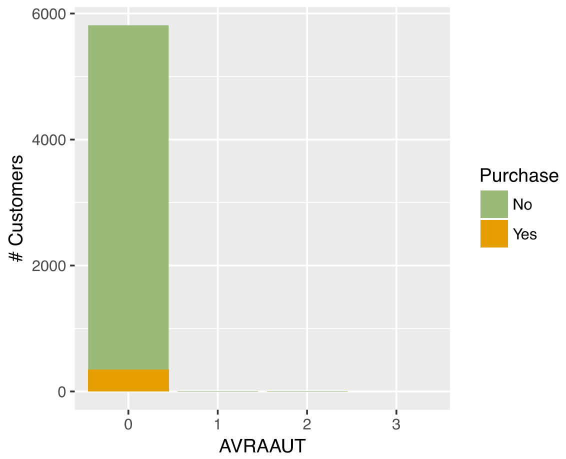
Customers
2000
1000
0
0
1
2
3
4
6
7
APERSAUT
Purchase
No
Yes



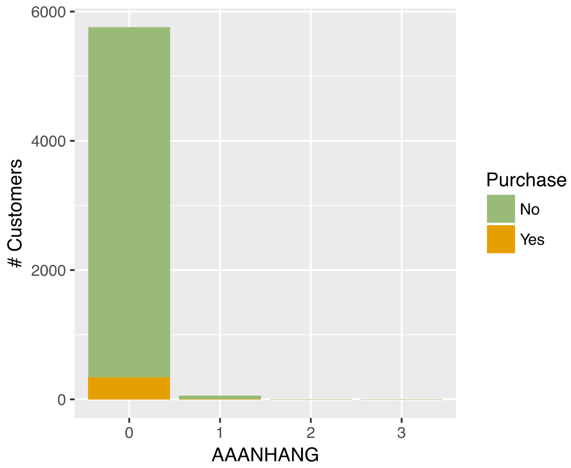
Customers
6000
4000
2000
0
0
1
2
3
4
ABESAUT
Purchase
No
Yes



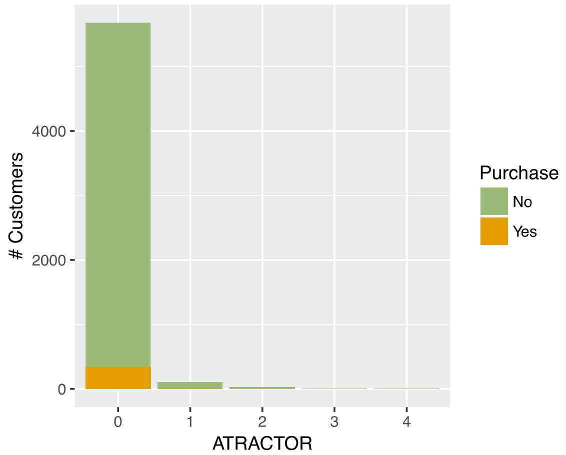
Customers
4000
2000
0
0
1
2
8
AMOTSCO
Purchase
No
Yes



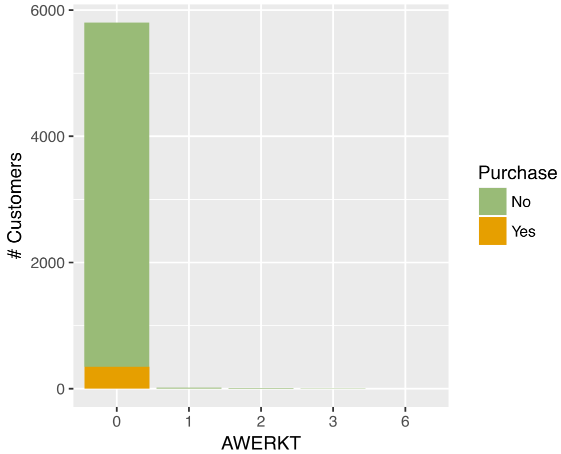
Customers
6000
4000
2000
0
0
1
2
3
AVRAAUT
Purchase
No
Yes



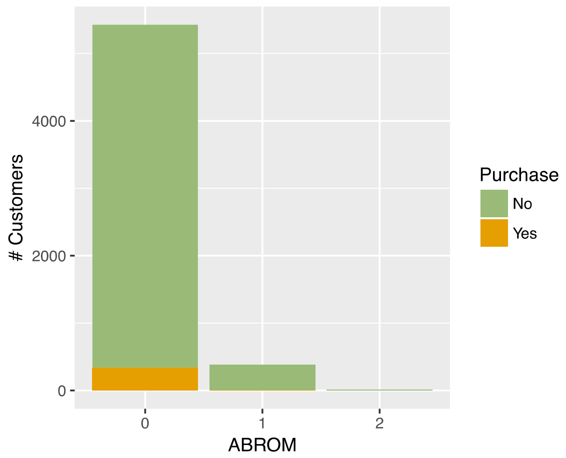
Customers
6000
4000
2000
0
0
1
2
3
AAANHANG
Purchase
No
Yes



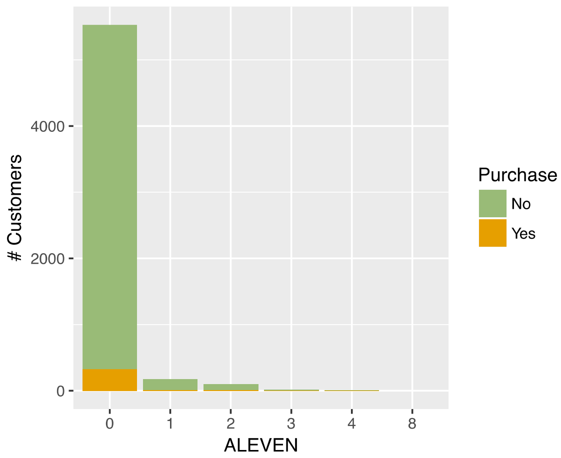
Customers
4000
2000
0
0
1
2
3
4
ATRACTOR
Purchase
No
Yes



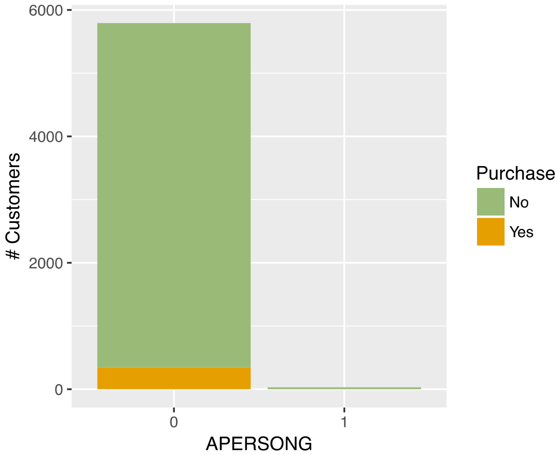
Customers
6000
4000
2000
0
0
1
2
3
6
AWERKT
Purchase
No
Yes

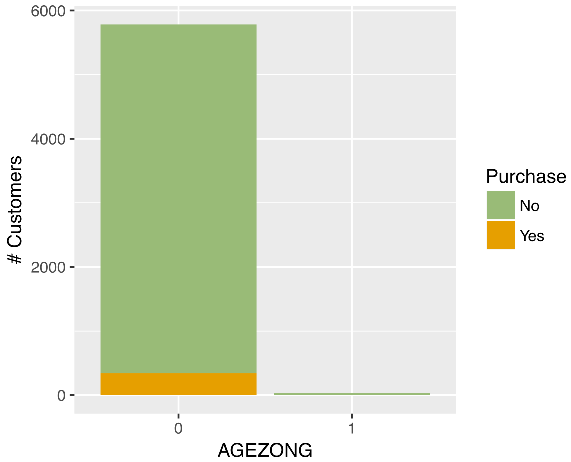


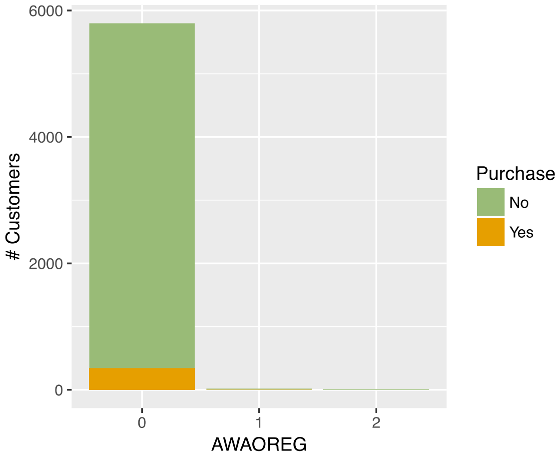
Customers
4000
2000
0
0
1
2
ABROM
Purchase
No
Yes

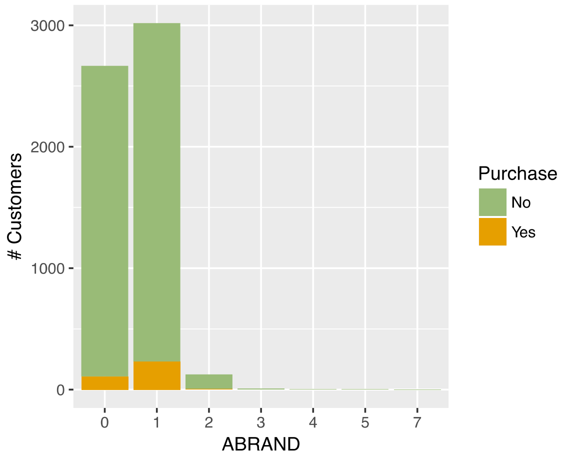


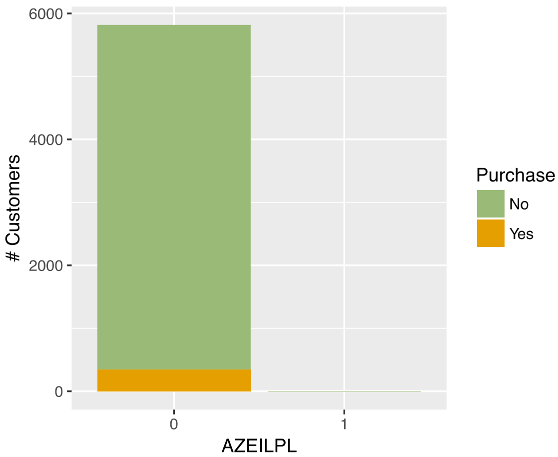
Customers
4000
2000
0
0
1
2
3
4
8
ALEVEN
Purchase
No
Yes

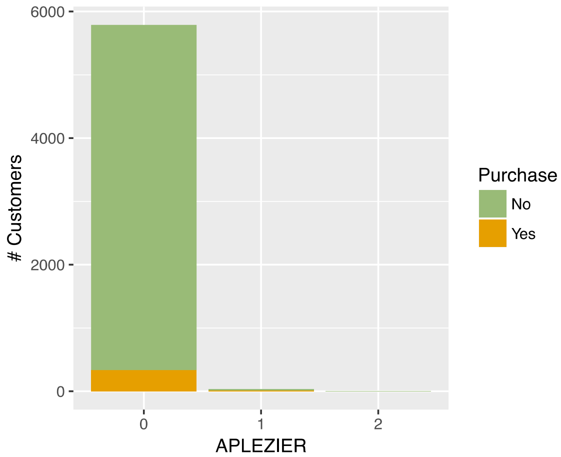

6000
4000
2000
0
Customers
0
1
APERSONG
Purchase
No
Yes

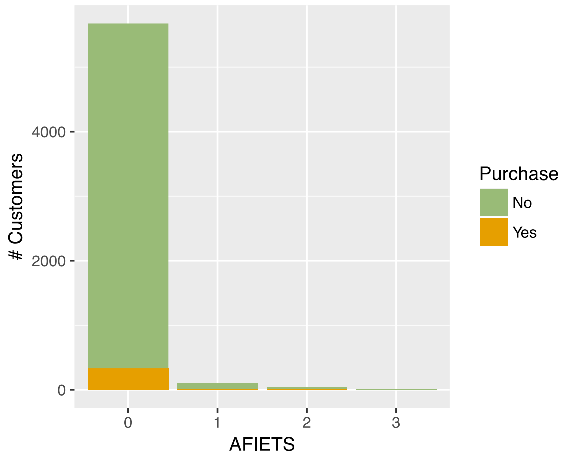

6000
4000
2000
0
Customers
0
1
AGEZONG
Purchase
No
Yes

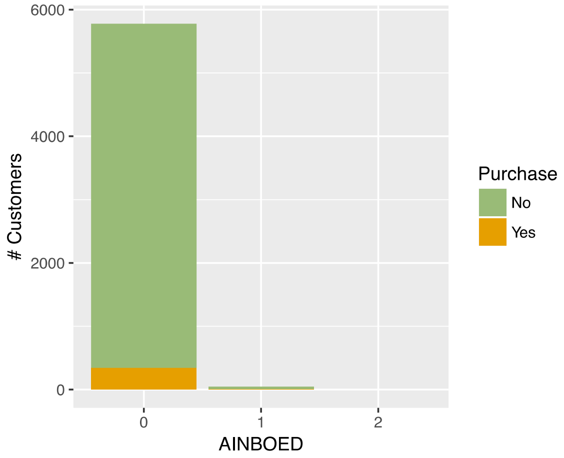

6000
4000
2000
0
Customers
0
1
2
AWAOREG
Purchase
No
Yes

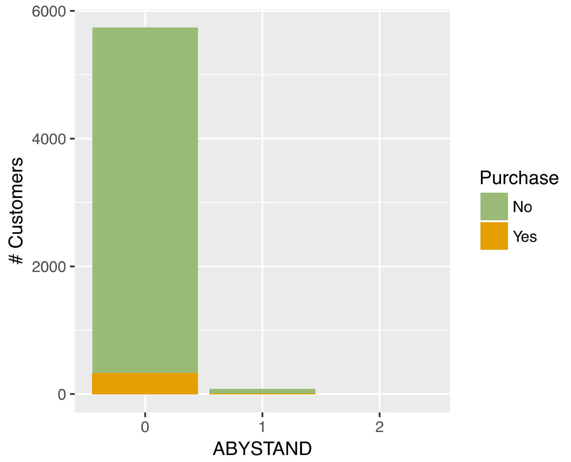

3000
2000
1000
0
Customers
0
1
2
3
4
5
7
ABRAND
Purchase
No
Yes


6000
4000
2000
0
Customers
0
1
AZEILPL
Purchase
No
Yes


6000
4000
2000
0
Customers
0
1
2
APLEZIER
Purchase
No
Yes







8.4. Spearman Correlation Between Predictor Variables and Target Variable

Variable	Spearman correlation with Target variable	Variable	Spearman correlation with Target variable
MOSTYPE	-0,058	PWAPART	0,095
MAANTHUI	-0,001	PWABEDR	0,001
MGEMOMV	0,035	PWALAND	-0,021
MGEMLEEF	0,004	PPERSAUT	0,164
MOSHOOFD	-0,062	PBESAUT	-0,007
MGODRK	0,02	PMOTSCO	0,01
MGODPR	0,035	PVRAAUT	-0,01
MGODOV	0,004	PAANHANG	0,015
MGODGE	-0,041	PTRACTOR	-0,016
MRELGE	0,068	PWERKT	-0,015
MRELSA	-0,027	PBROM	-0,045
MRELOV	-0,058	PLEVEN	0,019
MFALLEEN	-0,052	PPERSONG	-0,009

MFGEKIND	0,008	PGEZONG	0,034
MFWEKIND	0,029	PWAOREG	0,03
MOPLHOOG	0,078	PBRAND	0,101
MOPLMIDD	0,04	PZEILPL	0,026
MOPLLAAG	-0,091	PPLEZIER	0,106
MBERHOOG	0,057	PFIETS	0,029
MBERZELF	0,027	PINBOED	0,019
MBERBOER	-0,057	PBYSTAND	0,068
MBERMIDD	0,041	AWAPART	0,09
MBERARBG	-0,051	AWABEDR	0,001
MBERARBO	-0,057	AWALAND	-0,021
MSKA	0,06	APERSAUT	0,149
MSKB1	0,031	ABESAUT	-0,007
MSKB2	0,007	AMOTSCO	0,01
MSKC	-0,048	AVRAAUT	-0,01
MSKD	-0,061	AAANHANG	0,015
MHHUUR	-0,081	ATRACTOR	-0,017
MHKOOP	0,08	AWERKT	-0,015
MAUT1	0,073	ABROM	-0,045
MAUT2	0,008	ALEVEN	0,019
MAUT0	-0,077	APERSONG	-0,008
MZFONDS	-0,056	AGEZONG	0,034
MZPART	0,055	AWAOREG	0,03
MINKM30	-0,081	ABRAND	0,069
MINK3045	-0,006	AZEILPL	0,026
MINK4575	0,07	APLEZIER	0,106
MINK7512	0,063	AFIETS	0,029
MINK123M	0,007	AINBOED	0,019
MINKGEM	0,1	ABYSTAND	0,068
MKOOPKLA	0,095		

Table 8.4 – Spearman correlations between independent variables and dependent variable

8.5. Spearman Correlation Between Independent Variables

The data source contains 85 independent variables and to show exact Spearman correlation between all 85 independent variables the Spearman correlation matrix has to be split into several sections. The following table shows the place of each sub-section in the whole matrix.

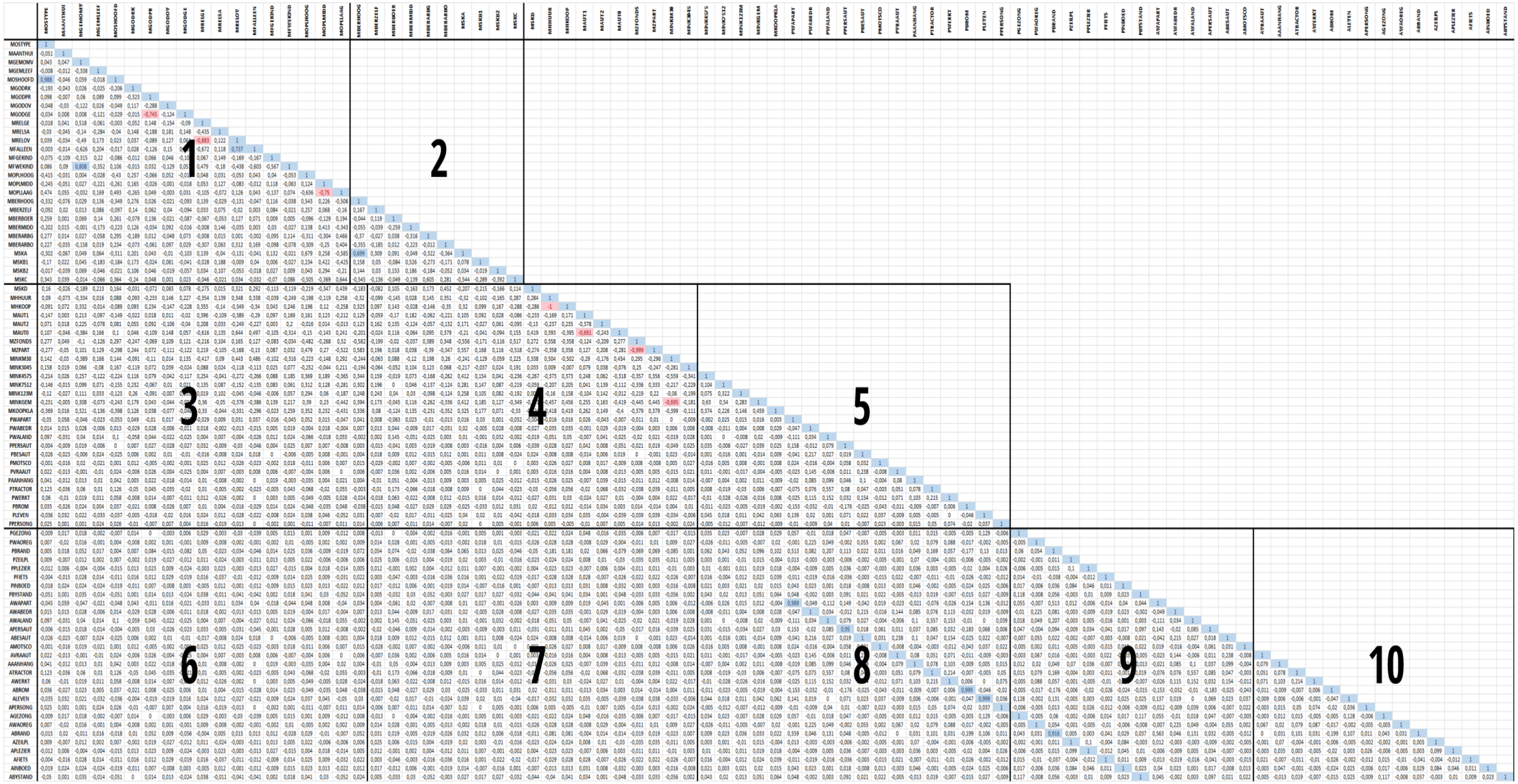


Table 8.5 – Subsections of Spearman's correlation matrix of all independent variables.

	MOSTYPE	MAANTHUI	MGEMOMV	MGEMLEEF	MOSHOOFD	MGODRK	MGODPR	MGODOV	MGODGE	MRELGE	MRELSA	MRELOV	MFALLEEN	MFGEKIND	MFWEKIND	MOPLHOOG	MOPLMIDD	MOPLLAAG
MOSTYPE	1																	
MAANTHUI	-0,051	1																
MGEMOMV	0,043	0,047	1															
MGEMLEEF	-0,008	-0,012	-0,338	1														
MOSHOOFD	0,988	-0,046	0,059	-0,018	1													
MGODRK	-0,193	-0,043	0,026	-0,025	-0,206	1												
MGODPR	0,098	-0,007	0,06	0,089	0,099	-0,323	1											
MGODOV	-0,048	-0,03	-0,122	0,026	-0,049	0,117	-0,288	1										
MGODGE	-0,034	0,008	0,008	-0,121	-0,029	-0,015	-0,745	-0,124	1									
MRELGE	-0,018	0,041	0,518	-0,061	-0,003	-0,052	0,148	-0,154	-0,09	1								
MRELSA	-0,03	-0,045	-0,14	-0,284	-0,04	0,148	-0,188	0,181	0,148	-0,435	1							
MRELOV	0,039	-0,034	-0,49	0,173	0,023	0,037	-0,089	0,127	0,061	-0,883	0,122	1						
MFALLEEN	-0,003	-0,014	-0,626	0,204	-0,017	0,028	-0,126	0,15	0,058	-0,672	0,118	0,737	1					
MFGEKIND	-0,075	-0,109	-0,315	0,22	-0,086	-0,012	0,066	0,046	-0,107	0,067	0,149	-0,169	-0,167	1				
MFWEKIND	0,086	0,09	0,808	-0,352	0,106	-0,015	0,032	-0,129	0,051	0,479	-0,18	-0,438	-0,603	-0,567	1			
MOPLHOOG	-0,415	-0,031	0,004	-0,028	-0,43	0,257	-0,066	0,052	-0,013	0,048	0,031	-0,053	0,043	0,04	-0,053	1		
MOPLMIDD	-0,245	-0,051	0,027	-0,221	-0,261	0,165	-0,026	-0,001	-0,018	0,053	0,127	-0,083	-0,012	0,118	-0,063	0,124	1	
MOPLLAAG	0,474	0,055	-0,032	0,169	0,493	-0,265	0,049	-0,003	0,031	-0,105	-0,072	0,126	0,043	-0,137	0,074	-0,636	-0,75	1
MBERHOOG	-0,332	-0,076	0,029	0,136	-0,349	0,276	0,026	-0,021	-0,093	0,139	-0,029	-0,131	-0,047	0,116	-0,038	0,543	0,226	-0,506
MBERZELF	-0,092	0,02	0,013	0,086	-0,097	0,14	0,062	0,04	-0,094	0,033	0,075	-0,02	0,003	0,084	-0,021	0,257	0,068	-0,16
MBERBOER	0,259	0,001	0,069	0,14	0,261	-0,079	0,136	-0,021	-0,087	-0,067	-0,053	0,127	0,071	0,009	0,005	-0,096	-0,129	0,194
MBERMIDD	-0,202	0,015	-0,001	-0,173	-0,223	0,126	-0,034	0,092	-0,016	-0,008	0,146	-0,035	0,003	0,03	-0,027	0,138	0,413	-0,343
MBERARBG	0,277	0,014	0,027	-0,058	0,295	-0,189	0,012	-0,048	0,073	-0,008	0,015	0,001	-0,002	-0,095	0,114	-0,311	-0,304	0,466
MBERARBO	0,227	-0,035	-0,158	0,019	0,234	-0,073	-0,061	0,097	0,029	-0,307	0,063	0,312	0,169	-0,098	-0,078	-0,309	-0,25	0,404
MSKA	-0,302	-0,067	0,049	0,064	-0,311	0,201	0,043	-0,01	-0,103	0,139	-0,04	-0,131	-0,041	0,132	-0,021	0,679	0,258	-0,585
MSKB1	-0,17	0,022	0,045	-0,183	-0,184	0,173	-0,024	0,081	-0,041	-0,028	0,188	-0,009	0,04	0,006	-0,027	0,234	0,422	-0,425
MSKB2	-0,017	-0,039	0,069	-0,046	-0,021	0,106	0,046	-0,019	-0,057	0,034	0,107	-0,053	-0,018	0,027	0,009	0,043	0,294	-0,21
MSKC	0,343	0,039	-0,014	-0,066	0,364	-0,24	0,048	0,001	0,023	-0,046	-0,021	0,034	-0,032	-0,07	0,086	-0,505	-0,369	0,644

Table 8.6 – 1st subsection of the Spearman correlation matrix

	MBERHOOG	MBERZELF	MBERBOER	MBERMIDD	MBERARBG	MBERARBO	MSKA	MSKB1	MSKB2	MSKC
MOSTYPE										
MAANTHUI										
MGEMOMV										
MGEMLEEF										
MOSHOOFD										
MGODRK										
MGODPR										
MGODOV										
MGODGE										
MRELGE										
MRELSA										
MRELOV										
MFALLEEN										
MFGEKIND										
MFWEKIND										
MOPLHOOG										
MOPLMIDD										
MOPLLAAG										
MBERHOOG	1									
MBERZELF	0,167	1								
MBERBOER	-0,044	0,119	1							
MBERMIDD	-0,055	-0,039	-0,259	1						
MBERARBG	-0,37	-0,027	0,038	-0,316	1					
MBERARBO	-0,355	-0,185	0,012	-0,223	-0,012	1				
MSKA	0,699	0,309	0,091	-0,049	-0,322	-0,364	1			
MSKB1	0,158	0,05	-0,084	0,526	-0,273	-0,171	0,078	1		
MSKB2	0,144	0,03	0,153	0,186	-0,184	-0,052	0,034	-0,019	1	
MSKC	-0,545	-0,136	-0,049	-0,139	0,605	0,281	-0,544	-0,289	-0,392	1

Table 8.7 – 2nd subsection of the Spearman correlation matrix

	MOSTYPE	MAANTHUI	MGEMOMV	MGEMLEEF	MOSHOOFD	MGODRK	MGODPR	MGODOV	MGODGE	MRELGE	MRELSA	MRELOV	MFALLEEN	MFGEKIND	MFWEKIND	MOPLHOOG	MOPLMIDD	MOPLLAAG	MBERHOOG
MSKD	0,16	-0,026	-0,189	0,213	0,164	-0,031	-0,072	0,083	0,078	-0,275	0,015	0,321	0,292	-0,113	-0,119	-0,219	-0,347	0,439	-0,183
MHHUUR	0,09	-0,073	-0,334	0,016	0,088	-0,093	-0,233	0,146	0,227	-0,354	0,139	0,348	0,338	-0,039	-0,249	-0,198	-0,119	0,258	-0,32
MHKOOP	-0,091	0,072	0,332	-0,014	-0,089	0,093	0,234	-0,147	-0,228	0,355	-0,14	-0,349	-0,34	0,043	0,246	0,196	0,12	-0,258	0,323
MAUT1	-0,147	0,003	0,213	-0,097	-0,149	-0,022	0,018	0,011	-0,02	0,396	-0,109	-0,389	-0,29	0,097	0,169	0,161	0,123	-0,212	0,129
MAUT2	0,071	0,018	0,225	-0,078	0,081	0,055	0,092	-0,106	-0,04	0,208	0,033	-0,249	-0,227	0,003	0,2	-0,016	0,014	-0,013	0,123
MAUT0	0,107	-0,046	-0,384	0,166	0,1	0,046	-0,109	0,148	0,057	-0,616	0,135	0,644	0,497	-0,105	-0,314	-0,15	-0,143	0,241	-0,201
MZFONDS	0,277	0,049	-0,1	-0,126	0,297	-0,247	-0,069	0,109	0,121	-0,216	0,104	0,165	0,127	-0,083	-0,034	-0,482	-0,268	0,52	-0,582
MZPART	-0,277	-0,05	0,101	0,129	-0,298	0,244	0,072	-0,111	-0,122	0,219	-0,105	-0,168	-0,13	0,087	0,032	0,479	0,27	-0,522	0,583
MINKM30	0,142	-0,03	-0,389	0,166	0,144	-0,091	-0,11	0,014	0,135	-0,417	0,09	0,443	0,486	-0,102	-0,316	-0,223	-0,148	0,292	-0,244
MINK3045	0,158	0,019	0,066	-0,08	0,167	-0,119	0,072	0,039	-0,024	0,088	0,024	-0,118	-0,113	0,025	0,077	-0,252	-0,044	0,211	-0,194
MINK4575	-0,214	0,026	0,257	-0,122	-0,224	0,116	0,079	-0,042	-0,117	0,254	-0,041	-0,272	-0,266	0,088	0,185	0,369	0,189	-0,365	0,344
MINK7512	-0,146	-0,015	0,099	0,071	-0,155	0,232	-0,067	0,01	0,021	0,135	0,087	-0,152	-0,135	0,083	0,061	0,312	0,128	-0,281	0,302
MINK123M	-0,12	-0,027	0,111	0,033	-0,123	0,26	-0,091	-0,007	0,053	0,019	0,102	-0,045	-0,046	-0,006	0,057	0,294	0,06	-0,187	0,248
MINKGEM	-0,231	-0,005	0,308	-0,073	-0,243	0,179	0,043	-0,044	-0,091	0,36	-0,05	-0,376	-0,386	0,139	0,217	0,39	0,23	-0,442	0,394
MKOOPKLA	-0,369	0,016	0,321	-0,136	-0,398	0,126	0,038	-0,077	-0,048	0,33	-0,044	-0,331	-0,296	-0,023	0,259	0,352	0,232	-0,431	0,336
PWAPART	-0,05	0,058	-0,046	-0,023	-0,053	0,049	-0,01	0,017	-0,023	-0,029	0,009	0,031	0,037	-0,016	-0,045	0,052	0,015	-0,047	0,041
PWABEDR	0,014	0,015	0,028	-0,006	0,013	-0,029	0,028	-0,006	-0,011	0,018	-0,002	-0,013	-0,015	0,005	0,019	-0,004	0,018	-0,004	0,007
PWALAND	0,097	-0,031	0,04	0,014	0,1	-0,058	0,044	-0,022	-0,025	0,004	0,007	-0,004	-0,026	0,012	0,024	-0,066	-0,018	0,055	-0,002
PPERSAUT	-0,004	-0,009	0,019	-0,006	0	0,007	0,027	-0,028	-0,027	0,032	-0,009	-0,03	-0,046	0,004	0,025	0,007	0,007	-0,008	0,003
PBESAUT	-0,026	-0,023	-0,006	0,024	-0,025	0,006	0,002	0,01	-0,01	-0,016	-0,008	0,024	0,018	0	-0,006	-0,005	0,008	-0,001	0,004
PMOTSCO	-0,001	-0,016	0,02	-0,021	0,001	0,012	-0,005	-0,002	-0,001	0,025	0,012	-0,026	-0,023	-0,002	0,018	-0,011	0,006	0,007	0,015
PVRAAUT	0,022	-0,013	-0,001	-0,01	0,024	-0,006	0,026	-0,004	-0,025	0,004	0,007	-0,003	0,008	0,006	-0,007	-0,004	0,006	0	0,006
PAANHANG	0,041	-0,012	0,013	0,01	0,042	0,003	0,022	-0,018	-0,014	0,01	-0,008	-0,002	0	0,019	-0,003	-0,035	0,004	0,021	0,004
PTRACTOR	0,123	-0,036	0,06	0,01	0,126	-0,05	0,045	-0,035	-0,02	0,01	-0,005	-0,002	-0,023	-0,005	0,043	-0,068	-0,02	0,055	-0,003
PWERKT	0,06	-0,01	0,019	0,011	0,058	-0,008	0,014	-0,007	-0,011	0,012	-0,026	-0,002	0	0,003	0,005	-0,049	-0,005	0,028	-0,024
PBROM	0,035	-0,026	0,024	0,004	0,037	-0,021	0,008	-0,026	0,007	0,01	0,004	-0,016	-0,029	0,014	0,024	-0,048	-0,035	0,048	-0,038
PLEVEN	-0,036	0,032	0,022	-0,033	-0,037	-0,005	-0,018	-0,02	0,016	0,024	0,012	-0,028	-0,022	-0,008	0,024	0,038	0,046	-0,052	0,031
PPERSONG	0,025	0,001	0,001	0,024	0,026	-0,01	-0,007	0,007	0,004	0,016	-0,019	-0,013	0	-0,002	0,001	-0,011	-0,007	0,011	0,014

Table 8.8 – 3rd subsection of the Spearman correlation matrix

	MBERZELF	MBERBOER	MBERMIDD	MBERARBG	MBERARBO	MSKA	MSKB1	MSKB2	MSKC	MSKD	MHHUUR	MHKOOP	MAUT1	MAUT2	MAUT0	MZFONDS	MZPART	MINKM30	MINK3045
MSKD	-0,082	0,105	-0,163	0,173	0,452	-0,207	-0,215	-0,166	0,114	1									
MHHUUR	-0,099	-0,145	0,028	0,145	0,351	-0,32	-0,102	-0,165	0,287	0,284	1								
MHKOOP	0,097	0,143	-0,028	-0,146	-0,35	0,32	0,099	0,167	-0,288	-0,286	-1	1							
MAUT1	-0,059	-0,17	0,182	-0,062	-0,221	0,105	0,092	0,028	-0,086	-0,233	-0,169	0,171	1						
MAUT2	0,162	0,135	-0,124	-0,057	-0,132	0,171	-0,027	0,061	-0,095	-0,13	-0,237	0,235	-0,378	1					
MAUT0	-0,024	0,116	-0,064	0,095	0,379	-0,21	-0,041	-0,094	0,155	0,419	0,393	-0,395	-0,691	-0,243	1				
MZFONDS	-0,199	-0,02	-0,037	0,389	0,348	-0,556	-0,171	-0,116	0,517	0,272	0,358	-0,358	-0,124	-0,209	0,277	1			
MZPART	0,196	0,018	0,038	-0,39	-0,347	0,557	0,168	0,116	-0,518	-0,274	-0,358	0,358	0,127	0,208	-0,281	-0,999	1		
MINKM30	-0,063	0,088	-0,12	0,198	0,26	-0,241	-0,129	-0,059	0,225	0,338	0,504	-0,502	-0,29	-0,176	0,454	0,295	-0,296	1	
MINK3045	-0,064	-0,052	0,104	0,123	0,068	-0,217	-0,037	0,024	0,191	0,033	0,009	-0,007	0,079	0,038	-0,076	0,25	-0,247	-0,281	1
MINK4575	0,159	-0,019	0,073	-0,168	-0,262	0,412	0,154	0,041	-0,236	-0,267	-0,373	0,373	0,248	0,062	-0,318	-0,357	0,356	-0,559	-0,341
MINK7512	0,196	0	0,046	-0,137	-0,124	0,281	0,147	0,087	-0,219	-0,059	-0,207	0,205	0,041	0,139	-0,112	-0,336	0,333	-0,217	-0,229
MINK123M	0,243	0,04	0,03	-0,098	-0,124	0,258	0,105	0,082	-0,192	0,009	-0,16	0,158	-0,104	0,142	-0,012	-0,219	0,22	-0,06	-0,199
MINKGEM	0,173	-0,045	0,116	-0,262	-0,336	0,412	0,185	0,127	-0,349	-0,341	-0,457	0,456	0,255	0,163	-0,419	-0,445	0,445	-0,695	-0,181
MKOOPKLA	0,08	-0,124	0,135	-0,231	-0,352	0,325	0,177	0,071	-0,33	-0,324	-0,418	0,419	0,262	0,149	-0,4	-0,379	0,379	-0,399	-0,111
PWAPART	0,008	-0,063	0,023	-0,01	-0,013	0,016	0,03	0,001	-0,032	-0,004	-0,016	0,016	0,026	-0,043	-0,007	-0,011	0,01	0	-0,009
PWABEDR	0,013	0,044	-0,009	0,017	-0,031	0,02	-0,005	0,028	-0,008	-0,027	-0,035	0,035	-0,001	0,029	-0,019	-0,004	0,003	0,006	0,008
PWALAND	0,002	0,145	-0,051	-0,025	0,003	0,01	-0,001	0,032	-0,002	-0,019	-0,051	0,05	-0,007	0,041	-0,025	-0,02	0,021	-0,019	0,028
PPERSAUT	-0,015	-0,041	0,003	-0,019	-0,008	0,003	-0,016	0,004	0,006	-0,039	-0,028	0,027	0,042	0,008	-0,051	-0,021	0,019	-0,049	0,025
PBESAUT	0,018	0,009	0,012	-0,015	0,012	0,001	0,011	0,008	-0,024	0,024	-0,008	0,008	-0,014	0,006	0,019	0	-0,001	0,023	-0,014
PMOTSCO	-0,029	-0,002	0,007	-0,002	-0,005	-0,006	0,011	0,01	0	0,003	-0,026	0,027	0,008	0,017	-0,009	0,008	-0,008	0,005	0,027
PVRAAUT	-0,007	0,036	0,002	-0,006	0,005	0,016	0,014	0	0,001	0,003	-0,016	0,016	0,004	0,008	-0,013	-0,005	0,005	-0,015	0,021
PAANHANG	-0,01	0,051	-0,004	-0,013	0,009	0,003	0,005	0,025	-0,012	-0,015	-0,026	0,025	-0,007	0,039	-0,015	-0,011	0,012	-0,008	0,014
PTRACTOR	-0,01	0,173	-0,066	-0,018	-0,008	0,009	0	0,044	-0,023	-0,03	-0,056	0,056	-0,02	0,068	-0,032	-0,038	0,039	-0,011	0,005
PWERKT	-0,018	0,063	-0,022	-0,008	0,012	-0,015	0,016	0,014	-0,012	-0,027	-0,031	0,03	-0,024	0,027	0,01	-0,004	0,004	0,022	-0,017
PBROM	-0,015	0,048	-0,027	0,029	0,029	-0,025	-0,033	0,012	0,031	0,02	-0,012	0,012	-0,014	0,034	0,003	0,014	-0,014	0,004	0,01
PLEVEN	-0,007	-0,02	0,017	-0,011	-0,025	0,04	0,02	0,01	-0,042	-0,018	-0,033	0,034	0,035	-0,004	-0,039	-0,039	0,039	-0,034	-0,006
PPERSONG	-0,006	0,007	-0,011	0,014	-0,007	0,02	0	0,005	-0,001	0,006	0,005	-0,005	-0,01	0,007	0,005	-0,014	0,013	-0,002	0,024

Table 8.9 – 4th subsection of the Spearman correlation matrix

	MINK4575	MINK7512	MINK123M	MINKGEM	MKOOPKLA	PWAPART	PWABEDR	PWALAND	PPERSAUT	PBESAUT	PMOTSCO	PVRAAUT	PAANHANG	PTRACTOR	PWERKT	PBROM	PLEVEN	PPERSONG
MSKD																		
MHHUUR																		
MHKOOP																		
MAUT1																		
MAUT2																		
MAUT0																		
MZFONDS																		
MZPART																		
MINKM30																		
MINK3045																		
MINK4575	1																	
MINK7512	0,104	1																
MINK123M	0,075	0,322	1															
MINKGEM	0,63	0,54	0,283	1														
MKOOPKLA	0,374	0,226	0,146	0,459	1													
PWAPART	-0,002	0,025	0,015	0,016	0,003	1												
PWABEDR	-0,008	-0,011	0,004	0,008	0,029	-0,047	1											
PWALAND	0,001	0	-0,008	0,02	-0,009	-0,111	0,034	1										
PPERSAUT	0,035	-0,008	-0,027	0,039	0,025	0,158	-0,012	0,079	1									
PBESAUT	0,001	-0,016	0,001	-0,014	0,009	-0,041	0,217	0,027	0,019	1								
PMOTSCO	-0,016	0,005	0,008	-0,001	0,008	0,024	-0,016	-0,004	0,058	0,032	1							
PVRAAUT	0,011	-0,001	-0,017	-0,004	-0,005	-0,023	0,145	-0,006	0,011	0,238	-0,008	1						
PAANHANG	-0,007	0,004	0,002	0,011	-0,009	-0,02	0,085	0,099	0,046	0,1	-0,004	0,08	1					
PTRACTOR	0,008	-0,019	-0,03	0,006	-0,007	-0,075	0,076	0,557	0,08	0,047	-0,003	0,051	0,078	1				
PWERKT	-0,01	-0,028	-0,026	-0,016	0,008	-0,025	0,115	0,152	0,032	0,154	-0,012	0,071	0,103	0,215	1			
PBROM	-0,011	-0,023	-0,005	-0,019	-0,002	-0,153	-0,032	-0,01	-0,176	-0,025	-0,043	-0,011	-0,009	-0,007	0,006	1		
PLEVEN	0,045	0,018	0,011	0,042	0,063	0,139	0,02	0,001	0,071	0,022	0,037	-0,009	0,005	-0,005	0	-0,046	1	
PPERSONG	-0,005	-0,012	-0,007	-0,012	-0,009	-0,01	-0,009	0,04	0,01	-0,007	0,023	-0,003	0,015	0,05	0,074	-0,02	0,037	1

Table 8.10 – 5th subsection of the Spearman correlation matrix

	MOSTYPE	MAANTHUI	MGEMOMV	MGEMLEEF	MOSHOOFD	MGODRK	MGODPR	MGODOV	MGODGE	MRELGE	MRELSA	MRELOV	MFALLEEN	MFGEKIND	MFWEKIND	MOPLHOOG	MOPLMIDD	MOPLLAAG	MBERHOOG
PGEZONG	-0,009	0,017	0,018	-0,002	-0,007	0,014	0	-0,003	0,006	0,029	-0,003	-0,03	-0,039	0,005	0,015	0,001	0,009	-0,012	0,008
PWAOREG	0,007	-0,02	0,016	-0,001	0,004	-0,008	0,002	0,001	-0,001	0,009	-0,008	-0,002	-0,001	-0,002	0,01	-0,005	0,002	0,002	0,009
PBRAND	0,005	0,018	0,052	0,017	0,004	0,007	0,084	-0,015	-0,082	0,05	-0,023	-0,034	-0,046	0,014	0,025	0,036	-0,009	-0,019	0,072
PZEILPL	0,009	-0,007	0,012	0,002	0,007	-0,002	0,019	-0,027	-0,012	0,011	-0,024	-0,003	-0,011	0,013	0,005	0,022	-0,006	-0,006	0,006
PPLEZIER	-0,012	0,006	-0,004	-0,004	-0,015	0,013	0,023	0,009	-0,024	-0,003	0,023	-0,003	-0,013	0,027	-0,015	0,004	0,018	-0,014	0,005
PFIETS	-0,004	-0,015	0,028	0,014	-0,011	0,016	0,012	0,029	-0,019	0,016	-0,037	-0,01	-0,012	-0,009	0,014	0,025	0,009	-0,031	0,022
PINBOED	-0,018	0,024	0,024	-0,024	-0,019	-0,011	0,007	-0,008	0,003	-0,005	0,012	-0,001	-0,012	-0,009	0,015	0,023	0,013	-0,022	0,012
PBYSTAND	-0,051	0,001	0,035	-0,014	-0,051	0,001	0,014	0,013	-0,024	0,038	-0,011	-0,041	-0,042	0,002	0,018	0,041	0,03	-0,052	0,024
AWAPART	-0,045	0,059	-0,047	-0,021	-0,048	0,043	-0,011	0,016	-0,021	-0,033	0,011	0,034	0,04	-0,018	-0,044	0,048	0,008	-0,04	0,034
AWABEDR	0,015	0,015	0,028	-0,006	0,014	-0,029	0,028	-0,006	-0,011	0,018	-0,002	-0,013	-0,015	0,005	0,019	-0,004	0,017	-0,004	0,007
AWALAND	0,097	-0,031	0,04	0,014	0,1	-0,059	0,045	-0,022	-0,025	0,004	0,007	-0,004	-0,027	0,012	0,024	-0,066	-0,018	0,055	-0,002
APERSAUT	-0,006	-0,015	0,018	-0,014	-0,004	-0,005	0,03	-0,026	-0,023	0,033	-0,005	-0,031	-0,045	-0,001	0,028	0,005	0,012	-0,008	-0,002
ABESAUT	-0,026	-0,023	-0,007	0,024	-0,025	0,006	0,002	0,01	-0,01	-0,017	-0,008	0,024	0,018	0	-0,006	-0,005	0,008	-0,001	0,004
AMOTSCO	-0,001	-0,016	0,019	-0,021	0,001	0,012	-0,005	-0,002	-0,001	0,025	0,012	-0,025	-0,023	-0,003	0,018	-0,011	0,006	0,007	0,015
AVRAAUT	0,022	-0,013	-0,001	-0,01	0,024	-0,006	0,026	-0,004	-0,025	0,004	0,007	-0,003	0,008	0,006	-0,007	-0,004	0,006	0	0,006
AAANHANG	0,041	-0,012	0,013	0,01	0,042	0,003	0,022	-0,018	-0,014	0,01	-0,008	-0,002	0	0,019	-0,003	-0,035	0,004	0,02	0,004
ATRACTOR	0,123	-0,036	0,06	0,01	0,126	-0,05	0,045	-0,035	-0,021	0,01	-0,005	-0,002	-0,023	-0,005	0,043	-0,068	-0,02	0,055	-0,003
AWERKT	0,06	-0,01	0,019	0,011	0,058	-0,008	0,014	-0,007	-0,011	0,012	-0,026	-0,002	0	0,003	0,005	-0,049	-0,005	0,028	-0,024
ABROM	0,036	-0,027	0,023	0,005	0,037	-0,021	0,008	-0,025	0,006	0,01	0,004	-0,015	-0,028	0,014	0,023	-0,049	-0,035	0,048	-0,038
ALEVEN	-0,035	0,032	0,021	-0,032	-0,036	-0,004	-0,019	-0,019	0,016	0,024	0,012	-0,027	-0,021	-0,009	0,024	0,037	0,045	-0,05	0,03
APERSONG	0,025	0,001	0,001	0,024	0,026	-0,01	-0,007	0,007	0,004	0,016	-0,019	-0,013	0	-0,002	0,001	-0,011	-0,007	0,011	0,014
AGEZONG	-0,009	0,017	0,018	-0,002	-0,007	0,014	0	-0,003	0,006	0,029	-0,003	-0,03	-0,039	0,005	0,015	0,001	0,009	-0,012	0,008
AWAOREG	0,007	-0,02	0,016	-0,001	0,004	-0,008	0,002	0,001	-0,001	0,009	-0,008	-0,002	-0,001	-0,002	0,01	-0,005	0,002	0,002	0,009
ABRAND	-0,015	0,02	-0,011	0,016	-0,018	0,01	0,052	0,009	-0,056	-0,004	0,005	0,015	0,013	0,012	-0,028	0,029	-0,01	-0,007	0,052
AZEILPL	0,009	-0,007	0,012	0,002	0,007	-0,002	0,019	-0,027	-0,012	0,011	-0,024	-0,003	-0,011	0,013	0,005	0,022	-0,006	-0,006	0,006
APLEZIER	-0,012	0,006	-0,004	-0,004	-0,015	0,013	0,023	0,009	-0,024	-0,003	0,023	-0,003	-0,013	0,027	-0,015	0,004	0,018	-0,014	0,005
AFIETS	-0,004	-0,016	0,028	0,014	-0,011	0,016	0,012	0,029	-0,019	0,016	-0,037	-0,011	-0,012	-0,009	0,014	0,025	0,009	-0,032	0,022
AINBOED	-0,019	0,024	0,024	-0,024	-0,019	-0,011	0,007	-0,008	0,003	-0,005	0,012	-0,001	-0,012	-0,009	0,015	0,023	0,013	-0,022	0,012
ABYSTAND	-0,05	0,001	0,035	-0,014	-0,051	0	0,014	0,013	-0,024	0,038	-0,011	-0,041	-0,041	0,002	0,018	0,041	0,03	-0,052	0,024

Table 8.11 – 6th subsection of the Spearman correlation matrix

	MBERZELF	MBERBOER	MBERMIDD	MBERARBG	MBERARBO	MSKA	MSKB1	MSKB2	MSKC	MSKD	MHHUUR	MHKOOP	MAUT1	MAUT2	MAUT0	MZFONDS	MZPART	MINKM30	MINK3045
PGEZONG	-0,013	0	-0,004	-0,002	-0,016	-0,001	0,005	0,001	-0,003	-0,021	-0,022	0,024	0,048	-0,016	-0,035	-0,006	0,007	-0,017	-0,015
PWAOREG	0,014	0,028	-0,001	-0,005	-0,013	-0,002	0,018	0,01	-0,015	-0,026	-0,028	0,028	-0,008	0,029	-0,004	-0,011	0,01	0,009	0,027
PBRAND	0,054	0,074	-0,02	-0,038	-0,064	0,065	0,013	0,025	-0,046	-0,05	-0,181	0,181	0,02	0,066	-0,079	-0,069	0,069	-0,085	0,001
PZEILPL	0,025	0,006	-0,015	0,004	-0,019	0,02	0,003	-0,01	-0,016	-0,023	-0,024	0,024	0,008	0,01	-0,03	-0,035	0,035	-0,011	0,005
PPLEZIER	0,012	-0,001	0,002	0,004	-0,012	0,011	0,007	-0,001	-0,002	0,004	-0,023	0,023	-0,007	0,006	0,004	-0,011	0,011	-0,01	0,003
PFIETS	0,003	-0,047	-0,003	-0,016	-0,036	0,016	0,001	-0,022	-0,019	-0,017	-0,028	0,028	0,028	-0,007	-0,026	-0,022	0,022	-0,026	-0,007
PINBOED	0,017	-0,012	0,006	-0,001	-0,019	0,014	-0,007	-0,016	0,001	-0,007	-0,013	0,013	0,031	0,008	-0,032	-0,003	0,003	-0,016	-0,008
PBYSTAND	0,005	-0,032	0,03	-0,052	-0,003	0,027	0,017	0,027	-0,032	-0,044	-0,041	0,041	0,034	0,001	-0,048	-0,033	0,033	-0,056	0,002
AWAPART	0,004	-0,061	0,02	-0,007	-0,008	0,01	0,027	-0,001	-0,026	0,003	-0,009	0,009	0,019	-0,045	0,001	-0,006	0,005	0,006	-0,012
AWABEDR	0,013	0,044	-0,009	0,017	-0,031	0,02	-0,005	0,028	-0,008	-0,027	-0,035	0,035	-0,001	0,029	-0,019	-0,004	0,003	0,006	0,008
AWALAND	0,002	0,145	-0,051	-0,025	0,003	0,01	-0,001	0,032	-0,002	-0,018	-0,051	0,05	-0,007	0,041	-0,025	-0,02	0,021	-0,019	0,028
APERSAUT	-0,02	-0,046	0,009	-0,014	-0,002	-0,003	-0,009	-0,003	0,011	-0,031	-0,011	0,011	0,045	0,002	-0,05	-0,017	0,016	-0,039	0,025
ABESAUT	0,018	0,009	0,012	-0,015	0,012	0,001	0,011	0,008	-0,024	0,024	-0,008	0,008	-0,014	0,006	0,019	0	-0,001	0,023	-0,014
AMOTSCO	-0,028	-0,002	0,007	-0,002	-0,004	-0,006	0,011	0,01	0	0,003	-0,026	0,027	0,008	0,017	-0,009	0,008	-0,008	0,006	0,026
AVRAAUT	-0,007	0,036	0,002	-0,006	0,005	0,016	0,014	0	0,001	0,003	-0,016	0,016	0,004	0,008	-0,013	-0,005	0,005	-0,015	0,021
AAANHANG	-0,01	0,05	-0,004	-0,013	0,009	0,003	0,005	0,025	-0,012	-0,015	-0,026	0,025	-0,007	0,039	-0,015	-0,011	0,012	-0,008	0,014
ATRACTOR	-0,01	0,173	-0,066	-0,018	-0,009	0,01	0	0,044	-0,023	-0,031	-0,056	0,056	-0,02	0,068	-0,032	-0,038	0,039	-0,011	0,005
AWERKT	-0,018	0,063	-0,022	-0,008	0,012	-0,015	0,016	0,014	-0,012	-0,027	-0,031	0,03	-0,024	0,027	0,01	-0,004	0,004	0,022	-0,017
ABROM	-0,015	0,048	-0,027	0,029	0,03	-0,025	-0,033	0,011	0,031	0,02	-0,011	0,011	-0,013	0,034	0,003	0,014	-0,014	0,004	0,011
ALEVEN	-0,007	-0,02	0,017	-0,01	-0,024	0,039	0,02	0,01	-0,04	-0,017	-0,032	0,032	0,035	-0,005	-0,039	-0,038	0,038	-0,033	-0,006
APERSONG	-0,006	0,007	-0,011	0,014	-0,007	0,02	0	0,005	-0,001	0,006	0,005	-0,005	-0,01	0,007	0,005	-0,014	0,013	-0,002	0,024
AGEZONG	-0,013	0	-0,004	-0,002	-0,016	-0,001	0,005	0,001	-0,003	-0,021	-0,022	0,024	0,048	-0,016	-0,035	-0,006	0,007	-0,017	-0,015
AWAOREG	0,014	0,028	-0,001	-0,005	-0,013	-0,002	0,018	0,01	-0,015	-0,026	-0,028	0,028	-0,008	0,029	-0,004	-0,011	0,01	0,009	0,027
ABRAND	0,031	0,019	-0,005	-0,019	-0,026	0,032	0,012	0,006	-0,018	-0,011	-0,081	0,081	-0,004	0,014	-0,014	-0,019	0,019	-0,023	0,007
AZEILPL	0,025	0,006	-0,015	0,004	-0,019	0,02	0,003	-0,01	-0,016	-0,023	-0,024	0,024	0,008	0,01	-0,03	-0,035	0,035	-0,011	0,005
APLEZIER	0,012	-0,001	0,002	0,004	-0,012	0,011	0,007	-0,001	-0,002	0,004	-0,023	0,023	-0,007	0,006	0,003	-0,011	0,011	-0,01	0,003
AFIETS	0,003	-0,046	-0,003	-0,016	-0,036	0,016	0,001	-0,022	-0,02	-0,017	-0,029	0,028	0,028	-0,007	-0,026	-0,022	0,022	-0,026	-0,007
AINBOED	0,017	-0,012	0,006	-0,001	-0,019	0,014	-0,007	-0,016	0,001	-0,007	-0,013	0,013	0,031	0,008	-0,032	-0,003	0,003	-0,016	-0,008
ABYSTAND	0,005	-0,033	0,03	-0,052	-0,003	0,027	0,017	0,027	-0,032	-0,044	-0,04	0,041	0,034	0,001	-0,048	-0,033	0,033	-0,056	0,002

Table 8.12 – 7th subsection of the Spearman correlation matrix

	MINK4575	MINK7512	MINK123M	MINKGEM	MKOOPKLA	PWAPART	PWABEDR	PWALAND	PPERSAUT	PBESAUT	PMOTSCO	PVRAAUT	PAANHANG	PTRACTOR	PWERKT	PBROM	PLEVEN	PPERSONG	PGEZONG
PGEZONG	0,035	0,023	-0,007	0,028	0,029	0,057	-0,01	0,018	0,047	-0,007	-0,005	-0,003	0,011	0,015	-0,005	-0,005	0,129	-0,006	1
PWAOREG	-0,026	-0,011	-0,005	-0,007	0,02	-0,001	0,225	0,049	-0,002	0,055	0,002	0,067	0,02	0,079	0,088	-0,017	-0,002	-0,005	-0,005
PBRAND	0,062	0,043	0,052	0,096	0,102	0,513	0,082	0,207	0,113	0,022	0,011	0,016	0,049	0,169	0,057	-0,177	0,13	0,013	0,06
PZEILPL	0,003	0,001	-0,01	0,015	-0,004	0,013	-0,003	-0,003	-0,006	-0,002	-0,005	-0,001	0,07	-0,004	-0,001	-0,006	-0,005	-0,002	-0,002
PPLEZIER	0,01	-0,001	0,011	0,019	0,018	-0,004	-0,009	0,005	0,036	-0,007	-0,003	-0,003	0,036	0,003	-0,005	-0,02	0,004	0,026	-0,006
PFIETS	0,016	-0,004	0,012	0,023	0,039	-0,011	-0,019	-0,016	-0,036	-0,003	-0,015	0,022	-0,007	-0,011	-0,01	-0,026	-0,002	-0,012	0,014
PINBOED	0,021	0,003	0,021	0,02	0,015	0,043	0,023	0,001	0,018	-0,008	0,013	-0,003	0,046	-0,002	-0,005	-0,024	0,025	-0,006	0,017
PBYSTAND	0,043	0,02	0,013	0,051	0,064	0,048	-0,002	0,003	0,091	0,021	0,022	-0,005	-0,013	0,019	-0,007	-0,015	0,027	-0,009	0,118
AWAPART	-0,006	0,026	0,015	0,012	-0,004	0,989	-0,049	-0,112	0,149	-0,042	0,019	-0,023	-0,021	-0,076	-0,026	-0,154	0,136	-0,012	0,055
AWABEDR	-0,008	-0,011	0,004	0,008	0,028	-0,047	1	0,034	-0,012	0,215	-0,016	0,144	0,085	0,076	0,115	-0,032	0,019	-0,009	-0,01
AWALAND	0,001	0	-0,008	0,02	-0,009	-0,111	0,034	1	0,079	0,027	-0,004	-0,006	0,1	0,557	0,153	-0,01	0	0,039	0,018
APERSAUT	0,031	-0,015	-0,034	0,027	0,03	0,153	-0,02	0,085	0,95	0,018	0,061	0,011	0,037	0,085	0,032	-0,183	0,068	0,006	0,047
ABESAUT	0,001	-0,016	0,001	-0,014	0,009	-0,041	0,216	0,027	0,019	1	0,031	0,238	0,1	0,047	0,154	-0,025	0,022	-0,007	-0,007
AMOTSCO	-0,016	0,005	0,008	-0,001	0,008	0,024	-0,016	-0,004	0,058	0,031	1	-0,008	-0,004	-0,003	-0,012	-0,043	0,037	0,022	-0,005
AVRAAUT	0,011	-0,001	-0,017	-0,004	-0,005	-0,023	0,145	-0,006	0,011	0,238	-0,008	1	0,08	0,051	0,071	-0,011	-0,009	-0,003	-0,003
AAANHANG	-0,007	0,004	0,002	0,011	-0,008	-0,019	0,085	0,099	0,046	0,099	-0,004	0,079	1	0,078	0,103	-0,009	0,005	0,015	0,012
ATRACTOR	0,008	-0,019	-0,03	0,006	-0,007	-0,075	0,075	0,557	0,08	0,047	-0,003	0,051	0,079	1	0,214	-0,007	-0,005	0,05	0,015
AWERKT	-0,01	-0,028	-0,026	-0,016	0,008	-0,025	0,115	0,152	0,032	0,154	-0,012	0,071	0,103	0,215	1	0,006	0	0,075	-0,005
ABROM	-0,011	-0,023	-0,005	-0,019	-0,004	-0,153	-0,032	-0,01	-0,176	-0,025	-0,043	-0,011	-0,009	-0,007	0,006	0,999	-0,046	-0,02	-0,005
ALEVEN	0,044	0,018	0,011	0,042	0,062	0,141	0,019	0	0,071	0,023	0,037	-0,009	0,006	-0,006	-0,001	-0,047	0,999	0,036	0,128
APERSONG	-0,005	-0,012	-0,007	-0,012	-0,009	-0,01	-0,009	0,04	0,01	-0,007	0,023	-0,003	0,015	0,05	0,074	-0,02	0,037	1	-0,006
AGEZONG	0,034	0,023	-0,007	0,028	0,029	0,057	-0,01	0,018	0,047	-0,007	-0,005	-0,003	0,012	0,015	-0,005	-0,005	0,129	-0,006	1
AWAOREG	-0,026	-0,011	-0,005	-0,007	0,02	-0,001	0,225	0,049	-0,002	0,055	0,002	0,067	0,02	0,079	0,088	-0,017	-0,002	-0,005	-0,005
ABRAND	0,009	0,023	0,036	0,033	0,022	0,559	0,046	0,131	0,048	-0,005	-0,012	0	0,031	0,101	0,031	-0,199	0,106	0,011	0,043
AZEILPL	0,003	0,001	-0,01	0,015	-0,004	0,013	-0,003	-0,003	-0,006	-0,002	-0,005	-0,001	0,07	-0,004	-0,001	-0,006	-0,005	-0,002	-0,002
APLEZIER	0,01	-0,001	0,011	0,019	0,018	-0,004	-0,009	0,005	0,036	-0,007	-0,003	-0,003	0,036	0,003	-0,005	-0,02	0,004	0,026	-0,006
AFIETS	0,016	-0,004	0,012	0,024	0,039	-0,011	-0,019	-0,016	-0,036	-0,003	-0,015	0,021	-0,007	-0,011	-0,01	-0,026	-0,002	-0,012	0,015
AINBOED	0,021	0,003	0,021	0,02	0,015	0,043	0,023	0,001	0,018	-0,008	0,013	-0,003	0,046	-0,001	-0,005	-0,024	0,025	-0,006	0,017
ABYSTAND	0,043	0,02	0,013	0,051	0,064	0,048	-0,002	0,003	0,092	0,021	0,022	-0,005	-0,013	0,019	-0,007	-0,015	0,027	-0,009	0,117

Table 8.13 – 8th subsection of the Spearman correlation matrix

	PWAOREG	PBRAND	PZEILPL	PPLEZIER	PFIETS	PINBOED	PBYSTAND	AWAPART	AWABEDR	AWALAND	APERSAUT	ABESAUT	AMOTSCO	AVRAAUT
PGEZONG														
PWAOREG	1													
PBRAND	0,054	1												
PZEILPL	-0,001	0,011	1											
PPLEZIER	-0,005	0,015	0,1	1										
PFIETS	-0,01	-0,038	-0,004	-0,012	1									
PINBOED	-0,006	0,036	0,084	0,046	0,011	1								
PBYSTAND	-0,008	0,056	-0,003	0,01	0,009	0,023	1							
AWAPART	-0,007	0,513	0,012	-0,006	-0,014	0,04	0,044	1						
AWABEDR	0,225	0,081	-0,003	-0,009	-0,019	0,023	-0,002	-0,049	1					
AWALAND	0,049	0,207	-0,003	0,005	-0,016	0,001	0,003	-0,111	0,034	1				
APERSAUT	-0,004	0,094	-0,009	0,034	-0,041	0,017	0,097	0,143	-0,02	0,085	1			
ABESAUT	0,055	0,022	-0,002	-0,007	-0,003	-0,008	0,021	-0,042	0,215	0,027	0,018	1		
AMOTSCO	0,002	0,011	-0,005	-0,003	-0,015	0,013	0,022	0,019	-0,016	-0,004	0,061	0,031	1	
AVRAAUT	0,067	0,016	-0,001	-0,003	0,022	-0,003	-0,005	-0,023	0,144	-0,006	0,011	0,238	-0,008	1
AAANHANG	0,02	0,049	0,07	0,036	-0,007	0,047	-0,013	-0,021	0,085	0,1	0,037	0,099	-0,004	0,079
ATRACTOR	0,079	0,169	-0,004	0,003	-0,011	-0,001	0,019	-0,076	0,076	0,557	0,085	0,047	-0,003	0,051
AWERKT	0,088	0,057	-0,001	-0,005	-0,01	-0,005	-0,007	-0,026	0,115	0,152	0,032	0,154	-0,012	0,071
ABROM	-0,017	-0,176	-0,006	-0,02	-0,026	-0,024	-0,015	-0,153	-0,032	-0,01	-0,183	-0,025	-0,043	-0,011
ALEVEN	-0,002	0,131	-0,005	0,003	-0,002	0,025	0,025	0,137	0,019	0	0,069	0,023	0,037	-0,009
APERSONG	-0,005	0,013	-0,002	0,026	-0,012	-0,006	-0,009	-0,012	-0,009	0,039	0,006	-0,007	0,022	-0,003
AGEZONG	-0,005	0,06	-0,002	-0,006	0,014	0,017	0,117	0,055	-0,01	0,018	0,047	-0,007	-0,005	-0,003
AWAOREG	1	0,054	-0,001	-0,005	-0,01	-0,006	-0,008	-0,007	0,225	0,049	-0,004	0,055	0,002	0,067
ABRAND	0,031	0,916	0,005	0,003	-0,041	0,029	0,037	0,563	0,046	0,131	0,032	-0,005	-0,012	0
AZEILPL	-0,001	0,011	1	0,1	-0,004	0,084	-0,003	0,012	-0,003	-0,003	-0,009	-0,002	-0,005	-0,001
APLEZIER	-0,005	0,015	0,099	1	-0,012	0,045	0,01	-0,006	-0,009	0,005	0,034	-0,007	-0,003	-0,003
AFIETS	-0,01	-0,037	-0,004	-0,012	1	0,011	0,009	-0,013	-0,019	-0,016	-0,041	-0,003	-0,015	0,021
AINBOED	-0,006	0,036	0,084	0,046	0,011	1	0,023	0,04	0,023	0,001	0,017	-0,008	0,013	-0,003
ABYSTAND	-0,008	0,056	-0,003	0,01	0,009	0,023	1	0,045	-0,002	0,003	0,097	0,021	0,022	-0,005

Table 8.14 – 9th subsection of the Spearman correlation matrix

	AAANHANG	ATRACTOR	AWERKT	ABROM	ALEVEN	APERSONG	AGEZONG	AWAOREG	ABRAND	AZEILPL	APLEZIER	AFIETS	AINBOED	ABYSTAND
PGEZONG														
PWAOREG														
PBRAND														
PZEILPL														
PPLEZIER														
PFIETS														
PINBOED														
PBYSTAND														
AWAPART														
AWABEDR														
AWALAND														
APERSAUT														
ABESAUT														
AMOTSCO														
AVRAAUT														
AAANHANG	1													
ATRACTOR	0,078	1												
AWERKT	0,103	0,214	1											
ABROM	-0,009	-0,007	0,006	1										
ALEVEN	0,006	-0,006	-0,001	-0,047	1									
APERSONG	0,015	0,05	0,074	-0,02	0,036	1								
AGEZONG	0,012	0,015	-0,005	-0,005	0,128	-0,006	1							
AWAOREG	0,02	0,079	0,087	-0,017	-0,002	-0,005	-0,005	1						
ABRAND	0,031	0,101	0,031	-0,199	0,107	0,011	0,043	0,031	1					
AZEILPL	0,07	-0,004	-0,001	-0,006	-0,005	-0,002	-0,002	-0,001	0,005	1				
APLEZIER	0,035	0,003	-0,005	-0,02	0,003	0,026	-0,006	-0,005	0,003	0,099	1			
AFIETS	-0,007	-0,011	-0,01	-0,026	-0,002	-0,012	0,015	-0,01	-0,041	-0,004	-0,012	1		
AINBOED	0,047	-0,001	-0,005	-0,024	0,025	-0,006	0,017	-0,006	0,029	0,084	0,046	0,011	1	
ABYSTAND	-0,013	0,019	-0,007	-0,015	0,025	-0,009	0,117	-0,008	0,037	-0,003	0,01	0,009	0,023	1

Table 8.15 – 10th subsection of the Spearman correlation matrix

8.6. PREDICTIVE MODELS RESULTS WITH VARIOUS VARIABLES COMBINATIONS

Nr.	Method	Variable Subset	Identified True Positives	Sensitivity	Precision	F-measure
1	Logistic Regression	1 - All socio-demographic & policies interactions	51	0,490	0,146	0,225
2	CART	1 - All socio-demographic & policies interactions	56	0,539	0,161	0,247
3	MLP - 1 neuron	1 - All socio-demographic & policies interactions	49	0,471	0,140	0,216
4	MLP - 2 neuron	1 - All socio-demographic & policies interactions	48	0,462	0,138	0,212
5	MLP - 3 neuron	1 - All socio-demographic & policies interactions	45	0,433	0,129	0,199
6	MLP - 4 neuron	1 - All socio-demographic & policies interactions	49	0,471	0,140	0,216
7	MLP - 5 neuron	1 - All socio-demographic & policies interactions	47	0,452	0,135	0,208
8	MLP - 6 neuron	1 - All socio-demographic & policies interactions	41	0,394	0,118	0,181
9	MLP - 7 neuron	1 - All socio-demographic & policies interactions	49	0,471	0,140	0,216
10	MLP - 8 neuron	1 - All socio-demographic & policies interactions	44	0,423	0,126	0,194
11	MLP - 9 neuron	1 - All socio-demographic & policies interactions	47	0,452	0,135	0,208
12	MLP - 10 neuron	1 - All socio-demographic & policies interactions	45	0,433	0,129	0,199
13	MLP - 11 neuron	1 - All socio-demographic & policies interactions	49	0,471	0,140	0,216
14	MLP - 12 neuron	1 - All socio-demographic & policies interactions	46	0,442	0,132	0,203
15	MLP - 13 neuron	1 - All socio-demographic & policies interactions	36	0,346	0,103	0,159
16	MLP - 14 neuron	1 - All socio-demographic & policies interactions	48	0,462	0,138	0,212
17	MLP - 15 neuron	1 - All socio-demographic & policies interactions	42	0,404	0,120	0,185
18	MLP - 16 neuron	1 - All socio-demographic & policies interactions	48	0,462	0,138	0,212
19	MLP - 17 neuron	1 - All socio-demographic & policies interactions	46	0,442	0,132	0,203
20	MLP - 18 neuron	1 - All socio-demographic & policies interactions	50	0,481	0,143	0,221
21	MLP - 19 neuron	1 - All socio-demographic & policies interactions	47	0,452	0,135	0,208
22	MLP - 20 neuron	1 - All socio-demographic & policies interactions	38	0,365	0,109	0,168
23	SVM gamma = 0.001 C = 10	1 - All socio-demographic & policies interactions	32	0,308	0,092	0,141
24	SVM gamma = 0.001 C = 20	1 - All socio-demographic & policies interactions	26	0,250	0,075	0,115
25	SVM gamma = 0.001 C = 30	1 - All socio-demographic & policies interactions	22	0,212	0,063	0,097
26	SVM gamma = 0.001 C = 40	1 - All socio-demographic & policies interactions	20	0,192	0,057	0,088
27	SVM gamma = 0.001 C = 50	1 - All socio-demographic & policies interactions	20	0,192	0,057	0,088
28	SVM gamma = 0.001 C = 100	1 - All socio-demographic & policies interactions	28	0,269	0,080	0,124
29	SVM gamma = 0.01 C = 10	1 - All socio-demographic & policies interactions	36	0,346	0,103	0,159
30	SVM gamma = 0.01 C = 20	1 - All socio-demographic & policies interactions	36	0,346	0,103	0,159
31	SVM gamma = 0.01 C = 30	1 - All socio-demographic & policies interactions	40	0,385	0,115	0,177
32	SVM gamma = 0.01 C = 40	1 - All socio-demographic & policies interactions	35	0,337	0,100	0,155

33	SVM gamma = 0.01 C = 50	1 - All socio-demographic & policies interactions	35	0,337	0,100	0,155
34	SVM gamma = 0.01 C = 100	1 - All socio-demographic & policies interactions	31	0,298	0,089	0,137
35	SVM gamma = 0.1 C = 10	1 - All socio-demographic & policies interactions	30	0,289	0,086	0,133
36	SVM gamma = 0.1 C = 20	1 - All socio-demographic & policies interactions	32	0,308	0,092	0,141
37	SVM gamma = 0.1 C = 30	1 - All socio-demographic & policies interactions	33	0,317	0,095	0,146
38	SVM gamma = 0.1 C = 40	1 - All socio-demographic & policies interactions	32	0,308	0,092	0,141
39	SVM gamma = 0.1 C = 50	1 - All socio-demographic & policies interactions	32	0,308	0,092	0,141
40	SVM gamma = 0.1 C = 100	1 - All socio-demographic & policies interactions	32	0,308	0,092	0,141
41	Logistic Regression	2 - Information Gain & Chi-Squared	50	0,481	0,143	0,221
42	CART	2 - Information Gain & Chi-Squared	56	0,539	0,161	0,247
43	MLP - 1 neuron	2 - Information Gain & Chi-Squared	51	0,490	0,146	0,225
44	MLP - 2 neuron	2 - Information Gain & Chi-Squared	49	0,471	0,140	0,216
45	MLP - 3 neuron	2 - Information Gain & Chi-Squared	49	0,471	0,140	0,216
46	MLP - 4 neuron	2 - Information Gain & Chi-Squared	55	0,529	0,158	0,243
47	MLP - 5 neuron	2 - Information Gain & Chi-Squared	54	0,519	0,155	0,238
48	MLP - 6 neuron	2 - Information Gain & Chi-Squared	55	0,529	0,158	0,243
49	MLP - 7 neuron	2 - Information Gain & Chi-Squared	53	0,510	0,152	0,234
50	MLP - 8 neuron	2 - Information Gain & Chi-Squared	54	0,519	0,155	0,238
51	MLP - 9 neuron	2 - Information Gain & Chi-Squared	49	0,471	0,140	0,216
52	MLP - 10 neuron	2 - Information Gain & Chi-Squared	50	0,481	0,143	0,221
53	MLP - 11 neuron	2 - Information Gain & Chi-Squared	53	0,510	0,152	0,234
54	MLP - 12 neuron	2 - Information Gain & Chi-Squared	53	0,510	0,152	0,234
55	MLP - 13 neuron	2 - Information Gain & Chi-Squared	51	0,490	0,146	0,225
56	MLP - 14 neuron	2 - Information Gain & Chi-Squared	54	0,519	0,155	0,238
57	MLP - 15 neuron	2 - Information Gain & Chi-Squared	55	0,529	0,158	0,243
58	MLP - 16 neuron	2 - Information Gain & Chi-Squared	53	0,510	0,152	0,234
59	MLP - 17 neuron	2 - Information Gain & Chi-Squared	52	0,500	0,149	0,230
60	MLP - 18 neuron	2 - Information Gain & Chi-Squared	53	0,510	0,152	0,234
61	MLP - 19 neuron	2 - Information Gain & Chi-Squared	47	0,452	0,135	0,208
62	MLP - 20 neuron	2 - Information Gain & Chi-Squared	51	0,490	0,146	0,225
63	SVM gamma = 0.001 C = 10	2 - Information Gain & Chi-Squared	24	0,231	0,069	0,106
64	SVM gamma = 0.001 C = 20	2 - Information Gain & Chi-Squared	22	0,212	0,063	0,097
65	SVM gamma = 0.001 C = 30	2 - Information Gain & Chi-Squared	19	0,183	0,054	0,084
66	SVM gamma = 0.001 C = 40	2 - Information Gain & Chi-Squared	25	0,240	0,072	0,110
67	SVM gamma = 0.001 C = 50	2 - Information Gain & Chi-Squared	19	0,183	0,054	0,084
68	SVM gamma = 0.001 C = 100	2 - Information Gain & Chi-Squared	26	0,250	0,075	0,115
69	SVM gamma = 0.01 C = 10	2 - Information Gain & Chi-Squared	31	0,298	0,089	0,137
70	SVM gamma = 0.01 C = 20	2 - Information Gain & Chi-Squared	34	0,327	0,097	0,150
71	SVM gamma = 0.01 C = 30	2 - Information Gain & Chi-Squared	30	0,289	0,086	0,133
72	SVM gamma = 0.01 C = 40	2 - Information Gain & Chi-Squared	28	0,269	0,080	0,124
73	SVM gamma = 0.01 C	2 - Information Gain & Chi-Squared	25	0,240	0,072	0,110

	= 50					
74	SVM gamma = 0.01 C = 100	2 - Information Gain & Chi-Squared	19	0,183	0,054	0,084
75	SVM gamma = 0.1 C = 10	2 - Information Gain & Chi-Squared	29	0,279	0,083	0,128
76	SVM gamma = 0.1 C = 20	2 - Information Gain & Chi-Squared	28	0,269	0,080	0,124
77	SVM gamma = 0.1 C = 30	2 - Information Gain & Chi-Squared	25	0,240	0,072	0,110
78	SVM gamma = 0.1 C = 40	2 - Information Gain & Chi-Squared	29	0,279	0,083	0,128
79	SVM gamma = 0.1 C = 50	2 - Information Gain & Chi-Squared	30	0,289	0,086	0,133
80	SVM gamma = 0.1 C = 100	2 - Information Gain & Chi-Squared	35	0,337	0,100	0,155
81	Logistic Regression	3 - Stepwise regression	49	0,471	0,140	0,216
82	CART	3 - Stepwise regression	53	0,510	0,152	0,234
83	MLP - 1 neuron	3 - Stepwise regression	53	0,510	0,152	0,234
84	MLP - 2 neuron	3 - Stepwise regression	52	0,500	0,149	0,230
85	MLP - 3 neuron	3 - Stepwise regression	49	0,471	0,140	0,216
86	MLP - 4 neuron	3 - Stepwise regression	45	0,433	0,129	0,199
87	MLP - 5 neuron	3 - Stepwise regression	52	0,500	0,149	0,230
88	MLP - 6 neuron	3 - Stepwise regression	50	0,481	0,143	0,221
89	MLP - 7 neuron	3 - Stepwise regression	49	0,471	0,140	0,216
90	MLP - 8 neuron	3 - Stepwise regression	48	0,462	0,138	0,212
91	MLP - 9 neuron	3 - Stepwise regression	47	0,452	0,135	0,208
92	MLP - 10 neuron	3 - Stepwise regression	51	0,490	0,146	0,225
93	MLP - 11 neuron	3 - Stepwise regression	46	0,442	0,132	0,203
94	MLP - 12 neuron	3 - Stepwise regression	47	0,452	0,135	0,208
95	MLP - 13 neuron	3 - Stepwise regression	45	0,433	0,129	0,199
96	MLP - 14 neuron	3 - Stepwise regression	51	0,490	0,146	0,225
97	MLP - 15 neuron	3 - Stepwise regression	53	0,510	0,152	0,234
98	MLP - 16 neuron	3 - Stepwise regression	53	0,510	0,152	0,234
99	MLP - 17 neuron	3 - Stepwise regression	39	0,375	0,112	0,172
100	MLP - 18 neuron	3 - Stepwise regression	51	0,490	0,146	0,225
101	MLP - 19 neuron	3 - Stepwise regression	53	0,510	0,152	0,234
102	MLP - 20 neuron	3 - Stepwise regression	39	0,375	0,112	0,172
103	SVM gamma = 0.001 C = 10	3 - Stepwise regression	23	0,221	0,066	0,102
104	SVM gamma = 0.001 C = 20	3 - Stepwise regression	17	0,164	0,049	0,075
105	SVM gamma = 0.001 C = 30	3 - Stepwise regression	21	0,202	0,060	0,093
106	SVM gamma = 0.001 C = 40	3 - Stepwise regression	14	0,135	0,040	0,062
107	SVM gamma = 0.001 C = 50	3 - Stepwise regression	20	0,192	0,057	0,088
108	SVM gamma = 0.001 C = 100	3 - Stepwise regression	22	0,212	0,063	0,097
109	SVM gamma = 0.01 C = 10	3 - Stepwise regression	31	0,298	0,089	0,137
110	SVM gamma = 0.01 C = 20	3 - Stepwise regression	14	0,135	0,040	0,062
111	SVM gamma = 0.01 C = 30	3 - Stepwise regression	21	0,202	0,060	0,093
112	SVM gamma = 0.01 C = 40	3 - Stepwise regression	23	0,221	0,066	0,102
113	SVM gamma = 0.01 C = 50	3 - Stepwise regression	20	0,192	0,057	0,088

114	SVM gamma = 0.01 C = 100	3 - Stepwise regression	25	0,240	0,072	0,110
115	SVM gamma = 0.1 C = 10	3 - Stepwise regression	26	0,250	0,075	0,115
116	SVM gamma = 0.1 C = 20	3 - Stepwise regression	30	0,289	0,086	0,133
117	SVM gamma = 0.1 C = 30	3 - Stepwise regression	31	0,298	0,089	0,137
118	SVM gamma = 0.1 C = 40	3 - Stepwise regression	31	0,298	0,089	0,137
119	SVM gamma = 0.1 C = 50	3 - Stepwise regression	28	0,269	0,080	0,124
120	SVM gamma = 0.1 C = 100	3 - Stepwise regression	32	0,308	0,092	0,141
121	Logistic Regression	4 - Boruta	47	0,452	0,135	0,208
122	CART	4 - Boruta	43	0,414	0,123	0,190
123	MLP - 1 neuron	4 - Boruta	48	0,462	0,138	0,212
124	MLP - 2 neuron	4 - Boruta	49	0,471	0,140	0,216
125	MLP - 3 neuron	4 - Boruta	48	0,462	0,138	0,212
126	MLP - 4 neuron	4 - Boruta	48	0,462	0,138	0,212
127	MLP - 5 neuron	4 - Boruta	38	0,365	0,109	0,168
128	MLP - 6 neuron	4 - Boruta	43	0,414	0,123	0,190
129	MLP - 7 neuron	4 - Boruta	36	0,346	0,103	0,159
130	MLP - 8 neuron	4 - Boruta	45	0,433	0,129	0,199
131	MLP - 9 neuron	4 - Boruta	46	0,442	0,132	0,203
132	MLP - 10 neuron	4 - Boruta	47	0,452	0,135	0,208
133	MLP - 11 neuron	4 - Boruta	46	0,442	0,132	0,203
134	MLP - 12 neuron	4 - Boruta	40	0,385	0,115	0,177
135	MLP - 13 neuron	4 - Boruta	45	0,433	0,129	0,199
136	MLP - 14 neuron	4 - Boruta	41	0,394	0,118	0,181
137	MLP - 15 neuron	4 - Boruta	39	0,375	0,112	0,172
138	MLP - 16 neuron	4 - Boruta	42	0,404	0,120	0,185
139	MLP - 17 neuron	4 - Boruta	47	0,452	0,135	0,208
140	MLP - 18 neuron	4 - Boruta	40	0,385	0,115	0,177
141	MLP - 19 neuron	4 - Boruta	40	0,385	0,115	0,177
142	MLP - 20 neuron	4 - Boruta	47	0,452	0,135	0,208
143	SVM gamma = 0.001 C = 10	4 - Boruta	20	0,192	0,057	0,088
144	SVM gamma = 0.001 C = 20	4 - Boruta	20	0,192	0,057	0,088
145	SVM gamma = 0.001 C = 30	4 - Boruta	18	0,173	0,052	0,080
146	SVM gamma = 0.001 C = 40	4 - Boruta	21	0,202	0,060	0,093
147	SVM gamma = 0.001 C = 50	4 - Boruta	19	0,183	0,054	0,084
148	SVM gamma = 0.001 C = 100	4 - Boruta	23	0,221	0,066	0,102
149	SVM gamma = 0.01 C = 10	4 - Boruta	29	0,279	0,083	0,128
150	SVM gamma = 0.01 C = 20	4 - Boruta	31	0,298	0,089	0,137
151	SVM gamma = 0.01 C = 30	4 - Boruta	29	0,279	0,083	0,128
152	SVM gamma = 0.01 C = 40	4 - Boruta	30	0,289	0,086	0,133
153	SVM gamma = 0.01 C = 50	4 - Boruta	33	0,317	0,095	0,146
154	SVM gamma = 0.01 C	4 - Boruta	27	0,260	0,077	0,119

	= 100					
155	SVM gamma = 0.1 C = 10	4 - Boruta	28	0,269	0,080	0,124
156	SVM gamma = 0.1 C = 20	4 - Boruta	33	0,317	0,095	0,146
157	SVM gamma = 0.1 C = 30	4 - Boruta	32	0,308	0,092	0,141
158	SVM gamma = 0.1 C = 40	4 - Boruta	31	0,298	0,089	0,137
159	SVM gamma = 0.1 C = 50	4 - Boruta	33	0,317	0,095	0,146
160	SVM gamma = 0.1 C = 100	4 - Boruta	35	0,337	0,100	0,155
161	Logistic Regression	5 - Total Mix	52	0,500	0,149	0,230
162	CART	5 - Total Mix	52	0,500	0,149	0,230
163	MLP - 1 neuron	5 - Total Mix	48	0,462	0,138	0,212
164	MLP - 2 neuron	5 - Total Mix	49	0,471	0,140	0,216
165	MLP - 3 neuron	5 - Total Mix	44	0,423	0,126	0,194
166	MLP - 4 neuron	5 - Total Mix	50	0,481	0,143	0,221
167	MLP - 5 neuron	5 - Total Mix	49	0,471	0,140	0,216
168	MLP - 6 neuron	5 - Total Mix	50	0,481	0,143	0,221
169	MLP - 7 neuron	5 - Total Mix	46	0,442	0,132	0,203
170	MLP - 8 neuron	5 - Total Mix	55	0,529	0,158	0,243
171	MLP - 9 neuron	5 - Total Mix	50	0,481	0,143	0,221
172	MLP - 10 neuron	5 - Total Mix	49	0,471	0,140	0,216
173	MLP - 11 neuron	5 - Total Mix	44	0,423	0,126	0,194
174	MLP - 12 neuron	5 - Total Mix	51	0,490	0,146	0,225
175	MLP - 13 neuron	5 - Total Mix	54	0,519	0,155	0,238
176	MLP - 14 neuron	5 - Total Mix	54	0,519	0,155	0,238
177	MLP - 15 neuron	5 - Total Mix	48	0,462	0,138	0,212
178	MLP - 16 neuron	5 - Total Mix	51	0,490	0,146	0,225
179	MLP - 17 neuron	5 - Total Mix	46	0,442	0,132	0,203
180	MLP - 18 neuron	5 - Total Mix	49	0,471	0,140	0,216
181	MLP - 19 neuron	5 - Total Mix	42	0,404	0,120	0,185
182	MLP - 20 neuron	5 - Total Mix	49	0,471	0,140	0,216
183	SVM gamma = 0.001 C = 10	5 - Total Mix	16	0,154	0,046	0,071
184	SVM gamma = 0.001 C = 20	5 - Total Mix	17	0,164	0,049	0,075
185	SVM gamma = 0.001 C = 30	5 - Total Mix	16	0,154	0,046	0,071
186	SVM gamma = 0.001 C = 40	5 - Total Mix	17	0,164	0,049	0,075
187	SVM gamma = 0.001 C = 50	5 - Total Mix	13	0,125	0,037	0,057
188	SVM gamma = 0.001 C = 100	5 - Total Mix	16	0,154	0,046	0,071
189	SVM gamma = 0.01 C = 10	5 - Total Mix	16	0,154	0,046	0,071
190	SVM gamma = 0.01 C = 20	5 - Total Mix	29	0,279	0,083	0,128
191	SVM gamma = 0.01 C = 30	5 - Total Mix	28	0,269	0,080	0,124
192	SVM gamma = 0.01 C = 40	5 - Total Mix	26	0,250	0,075	0,115
193	SVM gamma = 0.01 C = 50	5 - Total Mix	27	0,260	0,077	0,119
194	SVM gamma = 0.01 C = 100	5 - Total Mix	29	0,279	0,083	0,128

195	SVM gamma = 0.1 C = 10	5 - Total Mix	33	0,317	0,095	0,146
196	SVM gamma = 0.1 C = 20	5 - Total Mix	30	0,289	0,086	0,133
197	SVM gamma = 0.1 C = 30	5 - Total Mix	35	0,337	0,100	0,155
198	SVM gamma = 0.1 C = 40	5 - Total Mix	36	0,346	0,103	0,159
199	SVM gamma = 0.1 C = 50	5 - Total Mix	36	0,346	0,103	0,159
200	SVM gamma = 0.1 C = 100	5 - Total Mix	36	0,346	0,103	0,159
201	Logistic Regression	6 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
202	CART	6 - All policies interactions & Socio-demographic mix	51	0,490	0,146	0,225
203	MLP - 1 neuron	6 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
204	MLP - 2 neuron	6 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
205	MLP - 3 neuron	6 - All policies interactions & Socio-demographic mix	47	0,452	0,135	0,208
206	MLP - 4 neuron	6 - All policies interactions & Socio-demographic mix	49	0,471	0,140	0,216
207	MLP - 5 neuron	6 - All policies interactions & Socio-demographic mix	40	0,385	0,115	0,177
208	MLP - 6 neuron	6 - All policies interactions & Socio-demographic mix	37	0,356	0,106	0,163
209	MLP - 7 neuron	6 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
210	MLP - 8 neuron	6 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
211	MLP - 9 neuron	6 - All policies interactions & Socio-demographic mix	39	0,375	0,112	0,172
212	MLP - 10 neuron	6 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
213	MLP - 11 neuron	6 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
214	MLP - 12 neuron	6 - All policies interactions & Socio-demographic mix	44	0,423	0,126	0,194
215	MLP - 13 neuron	6 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
216	MLP - 14 neuron	6 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
217	MLP - 15 neuron	6 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
218	MLP - 16 neuron	6 - All policies interactions & Socio-demographic mix	46	0,442	0,132	0,203
219	MLP - 17 neuron	6 - All policies interactions & Socio-demographic mix	47	0,452	0,135	0,208
220	MLP - 18 neuron	6 - All policies interactions & Socio-demographic mix	43	0,414	0,123	0,190
221	MLP - 19 neuron	6 - All policies interactions & Socio-demographic mix	46	0,442	0,132	0,203
222	MLP - 20 neuron	6 - All policies interactions & Socio-demographic mix	47	0,452	0,135	0,208
223	SVM gamma = 0.001 C = 10	6 - All policies interactions & Socio-demographic mix	14	0,135	0,040	0,062
224	SVM gamma = 0.001 C = 20	6 - All policies interactions & Socio-demographic mix	15	0,144	0,043	0,066
225	SVM gamma = 0.001 C = 30	6 - All policies interactions & Socio-demographic mix	18	0,173	0,052	0,080
226	SVM gamma = 0.001 C = 40	6 - All policies interactions & Socio-demographic mix	16	0,154	0,046	0,071
227	SVM gamma = 0.001 C = 50	6 - All policies interactions & Socio-demographic mix	18	0,173	0,052	0,080
228	SVM gamma = 0.001 C = 100	6 - All policies interactions & Socio-demographic mix	16	0,154	0,046	0,071
229	SVM gamma = 0.01 C = 10	6 - All policies interactions & Socio-demographic mix	30	0,289	0,086	0,133
230	SVM gamma = 0.01 C	6 - All policies interactions & Socio-	25	0,240	0,072	0,110

	= 20	demographic mix				
231	SVM gamma = 0.01 C = 30	6 - All policies interactions & Socio-demographic mix	25	0,240	0,072	0,110
232	SVM gamma = 0.01 C = 40	6 - All policies interactions & Socio-demographic mix	26	0,250	0,075	0,115
233	SVM gamma = 0.01 C = 50	6 - All policies interactions & Socio-demographic mix	29	0,279	0,083	0,128
234	SVM gamma = 0.01 C = 100	6 - All policies interactions & Socio-demographic mix	29	0,279	0,083	0,128
235	SVM gamma = 0.1 C = 10	6 - All policies interactions & Socio-demographic mix	31	0,298	0,089	0,137
236	SVM gamma = 0.1 C = 20	6 - All policies interactions & Socio-demographic mix	32	0,308	0,092	0,141
237	SVM gamma = 0.1 C = 30	6 - All policies interactions & Socio-demographic mix	32	0,308	0,092	0,141
238	SVM gamma = 0.1 C = 40	6 - All policies interactions & Socio-demographic mix	34	0,327	0,097	0,150
239	SVM gamma = 0.1 C = 50	6 - All policies interactions & Socio-demographic mix	36	0,346	0,103	0,159
240	SVM gamma = 0.1 C = 100	6 - All policies interactions & Socio-demographic mix	35	0,337	0,100	0,155
241	Logistic Regression	7 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
242	CART	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
243	MLP - 1 neuron	7 - All policies interactions & Socio-demographic mix	49	0,471	0,140	0,216
244	MLP - 2 neuron	7 - All policies interactions & Socio-demographic mix	47	0,452	0,135	0,208
245	MLP - 3 neuron	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
246	MLP - 4 neuron	7 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
247	MLP - 5 neuron	7 - All policies interactions & Socio-demographic mix	45	0,433	0,129	0,199
248	MLP - 6 neuron	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
249	MLP - 7 neuron	7 - All policies interactions & Socio-demographic mix	54	0,519	0,155	0,238
250	MLP - 8 neuron	7 - All policies interactions & Socio-demographic mix	47	0,452	0,135	0,208
251	MLP - 9 neuron	7 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
252	MLP - 10 neuron	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
253	MLP - 11 neuron	7 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
254	MLP - 12 neuron	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
255	MLP - 13 neuron	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
256	MLP - 14 neuron	7 - All policies interactions & Socio-demographic mix	53	0,510	0,152	0,234
257	MLP - 15 neuron	7 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
258	MLP - 16 neuron	7 - All policies interactions & Socio-demographic mix	48	0,462	0,138	0,212
259	MLP - 17 neuron	7 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
260	MLP - 18 neuron	7 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
261	MLP - 19 neuron	7 - All policies interactions & Socio-demographic mix	50	0,481	0,143	0,221
262	MLP - 20 neuron	7 - All policies interactions & Socio-demographic mix	52	0,500	0,149	0,230
263	SVM gamma = 0.001 C = 10	7 - All policies interactions & Socio-demographic mix	22	0,212	0,063	0,097
264	SVM gamma = 0.001 C = 20	7 - All policies interactions & Socio-demographic mix	30	0,289	0,086	0,133
265	SVM gamma = 0.001 C = 30	7 - All policies interactions & Socio-demographic mix	24	0,231	0,069	0,106

266	SVM gamma = 0.001 C = 40	7 - All policies interactions & Socio-demographic mix	18	0,173	0,052	0,080
267	SVM gamma = 0.001 C = 50	7 - All policies interactions & Socio-demographic mix	25	0,240	0,072	0,110
268	SVM gamma = 0.001 C = 100	7 - All policies interactions & Socio-demographic mix	22	0,212	0,063	0,097
269	SVM gamma = 0.01 C = 10	7 - All policies interactions & Socio-demographic mix	24	0,231	0,069	0,106
270	SVM gamma = 0.01 C = 20	7 - All policies interactions & Socio-demographic mix	25	0,240	0,072	0,110
271	SVM gamma = 0.01 C = 30	7 - All policies interactions & Socio-demographic mix	31	0,298	0,089	0,137
272	SVM gamma = 0.01 C = 40	7 - All policies interactions & Socio-demographic mix	28	0,269	0,080	0,124
273	SVM gamma = 0.01 C = 50	7 - All policies interactions & Socio-demographic mix	20	0,192	0,057	0,088
274	SVM gamma = 0.01 C = 100	7 - All policies interactions & Socio-demographic mix	19	0,183	0,054	0,084
275	SVM gamma = 0.1 C = 10	7 - All policies interactions & Socio-demographic mix	18	0,173	0,052	0,080
276	SVM gamma = 0.1 C = 20	7 - All policies interactions & Socio-demographic mix	22	0,212	0,063	0,097
277	SVM gamma = 0.1 C = 30	7 - All policies interactions & Socio-demographic mix	23	0,221	0,066	0,102
278	SVM gamma = 0.1 C = 40	7 - All policies interactions & Socio-demographic mix	25	0,240	0,072	0,110
279	SVM gamma = 0.1 C = 50	7 - All policies interactions & Socio-demographic mix	27	0,260	0,077	0,119
280	SVM gamma = 0.1 C = 100	7 - All policies interactions & Socio-demographic mix	29	0,279	0,083	0,128

Table 8.16 – Predictive models results with various variables combinations

8.7. Models Results with Different Seeds

Nr.	Model	Seed	Identified True Positives	Sensitivity	Precision	F-measure
1	Logistic Regression	999	48	0,462	0,138	0,212
2	CART	999	50	0,481	0,143	0,221
3	MLP - 1 neuron	999	49	0,471	0,140	0,216
4	MLP - 2 neuron	999	47	0,452	0,135	0,208
5	MLP - 3 neuron	999	50	0,481	0,143	0,221
6	MLP - 4 neuron	999	48	0,462	0,138	0,212
7	MLP - 5 neuron	999	45	0,433	0,129	0,199
8	MLP - 6 neuron	999	50	0,481	0,143	0,221
9	MLP - 7 neuron	999	54	0,519	0,155	0,238
10	MLP - 8 neuron	999	47	0,452	0,135	0,208
11	MLP - 9 neuron	999	48	0,462	0,138	0,212
12	MLP - 10 neuron	999	50	0,481	0,143	0,221
13	MLP - 11 neuron	999	52	0,500	0,149	0,230
14	MLP - 12 neuron	999	50	0,481	0,143	0,221
15	MLP - 13 neuron	999	50	0,481	0,143	0,221
16	MLP - 14 neuron	999	53	0,510	0,152	0,234
17	MLP - 15 neuron	999	52	0,500	0,149	0,230
18	MLP - 16 neuron	999	48	0,462	0,138	0,212
19	MLP - 17 neuron	999	52	0,500	0,149	0,230

20	MLP - 18 neuron	999	52	0,500	0,149	0,230
21	MLP - 19 neuron	999	50	0,481	0,143	0,221
22	MLP - 20 neuron	999	52	0,500	0,149	0,230
23	SVM gamma = 0.001 C = 10	999	22	0,212	0,063	0,097
24	SVM gamma = 0.001 C = 20	999	30	0,288	0,086	0,132
25	SVM gamma = 0.001 C = 30	999	24	0,231	0,069	0,106
26	SVM gamma = 0.001 C = 40	999	18	0,173	0,052	0,079
27	SVM gamma = 0.001 C = 50	999	25	0,240	0,072	0,110
28	SVM gamma = 0.001 C = 100	999	22	0,212	0,063	0,097
29	SVM gamma = 0.01 C = 10	999	24	0,231	0,069	0,106
30	SVM gamma = 0.01 C = 20	999	25	0,240	0,072	0,110
31	SVM gamma = 0.01 C = 30	999	31	0,298	0,089	0,137
32	SVM gamma = 0.01 C = 40	999	28	0,269	0,080	0,124
33	SVM gamma = 0.01 C = 50	999	20	0,192	0,057	0,088
34	SVM gamma = 0.01 C = 100	999	19	0,183	0,054	0,084
35	SVM gamma = 0.1 C = 10	999	18	0,173	0,052	0,079
36	SVM gamma = 0.1 C = 20	999	22	0,212	0,063	0,097
37	SVM gamma = 0.1 C = 30	999	23	0,221	0,066	0,102
38	SVM gamma = 0.1 C = 40	999	25	0,240	0,072	0,110
39	SVM gamma = 0.1 C = 50	999	27	0,260	0,077	0,119
40	SVM gamma = 0.1 C = 100	999	29	0,279	0,083	0,128
41	Logistic Regression	111	47	0,452	0,135	0,208
42	CART	111	42	0,404	0,120	0,185
43	MLP - 1 neuron	111	11	0,106	0,032	0,049
44	MLP - 2 neuron	111	47	0,452	0,135	0,208
45	MLP - 3 neuron	111	50	0,481	0,143	0,221
46	MLP - 4 neuron	111	47	0,452	0,135	0,208
47	MLP - 5 neuron	111	50	0,481	0,143	0,221
48	MLP - 6 neuron	111	44	0,423	0,126	0,194
49	MLP - 7 neuron	111	48	0,462	0,138	0,212
50	MLP - 8 neuron	111	47	0,452	0,135	0,208
51	MLP - 9 neuron	111	44	0,423	0,126	0,194
52	MLP - 10 neuron	111	45	0,433	0,129	0,199
53	MLP - 11 neuron	111	49	0,471	0,140	0,216
54	MLP - 12 neuron	111	47	0,452	0,135	0,208
55	MLP - 13 neuron	111	48	0,462	0,138	0,212
56	MLP - 14 neuron	111	42	0,404	0,120	0,185
57	MLP - 15 neuron	111	45	0,433	0,129	0,199
58	MLP - 16 neuron	111	45	0,433	0,129	0,199
59	MLP - 17 neuron	111	42	0,404	0,120	0,185
60	MLP - 18 neuron	111	43	0,413	0,123	0,190
61	MLP - 19 neuron	111	44	0,423	0,126	0,194
62	MLP - 20 neuron	111	45	0,433	0,129	0,199

63	SVM gamma = 0.001 C = 10	111	24	0,231	0,069	0,106
64	SVM gamma = 0.001 C = 20	111	22	0,212	0,063	0,097
65	SVM gamma = 0.001 C = 30	111	20	0,192	0,057	0,088
66	SVM gamma = 0.001 C = 40	111	22	0,212	0,063	0,097
67	SVM gamma = 0.001 C = 50	111	24	0,231	0,069	0,106
68	SVM gamma = 0.001 C = 100	111	33	0,317	0,095	0,146
69	SVM gamma = 0.01 C = 10	111	27	0,260	0,077	0,119
70	SVM gamma = 0.01 C = 20	111	29	0,279	0,083	0,128
71	SVM gamma = 0.01 C = 30	111	29	0,279	0,083	0,128
72	SVM gamma = 0.01 C = 40	111	30	0,288	0,086	0,132
73	SVM gamma = 0.01 C = 50	111	28	0,269	0,080	0,124
74	SVM gamma = 0.01 C = 100	111	19	0,183	0,054	0,084
75	SVM gamma = 0.1 C = 10	111	34	0,327	0,097	0,150
76	SVM gamma = 0.1 C = 20	111	37	0,356	0,106	0,163
77	SVM gamma = 0.1 C = 30	111	39	0,375	0,112	0,172
78	SVM gamma = 0.1 C = 40	111	39	0,375	0,112	0,172
79	SVM gamma = 0.1 C = 50	111	38	0,365	0,109	0,168
80	SVM gamma = 0.1 C = 100	111	40	0,385	0,115	0,177
81	Logistic Regression	222	45	0,433	0,129	0,199
82	CART	222	41	0,394	0,117	0,181
83	MLP - 1 neuron	222	42	0,404	0,120	0,185
84	MLP - 2 neuron	222	43	0,413	0,123	0,190
85	MLP - 3 neuron	222	45	0,433	0,129	0,199
86	MLP - 4 neuron	222	45	0,433	0,129	0,199
87	MLP - 5 neuron	222	44	0,423	0,126	0,194
88	MLP - 6 neuron	222	46	0,442	0,132	0,203
89	MLP - 7 neuron	222	42	0,404	0,120	0,185
90	MLP - 8 neuron	222	43	0,413	0,123	0,190
91	MLP - 9 neuron	222	42	0,404	0,120	0,185
92	MLP - 10 neuron	222	43	0,413	0,123	0,190
93	MLP - 11 neuron	222	44	0,423	0,126	0,194
94	MLP - 12 neuron	222	43	0,413	0,123	0,190
95	MLP - 13 neuron	222	42	0,404	0,120	0,185
96	MLP - 14 neuron	222	40	0,385	0,115	0,177
97	MLP - 15 neuron	222	46	0,442	0,132	0,203
98	MLP - 16 neuron	222	46	0,442	0,132	0,203
99	MLP - 17 neuron	222	44	0,423	0,126	0,194
100	MLP - 18 neuron	222	44	0,423	0,126	0,194
101	MLP - 19 neuron	222	40	0,385	0,115	0,177
102	MLP - 20 neuron	222	43	0,413	0,123	0,190
103	SVM gamma = 0.001 C = 10	222	32	0,308	0,092	0,141
104	SVM gamma = 0.001 C = 20	222	25	0,240	0,072	0,110
105	SVM gamma = 0.001 C = 30	222	26	0,250	0,075	0,115

106	SVM gamma = 0.001 C = 40	222	26	0,250	0,075	0,115
107	SVM gamma = 0.001 C = 50	222	25	0,240	0,072	0,110
108	SVM gamma = 0.001 C = 100	222	29	0,279	0,083	0,128
109	SVM gamma = 0.01 C = 10	222	20	0,192	0,057	0,088
110	SVM gamma = 0.01 C = 20	222	22	0,212	0,063	0,097
111	SVM gamma = 0.01 C = 30	222	26	0,250	0,075	0,115
112	SVM gamma = 0.01 C = 40	222	34	0,327	0,097	0,150
113	SVM gamma = 0.01 C = 50	222	24	0,231	0,069	0,106
114	SVM gamma = 0.01 C = 100	222	35	0,337	0,100	0,155
115	SVM gamma = 0.1 C = 10	222	28	0,269	0,080	0,124
116	SVM gamma = 0.1 C = 20	222	26	0,250	0,075	0,115
117	SVM gamma = 0.1 C = 30	222	23	0,221	0,066	0,102
118	SVM gamma = 0.1 C = 40	222	25	0,240	0,072	0,110
119	SVM gamma = 0.1 C = 50	222	25	0,240	0,072	0,110
120	SVM gamma = 0.1 C = 100	222	25	0,240	0,072	0,110
121	Logistic Regression	333	33	0,317	0,095	0,146
122	CART	333	33	0,317	0,095	0,146
123	MLP - 1 neuron	333	31	0,298	0,089	0,137
124	MLP - 2 neuron	333	33	0,317	0,095	0,146
125	MLP - 3 neuron	333	35	0,337	0,100	0,155
126	MLP - 4 neuron	333	39	0,375	0,112	0,172
127	MLP - 5 neuron	333	32	0,308	0,092	0,141
128	MLP - 6 neuron	333	28	0,269	0,080	0,124
129	MLP - 7 neuron	333	33	0,317	0,095	0,146
130	MLP - 8 neuron	333	27	0,260	0,077	0,119
131	MLP - 9 neuron	333	32	0,308	0,092	0,141
132	MLP - 10 neuron	333	35	0,337	0,100	0,155
133	MLP - 11 neuron	333	30	0,288	0,086	0,132
134	MLP - 12 neuron	333	36	0,346	0,103	0,159
135	MLP - 13 neuron	333	32	0,308	0,092	0,141
136	MLP - 14 neuron	333	33	0,317	0,095	0,146
137	MLP - 15 neuron	333	31	0,298	0,089	0,137
138	MLP - 16 neuron	333	29	0,279	0,083	0,128
139	MLP - 17 neuron	333	37	0,356	0,106	0,163
140	MLP - 18 neuron	333	33	0,317	0,095	0,146
141	MLP - 19 neuron	333	33	0,317	0,095	0,146
142	MLP - 20 neuron	333	30	0,288	0,086	0,132
143	SVM gamma = 0.001 C = 10	333	34	0,327	0,097	0,150
144	SVM gamma = 0.001 C = 20	333	30	0,288	0,086	0,132
145	SVM gamma = 0.001 C = 30	333	27	0,260	0,077	0,119
146	SVM gamma = 0.001 C = 40	333	23	0,221	0,066	0,102
147	SVM gamma = 0.001 C = 50	333	23	0,221	0,066	0,102
148	SVM gamma = 0.001 C = 100	333	24	0,231	0,069	0,106

149	SVM gamma = 0.01 C = 10	333	34	0,327	0,097	0,150
150	SVM gamma = 0.01 C = 20	333	36	0,346	0,103	0,159
151	SVM gamma = 0.01 C = 30	333	32	0,308	0,092	0,141
152	SVM gamma = 0.01 C = 40	333	26	0,250	0,075	0,115
153	SVM gamma = 0.01 C = 50	333	30	0,288	0,086	0,132
154	SVM gamma = 0.01 C = 100	333	34	0,327	0,097	0,150
155	SVM gamma = 0.1 C = 10	333	22	0,212	0,063	0,097
156	SVM gamma = 0.1 C = 20	333	22	0,212	0,063	0,097
157	SVM gamma = 0.1 C = 30	333	23	0,221	0,066	0,102
158	SVM gamma = 0.1 C = 40	333	22	0,212	0,063	0,097
159	SVM gamma = 0.1 C = 50	333	21	0,202	0,060	0,093
160	SVM gamma = 0.1 C = 100	333	21	0,202	0,060	0,093
161	Logistic Regression	444	43	0,413	0,123	0,190
162	CART	444	45	0,433	0,129	0,199
163	MLP - 1 neuron	444	14	0,135	0,040	0,062
164	MLP - 2 neuron	444	44	0,423	0,126	0,194
165	MLP - 3 neuron	444	43	0,413	0,123	0,190
166	MLP - 4 neuron	444	41	0,394	0,117	0,181
167	MLP - 5 neuron	444	42	0,404	0,120	0,185
168	MLP - 6 neuron	444	42	0,404	0,120	0,185
169	MLP - 7 neuron	444	41	0,394	0,117	0,181
170	MLP - 8 neuron	444	41	0,394	0,117	0,181
171	MLP - 9 neuron	444	41	0,394	0,117	0,181
172	MLP - 10 neuron	444	43	0,413	0,123	0,190
173	MLP - 11 neuron	444	40	0,385	0,115	0,177
174	MLP - 12 neuron	444	46	0,442	0,132	0,203
175	MLP - 13 neuron	444	45	0,433	0,129	0,199
176	MLP - 14 neuron	444	44	0,423	0,126	0,194
177	MLP - 15 neuron	444	39	0,375	0,112	0,172
178	MLP - 16 neuron	444	44	0,423	0,126	0,194
179	MLP - 17 neuron	444	35	0,337	0,100	0,155
180	MLP - 18 neuron	444	42	0,404	0,120	0,185
181	MLP - 19 neuron	444	41	0,394	0,117	0,181
182	MLP - 20 neuron	444	39	0,375	0,112	0,172
183	SVM gamma = 0.001 C = 10	444	26	0,250	0,075	0,115
184	SVM gamma = 0.001 C = 20	444	29	0,279	0,083	0,128
185	SVM gamma = 0.001 C = 30	444	29	0,279	0,083	0,128
186	SVM gamma = 0.001 C = 40	444	32	0,308	0,092	0,141
187	SVM gamma = 0.001 C = 50	444	27	0,260	0,077	0,119
188	SVM gamma = 0.001 C = 100	444	38	0,365	0,109	0,168
189	SVM gamma = 0.01 C = 10	444	25	0,240	0,072	0,110
190	SVM gamma = 0.01 C = 20	444	25	0,240	0,072	0,110
191	SVM gamma = 0.01 C = 30	444	23	0,221	0,066	0,102

192	SVM gamma = 0.01 C = 40	444	30	0,288	0,086	0,132
193	SVM gamma = 0.01 C = 50	444	24	0,231	0,069	0,106
194	SVM gamma = 0.01 C = 100	444	28	0,269	0,080	0,124
195	SVM gamma = 0.1 C = 10	444	20	0,192	0,057	0,088
196	SVM gamma = 0.1 C = 20	444	23	0,221	0,066	0,102
197	SVM gamma = 0.1 C = 30	444	23	0,221	0,066	0,102
198	SVM gamma = 0.1 C = 40	444	27	0,260	0,077	0,119
199	SVM gamma = 0.1 C = 50	444	28	0,269	0,080	0,124
200	SVM gamma = 0.1 C = 100	444	26	0,250	0,075	0,115
201	Logistic Regression	555	39	0,375	0,112	0,172
202	CART	555	40	0,385	0,115	0,177
203	MLP - 1 neuron	555	32	0,308	0,092	0,141
204	MLP - 2 neuron	555	35	0,337	0,100	0,155
205	MLP - 3 neuron	555	42	0,404	0,120	0,185
206	MLP - 4 neuron	555	38	0,365	0,109	0,168
207	MLP - 5 neuron	555	39	0,375	0,112	0,172
208	MLP - 6 neuron	555	39	0,375	0,112	0,172
209	MLP - 7 neuron	555	41	0,394	0,117	0,181
210	MLP - 8 neuron	555	41	0,394	0,117	0,181
211	MLP - 9 neuron	555	38	0,365	0,109	0,168
212	MLP - 10 neuron	555	39	0,375	0,112	0,172
213	MLP - 11 neuron	555	39	0,375	0,112	0,172
214	MLP - 12 neuron	555	41	0,394	0,117	0,181
215	MLP - 13 neuron	555	37	0,356	0,106	0,163
216	MLP - 14 neuron	555	40	0,385	0,115	0,177
217	MLP - 15 neuron	555	40	0,385	0,115	0,177
218	MLP - 16 neuron	555	39	0,375	0,112	0,172
219	MLP - 17 neuron	555	42	0,404	0,120	0,185
220	MLP - 18 neuron	555	41	0,394	0,117	0,181
221	MLP - 19 neuron	555	42	0,404	0,120	0,185
222	MLP - 20 neuron	555	40	0,385	0,115	0,177
223	SVM gamma = 0.001 C = 10	555	24	0,231	0,069	0,106
224	SVM gamma = 0.001 C = 20	555	41	0,394	0,117	0,181
225	SVM gamma = 0.001 C = 30	555	35	0,337	0,100	0,155
226	SVM gamma = 0.001 C = 40	555	33	0,317	0,095	0,146
227	SVM gamma = 0.001 C = 50	555	34	0,327	0,097	0,150
228	SVM gamma = 0.001 C = 100	555	35	0,337	0,100	0,155
229	SVM gamma = 0.01 C = 10	555	21	0,202	0,060	0,093
230	SVM gamma = 0.01 C = 20	555	28	0,269	0,080	0,124
231	SVM gamma = 0.01 C = 30	555	24	0,231	0,069	0,106
232	SVM gamma = 0.01 C = 40	555	17	0,163	0,049	0,075
233	SVM gamma = 0.01 C = 50	555	21	0,202	0,060	0,093
234	SVM gamma = 0.01 C = 100	555	13	0,125	0,037	0,057

235	SVM gamma = 0.1 C = 10	555	17	0,163	0,049	0,075
236	SVM gamma = 0.1 C = 20	555	20	0,192	0,057	0,088
237	SVM gamma = 0.1 C = 30	555	23	0,221	0,066	0,102
238	SVM gamma = 0.1 C = 40	555	18	0,173	0,052	0,079
239	SVM gamma = 0.1 C = 50	555	20	0,192	0,057	0,088
240	SVM gamma = 0.1 C = 100	555	17	0,163	0,049	0,075
241	Logistic Regression	666	42	0,404	0,120	0,185
242	CART	666	44	0,423	0,126	0,194
243	MLP - 1 neuron	666	14	0,135	0,040	0,062
244	MLP - 2 neuron	666	38	0,365	0,109	0,168
245	MLP - 3 neuron	666	38	0,365	0,109	0,168
246	MLP - 4 neuron	666	43	0,413	0,123	0,190
247	MLP - 5 neuron	666	44	0,423	0,126	0,194
248	MLP - 6 neuron	666	45	0,433	0,129	0,199
249	MLP - 7 neuron	666	40	0,385	0,115	0,177
250	MLP - 8 neuron	666	40	0,385	0,115	0,177
251	MLP - 9 neuron	666	42	0,404	0,120	0,185
252	MLP - 10 neuron	666	41	0,394	0,117	0,181
253	MLP - 11 neuron	666	42	0,404	0,120	0,185
254	MLP - 12 neuron	666	39	0,375	0,112	0,172
255	MLP - 13 neuron	666	45	0,433	0,129	0,199
256	MLP - 14 neuron	666	39	0,375	0,112	0,172
257	MLP - 15 neuron	666	37	0,356	0,106	0,163
258	MLP - 16 neuron	666	41	0,394	0,117	0,181
259	MLP - 17 neuron	666	45	0,433	0,129	0,199
260	MLP - 18 neuron	666	39	0,375	0,112	0,172
261	MLP - 19 neuron	666	42	0,404	0,120	0,185
262	MLP - 20 neuron	666	41	0,394	0,117	0,181
263	SVM gamma = 0.001 C = 10	666	28	0,269	0,080	0,124
264	SVM gamma = 0.001 C = 20	666	31	0,298	0,089	0,137
265	SVM gamma = 0.001 C = 30	666	36	0,346	0,103	0,159
266	SVM gamma = 0.001 C = 40	666	32	0,308	0,092	0,141
267	SVM gamma = 0.001 C = 50	666	35	0,337	0,100	0,155
268	SVM gamma = 0.001 C = 100	666	29	0,279	0,083	0,128
269	SVM gamma = 0.01 C = 10	666	27	0,260	0,077	0,119
270	SVM gamma = 0.01 C = 20	666	33	0,317	0,095	0,146
271	SVM gamma = 0.01 C = 30	666	25	0,240	0,072	0,110
272	SVM gamma = 0.01 C = 40	666	40	0,385	0,115	0,177
273	SVM gamma = 0.01 C = 50	666	20	0,192	0,057	0,088
274	SVM gamma = 0.01 C = 100	666	17	0,163	0,049	0,075
275	SVM gamma = 0.1 C = 10	666	27	0,260	0,077	0,119
276	SVM gamma = 0.1 C = 20	666	27	0,260	0,077	0,119
277	SVM gamma = 0.1 C = 30	666	30	0,288	0,086	0,132

278	SVM gamma = 0.1 C = 40	666	25	0,240	0,072	0,110
279	SVM gamma = 0.1 C = 50	666	27	0,260	0,077	0,119
280	SVM gamma = 0.1 C = 100	666	24	0,231	0,069	0,106
281	Logistic Regression	777	43	0,413	0,123	0,190
282	CART	777	44	0,423	0,126	0,194
283	MLP - 1 neuron	777	42	0,404	0,120	0,185
284	MLP - 2 neuron	777	43	0,413	0,123	0,190
285	MLP - 3 neuron	777	44	0,423	0,126	0,194
286	MLP - 4 neuron	777	43	0,413	0,123	0,190
287	MLP - 5 neuron	777	41	0,394	0,117	0,181
288	MLP - 6 neuron	777	42	0,404	0,120	0,185
289	MLP - 7 neuron	777	36	0,346	0,103	0,159
290	MLP - 8 neuron	777	40	0,385	0,115	0,177
291	MLP - 9 neuron	777	43	0,413	0,123	0,190
292	MLP - 10 neuron	777	41	0,394	0,117	0,181
293	MLP - 11 neuron	777	40	0,385	0,115	0,177
294	MLP - 12 neuron	777	37	0,356	0,106	0,163
295	MLP - 13 neuron	777	45	0,433	0,129	0,199
296	MLP - 14 neuron	777	36	0,346	0,103	0,159
297	MLP - 15 neuron	777	35	0,337	0,100	0,155
298	MLP - 16 neuron	777	40	0,385	0,115	0,177
299	MLP - 17 neuron	777	39	0,375	0,112	0,172
300	MLP - 18 neuron	777	39	0,375	0,112	0,172
301	MLP - 19 neuron	777	40	0,385	0,115	0,177
302	MLP - 20 neuron	777	42	0,404	0,120	0,185
303	SVM gamma = 0.001 C = 10	777	26	0,250	0,075	0,115
304	SVM gamma = 0.001 C = 20	777	22	0,212	0,063	0,097
305	SVM gamma = 0.001 C = 30	777	23	0,221	0,066	0,102
306	SVM gamma = 0.001 C = 40	777	32	0,308	0,092	0,141
307	SVM gamma = 0.001 C = 50	777	28	0,269	0,080	0,124
308	SVM gamma = 0.001 C = 100	777	25	0,240	0,072	0,110
309	SVM gamma = 0.01 C = 10	777	26	0,250	0,075	0,115
310	SVM gamma = 0.01 C = 20	777	36	0,346	0,103	0,159
311	SVM gamma = 0.01 C = 30	777	27	0,260	0,077	0,119
312	SVM gamma = 0.01 C = 40	777	30	0,288	0,086	0,132
313	SVM gamma = 0.01 C = 50	777	33	0,317	0,095	0,146
314	SVM gamma = 0.01 C = 100	777	36	0,346	0,103	0,159
315	SVM gamma = 0.1 C = 10	777	26	0,250	0,075	0,115
316	SVM gamma = 0.1 C = 20	777	29	0,279	0,083	0,128
317	SVM gamma = 0.1 C = 30	777	31	0,298	0,089	0,137
318	SVM gamma = 0.1 C = 40	777	34	0,327	0,097	0,150
319	SVM gamma = 0.1 C = 50	777	32	0,308	0,092	0,141
320	SVM gamma = 0.1 C = 100	777	31	0,298	0,089	0,137

321	Logistic Regression	888	48	0,462	0,138	0,212
322	CART	888	44	0,423	0,126	0,194
323	MLP - 1 neuron	888	7	0,067	0,020	0,031
324	MLP - 2 neuron	888	47	0,452	0,135	0,208
325	MLP - 3 neuron	888	45	0,433	0,129	0,199
326	MLP - 4 neuron	888	42	0,404	0,120	0,185
327	MLP - 5 neuron	888	47	0,452	0,135	0,208
328	MLP - 6 neuron	888	51	0,490	0,146	0,225
329	MLP - 7 neuron	888	51	0,490	0,146	0,225
330	MLP - 8 neuron	888	49	0,471	0,140	0,216
331	MLP - 9 neuron	888	49	0,471	0,140	0,216
332	MLP - 10 neuron	888	46	0,442	0,132	0,203
333	MLP - 11 neuron	888	51	0,490	0,146	0,225
334	MLP - 12 neuron	888	55	0,529	0,158	0,243
335	MLP - 13 neuron	888	46	0,442	0,132	0,203
336	MLP - 14 neuron	888	48	0,462	0,138	0,212
337	MLP - 15 neuron	888	50	0,481	0,143	0,221
338	MLP - 16 neuron	888	51	0,490	0,146	0,225
339	MLP - 17 neuron	888	48	0,462	0,138	0,212
340	MLP - 18 neuron	888	54	0,519	0,155	0,238
341	MLP - 19 neuron	888	46	0,442	0,132	0,203
342	MLP - 20 neuron	888	50	0,481	0,143	0,221
343	SVM gamma = 0.001 C = 10	888	33	0,317	0,095	0,146
344	SVM gamma = 0.001 C = 20	888	31	0,298	0,089	0,137
345	SVM gamma = 0.001 C = 30	888	34	0,327	0,097	0,150
346	SVM gamma = 0.001 C = 40	888	22	0,212	0,063	0,097
347	SVM gamma = 0.001 C = 50	888	31	0,298	0,089	0,137
348	SVM gamma = 0.001 C = 100	888	34	0,327	0,097	0,150
349	SVM gamma = 0.01 C = 10	888	28	0,269	0,080	0,124
350	SVM gamma = 0.01 C = 20	888	30	0,288	0,086	0,132
351	SVM gamma = 0.01 C = 30	888	40	0,385	0,115	0,177
352	SVM gamma = 0.01 C = 40	888	31	0,298	0,089	0,137
353	SVM gamma = 0.01 C = 50	888	23	0,221	0,066	0,102
354	SVM gamma = 0.01 C = 100	888	39	0,375	0,112	0,172
355	SVM gamma = 0.1 C = 10	888	32	0,308	0,092	0,141
356	SVM gamma = 0.1 C = 20	888	32	0,308	0,092	0,141
357	SVM gamma = 0.1 C = 30	888	29	0,279	0,083	0,128
358	SVM gamma = 0.1 C = 40	888	31	0,298	0,089	0,137
359	SVM gamma = 0.1 C = 50	888	29	0,279	0,083	0,128
360	SVM gamma = 0.1 C = 100	888	31	0,298	0,089	0,137
361	Logistic Regression	9999	48	0,462	0,138	0,212
362	CART	9999	43	0,413	0,123	0,190
363	MLP - 1 neuron	9999	42	0,404	0,120	0,185

364	MLP - 2 neuron	9999	43	0,413	0,123	0,190
365	MLP - 3 neuron	9999	43	0,413	0,123	0,190
366	MLP - 4 neuron	9999	48	0,462	0,138	0,212
367	MLP - 5 neuron	9999	47	0,452	0,135	0,208
368	MLP - 6 neuron	9999	45	0,433	0,129	0,199
369	MLP - 7 neuron	9999	44	0,423	0,126	0,194
370	MLP - 8 neuron	9999	42	0,404	0,120	0,185
371	MLP - 9 neuron	9999	42	0,404	0,120	0,185
372	MLP - 10 neuron	9999	45	0,433	0,129	0,199
373	MLP - 11 neuron	9999	45	0,433	0,129	0,199
374	MLP - 12 neuron	9999	46	0,442	0,132	0,203
375	MLP - 13 neuron	9999	40	0,385	0,115	0,177
376	MLP - 14 neuron	9999	44	0,423	0,126	0,194
377	MLP - 15 neuron	9999	45	0,433	0,129	0,199
378	MLP - 16 neuron	9999	41	0,394	0,117	0,181
379	MLP - 17 neuron	9999	42	0,404	0,120	0,185
380	MLP - 18 neuron	9999	41	0,394	0,117	0,181
381	MLP - 19 neuron	9999	44	0,423	0,126	0,194
382	MLP - 20 neuron	9999	39	0,375	0,112	0,172
383	SVM gamma = 0.001 C = 10	9999	19	0,183	0,054	0,084
384	SVM gamma = 0.001 C = 20	9999	28	0,269	0,080	0,124
385	SVM gamma = 0.001 C = 30	9999	28	0,269	0,080	0,124
386	SVM gamma = 0.001 C = 40	9999	25	0,240	0,072	0,110
387	SVM gamma = 0.001 C = 50	9999	31	0,298	0,089	0,137
388	SVM gamma = 0.001 C = 100	9999	24	0,231	0,069	0,106
389	SVM gamma = 0.01 C = 10	9999	26	0,250	0,075	0,115
390	SVM gamma = 0.01 C = 20	9999	23	0,221	0,066	0,102
391	SVM gamma = 0.01 C = 30	9999	20	0,192	0,057	0,088
392	SVM gamma = 0.01 C = 40	9999	19	0,183	0,054	0,084
393	SVM gamma = 0.01 C = 50	9999	20	0,192	0,057	0,088
394	SVM gamma = 0.01 C = 100	9999	25	0,240	0,072	0,110
395	SVM gamma = 0.1 C = 10	9999	27	0,260	0,077	0,119
396	SVM gamma = 0.1 C = 20	9999	30	0,288	0,086	0,132
397	SVM gamma = 0.1 C = 30	9999	29	0,279	0,083	0,128
398	SVM gamma = 0.1 C = 40	9999	30	0,288	0,086	0,132
399	SVM gamma = 0.1 C = 50	9999	31	0,298	0,089	0,137
400	SVM gamma = 0.1 C = 100	9999	31	0,298	0,089	0,137

Table 8.17 – Produced predictive models results with different partitioned datasets

8.8. R CODE

```
############## The main analysis of data ##############

# Reading data
setwd("~/Documents/Master Thesis/Practical part")
Caravan <- read.table("ticdata2000.txt")

# Used packages
library(reshape)
library(Hmisc)
library(ggplot2)
library(corrplot)
library(dplyr)
library(caret)
library(gridExtra)
library(MASS)
library(ROSE)
library(rpart)
library(RSNNS)
library(e1071)
library(Boruta)
library(FSelector)
library(mlr)
library(ggvis)
library(rJava)


# Renaming variables
Caravan <- rename(Caravan,c(V1="MOSTYPE"))
Caravan <- rename(Caravan,c(V2="MAANTHUI"))
Caravan <- rename(Caravan,c(V3="MGEMOMV"))
Caravan <- rename(Caravan,c(V4="MGEMLEEF"))
Caravan <- rename(Caravan,c(V5="MOSHOOFD"))
Caravan <- rename(Caravan,c(V6="MGODRK"))
Caravan <- rename(Caravan,c(V7="MGODPR"))
Caravan <- rename(Caravan,c(V8="MGODOV"))
Caravan <- rename(Caravan,c(V9="MGODGE"))
Caravan <- rename(Caravan,c(V10="MRELGE"))
Caravan <- rename(Caravan,c(V11="MRELSA"))
Caravan <- rename(Caravan,c(V12="MRELOV"))
Caravan <- rename(Caravan,c(V13="MFALLEEN"))
Caravan <- rename(Caravan,c(V14="MFGEKIND"))
Caravan <- rename(Caravan,c(V15="MFWEKIND"))
Caravan <- rename(Caravan,c(V16="MOPLHOOG"))
Caravan <- rename(Caravan,c(V17="MOPLMIDD"))
Caravan <- rename(Caravan,c(V18="MOPLLAAG"))
Caravan <- rename(Caravan,c(V19="MBERHOOG"))
Caravan <- rename(Caravan,c(V20="MBERZELF"))
Caravan <- rename(Caravan,c(V21="MBERBOER"))
Caravan <- rename(Caravan,c(V22="MBERMIDD"))
Caravan <- rename(Caravan,c(V23="MBERARBG"))
Caravan <- rename(Caravan,c(V24="MBERARBO"))
Caravan <- rename(Caravan,c(V25="MSKA"))
Caravan <- rename(Caravan,c(V26="MSKB1"))
Caravan <- rename(Caravan,c(V27="MSKB2"))
Caravan <- rename(Caravan,c(V28="MSKC"))
Caravan <- rename(Caravan,c(V29="MSKD"))
Caravan <- rename(Caravan,c(V30="MHHUUR"))
Caravan <- rename(Caravan,c(V31="MHKOOP"))
Caravan <- rename(Caravan,c(V32="MAUT1"))
Caravan <- rename(Caravan,c(V33="MAUT2"))
Caravan <- rename(Caravan,c(V34="MAUT0"))
Caravan <- rename(Caravan,c(V35="MZFONDS"))
Caravan <- rename(Caravan,c(V36="MZPART"))
Caravan <- rename(Caravan,c(V37="MINKM30"))
```

```r
Caravan <- rename(Caravan,c(V38="MINK3045"))
Caravan <- rename(Caravan,c(V39="MINK4575"))
Caravan <- rename(Caravan,c(V40="MINK7512"))
Caravan <- rename(Caravan,c(V41="MINK123M"))
Caravan <- rename(Caravan,c(V42="MINKGEM"))
Caravan <- rename(Caravan,c(V43="MKOOPKLA"))
Caravan <- rename(Caravan,c(V44="PWAPART"))
Caravan <- rename(Caravan,c(V45="PWABEDR"))
Caravan <- rename(Caravan,c(V46="PWALAND"))
Caravan <- rename(Caravan,c(V47="PPERSAUT"))
Caravan <- rename(Caravan,c(V48="PBESAUT"))
Caravan <- rename(Caravan,c(V49="PMOTSCO"))
Caravan <- rename(Caravan,c(V50="PVRAAUT"))
Caravan <- rename(Caravan,c(V51="PAANHANG"))
Caravan <- rename(Caravan,c(V52="PTRACTOR"))
Caravan <- rename(Caravan,c(V53="PWERKT"))
Caravan <- rename(Caravan,c(V54="PBROM"))
Caravan <- rename(Caravan,c(V55="PLEVEN"))
Caravan <- rename(Caravan,c(V56="PPERSONG"))
Caravan <- rename(Caravan,c(V57="PGEZONG"))
Caravan <- rename(Caravan,c(V58="PWAOREG"))
Caravan <- rename(Caravan,c(V59="PBRAND"))
Caravan <- rename(Caravan,c(V60="PZEILPL"))
Caravan <- rename(Caravan,c(V61="PPLEZIER"))
Caravan <- rename(Caravan,c(V62="PFIETS"))
Caravan <- rename(Caravan,c(V63="PINBOED"))
Caravan <- rename(Caravan,c(V64="PBYSTAND"))
Caravan <- rename(Caravan,c(V65="AWAPART"))
Caravan <- rename(Caravan,c(V66="AWABEDR"))
Caravan <- rename(Caravan,c(V67="AWALAND"))
Caravan <- rename(Caravan,c(V68="APERSAUT"))
Caravan <- rename(Caravan,c(V69="ABESAUT"))
Caravan <- rename(Caravan,c(V70="AMOTSCO"))
Caravan <- rename(Caravan,c(V71="AVRAAUT"))
Caravan <- rename(Caravan,c(V72="AAANHANG"))
Caravan <- rename(Caravan,c(V73="ATRACTOR"))
Caravan <- rename(Caravan,c(V74="AWERKT"))
Caravan <- rename(Caravan,c(V75="ABROM"))
Caravan <- rename(Caravan,c(V76="ALEVEN"))
Caravan <- rename(Caravan,c(V77="APERSONG"))
Caravan <- rename(Caravan,c(V78="AGEZONG"))
Caravan <- rename(Caravan,c(V79="AWAOREG"))
Caravan <- rename(Caravan,c(V80="ABRAND"))
Caravan <- rename(Caravan,c(V81="AZEILPL"))
Caravan <- rename(Caravan,c(V82="APLEZIER"))
Caravan <- rename(Caravan,c(V83="AFIETS"))
Caravan <- rename(Caravan,c(V84="AINBOED"))
Caravan <- rename(Caravan,c(V85="ABYSTAND"))
Caravan <- rename(Caravan,c(V86="Purchase"))


# Convert target variable to factor with yes and no
Caravan$Purchase <- factor(Caravan$Purchase,
                levels = c(0,1),
                labels = c("No", "Yes"))
str(Caravan$Purchase)

# Basic analysis
summary(Caravan)
dim(Caravan)
names(Caravan)
describe(Caravan)
str(Caravan)
levels(Caravan[,86]) # target variable has to be factor
dim(Caravan)

# Checking if there is missing data
```

```
number_rows <- nrow(Caravan)
complete_rows  <- sum(complete.cases(Caravan))
number_rows==complete_rows #no missing data

# checking insurance holders in the data set
describe(Caravan[86]=='No')
describe(Caravan[86]=='Yes') # very imbalanced dataset

# Purchase plot
ggplot(Caravan, aes(x = Caravan$Purchase)) +  geom_bar(fill="#99BB77") +
  labs(title="Purchase of Caravan Policy ") +
  labs(x="Caravan Policy", y="# Customers") +
  theme(plot.title = element_text(hjust = 0.5))

# Histograms by target variable, run separately for all variables
n <- 1
ggplot(Caravan, aes(x = factor(Caravan[,n]),  fill = Purchase)) + geom_bar() +
  scale_fill_manual(values=c("#99BB77", "#E69F00")) +
  labs(x=names(Caravan[n]), y="# Customers")


######## creating excel file with insurance holders in each variable group #########
Caravan1 <- Caravan

# target variable with 0 and 1
Caravan1$Purchase_num <- ifelse(Caravan1$Purchase=='No',0,1)

for (i in 1:85){
  Caravan1[,i] <- factor(Caravan1[,i])
}


# Save variable names in vector
variables <- colnames(Caravan1)

# create data frame for results
DF <- as.data.frame(c(1:5))
DF <- t(DF)
colnames(DF) <- c("Level", "Variable" ,"Number", "Percentage", "Standard Deviation")
DF <- DF[-1,]

# fulfill the database with the results
for (n in 1:85){
  DFresults <- data.frame(
    level=levels(Caravan1[,n]),
    variable=variables[n],
    number=tapply(Caravan1$Purchase_num, Caravan1[,n], length),
    percentage=round(tapply(Caravan1$Purchase_num, Caravan1[,n], mean)*100,2),
    sd=tapply(Caravan1$Purchase_num, Caravan1[,n], sd)
  )
DF<-  rbind(DF,DFresults)
}

###############################################################################


#################### Spearman Correlation ##############################

CaravanNum <- within(Caravan, Purchase <- as.numeric(Purchase)) # target variable converts to numeric
SpearmanCorTarget <- cor(CaravanNum[1:85],CaravanNum[86], method="spearman") # correlation between target variable and
 independent variables

SpearmanCorData <- cor(CaravanNum, method="spearman", use="complete.obs") # correlation between all variables

corrplot(SpearmanCorData, method="ellipse", order = "hclust", tl.cex=0.7) # correlation plot

write.csv2(SpearmanCorData, "SpearmanCorDataFinal.csv")
```

```r
write.csv2(SpearmanCorTarget, "SpearmanCorTargetFinal.csv")

################### ELIMINATION OF VARIABLES  ##########################
#Elimination of variables that are correlated to each other, and also looking to variables correlations with target variable

CaravanV2<-Caravan

#1 not useful variables with religion
#--   MGODRK, MGODPR, MGODOV, MGODGE

#2 married, single and etc
cor_martial_var <- cor(Caravan[c("MRELGE","MRELSA", "MRELOV", "MFALLEEN")], method="spearman")
# leave only variable that presents singles, then we know if single or no
#-- MRELGE, MRELSA, MRELOV

#3 have kids or no
cor_kids_var <- cor(Caravan[c("MFGEKIND", "MFWEKIND")], method="spearman")
# leave variable MFWEKIND with children because higher correlation with target variable
#--  MFGEKIND

#4 education
cor_education_var <- cor(Caravan[c("MOPLHOOG", "MOPLMIDD", "MOPLLAAG")], method="spearman")
# leave all

#5 group
cor_group_var <- cor(Caravan[c("MBERHOOG", "MBERZELF", "MBERBOER", "MBERMIDD", "MBERARBG",
                     "MBERARBO")], method="spearman")
# variables not highly correlated

#6 socialClass
cor_social_var <- cor(Caravan[c("MSKA","MSKB1", "MSKB2", "MSKC", "MSKD")],
               method="spearman")
# variables not highly correlated

#7 socialClass with Group variables
cor_group_social_var <- cor(Caravan[c("MBERHOOG", "MBERZELF", "MBERBOER", "MBERMIDD", "MBERARBG",
                       "MBERARBO", "MSKA","MSKB1", "MSKB2", "MSKC", "MSKD")],
                  method="spearman")

#8 social and income
cor_social_income_var <- cor(Caravan[c("MSKA","MSKB1", "MSKB2", "MSKC", "MSKD", "MINKM30",
                        "MINK3045", "MINK4575", "MINK7512", "MINK123M")],
                  method="spearman")

#9 incomes and avgIncome
cor_incomes_var <- cor(Caravan[c("MINKM30", "MINK3045", "MINK4575", "MINK7512", "MINK123M",
                     "MINKGEM", "MKOOPKLA")], method="spearman")

#10 rented home,
cor_rented_var <- cor(Caravan[c("MHHUUR", "MHKOOP")], method="spearman")
#very higly correlated, leave MHKOOP Home owners
#-- MHHUUR

#11 cars
cor_cars_var <- cor(Caravan[c("MAUT1", "MAUT2", "MAUT0")], method="spearman")
# leave variable no car, then we know has at least one car or no
# MAUT1, MAUT2

#12 healthInsurnceType
cor_healthInsurance_var <- cor(Caravan[c("MZFONDS", "MZPART")], method="spearman")
# highly correlated, leave MZPART, and we will know another one part that has public health insurance
# MZFONDS

#13 Customer type
cor_customerType_var <- cor(Caravan[c("MOSTYPE", "MOSHOOFD")], method="spearman")

# MGODRK, MGODPR, MGODOV, MGODGE, MRELGE, MRELSA, MRELOV, MFGEKIND, MHHUUR, MAUT1, MAUT2, MZFONDS
```

```
CaravanV2 <- subset(Caravan, select = -c(MGODRK, MGODPR, MGODOV, MGODGE, MRELGE, MRELSA,
                              MRELOV, MFGEKIND, MHHUUR, MAUT1, MAUT2, MZFONDS))

#14 Number and contribution of policies correlation
colnames(CaravanV2)
names(CaravanV2[32])
cor_policies_vars <- cor(CaravanV2[32:73], method="spearman") # very higly correlated
write.csv2(cor_policies_vars, "PoliciesCorrelationFinal.csv")

# create correlation plot
corrplot(cor_policies_vars, method="circle")
corrplot(cor_policies_vars, method="ellipse", tl.cex=0.7)

################# VARIABLES Transformation - CaravanV3 ##################################

dim(CaravanV2)
names(CaravanV2[74]) #74 variable Purchase
CaravanV3<-CaravanV2

# Create Product-Contribution interaction variables
CaravanV3$IAANHANG <- CaravanV3$PAANHANG*CaravanV3$AAANHANG
CaravanV3$ITRACTOR <- CaravanV3$PTRACTOR*CaravanV3$ATRACTOR
CaravanV3$IWERKT <- CaravanV3$PWERKT*CaravanV3$AWERKT
CaravanV3$IBROM <- CaravanV3$PBROM*CaravanV3$ABROM
CaravanV3$ILEVEN <- CaravanV3$PLEVEN*CaravanV3$ALEVEN
CaravanV3$IPERSONG <- CaravanV3$PPERSONG*CaravanV3$APERSONG
CaravanV3$IGEZONG <- CaravanV3$PGEZONG*CaravanV3$AGEZONG
CaravanV3$IWAOREG <- CaravanV3$PWAOREG*CaravanV3$AWAOREG
CaravanV3$IBRAND <- CaravanV3$PBRAND*CaravanV3$ABRAND
CaravanV3$IZEILPL <- CaravanV3$PZEILPL*CaravanV3$AZEILPL
CaravanV3$IPLEZIER <- CaravanV3$PPLEZIER*CaravanV3$APLEZIER
CaravanV3$IFIETS <- CaravanV3$PFIETS*CaravanV3$AFIETS
CaravanV3$IINBOED <- CaravanV3$PINBOED*CaravanV3$AINBOED
CaravanV3$IBYSTAND <- CaravanV3$PBYSTAND*CaravanV3$ABYSTAND

dim(CaravanV3)
names(CaravanV3)
CaravanV3 <- subset(CaravanV3, select = -c(32:73))
names(CaravanV3[32]) # 32 Purchase

###########################################################################################
############################### Variable Selection #################################
###########################################################################################

######## INFORMATION GAIN and Chi-Squared ########

trainTask <- makeClassifTask(data = CaravanV3,target = "Purchase", positive = "Yes")
#Feature importance
im_feat <- generateFilterValuesData(trainTask, method = c("information.gain","chi.squared"))
#plots
plotFilterValues(im_feat,n.show = 14, feat.type.cols = TRUE) + scale_fill_manual(values="#99BB77")
#data
im_feat$data
#write
write.csv2(im_feat$data, "informationGain_ChiSquared.csv")

#another way
im2<-information.gain(Purchase~.,CaravanV3)

###### STEPWISE GLM ######

fullGLM <- glm(Purchase~., CaravanV3, family=binomial)
summary(fullGLM)
step <- stepAIC(fullGLM, directions='both')
step$anova
summary(step)
#final model Call: glm(formula = Purchase ~ MGEMLEEF + MOPLMIDD + MOPLLAAG + MBERBOER +
```

```
# MBERMIDD + MSKC + MHKOOP + MAUT0 + MINK123M + MINKGEM + IWERKT +
#IBROM + IWAOREG + IBRAND + IPLEZIER + IFIETS + IBYSTAND,
#family = binomial, data = CaravanV3)


######### Boruta ##########

set.seed(123)
boruta.train <- Boruta(Purchase~., data = CaravanV3, doTrace = 2)
print(boruta.train)
final.boruta <- TentativeRoughFix(boruta.train)

getSelectedAttributes(final.boruta, withTentative = F)
# [1] "MOSTYPE"  "MGEMOMV"  "MGEMLEEF" "MOSHOOFD" "MFALLEEN" "MFWEKIND" "MOPLHOOG" "MOPLMIDD"
# [9] "MOPLLAAG" "MBERHOOG" "MBERZELF" "MBERBOER" "MBERMIDD" "MBERARBG" "MBERARBO" "MSKA"
# [17] "MSKB1"    "MSKB2"    "MSKC"     "MSKD"     "MHKOOP"   "MAUT0"    "MZPART"   "MINKM30"
# [25] "MINK3045" "MINK4575" "MINK7512" "MINK123M" "MINKGEM"  "MKOOPKLA" "ITRACTOR" "IBROM"
# [33] "IBRAND"   "IPLEZIER" "IBYSTAND"

boruta.df <- attStats(final.boruta)

write.csv2(boruta.df, "borutadf.csv")

plot(boruta.train, xlab = "", xaxt = "n")
lz<-lapply(1:ncol(boruta.train$ImpHistory),function(i)
  boruta.train$ImpHistory[is.finite(boruta.train$ImpHistory[,i]),i])
names(lz) <- colnames(boruta.train$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
     at = 1:ncol(boruta.train$ImpHistory), cex.axis = 0.7)


######################################################################################
################################## Variable Subsets ##################################
######################################################################################

#----- Information Gain and Chi-Squared -----#
#select all that information gain is not zero

CaravanSet1 <- subset(CaravanV3, select = c(IBRAND, MINKGEM, MKOOPKLA, MINKM30, MOSTYPE, MOSHOOFD,
                        MOPLLAAG, MHKOOP, MINK4575, MAUT0, IPLEZIER, MOPLHOOG,
                        MSKA,MSKD, Purchase))
#Purchase 15 variable

# checking selected variables correlation
SpearmanCorCaravanSet1 <- cor(CaravanSet1[,-15], method="spearman", use="complete.obs")
corrplot(SpearmanCorCaravanSet1, method="ellipse", order = "hclust", tl.cex=0.7)

#----- Regression Stepwise -----#
# select final model

CaravanSet2 <- subset(CaravanV3, select = c(MGEMLEEF, MOPLMIDD, MOPLLAAG, MBERBOER, MBERMIDD, MSKC,
                        MHKOOP, MAUT0, MINK123M, MINKGEM, IWERKT, IBROM, IWAOREG,
                        IBRAND, IPLEZIER, IFIETS, IBYSTAND, Purchase))
#Purchase variable 18

SpearmanCorCaravanSet2 <- cor(CaravanSet2[,-18], method="spearman", use="complete.obs")
corrplot(SpearmanCorCaravanSet2, method="ellipse", order = "hclust", tl.cex=0.7)

#----- Boruta -----#
# select all confirmed variables

CaravanSet3 <- subset(CaravanV3, select = c(IPLEZIER, MOSTYPE, MOPLLAAG, MOPLHOOG, MSKA, MOSHOOFD,
                        MFALLEEN, MSKC, IBRAND, MBERARBO, MOPLMIDD, MINKM30,
                        MBERARBG, MZPART, MINKGEM, MKOOPKLA, MAUT0, MINK7512,
                        MFWEKIND, MHKOOP, MBERMIDD, MBERHOOG, MBERBOER, MSKD,
                        MGEMLEEF, MINK4575, MSKB1, MINK3045, MBERZELF, MGEMOMV,
```

```
                        MSKB2, ITRACTOR, IBROM, IBYSTAND, Purchase))
#Purchase variable 35

SpearmanCorCaravanSet3 <- cor(CaravanSet3[,-35], method="spearman", use="complete.obs")
corrplot(SpearmanCorCaravanSet3, method="ellipse", order = "hclust", tl.cex=0.7)

#----- Variable Mix based on all variable selection methods -----#

### MIXED 1 - from everything, everything mixed
CaravanMixed1 <-  subset(CaravanV3, select = c(IPLEZIER, MOSTYPE, MOPLLAAG, IBRAND, MBERARBO,
                          MZPART, MINKGEM, MKOOPKLA, MAUT0, MHKOOP, MBERMIDD,
                          MGEMLEEF, ITRACTOR, IBROM, IBYSTAND, MINK123M,
                          IZEILPL, Purchase))
#Purchase variable 18

SpearmanCorCaravanMixed1 <- cor(CaravanMixed1[,-18], method="spearman", use="complete.obs")
corrplot(SpearmanCorCaravanMixed1, method="ellipse", order = "hclust", tl.cex=0.7)

### MIXED 2 - all interaction and mixed sociodemographic
CaravanMixed2 <-  subset(CaravanV3, select = c(MOSTYPE, MOPLLAAG, MBERARBO, MZPART, MINKGEM,
                          MKOOPKLA,MAUT0, MHKOOP, MBERMIDD, MGEMLEEF,
                          MINK123M, IPLEZIER, IBRAND, ITRACTOR, IBROM,
                          IBYSTAND, IWAOREG, IGEZONG, IFIETS, IAANHANG,
                          IINBOED, ILEVEN, IWERKT, IPERSONG, IZEILPL,
                          Purchase))
#Purchase variable 26

SpearmanCorCaravanMixed2 <- cor(CaravanMixed2[,-26], method="spearman", use="complete.obs")
corrplot(SpearmanCorCaravanMixed2, method="ellipse", order = "hclust", tl.cex=0.7)


### MIXED 3 - all interaction and power purchasing and some other from information gain
###(based on logic)
CaravanMixed3 <-  subset(CaravanV3, select = c(MKOOPKLA, MINKGEM, MOPLLAAG, MHKOOP, MAUT0,
                          IPLEZIER, IBRAND, ITRACTOR, IBROM,
                          IBYSTAND, IWAOREG, IGEZONG, IFIETS, IAANHANG,
                          IINBOED, ILEVEN, IWERKT, IPERSONG, IZEILPL,
                          Purchase))


SpearmanCorCaravanMixed3 <- cor(CaravanMixed3[,-20], method="spearman", use="complete.obs")
corrplot(SpearmanCorCaravanMixed3, method="ellipse", order = "hclust", tl.cex=0.7)


#######################################################################################
############################    Splitting Data    #################################
#######################################################################################
s <- 999
set.seed(s)

#By default, createDataPartition does a stratified random split of the data.
train.index <- createDataPartition(CaravanV3$Purchase, p = .7, list = FALSE)
str(train.index)

#1
TrainCaravanV3 <- CaravanV3[train.index,]
TestCaravanV3 <- CaravanV3[-train.index,]

describe(TrainCaravanV3$Purchase=='Yes') #Train: 4076 customers, 244-Yes, 3832-No
describe(TestCaravanV3$Purchase=='Yes')  # Test: 1746 customers, 104-Yes, 1642-No


#2
TrainCaravanSet1 <- CaravanSet1[train.index,]
TestCaravanSet1 <- CaravanSet1[-train.index,]

#3
```

```
TrainCaravanSet2 <- CaravanSet2[train.index,]
TestCaravanSet2 <- CaravanSet2[-train.index,]

#4
TrainCaravanSet3 <- CaravanSet3[train.index,]
TestCaravanSet3 <- CaravanSet3[-train.index,]

#5
TrainCaravanMixed1 <- CaravanMixed1[train.index,]
TestCaravanMixed1 <- CaravanMixed1[-train.index,]

#6
TrainCaravanMixed2 <- CaravanMixed2[train.index,]
TestCaravanMixed2 <- CaravanMixed2[-train.index,]

#7
TrainCaravanMixed3 <- CaravanMixed3[train.index,]
TestCaravanMixed3 <- CaravanMixed3[-train.index,]


##########################################################################################
###########################     Preparing Modelling     ###############################
##########################################################################################

# for the methods that need to separate target variable
ytrain <- ifelse(TrainCaravanV3$Purchase=="Yes",1,0)
ytest <- ifelse(TestCaravanV3$Purchase=="Yes",1,0)

# the number of true positives in the test set
test_truepositives <- sum(ytest)

trainpredictions <- as.data.frame(ytrain)
testpredictions <- as.data.frame(ytest)
colnames(trainpredictions) <- "trainy"
colnames(testpredictions) <- "testy"


# ceate data frame for results
combinedresults <- as.data.frame(c(1:9))
combinedresults <- t(combinedresults)
colnames(combinedresults) <- c("Method_Name", "Variable_Set" ,"Test_Set_True_Positives", "TP_Identified", "Baseline", "Lift",
 "Sensitivity", "Precision", "F-measure")
combinedresults <- combinedresults[-1,]


#select top 20% of observations with highest probability to purchase insurance
# test set has 1746 rows
testsetsize <- nrow(TestCaravanV3)

# 20% is 349 predictions:
top20_test <- round((testsetsize*.2), digits=0)

# selecting 349 observations from the data randomly, 21 pcorrectly identified customers should appear:
positivesCustomers <- nrow(CaravanV3[CaravanV3$Purchase=="Yes",]) #348 Yes in total
positivesrate <- round(positivesCustomers/nrow(CaravanV3), digits=4) #percentage of Yes in the dataset
positivesrate*top20_test
# 20.8702 round to 21
Baseline <- 21
# the model has to reached more tahn 21, otherwise the model is bad



# function for results
evaluate <- function(methodname, variableset, predictions, combinedresults){
 lift <- as.data.frame(cbind(ytest, predictions))
 colnames(lift) <- c("y", "yhat")
 order <- lift[order(lift$yhat,decreasing=TRUE),]
```

```
 liftcut <- order[1:349,]
 TP <- sum(liftcut$y)
 Lift <- TP-Baseline
 Sensitivity <- sum(liftcut$y)/sum(ytest)
 Precision <- sum(liftcut$y)/349
 FMeasure <- (2*(sum(liftcut$y)/349)*(sum(liftcut$y)/sum(ytest)))/((sum(liftcut$y)/349)+(sum(liftcut$y)/sum(ytest)))
 results <- as.vector(c(methodname, variableset, test_truepositives, TP, Baseline, Lift, Sensitivity,
                   Precision, FMeasure))
}


###########################################################################################
##################################     Modelling    ##################################
###########################################################################################

######################### Selecting variables combination  ##########################

########## 1 - All socio-demographic & policies interactions - CaravanV3 ###########

variableset <- "1 - All socio-demographic & policies interactions"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanV3)=="Purchase")

glm1 <- glm(Purchase~., data=TrainCaravanV3, family=binomial)
predictions <- predict(glm1, newdata=TestCaravanV3[,-32], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

# ROC curve
with(TestCaravanV3,roc.curve(Purchase,predictions,col=1))

#--- Decision Tree CART ---#

set.seed(s)
methodname <- "CART - cp=0.0022, minsplit=9"

cart1 <- rpart(Purchase~.,data=TrainCaravanV3, method="class",
          control = rpart.control(cp = 0.0022, minsplit = 9))
predictions<- as.data.frame(predict(cart1, newdata=TestCaravanV3[,-32], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


methodname <- "CART - default"
cart1.1 <- rpart(Purchase~.,data=TrainCaravanV3, method="class")
predictions<- as.data.frame(predict(cart1.1, newdata=TestCaravanV3[,-32], type="prob"))[,2]
results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart1)
printcp(cart1)

#automatically select the complexity parameter associated with the smallest cross-validated error.
prunecart1.1<- prune(cart1.1, cp=cart1.1$cptable[which.min(cart1.1$cptable[,"xerror"]),"CP"])
predictions<- as.data.frame(predict(prunecart1.1, newdata=TestCaravanV3[,-32]))[,2]
results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

# ROC curve
with(TestCaravanV3,roc.curve(Purchase,predictions,col=1))
```

```
#--- Neural Network - MLP ---#
# package RSNNS, function mlp()

activationFunctions <- RSNNS::getSnnsRFunctionTable()
#Std_Backpropagation , BackpropBatch, BackpropChunk, BackpropClassJogChunk, BackpropMomentum
#BackpropWeightDecay, TimeDelayBackprop
#learnFunc="Std_Backpropagation"

set.seed(s)

for (i in 1:20){
 methodname <- paste("MLP -", i, "neuron",sep=" ")
 setSnnsRSeedValue(s)
 mlpi <- mlp(TrainCaravanV3[,-32], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
       initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanV3[,-32], targetsTest = ytest)
 predictions <- predict(mlpi, TestCaravanV3[,-32])
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}

#methodname <- "MLP - n=10, c(0.3, 0.000023)"
#mlp1 <- mlp(TrainCaravanV3[,-32], ytrain, size=10, maxit = 300, learnFunc="Std_Backpropagation",
#          learnFuncParams = c(0.3, 0.000023),  hiddenActFunc = "Act_TanH",
#          outputActFunc = "Act_Logistic",
#          inputsTest = TestCaravanV3[,-32], targetsTest = ytest)
#predictions <- predict(mlp1, TestCaravanV3[,-32])


# ROC curve
with(TestCaravanV3,roc.curve(Purchase,predictions,col=1))


##########--- Support Vector Machine svm ---##########

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
for (i in 1:6) {
 methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
 set.seed(s)
 svmi <-e1071:::svm(Purchase~., data = TrainCaravanV3, type = "C-classification",
         kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
         probability = TRUE)
 predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanV3[,-32], probability = TRUE), "probabilities"))[,2]
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}
}

#####
#wts <-  100 / table(TrainCaravanV3$Purchase)
#svm4 <-e1071:::svm(Purchase~., data = TrainCaravanV3,
#          probability = TRUE)
#predictions <- as.data.frame(attr(predict(svm4, newdata=TestCaravanV3[,-32], probability = TRUE), "probabilities"))[,2]
#require(e1071)
#mytune <- tune(svm, Purchase~., data=TrainCaravanV3,class.weights = wts,
#        probability=TRUE, ranges=list(gamma=2^(-8:0), cost=10^(-2:4)),
#        scale=FALSE,
#        tunecontrol=tune.control(best.model=TRUE, performances=TRUE, sampling="cross",
#                     cross=5))

#obj <- tune(svm, train.x=TrainCaravanV3[,-32], train.y=ytrain, kernel="radial",
#      ranges=list(cost=10^(-2:2), gamma=seq(0, 100, 0.5)),
#      tunecontrol = tune.control(sampling = "fix"))
#best parameters:
#  cost gamma
```

```
#   1   0.5
#best performance: 0.06056629


########## 2 - Information Gain & Chi-Squared - CaravanSet1 #############

variableset <- "Information Gain"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanSet1)=="Purchase")

glm2.1 <- glm(Purchase~., data=TrainCaravanSet1, family=binomial)
predictions <- predict(glm2.1, newdata=TestCaravanSet1[,-15], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

##########--- Decision Tree CART ---##########

methodname <- "CART"

set.seed(s)
cart2.1 <- rpart(Purchase~.,data=TrainCaravanSet1, method="class",
            control = rpart.control(cp = 0.001, minsplit = 12))
predictions<- as.data.frame(predict(cart2.1, newdata=TestCaravanSet1[,-15], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


set.seed(S)
methodname <- "CART - default"
cart2.2 <- rpart(Purchase~.,data=TrainCaravanSet1, method="class")
predictions<- as.data.frame(predict(cart2.2, newdata=TestCaravanSet1[,-15], type="prob"))[,2]
results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart2.1)
printcp(cart2.1)
cart2.1$cptable

# automatically select the complexity parameter associated with the smallest cross-validated error.
prunecart1.1<- prune(cart1.1, cp=cart1.1$cptable[which.min(cart1.1$cptable[,"xerror"]),"CP"])
predictions<- as.data.frame(predict(prunecart1.1, newdata=TestCaravanV3[,-32]))[,2]

#----- MLP ------#

for (i in 1:20){
  methodname <- paste("MLP -", i, "neuron",sep=" ")
  setSnnsRSeedValue(s)
  mlpi <- mlp(TrainCaravanSet1[,-15], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
          initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanSet1[,-15], targetsTest = ytest)
  predictions <- predict(mlpi, TestCaravanSet1[,-15])
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
  combinedresults <- rbind(combinedresults, results1)
}


#--- SVM ---#

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
```

```
for (ii in 1:3) {
 for (i in 1:6) {
   methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
 set.seed(s)
 svmi <-e1071:::svm(Purchase~., data = TrainCaravanSet1, type = "C-classification",
                  kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
                  probability = TRUE)
 predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanSet1[,-15], probability = TRUE), "probabilities"))[,2]
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}
}


########## 3 - Stepwise regression - CaravanSet2 ##############################

variableset <- "3 - Stepwise regression"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanSet2)=="Purchase")

glm3.1 <- glm(Purchase~., data=TrainCaravanSet2, family=binomial)
predictions <- predict(glm3.1, newdata=TestCaravanSet2[,-18], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


#--- Decision Tree CART ---#

methodname <- "CART"
set.seed(S)
cart3.1 <- rpart(Purchase~.,data=TrainCaravanSet2, method="class",
             control = rpart.control(cp = 0.001, minsplit = 12))
predictions<- as.data.frame(predict(cart3.1, newdata=TestCaravanSet2[,-18], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


set.seed(S)
methodname <- "CART - default"
cart3.2 <- rpart(Purchase~.,data=TrainCaravanSet2, method="class")
predictions<- as.data.frame(predict(cart3.2, newdata=TestCaravanSet2[,-18], type="prob"))[,2]
results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart3.1)
printcp(cart3.1)
cart3.1$cptable

#----- MLP ------#

for (i in 1:20){
 methodname <- paste("MLP -", i, "neuron",sep=" ")
 setSnnsRSeedValue(s)
 mlpi <- mlp(TrainCaravanSet2[,-18], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
       initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanSet2[,-18], targetsTest = ytest)
 predictions <- predict(mlpi, TestCaravanSet2[,-18])
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}
```

```r
#--- SVM ---#

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
 for (i in 1:6) {
  methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
  set.seed(s)
 svmi <-e1071:::svm(Purchase~., data = TrainCaravanSet2, type = "C-classification",
                 kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
                 probability = TRUE)
 predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanSet2[,-18], probability = TRUE), "probabilities"))[,2]
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}
}


########## 4 - Boruta - CaravanSet3 ##############################

variableset <- "4 - Boruta"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanSet3)=="Purchase")

glm4.1 <- glm(Purchase~., data=TrainCaravanSet3, family=binomial)
predictions <- predict(glm4.1, newdata=TestCaravanSet3[,-35], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


#--- Decision Tree CART ---#

methodname <- "CART"
set.seed(s)
cart4.1 <- rpart(Purchase~.,data=TrainCaravanSet3, method="class",
              control = rpart.control(cp = 0.0011, minsplit = 12))
predictions<- as.data.frame(predict(cart4.1, newdata=TestCaravanSet3[,-35], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart4.1)
printcp(cart1)
cart4.1$cptable


#----- MLP ------#

for (i in 1:20){
 methodname <- paste("MLP -", i, "neuron",sep=" ")
 setSnnsRSeedValue(s)
 mlpi <- mlp(TrainCaravanSet3[,-35], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
        initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanSet3[,-35], targetsTest = ytest)
 predictions <- predict(mlpi, TestCaravanSet3[,-35])
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}

#--- SVM ---#
```

```
GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
 for (i in 1:6) {
  methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
  set.seed(s)
 svmi <-e1071:::svm(Purchase~., data = TrainCaravanSet3, type = "C-classification",
                kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
                probability = TRUE)
 predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanSet3[,-35], probability = TRUE), "probabilities"))[,2]
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}
}


########## 5 - Total Mix - CaravanMixed1 ###############################

variableset <- "5 - Total Mix"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanMixed1)=="Purchase")

glm5.1 <- glm(Purchase~., data=TrainCaravanMixed1, family=binomial)
predictions <- predict(glm5.1, newdata=TestCaravanMixed1[,-18], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

#--- Decision Tree CART ---#

methodname <- "CART"
set.seed(s)
cart5.1 <- rpart(Purchase~.,data=TrainCaravanMixed1, method="class",
             control = rpart.control(cp = 0.0017, minsplit = 9))
predictions<- as.data.frame(predict(cart5.1, newdata=TestCaravanMixed1[,-18], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart5.1)
printcp(cart1)
cart5.1$cptable

#----- MLP ------#

for (i in 1:20){
 methodname <- paste("MLP -", i, "neuron",sep=" ")
 setSnnsRSeedValue(s)
 mlpi <- mlp(TrainCaravanMixed1[,-18], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
          initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanMixed1[,-18], targetsTest = ytest)
 predictions <- predict(mlpi, TestCaravanMixed1[,-18])
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}


#--- SVM ---#

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
 for (i in 1:6) {
```

```
    methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
    set.seed(s)
  svmi <-e1071:::svm(Purchase~., data = TrainCaravanMixed1, type = "C-classification",
                  kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
                  probability = TRUE)
  predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanMixed1[,-18], probability = TRUE), "probabilities"))[,2]
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
  combinedresults <- rbind(combinedresults, results1)
  }
}


######## 6 - All policies interactions & Socio-demographic mix - CaravanMixed2 #######

variableset <- "6 - All policies interactions & Socio-demographic mix"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanMixed2)=="Purchase")

glm6.1 <- glm(Purchase~., data=TrainCaravanMixed2, family=binomial)
predictions <- predict(glm6.1, newdata=TestCaravanMixed2[,-26], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


#--- Decision Tree CART ---#

methodname <- "CART"
set.seed(s)
cart6.1 <- rpart(Purchase~.,data=TrainCaravanMixed2, method="class",
              control = rpart.control(cp = 0.0019, minsplit = 7))
predictions<- as.data.frame(predict(cart6.1, newdata=TestCaravanMixed2[,-26], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart6.1)
printcp(cart6.1)
cart6.1$cptable

#----- MLP ------#

for (i in 1:20){
  methodname <- paste("MLP -", i, "neuron",sep=" ")
  setSnnsRSeedValue(s)
  mlpi <- mlp(TrainCaravanMixed2[,-26], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
          initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanMixed2[,-26], targetsTest = ytest)
  predictions <- predict(mlpi, TestCaravanMixed2[,-26])
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
  combinedresults <- rbind(combinedresults, results1)
}

#--- SVM ---#

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
  for (i in 1:6) {
    methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
    set.seed(s)
  svmi <-e1071:::svm(Purchase~., data = TrainCaravanMixed2, type = "C-classification",
                  kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
```

```
                 probability = TRUE)
  predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanMixed2[,-26], probability = TRUE), "probabilities"))[,2]
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
  combinedresults <- rbind(combinedresults, results1)
}
}


####### 7 - All policies interactions & Socio-demographic mix - CaravanMixed3 ########

variableset <- "7 - All policies interactions & Socio-demographic mix"

#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"
which(colnames(TestCaravanMixed3)=="Purchase")

glm7.1 <- glm(Purchase~., data=TrainCaravanMixed3, family=binomial)
predictions <- predict(glm7.1, newdata=TestCaravanMixed3[,-20], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


#--- Decision Tree CART ---#

methodname <- "CART"
set.seed(s)
cart7.1 <-  rpart(Purchase~.,data=TrainCaravanMixed4, method="class",
              control = rpart.control(cp = 0.0014, minsplit = 9))
predictions<- as.data.frame(predict(cart7.1, newdata=TestCaravanMixed3[,-20], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart7.1)
printcp(cart7.1)
cart7.1$cptable


#----- MLP ----#

for (i in 1:20){
  methodname <- paste("MLP -", i, "neuron",sep=" ")
  setSnnsRSeedValue(s)
  mlpi <- mlp(TrainCaravanMixed3[,-20], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),
        initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanMixed3[,-20], targetsTest = ytest)
  predictions <- predict(mlpi, TestCaravanMixed3[,-20])
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
  combinedresults <- rbind(combinedresults, results1)
}

#--- SVM ---#

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
  for (i in 1:6) {
    methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
    set.seed(s)
  svmi <-e1071:::svm(Purchase~., data = TrainCaravanMixed3, type = "C-classification",
                 kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
                 probability = TRUE)
  predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanMixed3[,-20], probability = TRUE), "probabilities"))[,2]
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
```

```
 combinedresults <- rbind(combinedresults, results1)
}
}

write.csv2(combinedresults, "999 - combinedresults 7sets.csv")

# Final variable subset is CaravanMixed3. Now we will run this subset on different seeds
# to get the best model.

###############################################################################
####################  Getting the best model  #############################
###############################################################################

#the seeds will be changed 10 times and the code below will run
#every time changing seed

s<-888
set.seed(s)

#By default, createDataPartition does a stratified random split of the data.
train.index <- createDataPartition(CaravanMixed4$Purchase, p = .7, list = FALSE)
str(train.index)

#8
TrainCaravanMixed3 <- CaravanMixed3[train.index,]
TestCaravanMixed3 <- CaravanMixed3[-train.index,]

describe(TrainCaravanMixed3$Purchase=='Yes') #Train: 4076 customers, 244-Yes, 3832-No
describe(TestCaravanMixed3$Purchase=='Yes')  # Test: 1746 customers, 104-Yes, 1642-No

variableset <- s

# for the methods that need to separate target variable
ytrain <- ifelse(TrainCaravanMixed3$Purchase=="Yes",1,0)
ytest <- ifelse(TestCaravanMixed3$Purchase=="Yes",1,0)

# the number of true positives in the test set
test_truepos <- sum(ytest)


#--- Logistic Regression GLM ---#

methodname <- "Logistic Regression"

set.seed(s)
glm8.1 <- glm(Purchase~., data=TrainCaravanMixed3, family=binomial)
predictions <- predict(glm8.1, newdata=TestCaravanMixed4[,-20], type='response')

summary(predictions)

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)


#--- Decision Tree CART ---#

methodname <- "CART"
set.seed(s)
cart8.1 <- rpart(Purchase~.,data=TrainCaravanMixed3, method="class",
            control = rpart.control(cp = 0.0002, minsplit = 12))
predictions<- as.data.frame(predict(cart8.1, newdata=TestCaravanMixed3[,-20], type="prob"))[,2]

results1 <- evaluate(methodname, variableset, predictions, combinedresults)
combinedresults <- rbind(combinedresults, results1)

plotcp(cart8.1)
printcp(cart8.1)
```

```
cart8.1$cptable

#----- MLP ------#

for (i in 1:20){
 methodname <- paste("MLP -", i, "neuron",sep=" ")
 setSnnsRSeedValue(s)
 mlpi <- mlp(TrainCaravanMixed3[,-20], ytrain, maxit=70, size=c(i),learnFuncParams = c(0.2, 0),

           initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanMixed3[,-20], targetsTest = ytest)
 predictions <- predict(mlpi, TestCaravanMixed4[,-20])
 results1 <- evaluate(methodname, variableset, predictions, combinedresults)
 combinedresults <- rbind(combinedresults, results1)
}


#--- SVM ---#

GammaValue <- c(0.001, 0.01, 0.1)
Cvalue <- c(10, 20, 30, 40, 50, 100)
for (ii in 1:3) {
 for (i in 1:6) {
  methodname <- paste("SVM", "gamma =", GammaValue[ii], "C =", Cvalue[i], sep=" ")
  set.seed(s)
  svmi <-e1071:::svm(Purchase~., data = TrainCaravanMixed3, type = "C-classification",
            kernel = "radial", gamma = GammaValue[ii], cost = Cvalue[i],
            probability = TRUE)
  predictions <- as.data.frame(attr(predict(svmi, newdata=TestCaravanMixed3[,-20], probability = TRUE), "probabilities"))[,2]
  results1 <- evaluate(methodname, variableset, predictions, combinedresults)
  combinedresults <- rbind(combinedresults, results1)
 }
}


write.csv2(combinedresults, "results of 10 different seeds.csv")


###### The best model MLP with 12 hidden neurons on 888 seed
s
setSnnsRSeedValue(s)
mlp_12n <- mlp(TrainCaravanMixed3[,-20], ytrain, maxit=70, size=c(12),learnFuncParams = c(0.2, 0),

          initFuncParams = c(-0.29, 0.29), inputsTest = TestCaravanMixed3[,-20], targetsTest = ytest)
predictions <- predict(mlp_12n, TestCaravanMixed3[,-20])
results1 <- evaluate(methodname, variableset, predictions, combinedresults)

# weighted SSE agains iterations plot
plotIterativeError(mlp_12n, ylim=c(200,245))
legend('topleft', c("Training Data", "Test Data"),col=c("black", "red"), lty=c(1,1), bty='n')
plotIterativeError(mlp_12n)
mlp_12n$IterativeTestError
summary(mlp_12n)
plotIterativeError(mlp_12n,log='xy')

# ROC curve
with(TestCaravanMixed4,roc.curve(Purchase,predictions,col=1))

####################################################################################
########################## Predictions on unseen data ###########################
####################################################################################

predictData <- read.table("ticeval2000.txt")
predictTarget <- read.table("tictgts2000.txt")

require(reshape)
predictData <- rename(predictData,c(V1="MOSTYPE"))
predictData <- rename(predictData,c(V2="MAANTHUI"))
```

```
predictData <- rename(predictData,c(V3="MGEMOMV"))
predictData <- rename(predictData,c(V4="MGEMLEEF"))
predictData <- rename(predictData,c(V5="MOSHOOFD"))
predictData <- rename(predictData,c(V6="MGODRK"))
predictData <- rename(predictData,c(V7="MGODPR"))
predictData <- rename(predictData,c(V8="MGODOV"))
predictData <- rename(predictData,c(V9="MGODGE"))
predictData <- rename(predictData,c(V10="MRELGE"))
predictData <- rename(predictData,c(V11="MRELSA"))
predictData <- rename(predictData,c(V12="MRELOV"))
predictData <- rename(predictData,c(V13="MFALLEEN"))
predictData <- rename(predictData,c(V14="MFGEKIND"))
predictData <- rename(predictData,c(V15="MFWEKIND"))
predictData <- rename(predictData,c(V16="MOPLHOOG"))
predictData <- rename(predictData,c(V17="MOPLMIDD"))
predictData <- rename(predictData,c(V18="MOPLLAAG"))
predictData <- rename(predictData,c(V19="MBERHOOG"))
predictData <- rename(predictData,c(V20="MBERZELF"))
predictData <- rename(predictData,c(V21="MBERBOER"))
predictData <- rename(predictData,c(V22="MBERMIDD"))
predictData <- rename(predictData,c(V23="MBERARBG"))
predictData <- rename(predictData,c(V24="MBERARBO"))
predictData <- rename(predictData,c(V25="MSKA"))
predictData <- rename(predictData,c(V26="MSKB1"))
predictData <- rename(predictData,c(V27="MSKB2"))
predictData <- rename(predictData,c(V28="MSKC"))
predictData <- rename(predictData,c(V29="MSKD"))
predictData <- rename(predictData,c(V30="MHHUUR"))
predictData <- rename(predictData,c(V31="MHKOOP"))
predictData <- rename(predictData,c(V32="MAUT1"))
predictData <- rename(predictData,c(V33="MAUT2"))
predictData <- rename(predictData,c(V34="MAUT0"))
predictData <- rename(predictData,c(V35="MZFONDS"))
predictData <- rename(predictData,c(V36="MZPART"))
predictData <- rename(predictData,c(V37="MINKM30"))
predictData <- rename(predictData,c(V38="MINK3045"))
predictData <- rename(predictData,c(V39="MINK4575"))
predictData <- rename(predictData,c(V40="MINK7512"))
predictData <- rename(predictData,c(V41="MINK123M"))
predictData <- rename(predictData,c(V42="MINKGEM"))
predictData <- rename(predictData,c(V43="MKOOPKLA"))
predictData <- rename(predictData,c(V44="PWAPART"))
predictData <- rename(predictData,c(V45="PWABEDR"))
predictData <- rename(predictData,c(V46="PWALAND"))
predictData <- rename(predictData,c(V47="PPERSAUT"))
predictData <- rename(predictData,c(V48="PBESAUT"))
predictData <- rename(predictData,c(V49="PMOTSCO"))
predictData <- rename(predictData,c(V50="PVRAAUT"))
predictData <- rename(predictData,c(V51="PAANHANG"))
predictData <- rename(predictData,c(V52="PTRACTOR"))
predictData <- rename(predictData,c(V53="PWERKT"))
predictData <- rename(predictData,c(V54="PBROM"))
predictData <- rename(predictData,c(V55="PLEVEN"))
predictData <- rename(predictData,c(V56="PPERSONG"))
predictData <- rename(predictData,c(V57="PGEZONG"))
predictData <- rename(predictData,c(V58="PWAOREG"))
predictData <- rename(predictData,c(V59="PBRAND"))
predictData <- rename(predictData,c(V60="PZEILPL"))
predictData <- rename(predictData,c(V61="PPLEZIER"))
predictData <- rename(predictData,c(V62="PFIETS"))
predictData <- rename(predictData,c(V63="PINBOED"))
predictData <- rename(predictData,c(V64="PBYSTAND"))
predictData <- rename(predictData,c(V65="AWAPART"))
predictData <- rename(predictData,c(V66="AWABEDR"))
predictData <- rename(predictData,c(V67="AWALAND"))
predictData <- rename(predictData,c(V68="APERSAUT"))
predictData <- rename(predictData,c(V69="ABESAUT"))
```

```
predictData <- rename(predictData,c(V70="AMOTSCO"))
predictData <- rename(predictData,c(V71="AVRAAUT"))
predictData <- rename(predictData,c(V72="AAANHANG"))
predictData <- rename(predictData,c(V73="ATRACTOR"))
predictData <- rename(predictData,c(V74="AWERKT"))
predictData <- rename(predictData,c(V75="ABROM"))
predictData <- rename(predictData,c(V76="ALEVEN"))
predictData <- rename(predictData,c(V77="APERSONG"))
predictData <- rename(predictData,c(V78="AGEZONG"))
predictData <- rename(predictData,c(V79="AWAOREG"))
predictData <- rename(predictData,c(V80="ABRAND"))
predictData <- rename(predictData,c(V81="AZEILPL"))
predictData <- rename(predictData,c(V82="APLEZIER"))
predictData <- rename(predictData,c(V83="AFIETS"))
predictData <- rename(predictData,c(V84="AINBOED"))
predictData <- rename(predictData,c(V85="ABYSTAND"))
predictData <- rename(predictData,c(V86="Purchase"))

predictData <- subset(predictData, select = -c(MGODRK, MGODPR, MGODOV, MGODGE, MRELGE, MRELSA,
                         MRELOV, MFGEKIND, MHHUUR, MAUT1, MAUT2, MZFONDS))

# Create Product-Contribution interaction variables
predictData$IAANHANG <- predictData$PAANHANG*predictData$AAANHANG
predictData$ITRACTOR <- predictData$PTRACTOR*predictData$ATRACTOR
predictData$IWERKT <- predictData$PWERKT*predictData$AWERKT
predictData$IBROM <- predictData$PBROM*predictData$ABROM
predictData$ILEVEN <- predictData$PLEVEN*predictData$ALEVEN
predictData$IPERSONG <- predictData$PPERSONG*predictData$APERSONG
predictData$IGEZONG <- predictData$PGEZONG*predictData$AGEZONG
predictData$IWAOREG <- predictData$PWAOREG*predictData$AWAOREG
predictData$IBRAND <- predictData$PBRAND*predictData$ABRAND
predictData$IZEILPL <- predictData$PZEILPL*predictData$AZEILPL
predictData$IPLEZIER <- predictData$PPLEZIER*predictData$APLEZIER
predictData$IFIETS <- predictData$PFIETS*predictData$AFIETS
predictData$IINBOED <- predictData$PINBOED*predictData$AINBOED
predictData$IBYSTAND <- predictData$PBYSTAND*predictData$ABYSTAND

names(predictData)

################### Final Variable subset ##################

## Mixed :: 7 - All policies interactions & Socio-demographic mix - CaravanMixed3 - predictMixed3
predictMixed3 <-  subset(predictData, select = c(MKOOPKLA, MINKGEM, MOPLLAAG, MHKOOP, MAUT0,
                          IPLEZIER, IBRAND, ITRACTOR, IBROM,
                          IBYSTAND, IWAOREG, IGEZONG, IFIETS, IAANHANG,
                          IINBOED, ILEVEN, IWERKT, IPERSONG, IZEILPL))

################### Evaluation Function for prediction data set ###########

truepos <- sum(predictTarget) #238

# create data frame
combinedresultsFinal <- as.data.frame(c(1:9))
combinedresultsFinal <- t(combinedresultsFinal)
colnames(combinedresultsFinal) <- c("Method_Name", "Variable_Set" ,"Test_Set_True_Positives", "TP_Identified", "Baseline",
 "Lift", "Sensitivity", "Precision", "F-measure")
combinedresultsFinal <- combinedresultsFinal[-1,]

BaselineFinal<-48

# create function to evaluate results
evaluateFinal <- function(methodname, variableset, predictions, combinedresultsFinal){
 liftFinal <- as.data.frame(cbind(predictTarget, predictions))
 colnames(liftFinal) <- c("y", "yhat")
 orderFinal <- liftFinal[order(liftFinal$yhat,decreasing=TRUE),]
 liftcutFinal <- orderFinal[1:800,]
 TPFinal <- sum(liftcutFinal$y)
```

```
 LiftFinal <- TPFinal-BaselineFinal
 SensitivityFinal <- sum(liftcutFinal$y)/sum(predictTarget)
 PrecisionFinal <- sum(liftcutFinal$y)/800
 FMeasureFinal <-
(2*(sum(liftcutFinal$y)/800)*(sum(liftcutFinal$y)/sum(predictTarget)))/((sum(liftcutFinal$y)/800)+(sum(liftcutFinal$y)/sum(predic
tTarget)))
 resultsFinal <- as.vector(c(methodname, variableset, truepos, TPFinal, BaselineFinal, LiftFinal,
                    SensitivityFinal, PrecisionFinal, FMeasureFinal))
}


predictions <- predict(mlp_12n, predictMixed3)
Finalresults1 <- evaluateFinal(methodname, variableset, predictions, combinedresultsFinal)
combinedresultsFinal <- rbind(combinedresultsFinal, Finalresults1)

write.csv2(combinedresultsFinal, "Prediction Results.csv")
```