GHENT UNIVERSITY

FACULTY OF SCIENCES

USING MACHINE LEARNING TECHNIQUES FOR ANALYZING SURVEY DATA

Bastjaan Beernaert

Master dissertation submitted to
obtain the degree of
Master of Statistical Data Analysis

Promotor: dr. Koen Plevoets

Academic year 2020 – 2021

GHENT UNIVERSITY

FACULTY OF SCIENCES

USING MACHINE LEARNING TECHNIQUES FOR ANALYZING SURVEY DATA

Bastjaan Beernaert

Master dissertation submitted to
obtain the degree of
Master of Statistical Data Analysis

Promotor: dr. Koen Plevoets

Academic year 2020 – 2021

Confidentiality Agreement

PERMISSION

The author and the promoter give permission to consult this master dissertation and to copy it or parts of it for personal use. Each other use falls under the restrictions of the copyright, in particular concerning the obligation to mention explicitly the source when using results of this master dissertation.

Student's name: Bastjaan Beernaert

Preface

As the process of writing my master's dissertation comes to an end, I reflect upon the things I have learned and the difficulties I have faced. It was a long process with many obstacles, but I am proud of what I can show today.

I need to thank my supervisor dr. Plevoets, who gave me the opportunity to develop my knowledge in both machine learning and Python programming. He came up with the subject, taking into consideration my desire to improve my knowledge in forementioned topics and his available knowledge. This motivated me to try my hardest.

Furthermore, I would like to reflect on the whole journey of my studies that has led to this moment. In my secondary education, I studied economy and languages. Since this programme only had three hours of mathematics and two hours of sciences, everyone recommended me to take the Bachelor of Science in Business Administration program. However, as a result of the support of several people, I chose to try my luck in the Business Engineering programme. While facing difficulties to close the gap in knowledge for both mathematics and science courses, I have been able to follow the standard learning path throughout my studies. I am very proud to have achieved this. To finish my academic career and wishing to deepen my knowledge in statistics and data science, I enrolled in the Master of Science in Statistical Data Analysis. Even though it was an exceptional year due to covid, I have not regretted my decision as I believe this second Master will help me greatly in my professional career. To conclude, I would like to thank the following people that helped me accomplish this:

- My girlfriend, who was the first person to push me to the Business Engineering programme and who always made me strive for excellence. Seeing her work hard every day to reach her own goals, she inspired me to do the same.
- The friends that I made along the way. We knew that we could all count on each other when we could not make it to a class or we had trouble understanding a certain subject. Our conversations made sure we knew that nobody was facing their problems alone and that we were in this together.
- Finally, I need to thank my parents. They allowed me to choose my own path and supported me in my decisions.

Thank you for your interest in this master's dissertation. I hope you enjoy reading it.

Table of Contents

List of Abbreviations

BERT	bidirectional encoder representations from transformers
CRISP-DM	cross-industry standard process for data mining
CV	cross-validation
DNN	dense neural network
DT	decision tree
FA	factor analysis
FN	false negatives
FP	false positives
GAN	generative adversarial network
KDA	kernel discriminant analysis
LDA	linear discriminant analysis
MCC	Matthews correlation coefficient
NLP	natural language processing
NODE	neural oblivious decision ensembles
OLR	ordinal logistic regression
PCA	principal component analysis
RBF	radial basic functions
RF	random forest
RFECV	recursive feature eliminated with cross-validation
SAINT	self-attention and intersample attention transformer
SH	successive halving
SVM	support vector machine
TN	true negatives
TP	true positives
TPE	tree parzen estimator
VIF	variance inflation factor
XGB	extreme gradient boosting

List of Figures

List of Tables

Abstract

Analysing survey data has been done for a long time. To find patterns in the answers, multivariate statistical techniques such as Principal Component Analysis or Factor Analysis are used in studies like psychometric examination, where they have shown great results. These techniques, however, make some huge assumptions about the data. For example, they both need variables that can combine in a linear manner so that linear transformations can be done and some applications of FA make distributional assumptions as well. Therefore, they will fail for other types of studies. This thesis has tried some techniques that enable to broaden the analysis of survey data. In particular, some Machine Learning techniques that are more robust and allow for non-linearities are proposed. In addition, some dimensionality reduction methods are discussed that can be used in place of PCA.

The data that was used in this study was the Young People Survey, where 1010 Slovakians answered 150 questions about many different aspects of life, ranging from music and movie preferences to phobias, views on life and spending habits. In addition, some demographic questions (e.g. questions about age, gender and education) were asked as well. The question about saving habits, Finances, was used in this study to show the relevance of ML in survey analysis. The models aimed to predict the answer to this question based on all other information. In particular, the questions that this thesis has tried to answer were: 'Is it possible to classify the spending habits of young people using information obtained from not directly related other questions?', 'What Machine Learning algorithms are performing the best when analysing survey data?' and 'What feature reduction methods are more robust than both PCA and FA?'.

It turned out that the Extreme Gradient Boosting algorithm without any dimensionality reduction performed the best, closely followed by the Random Forest algorithm. The dimensionality reduction methods proved to have no benefit on predicting performance here, except for PCA in the case of using Support Vector Machines. However, one could argue that these feature reduction methods would have had more success if the hyperparameter selection was not as optimized.

Concerning the hyperparameter optimization techniques, it was observed that TPE was able to surpass the performance of Grid Search and Randomized Search if enough iterations were allowed and the technique did not get stuck in a local optimum.

The best models could then be used for interpretation by checking the importance of each variable for predicting the dependent variable (Finance) using metrics such as gain, coverage, weight and mean decrease in Gini. As a non-exhaustive example, it was observed that there are two groups of variables that obtain high importance values; those that make a distinction between both responsible young people who prioritise workload, focus on achievements, take time to make decisions, think ahead and are reliable opposed to young people who like to party, enjoy some drinks, spend money on looks and having trouble getting up in the morning. The Ordinary Logistic Regression confirmed these conclusions and gave some additional insights concerning the significance and direction of those variables. Furthermore, it was observed that being an only child or having a doctorate degree also increases the odds of a young Slovakian person to be a saver.

If the goal of the user is to increase predicting performance and is willing to sacrifice some interpretability, then one can use a combination of different types of ML models. This is called ensemble learning. In particular, the three classifiers mentioned above will be combined (all possible combinations), along with a version with some other classifiers using their default hyperparameters. Both voting and stacking ensembles were used. The first method uses another ML algorithm (called the meta-model) to aggregate the outputs of the base models, while the latter just chooses the option that has either been predicted the most (hard voting with majority rule) or which has the highest sum of predicted probabilities (soft voting). It turned out that combining RF and XGB using soft voting improved predicting performance.

Keywords: Survey Analysis, Principal Component Analysis, Factor Analysis, Machine Learning, Random Forest, Support Vector Machine, Extreme Gradient Boosting, Dimensionality Reduction, Hyperparameter Optimization, TPE, Ordinal Logistic Regression, Ensemble

Chapter 1: Introduction

A questionnaire or survey is a list of questions that can be used by both researchers and businesses to uncover new information. These questions can be varied, covering many different topics and using different formats. The biggest distinction is the one between open and closed questions (sometimes also distinguished as qualitative and quantitative data respectively). In this first category, people are given an empty text box so that they are free to respond however they like. Meanwhile, closed questions are often used to ask respondents how they feel about something or how much they agree with a given statement. To measure this, these questions often use the Likert scale, named after psychologist Rensis Likert (Rinker, 2014). This is a symmetric agree-disagree scale with an odd number of options. For example, there can be seven options ranging from Agree to Disagree, with the fourth option being Indecisive. Finally, other kinds of closed questions are categorical, where you also have to choose between some distinct answers. The difference lies in the fact that the answers have no order. These type of questions are often used for demographic information (e.g. gender and highest obtained degree).

Surveys have been conducted for a long time. The national census, the most famous public survey in the United States of America, has been held since 1790. Therefore, many different ways of analysing survey data have been proposed. These methods differ a lot in complexity. A simple form of analysis is just visualizing a question using bar charts, cross-tabulation charts or using a simple statistic like the mean and mode.
Very often, hypothesis testing is performed. There, statistical tests are computed to check the statistical significance of the results to know whether the obtained results are representative for the whole population and not just by pure chance. In other words, it is used to be certain that your results are meaningful.

However, it is possible that the surveyor wants a more thorough analysis of the data to find patterns in the answers. This can be achieved with multivariate statistical techniques such as Principal Component Analysis (PCA) or Factor Analysis (FA). These techniques have the property that they compute linear transformations of the data. Certain versions of FA also make distributional assumptions.

In this dissertation, the goal will be to explore the potential benefits of multivariate statistical techniques that are able to handle non-linearities and are robust to outliers for data that has these characteristics (e.g. surveys). For this, several machine learning algorithms will be used. Another advantage of these techniques is that they can handle both numerical and categorical data. Combined with Natural Language Processing (NLP) tools, they can analyse both Likert-scale data from closed survey questions (often represented as numeric variables) and frequency data resulting from open survey questions (which can be represented as both numeric and categorical variables) as well. For example, some NLP algorithm can classify texts from open questions into certain categories (i.e. topic classification), after which these classifications can be used as additional features to the main machine learning algorithm.
Furthermore, some feature reduction methods will be analysed. These are used to lower the dimensionality of the data, which leads to faster computations and potentially improved performance of the machine learning algorithms. PCA served as the point of reference in this comparison.

In Chapter 2, the focus will be on formulating the exact analysis that will be performed in this study and the data that will be used. The theoretical concepts behind all the machine learning algorithms and feature reductions, along with how to implement and evaluate them, will then be discussed in Chapter 3.

In Chapter 4, all the techniques mentioned in Chapter 3 will be performed on a particular survey. The results of the study will be shown and analysed.

At the end of the dissertation, a conclusion will be written along with some limitations and potential future research.

Chapter 2: Data and Research Question

2.1 Research Question

The main focus of this research was on predictive modelling. In particular, multi-class classification was done. To achieve this, it was decided to take one variable in the Young Peoples Survey dataset and treat it as the variable that needs to be predicted on the basis of several other variables. The Young People Survey will be described in the next section. After some consideration, it was decided to opt for the 'Finances' variable, which asks the surveyee whether they try to save all the money they can ranging from 1 (strongly disagree) to 5 (strongly agree). The responses to that question were put in three classes to decrease the complexity of the problem (cfr. infra). As such, the research questions of this dissertation are 'Is it possible to classify the spending habits of young people using information obtained from not directly related other questions?', 'What Machine Learning algorithms are performing the best when analysing survey data?' and 'What feature reduction methods are more robust than both PCA and FA?'.

2.2 Data

The survey that is used in this research was retrieved from the popular Data Science website Kaggle and is called the 'Young People Survey' (Sabo, 2013). This survey was conducted at the Faculty of Sciences and Economic Sciences of the Comenius University in Bratislava. In this survey, all participants were of Slovakian nationality and aged between 15 and 30. The participants were asked about many different aspects of life, ranging from music and movie preferences to phobias, views on life and spending habits. In addition, some demographic questions (e.g. questions about age, gender and education) were asked as well. In total, this resulted in 150 questions that were filled in by 1010 people (411 male, 593 female). All these questions will be used as the variables in our models. Out of the 150 features, 11 are categorical and 139 are continuous (Likert-scale integers or floats). A full list of all variables, along with some extra information and descriptive statistics, can be found in Appendix A. Furthermore, a more thorough exploration of the data will be done in a later section (see Data Preparation).

Chapter 3: Methodology

Since this is a data analysis project, it was a logical decision to opt for the Cross-industry standard process for data mining (CRISP-DM). CRISP-DM consists out of six iterative phases:

1. **Problem Understanding:** Determine objectives, taking into account the domain and the current situation. Furthermore, determine data mining goals and produce project plan that aid in reaching the objectives. Often referred to as Business Understanding.
2. **Data Understanding:** Collect initial data, describe and explore data to get familiar with it and verify data size and quality.
3. **Data Preparation** Generally, this is seen as the most time-consuming phase. This phase contains all the steps necessary to get from our raw data to our final dataset (called base table). A non-exhaustive list of steps is selecting data, cleaning data, constructing data, integrating data and formatting data.
4. **Modelling:** Select modelling techniques, build models, optimize the hyperparameters of the models. Sometimes, a combination of modelling techniques can be used as well. Here, both the machine learning algorithms and dimensionality reductions methods will be discussed.
5. **Evaluation:** Evaluate results of the models obtained in the fourth step, review process and determine next steps.
6. **Deployment:** Produce final report. Be sure to organize the obtained results in such a way that they can be deployed in real life situations.

(Data Science Project Management, s.d.)

As the framework is a cross-industry standard, it can be implemented in any data analysis no matter what domain one is operating in. Furthermore, it is important to know that it is not necessary to go through the phases in this particular order.

Each separate phase will be discussed, some more extensively than others. The CRISP-DM is visualized on Figure 1. As can be seen, the process does not end after the Deployment phase but goes back to other phases for improvement. This explains its iterative nature (Van den Poel, 2018).

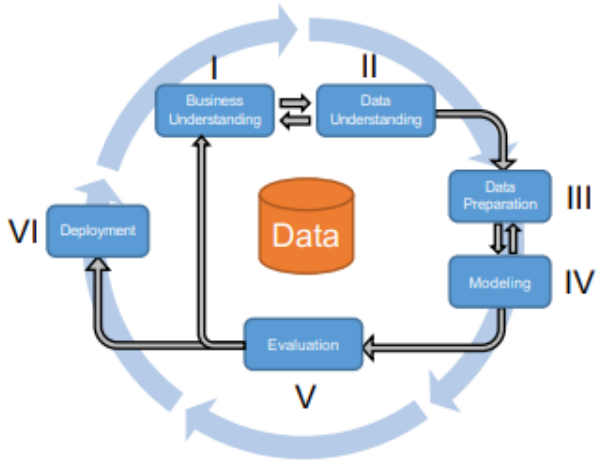

Figure 1 CRISP-DM (Van den Poel, 2018)

The popularity of the CRISP-DM model for analytics, data mining and data science projects has been proved by multiple questionnaires, including the poll of Piatetsky (2014) where 43% of respondents say that they use CRISP-DM, while the second biggest specific methodology is SEMMA with only 8.5%.

3.1 Business Understanding

This master's dissertation aims to research innovative methods to analyse survey data instead of using simple visualizations, hypothesis testing or multivariate analyses that have assumptions which may be too strict (e.g. PCA and FA needing linearity, FA also making distributional assumptions). The focus will be on predictive modelling, as several machine learning algorithms and feature reduction techniques are used to predict the financing habits of young Slovakian people as accurately as possible. However, both FA and PCA will still be used in this study to show that the proposed methods are complementary to the traditional multivariate analyses.

3.2 Data Understanding

3.2.1 Exploratory Data Analysis

As mentioned in Chapter 2, there are 150 variables in total, of which 11 are categorical and 139 are numeric (integers for the Likert-scale questions and floats for the demographic ones). While checking the numeric variables, one could see that all the Likert-scale variables are correct: all are integers (so no decimals) between 1 and 5. To be able to compare questions concerning different themes, it was also made certain that 1 meant either disagreeing or disliking and 5 meant either agreeing or liking a lot. It seems that for most Likert-scale questions, all options have at least been chosen once (i.e. 5 unique values in the dataset). In fact, the only Likert-scale question that does not have all options chosen is the 'Fun with friends' variable (i.e. 'How interested are you in socializing?'). The other non-Likert-scale numeric variables seem to have a bigger amount of unique values, ranging from 8 (Number of siblings) to 69 (Weight). The number of unique values could have been larger if the survey asked for more precision in the height or weight (i.e. numbers to one or more decimal places), but that could have the downside that less people would fill it in because they don't want to go through the burden of finding out these figures exactly, or people would just fill in something random leading to inaccurate responses. With an average of 4.73, it seems most young people in Slovakia are fond of listening to music. It is also the question where most surveyees agreed, having the lowest standard deviation (0.66). On the other hand, they do not enjoy writing poetry as this only received a mean of 1.90. The question where most people were disagreeing, was whether they try to always vote during elections with a standard deviation of 1.57. However, one could argue that this should have been a binary question (i.e. either yes or no). Therefore, it can be useful to mention that the most divergent responses for a ‘true’ Likert-scale question was the interest in pets, with a standard deviation of 1.55. The descriptive statistics of all numeric variables can be found in the second section of Appendix A.

For the categorical variables, it is interesting to look at the distribution of the number of responses for each possibility. Most distributions are in line with what could be expected, with the majority of the people surveyed being social drinkers, right handed and not an only child. It seems that all options have been chosen at least once as well. Interestingly, more than 60 percent had only achieved secondary school as their highest level of education. This is likely due to the fact that many surveyees were still attending college at the time of the survey. To check the distribution of all categorical variables, the reader is referred to Appendix B.

The correlation plot on Figure 2 shows the correlation between all original 150 questions. It can be observed that some variables have a rather high (>0.5) correlation between one another. Regarding the variable that is most interesting to this research, Finances, the correlations are actually rather low. The top three absolute values for the correlation between variables are only 0.298, 0.257 and 0.206.

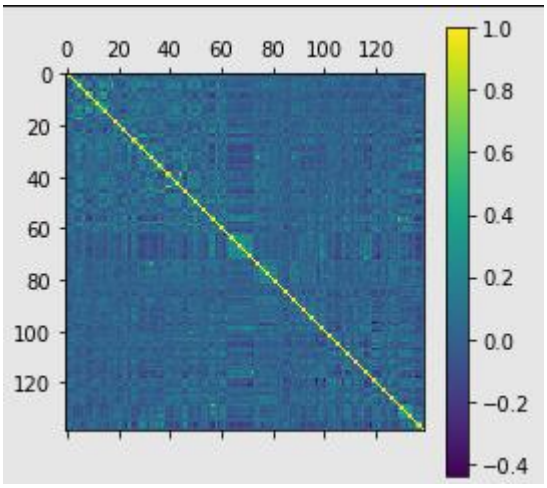

Figure 2 Correlation Plot

Most variables have some missing values, i.e. people who chose not to fill in a certain question. The questions that were not filled in the most were height and weight with both 20 non-responses, which can be expected as it is a delicate question for some. The Likert-scale question that was left empty the most was the one asking about the interest in sport and leisure activities (15 non-responses). Figure 3 shows the missing values on an observational level (= filled in surveys). The right side of the figure shows the number of filled in variables, while the white horizontal stripes represent the columns with missing values. Most surveys had no missing value, with the most missing values for a single survey being 9 (i.e. 141 non-missing variables). Handling these missing values will be done in the Data Preparation phase (cfr. infra).

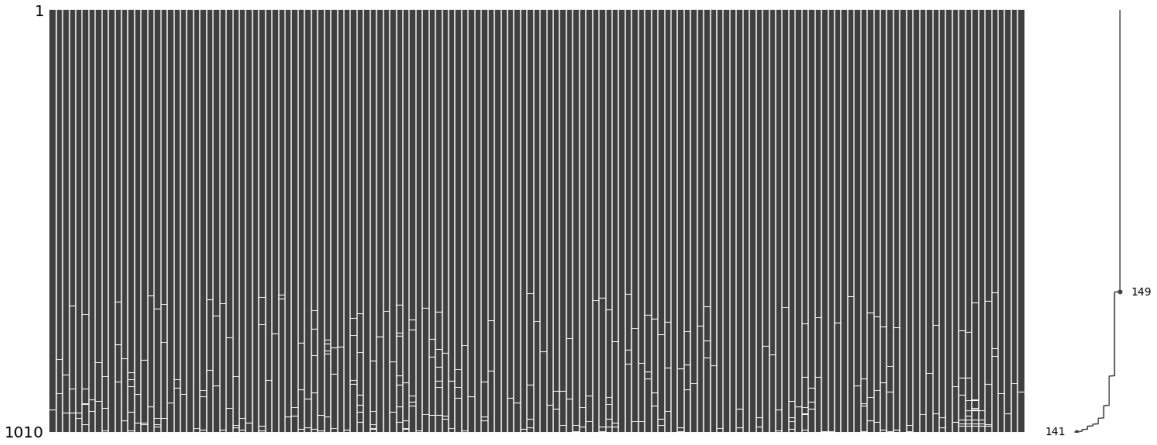

Figure 3 Missing Values on observation level

3.2.2 Software and Hardware used

For this master's dissertation, Python was chosen for all but the Ordinal Logistic Regression (cfr. supra), for which R was preferred as it is easier to check the model's assumptions using the packages that are available in R. The packages that were used in this dissertation, along with their specific purpose and version, are shown below in Table 1.

Table 1 Python Packages

Package	Version	Usage
Python		
pandas	1.2.2	High-performance, easy-to-use data structures and data analysis tools
numpy	1.20.1	Array processing for numbers, strings, records and objects
matplotlib	3.3.4	Publication quality figures in Python
seaborn	0.11.1	Statistical data visualization
dill	0.3.3	Serialize (almost) all of Python
missingno	0.4.2	Missing data visualization module for Python
joblib	1.0.1	Lightweight pipelining
plotly	5.1.0	Interactive, browser-based graphing library
optuna	2.8.0	Hyperparameter optimization
scikit-learn	0.23.2	A set of Python modules for machine learning and data mining
xgboost	1.3.3	Extreme Gradient Boosting
hyperopt	0.2.5	Hyperparameter optimization
hpsklearn	0.1.0	Hyperparameter optimization
scipy	1.6.0	Scientific library for numerical integration, optimization, linear algebra, interpolation and statistics
factor_analyzer	0.3.2	Factor Analysis
kfda	0.1.1	Kernel Fischer Discriminant Analysis
R		
MASS	7.3.53	Ordered Logistic Regression
stargazer	5.2.2	Visualize summary of regression model
formattable	0.2.1	Table visualizations
car	3.0.10	Variance Inflation Factor for multi-collinearity + Check assumption of proportional odds

3.3 Data Preparation

After the exploratory data analysis of the previous section, it was clear that some things had to be changed in the data. The first thing that had to be done was changing the type of certain features. Namely, features with type object were changed to type category so that they could be one-hot encoded later on (cfr. infra). In addition, the Likert scale variables currently have values ranging from 1 to 5. It was decided to change this to -2 to 2 for easier interpretability. As this is just a linear transformation, this does not affect the results of the analysis. To do this, all Likert scale features are retrieved after which three is subtracted from each value.

Furthermore, there were also two outliers that were found by checking the distribution of the numeric non-Likert-scale questions. For the Height, there is one person who appears to be shorter than 80cm. On the other hand, there seems to be one person that has 10 siblings. After looking at the observations where the outliers occur, it seems that that the person accidentally wrote 62cm instead of 162cm. This was manually corrected. The number of siblings was a bit harder to check, but it was assumed that only 1 instead of 10 siblings was meant. As an example, the boxplot of Height before and after changing the outlier is shown on Figure 4. One can observe that the distribution now better resembles a Normal distribution.

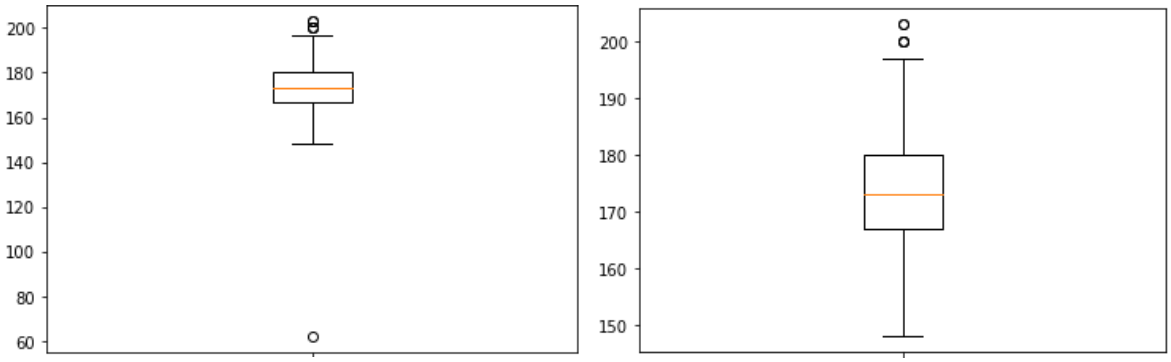

Figure 4 Boxplot Height before (left) and after (right) handling outlier

Finally, the type of the dependent variable (Finances) was changed to categorical. As there were very few outliers in this variable (three), it was decided to go for a very simple forward fill imputation, where the last valid observation will be propagated forward. For the other variables, another imputation was used (which will be explained later). As mentioned before, there will only be three classes during our problem. Therefore, the answers -2 and -1 were changed to -1, 0 stayed 0, and 1 and 2 became both 1. They can be interpreted as being a big saver (1), mediocre saver (0) or a spender (-1). This resulted in the distribution of Figure 5.

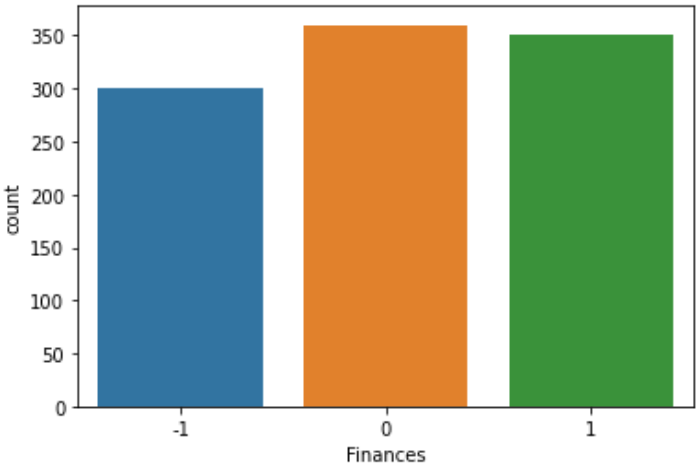

Figure 5 Distribution of Dependent Variable

It can be observed that the distribution is rather balanced. However, it was still decided to deal with the class imbalance to make the model as accurate as possible for each class. Furthermore, the attentive reader will have noticed that not all pre-processing steps have been done in this phase. This is because some of the steps are only done after the training-test split to avoid data leakage. More explanation on this will be given in the Modelling phase.

3.4 Modelling

3.4.1 Classification Algorithms

The predictive modelling process aims at finding generalizable patterns in the data. To get the most value from machine learning, one needs to know how to pair the best algorithms with the right data (SAS Analytics Insights, 2020).

For this master's dissertation, the decision was made to go for a supervised approach, in particular using multi-class classification. A summary of some possible modelling methods are proposed to be certain that the right algorithm is used. One is inherently multiclass (Random Forest), while the others are inherently binary classifiers (Support Vector Machine and Extreme Gradient Boosting). To make these classifiers multiclass, either the one-vs-rest or one-vs-all will be used (cfr. infra). Each machine learning method that was tested and their hyperparameters are discussed below. A model hyperparameter is a characteristic of a model that is external to the model itself, which means that its value cannot be estimated from the data. The value has to be set before the prediction process can begin. By tuning them, the optimal hyperparameters will result in the most 'accurate' predictions (Joseph, 2018).

3.4.1.1 Support Vector Machine

A support vector machine (SVM) performs classification in a linear way by finding the line (two dimensions), plane (three dimensions) or hyperplane (more than three dimensions) that maximizes the margin between the classes. However, a nonlinear region can separate the groups more efficiently in some situations. SVM handles this by using a kernel function (nonlinear) to map the data into a different space where a hyperplane (linear) can not be used. In other words, a non-linear function is learned by a linear learning machine in a high-dimensional feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space.

The kernel function transforms the data into a higher dimensional feature space to enable the performance of linear separation. This is referred to as kernel trick. An example can be seen below in Figure 6.

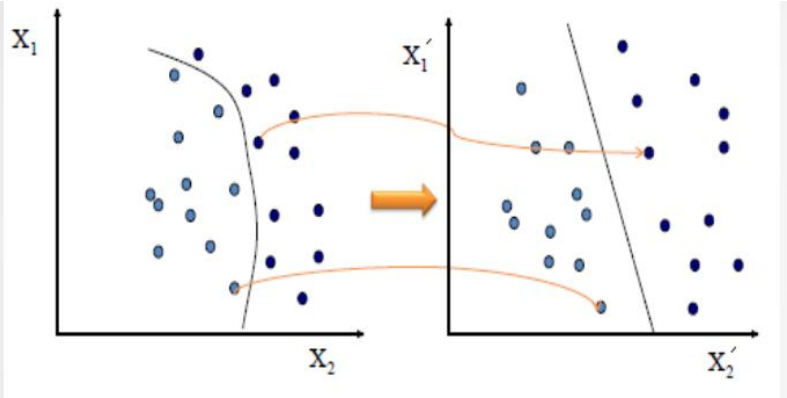

Figure 6 Example SVM (Sadawi, s.d.)

Since this algorithm is inherently binary, the one-vs-one strategy was used. With this strategy, one classifier is constructed per pair of classes. It then chooses the class which obtained the most votes. In case of a tie, the class with the highest aggregate classification confidence is chosen. As the number of classifiers needed to fit equals $\frac{n_{classes} * (n_{classes} - 1)}{2}$, this can be computationally expensive. However, in this research, it only led to six classifiers. Furthermore, it is also the better option for kernel algorithms as they often don't scale well with the number of observations. With one-vs-one, each model only involves a smaller subset of the data (Brownlee, 2020).

The success of performing SVM depends on the right choice of kernel function, the regularization parameter, the gamma parameter and the degree.

Examples of kernel functions are the polynomial kernel and the exponential kernel. The polynomial kernel can be written as $K(x, x_i) = 1 + sum(x * x_i)^d$, where d is a positive integer. The exponential kernel can be written as $K(x, x_i) = e^{(-\gamma * sum((x - x_i{}^2))}$, with x the input and x_i the support vector.

The regularization parameter, also known as C, tells the SVM algorithm how much misclassifying in each training example should be avoided. If C has a rather high value, the optimization will go for a smaller-margin hyperplane if it does a better job of getting all the training points classified correctly. On the other hand, a smaller value for C will cause the optimizer to look for a larger-margin separating hyperplane even if that leads to more misclassifications.

The gamma parameter (which is a different gamma than the one in the exponential kernel) defines how far the influence of one observation reaches. Low values mean that points far away from the plausible separation line are considered in the calculation of said separation line. When gamma is high, only points close to the plausible line are considered.

Finally, the degree is a parameter that is only applicable to polynomial kernels. Higher-degree polynomial kernels allow for a more flexible decision boundary (Ben-Hur et al., 2008).

3.4.1.2 Random Forest

Random forest is an adaptation of the decision tree ensemble technique. It is often seen as an improvement of the bootstrap aggregation or bagging method. It works the same way as bagging, as it also takes multiple subsets out of the training data set, runs the decision tree algorithm on each subset and then average all results to get the final prediction. For each observation, the predicted class by each of the trees is recorded and the most common class is chosen for that observation. This is called the majority vote.

Using bagging, the variance will be reduced from σ^2 to $\frac{\sigma^2}{n}$ (with n the number of trees) compared to the decision tree assuming independent and identically distributed data. However, as the trees use similar data, they will actually be correlated leading to a higher variance than the formula mentioned. Furthermore, the correlation will also increase bias. (Hastie et al., 2008).

Random forest allows to improve both variance and bias compared to the regular bagging method . It is able to decrease the correlation between trees by randomly taking m out of the p predictors at each split. The value of m is often $\sqrt{p}$. This is done to inhibit that the same very strong predictor is chosen at each split.

Averaging many less correlated trees will reduce variance much more than averaging many highly correlated trees. The bias will decrease thanks to the lower correlation as well.
In summation, the random forest model allows to drastically reduce the variance of the decision tree, while only receiving a minimal increase in bias. Thus, the overall performance is improved. The biggest disadvantage is the huge decrease in interpretability, as it is difficult to visualize for a huge number of trees (James et al., 2017).
For this classifier, it will also be investigated whether scaling the features has any impact on the prediction performance. It is true that RF is based on tree partitioning algorithms, where a collection of partition rules is obtained which should not change with scaling (the trees thus only see ranks in the features). However, RF will tend to favour highly variable continuous predictors to split, since there are more opportunities to partition the data (even if only a subset of the variables is used in each individual tree). This leads to some highly variable features to get an unjustified large importance. Since we might want to take a look at the importance of each individual feature in the prediction, it was decided to try scaling (Strobl et al, 2007).
Random forest has some hyperparameters that need to be tuned. Some examples for random forest are the number of trees, the splitting criterion to consider, the maximum depth of a tree, the splitting criterion and the minimum sample split or leaf. These last two indicate the minimum (absolute or relative) amount of samples that need to be present in both parts after splitting a node or at the bottom node respectively. The splitting criterion will be explained when discussing the importance of the features in this algorithm (cfr. infra).

3.4.1.3 Gradient Boosting

The (extreme) gradient boosting algorithm is another decision tree ensemble technique. However, it works on the principle of boosting instead of the bagging in the random forest method.
Just like in bagging, boosting works with multiple trees to then combine them to create a single predictive model. However, the way that the trees are built differs significantly. The trees are grown sequentially instead of independently. This means that each tree is grown using information from the previously grown trees, with the model precision being improved after each iteration. Furthermore, each tree is fit on the original full data set instead of involving bootstrap sampling (there is a hyperparameter for that though, cfr. infra). In this dissertation, the extreme gradient boosting version was performed.

It works on the same principle as the general gradient boosting, but with a more regularized model formalization to control for overfitting[1]. As this method is also inherently binary, the one-vs-rest strategy was used here to make it multiclass. With this strategy, each class is fitted against all other classes for each classifier. Compared to the one-vs-one strategy, this saves computation time as only three classifiers will be needed (one for each class). Furthermore, it is also more interpretable as each class is only represented by one classifier. This is the default strategy for this method and it is also the default for almost all classifiers.

Similar to random forest, boosting has hyperparameters that need to be tuned. Some examples are (Baboota & Kaur, 2018):

1. The number of trees. With this method, there is danger of overfitting when this is too large.
2. The shrinkage parameter, which controls the rate at which the method learns. Most of the time, it ranges between 0.01 and 0.001. Having a small will often lead to requiring a large number of trees to get a good model performance.
3. The number of splits in each tree d. It is also called the interaction depth, as it controls the interaction order of the boosted model. D splits can involve at most d variables. If d = 1, each tree is a stump consisting of a single split. Thus, a higher value for d will result in a more complex boosted ensemble.
4. Gamma, which regularises the model using across trees information. It shows by how much the loss has to be reduced after a split, for that split to actually done. The higher the value, the higher the regularization.
5. The sampling of the dataset at each boosting round. Instead of using all the data every time, one can build a tree on slightly different data for each step. This way, the model is less likely to overfit. One can subset on either the fraction of observations (subsample) or on the fraction of features (colsample_bytree) to be used.

3.4.1.4 Ensemble Learning

The composite models that are referred to here are not those that form ensembles out of the same base model type (e.g. an ensemble of trees like Random Forest), but to a combination of different types of ML models.

[1] Note that we the xgboost package was used instead of the standard scikit-learn because it can be parallelized, it is memory-efficient and it also uses second derivates to find the optimal constant in each terminal node, along with advanced regularization to improve model generalization (Brownlee, 2016).

In particular, the three optimized classifiers mentioned above will be combined (all possible combinations), along with a version with some other classifiers using their default hyperparameters. These were Logistic Regression, Classification Tree and Gaussian Naïve Bayes. As these classifiers were not optimized and only used for this single purpose, it was decided not to give them an extensive explanation. The interested reader is referred to Vanghese (2018).

There are some ways to combine the different classifiers (often called base models). In this dissertation, both stacking (or stacked generalisation) and voting will be performed. The first method uses another ML algorithm (called the meta-model) to aggregate the outputs of the base models, while the latter just chooses the option that has either been predicted the most (hard voting with majority rule) or which has the highest sum of predicted probabilities (soft voting). The latter is preferred if the classifiers are well-calibrated (Lones, 2021).

3.4.2 Dimensionality Reduction Techniques

Dimensionality reduction techniques were tried in an attempt to both further improve the performance of our machine learning algorithms[2], and to show several other techniques than the ones that are currently often used (Factor Analysis and Principal Component Analysis). To be more precise, the classifiers mentioned in the previous section will be performed on the output of the techniques that are mentioned in this section. The dimensionality reduction techniques can be divided according to two criteria: whether they are a feature extraction or feature selection method and whether it is a supervised or unsupervised technique. The former distinction checks whether the original features are maintained (feature selection) or whether some transformation takes place where the data is projected on a new feature space with as goal to maintain most of the relevant information (feature extraction).

Meanwhile, the latter distinction checks whether the reduction uses a dependent variable to base its reduction on (supervised) or not (unsupervised). In this study, this variable is Finances.

3.4.2.1 Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised linear transformation technique. It is used in many fields.

[2] Sometimes, dimensionality reduction is used even though it decreases performance. This is done from a computational perspective: in real life situations, it is often needed for models to be computed fast (sometimes even real-time). To ensure that the complexity of the model does not need to be reduced too much, the amount of features is reduced.

It aims to find the directions of maximum variance in high-dimensional data, after which it projects the data onto a new subspace. The axes of the new subspace are the directions of maximum variance and they are orthogonal to each other, meaning that the projections (axes) are uncorrelated to one another. The first principal component will have the greatest variance, the second principal component the second greatest variance and so on. For a more elaborate explanation, including the math behind the technique, the reader is referred to Raschka & Mirjalili (2019).

As mentioned before, PCA has the downside that it is unable to handle both numeric and categorical variables. The reason for that is that PCA is designed for continuous variables; it tries to minimize variance (or squared deviations). The concept of squared deviations breaks down when you have binary variables. One can use dummy variables to replace the categorical variables so PCA does work (which is done in this dissertation), but it is recommended to use another technique like Multiple Correspondence Analysis in that case instead. This is an extension of PCA and can be viewed as a way to code categorical variables into a set of continuous variables (Husson et al., 2010). However, this technique will not be elaborated upon further as the focus lies on applying Machine Learning techniques.

3.4.2.2 Factor Analysis

Factor Analysis (FA) is not like the other techniques mentioned here, in the sense that it can not be used as a dimensionality reduction before a classifier. Its analysis stands on its own. FA is used to identify the structure underlying the variables and to estimate scores to measure latent factors themselves. In Factor Analysis, only the shared variance is analysed in contrast to PCA where all the observed variance is analysed. Furthermore, FA explicitly assumes the existence of latent factors underlying the observed data. PCA instead seeks to identify variables that are composites of the observed variables (Bock, s.d.). To conclude, the underlying factors in FA are labelable and interpretable compared to the uninterpretable PCA components (Navlani, 2019).

Before FA is performed, it is important to evaluate whether there can be factors found in the dataset. Two tests that will be tried to check this are Bartlett's Test and Kaiser-Meyer-Olkin (KMO) Test. Bartlett’s Test uses the correlation matrix against the identity matrix (i.e. ones on diagonals and zeroes elsewhere) to check if the observed variables have enough correlation between them. Ideally, the test is statistically significant.

Meanwhile, KMO measures the adequacy both for each observed variable and the complete model by estimating the proportion of variance that might be a common variance among all observed variables. The values range between 0 and 1, with a value of 0.60 or above often considered as a good threshold. With a p value smaller than 0.0001 and a KMO of 0.766, FA may be performed.
The first part of FA is choosing the number of factors. The Kaiser criterion can be used for this, which is very straightforward. All it does is only keeping the factors with eigenvalues greater than 1. In a standard normal distribution with mean 0 and standard deviation 1, the variance will be 1. Since the data is standard scaled, the variance of a feature is 1. This is the reason for selecting factors whose eigenvalues (variance) are greater than 1 i.e. the factors which explain more variance than a single observed variable. (Babu D., 2020)

After this, FA is performed again with the correct amount of factors. Then, the loadings of each variable to each factor are checked to see which variables relate to which factors. To obtain these loadings, a rotation strategy is used so that the space with the loadings (represented as points) shows a clear pattern, i.e. factors that are clearly marked by high loadings for some variables and low loadings for others. Put simply, rotations help by minimizing the complexity of the loadings and makes them easier to interpret.
The strategy used here is varimax, which maximizes the sum of the variances of the squared loadings as all the coefficients will be either large or near zero, with few intermediate values. The goal is to associate each variable to at most one factor.
To conclude the FA process, one can see the total/cumulative amount of variance explained by these first X factors and the communalities. This last metric is the proportion of each variable's variance that can be explained by the factors.

3.4.2.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is another linear transformation technique that performs feature extraction. The general concept is very similar to PCA, but LDA is supervised.
The goal in LDA is to find the feature subspace that optimizes class separability instead of maximum variance in the dataset (i.e. PCA).

3.4.2.4 Kernel Discriminant Analysis

Kernel Discriminant Analysis (KDA) is a modification of LDA which allows for discriminant analysis in high dimensionality using the kernel trick, similar to what was done earlier in the SVM classifier.

3.4.2.5 Recursive Feature Elimination and Cross-Validated Selection

Recursive Feature Elimination (RFE) is a wrapper-type feature selection algorithm. This means that it selects a subset of the most relevant features for a dataset using a certain machine learning algorithm in the core of the model. The features are ranked by importance based on the model, after which the least important features are discarded and the model is re-fitted. With regular RFE, deciding the number of features one wants to end up with is a hyperparameter that needs to be chosen. To get this number automatically, Cross-Validated Selection is added to RFE (RFECV). In this research, 5-fold RFECV was used. This means that for each split, the training set will be transformed by RFE 5 times. This is done for all possible amount of features, after which RFECV transforms the entire set using the best scoring number of features.
The importances are taken from the fitted model using either a coefficient or feature importances attribute. Since only the linear kernel of the SVM model supports such attributes, RFECV was not attempted for this algorithm as this kernel proved to be nonoptimal.

3.4.3 Regression Technique

Regression was performed with the goal to be able to look at the direction and size of the influence of certain variables. The exact model that was performed, is called the Ordinal Logistic Regression (OLR) model.
This model was chosen because the logistic function can be used to model the odds of the three discrete outcomes (-1, 0, 1) as a linear combination of the independent variables. One could argue that a multinomial logit model would also be suitable, but this model does not take into account that our dependent variable is ordinal. Thus, the ordinal logit model fulfils both the ordinality and discreteness requirements and can produce probability estimates for such outcomes (Davidson & MacKinnon, 2004). The fundamentals of the model will now be elaborated upon, using the research of Heino and Sillanpää (2013).
The ordinal logit model tries to create a connection between the independent variables (i.e. the features) and a discrete dependent variable by means of an unobservable continuous dependent variable.

The connection is achieved by assuming that the continuous variable is a function of the independent variables and a disturbance (or randomness) with known distribution properties. Each value of this continuous dependent variable relates to a set of probabilities for each possible value of the discrete dependent variable.

For each part of the model, some assumptions need to be made. For the disturbance, a standardized logistic distribution is assumed. This is a symmetric distribution similar to the normal distribution. Concerning the independent variables, it is assumed that a linear relationship exists between them and the unobservable continuous variable. To be more exact, the unobservable continuous variable is the sum of the disturbance and the product of the independent variables with an equally long vector of constants. Mathematically, this can be written as:

$$y^* = \boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{X} + \varepsilon,$$

with y^* the unobservable continuous variable, $\boldsymbol{X}$ a vector of n independent variables and $\boldsymbol{\beta}^{\boldsymbol{T}}$ a vector of n constants (Davidson & MacKinnon, 2004). Finally, the relation between the unobservable continuous variable y^* and the observable discrete variable y is assumed by giving y^* some constant threshold points $\hat{\kappa}_k$. The value of y then depends on whether y^* has crossed a particular threshold. Thus, the probability of observing a certain state of the discrete dependent variable y depends on $\boldsymbol{\beta}^{\boldsymbol{T}}\boldsymbol{X}$ and the constant threshold values $\hat{\kappa}_k$.

As the expression of these probabilities is built around the cumulative distribution function of the standardized logistic distribution, it is implied that y also depends on the standardized logistic distribution.

For a more detailed explanation of the fundamentals, including mathematical formulations for the distribution functions and the probabilities of each state, the reader is referred to [23].

It should be noted that it was decided to only have a look at the main effects in this research without checking the interaction effects, i.e. to compute an additive model. This way, the coefficient of one variable can be interpreted as the effect of that variable independent of the other variables[3]. Furthermore, this keeps the model at a reasonable complexity. Introducing interaction terms could lead to the curse of dimensionality, as the number of features would quickly become greater than the number of variables that are available in this study.

[3] More on interpretation in the Deployment section (cfr. infra).

3.5 Evaluation

3.5.1 Classification

In this phase, the performance of the trained models is evaluated. Several different evaluation metrics will be elaborated upon for the classification techniques.

The confusion matrix shows the combination of the actual and predicted classes. The following explanation is inspired by Sadawi (s.d.).

Each row represents the number of actual observations in a category, while each column represents the predicted number of observations in a category. It is a good measure to understand which classes are most easily confused, hence its name. An example of a confusion matrix for this research can be found below in Table 2. The numbers are just random as an example.

Table 2 Confusion Matrix

	Predicted -1	Predicted 0	Predicted 1
Actual -1	197 (TP_{-1})	46 ($E_{-1,0}$)	47 ($E_{-1,1}$)
Actual 0	57 ($E_{0,-1}$)	96 (TP_0)	36 ($E_{0,1}$)
Actual 1	72 ($E_{1,-1}$)	45 ($E_{1,0}$)	44 (TP_1)

The true positives (TP's), true negatives (TN's), false positives (FP's) and false negatives (FN's) are needed be able to calculate the performance metrics.

The diagonal shows the TP's. These are the observations where the predicted category is the same as the actual category. All the other cells are errors (E's).

The total number of FN's for a class are all the instances which were classified as another class, but are actually that class. In the matrix, they are the sum of values in the corresponding row (excluding the TP's).

The total number of FP's for a class are all the instances that are classified as that class, but actually are another class. In the matrix, it is the sum of values in the corresponding column (excluding the TP's).

The total number of TN's for a certain class will be the sum of all columns and rows excluding that class's column and row. Some interesting metrics can now be calculated.

The first metric is the precision of each category based on the model. It is calculated as:

$p = \frac{TP}{TP+FP}$.

In the example, the precision of the -1 class gives: $p_{-1} = \frac{TP_{-1}}{TP_{-1}+E_{0,-1}+E_{1,-1}}$.

The second metric is recall. It corresponds to the true-positive rate of the considered class. It is calculated as:

$r = \frac{TP}{TP+FN}$.

The denominator is the total number of test examples of the considered class (i.e. the row of that class in the confusion matrix). In the example, the recall of the -1 class gives:

$r_{-1} = \frac{TP_{-1}}{TP_{-1}+E_{-1,0}+E_{-1,1}}$.

Another metric we can derive from the confusion matrix is the F1 score. The F1 score is a weighted harmonic mean of recall and precision and is normalized between 0 or 1. A score of 1 shows a perfect precision and recall, while the worst possible value is 0. The benefit of F1 score is the largest when there is an uneven class distribution. The metric is calculated as (Sadawi, s.d.).:

$$F_1 = \left(\frac{2}{recall^{-1}+precision^{-1}}\right) = 2\ x\ \frac{precision\ x\ recall}{precision+recall}$$

Finally, Matthews Correlation Coefficient (MCC) evaluates multiclass classifiers by computing the correlation coefficient between the observed and predicted classifications. Its value lies between -1 and 1. 1 indicates a perfect prediction, 0 is similar to a random prediction and -1 shows an inverse prediction, in which case you can reverse the classifier's outcome to get the ideal classifier. MCC is also symmetric, so no class is more important than the other. Computing the MCC goes as follows (Shmueli, 2019):

$$MCC = \frac{TPxTN-FPxFN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}.$$

Chicco & Jurman (2020) argue that F1 can show overoptimistic results, especially on imbalanced datasets. They believe that the Matthews Correlation Coefficient (MCC) is a more reliable statistical measure since it only outputs a high score when all four quadrants in the confusion matrix (true positives, false negatives, true negatives and false positives) have good results proportional to the size of both positive and negative elements in the dataset. However, one could argue that the cost of a false positive or a false negative is the same in this research, making it no problem to give equal weight to precision and recall (as the F1-score does).

There are different kinds of averages that can be adopted to extract a single number for each metric, since the metrics above are calculated for each class separately.

Macro averaging computes the score for each class separately and then takes the average value. This method is used when the it is preferred to put the same emphasis on all classes. For the recall, this comes down to:

$$r = \frac{r_{-1}+r_0+r_1}{3}.$$

On the other hand, micro averaging calculates the measure from the grand total of the numerator and denominator. In other words, the individual true positives, false positives and false negatives of the system for different classes are summed up.

Micro averaging is useful when you want to bias your classes towards the most populated class. If recall is again taken as example, the formula is:

$$r = \frac{TP_{-1} + TP_0 + TP_1}{TP_{-1} + E_{-1,0} + E_{-1,1} + TP_0 + E_{0,-1} + E_{0,1} + TP_1 + E_{1,-1} + E_{1,0}}.$$

One important point to note is that in a multiclass setting, micro averaged precision and recall will always be the same as every single false prediction (or error) will be a false positive when calculating precision and a false negative when calculating recall (Lanaro, 2016). This leads to both also being equal to the overall accuracy.

Finally, the weighted average calculates the metrics for each label and then finds their average weighted by the number of true instances for each label (also called support). Compared to macro averaging, this metric accounts for the label imbalance.

3.5.2 Regression

The metric that is often used with OLR is the Akaike information criterion. This metric attempts to measure the relative amount of information that is lost by a given model. This measurement is performed with a trade-off between the goodness of fit and the simplicity of the model, so both overfitting and underfitting are taken into consideration. The absolute value is not to be interpreted, but is used to compare the performance of separate models. A lower AIC metric is a good indication of a better model if both models use the same data (Kolassa, 2014). However, as only one OLR model will be computed, it does not make sense to use AIC here.

Hence, the evaluation of the regression will be focussed on checking whether the model complies with four assumptions of OLR (Lee, 2019):

1. The dependent variables are ordered.
2. The independent variables are either continuous, categorical or ordinal.
3. No multi-collinearity.
4. Proportional odds.

As mentioned in section 3.4.3, the dependent variable (Finances) is ordered. The second assumption will also be fulfilled, as all our independent variables fall in the mentioned categories.

Multi-collinearity should be avoided as well. This phenomenon occurs when there is high correlation between the independent variables and leads to unreliable estimates of the regression coefficients. To check for multi-collinearity, the variance inflation factor (VIF) will be computed. The VIF is the quotient of the variance in a model with multiple terms by the variance of a model with one term alone. It gives an index that shows how much the standard error of a coefficient is increased because of collinearity. A value above 10 is a sign of high multi-collinearity (Kutner et al., 2004). However, other research mentions 5 as the highest acceptable value (e.g. Akers, 2018).

Finally, the assumption of proportional odds means that the relationship between each pair of outcome groups needs to be equal. This results in only one set of coefficients and that thus only one model is needed. This assumption will be checked with Brant's test, which measures whether the observed deviations from the OLR model are greater than what could be attributed to chance alone. A significant test statistic shows evidence that the parallel regression assumption has been violated (Brant, 1990).

Variables that violate any of these four assumptions will be removed from the equation to make certain that our final model is correct.

3.6 Deployment

The techniques of the modelling phase and the evaluation metrics of the evaluation phase will now be used on the data that was explained in the previous phases.

3.6.1 Validation

The process of performing the analysis was approximately the same for all algorithms. Often, the first thing that is done is to split the dataset in a training and test set. Then, k-fold cross-validation (CV) is performed. In k-fold CV, the observations in the training set are split in to k groups (or folds) of equal size. The first fold is then seen as a validation set, while the model is fit on the other k-1 groups. After fitting the model, the test error is quantified on the first fold using the number of misclassified observations (for example the F1 score, cfr. supra). The process is repeated until each fold has been used as a validation set. To get the model's final performance, the average test error is computed. To decide the number of k, a trade-off has to be made between the importance of computation time, bias and variance. The lower the number of groups, the lower the computational effort. The more observations that are used for fitting the model, the less biased the estimates of the test error will be (meaning they will be less overestimated). Leave-one-out cross-validation will thus score really well on bias. However, its variance will be higher than using k-fold cross-validation with a smaller amount of groups, as a smaller amount of groups will be less correlated with each other since there will be less overlap between training sets. The mean of highly correlated variables will have higher variance than less correlated variables (James et al., 2017). However, there is a downside to this method, as the same data is used to tune model parameters and evaluate model performance. This may lead to an optimistically biased evaluation, getting a poor estimation of errors in training or test data due to information leakage (Kumar, 2020).

To avoid this problem, in addition with the fact that it is recommended when the dataset is rather small, nested cross-validation effectively uses a series of train and test splits with an inner and outer loop. First, in the outer loop, the dataset is split into a training and test set. Here, it was opted to go for stratified k-fold to take into account the class imbalance. This ensures that one particular class is not overrepresented in either the training or test data. Then, some additional pre-processing is done.

To be more precise, the missing values are imputed by using the most frequent value for that variable, the categorical variables are one-hot encoded to make them binary variables[4] and the variables are standardized by subtracting the mean and scaling to unit variance. It is important that all these pre-processing steps are only fitted on the training data to avoid data leakage.

Next, the inner-CV is applied to the k-1 training folds or groups from the outer CV. The set of hyperparameters of the particular model and feature reduction technique[5] are optimized (cfr. infra) and is then used to configure the model. Here, the best model is decided by the weighted F1-score. The best model returned from CV is then evaluated using the last fold or group. In other words, the best model is used to predict the labels of the test set of that particular split. Based on these predictions, several metrics are computed: balanced accuracy (which is basically the average of recall obtained on each class), weighted precision, weighted recall, weighted F1-score and MCC.

This method is repeated k times, and the final CV score is computed by taking the mean of all k scores. Because the dataset is rather small, the values for k for both inner and outer CV were made not too small so that there was enough data to train on. For the outer loop, a value of k=10 was decided while for the inner loop it was put at k=5. This means that the inner procedure was done ten times, leading to ten results for each metric. Furthermore, the model optimization in the inner loop used 5-fold CV to find the best hyperparameters.

The downside of using nested CV is that it dramatically increases the training and evaluation computation times of the models. If $n*k$ models are trained for non-nested CV, then the number of models to be trained increases to $k*n*k$. However, as we don't have a lot of observations, the computation times should still be feasible.

For each of the classifiers, we had to make sure that the models take class imbalance into account. For RF and SVM, there is a hyperparameter that automatically adjusts weights inversely proportional to class frequencies in the input data. For XGB, we had to use a separate function to estimate sample weights by class (but it uses the same formula as RF and SVM). This was then entered at the fitting phase.

[4] Note that RF, XGB and the non-linear kernels of SVM are able to work with categorical variables without one-hot encoding.

[5] It is possible to put all the models that one wants to compute into one nested cross-validation procedure, but this would mean that there would be no score for each separate machine learning algorithm obtained as the nested CV will only output the ML algorithm that performs best for this particular problem. That is why it was decided to only test one algorithm in each loop, so that there is a score for each machine learning algorithm that is tested. This way, the different techniques can be compared.

3.6.2 Hyperparameter optimization

There are several techniques that can be used for the hyperparameter optimization in combination with the nested cross-validation process.

The first two options are non-automated hyperparameter tuning techniques. In Grid Search, a grid of hyperparameters values for each of the hyperparameters is set up. All possible combinations of the parameters in the grid will be tried. Meanwhile, Randomized Search will have a range of hyperparameter values as input but will only try a small subset of them based on the number of iterations that is decided to be done. Therefore, Grid Search will often be slower than Random Search when many values for each hyperparameter are given, but it can lead to better results. On the contrary, Random Search is often faster but potentially misses some important points in the search space. However, both Random and Grid search pay no attention to past results at all and would keep searching across the entire range of the number of estimators even though it's clear the optimal answer (probably) lies in a small region (Koehrsen, 2018). Potentially, one could first try Random Search to get an idea which parameters work best and then form a grid based on them to find the optimal combination.

Two other options can be labelled as automated hyperparameter tuning techniques. Bayesian Optimization uses probability theory to find the minimum of a function. It can reduce the number of search iterations by choosing the input values bearing in mind the past outcomes. Compared to randomized grid search, this method offers the advantage that it considers the structure of the search space to optimize computation time. Example of Python packages to implement this are Hyperopt and Optuna. Optuna has the advantage that it can prune trials, which is a form of early-stopping which terminates unpromising trials early to save computing time for more promising trials. In their research, Bergstra et al. (2011) concluded that the Tree-structured Parzen Estimator Approach (TPE) exceeds the performance of a brute-force random search in two difficult hyperparameter optimization tasks involving deep belief networks. Furthermore, they are more practical since they take less time to compute. Finally, Genetic Algorithms try to apply natural selection mechanisms to data science. This is done by first computing some models with some predefined hyperparameters. Then, some offspring are generated that are close to the hyperparameters of the best models until we have the same amount of models as first. This process is repeated several times. In the end, only the best models should survive the process.

It should be noted that the results are highly dependent on the chosen grid space and the dataset. That is why it is always better to try at least some of them instead of going for one particular technique (Ippolito, 2019).

During their research, Wendt et al. (2020) concluded that Hyperopt is their selection as the best Bayesian method. Furthermore, Hyperopt is faster than random search. However, the random search was more reliable than Hyperopt. In other words: 'While Hyperopt reaches the grid search score slightly faster than random search, one can be much more confident that the random search reaches it at the end of the test'. Furthermore, these Bayesian methods have the downside that they have the potential to get stuck in a local optimal point. For example, a SVM algorithm that only optimizes a linear kernel.

To conclude, Successive Halving (SH) is mentioned as well since it has much potential: it is able to find parameter combinations that are just as accurate as grid search, in much less time. SH is an iterative selection process where all candidates (the parameter combinations) are evaluated with a small amount of resources at the first iteration. Only some of these candidates are selected for the next iteration, which will be allocated more resources. For parameter tuning, the resource is typically the number of training samples, but it can also be an arbitrary numeric parameter such as the number of trees in a random forest. However, its potential lies for the most part in hyperparameter optimization situations where the data set is much bigger than the one in this study. If it were to be used here, it would be with a numeric parameter as resource (Scikit-learn developers, 2020).

3.6.3 Feature Importances

While having the ability to predict the response of a certain question (e.g. Finances) is useful on its own, gaining more interpretation of the machine learning models would be beneficial. Similar to what was mentioned in RFECV (cfr. supra), the importance of each feature in predicting the dependent variable can be computed. However, this computation differs for the classifiers that were used.

For Extreme Gradient Boosting, there are three metrics available to measure feature importance:

1. **Gain:** This is the contribution of the corresponding variable and is calculated by taking each variable's contribution for each tree in the model.

2. **Cover(age):** The number of observations related to this variable. For example, if you have 100 observations, 4 features and 3 trees, and suppose feature1 is used to decide the leaf node for 10, 5, and 2 observations in tree1, tree2 and tree3 respectively; then the metric will count cover for this feature as 10+5+2 = 17 observations..
3. **Weight:** Also known as frequency. The relative number of times a particular variable occurs in the trees of the model.
   (Abu-Rmileh, 2019)

The weight will often be low for binary variables, as they can be used at most once to split in each tree. Therefore, gain and coverage are preferred.

Meanwhile, Random forest uses either the mean decrease in Gini impurity or the information gain to obtain the importance of the variables. Both measure the quality of a split.
The first metric measures the total decrease in node impurity from splitting on the variable, averaged over all trees. Impurity measures how often a randomly chosen record from the data set, used to train the model, will be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset. Thus, it is the probability of a new record being incorrectly classified at a given node in a tree, based on the training data. The larger the value of the mean decrease in Gini, the better (Han, Guo & Yu, 2016).
Meanwhile, information gain measures the reduction in entropy by splitting a dataset according to some value of a random variable. Entropy can be described as the amount of variance (and thus uncertainty) that the data has. The higher the variance, the higher the entropy. Thus, the performance of potential splits is calculated by subtracting the weighted entropies of each branch (i.e. weighting the entropy of each branch by the relative amount of observations in the split) from the original entropy. The split that maximizes this calculation is then chosen (Zhou, 2019).
Both metrics can be used in this multi-class setting. Furthermore, Raileanu & Stoffel (2004) argue that they will lead to similar performance as the metrics only disagreed in 2% of all cases, leading to the conclusion that there is no significant difference between the two criteria. As these metrics are part of the hyperparameter optimization of the RF model as well, it was decided to go for the metric that was used as splitting criterion in the best performing model.

3.6.4 Significance Features and Odds Ratios

Once the most important variables from the classifying algorithms are determined and the assumptions of the OLR model have been checked, one can look at the results of said OLR model. The model will output a coefficient for each variable along with a p-value.

An asterisk behind the coefficient of a variable means that the variable is significant on the 0.10 (one asterisk), 0.05 (two asterisks) or 0.01 (three asterisks) level. The p-value is the probability of obtaining an effect at least as extreme as the one in the sample data, assuming that the null hypothesis is correct (The Minitab Blog, 2014). Thus, a small p-value is strong evidence against the null hypothesis. The null hypothesis here is that the independent variable has no influence on saving habits. For example, if the p-value is below 0.01, then the variable is significant at the 0.01 level. The null hypothesis can then be rejected and it can be assumed that the variable does have a significant (positive or negative) influence on the saving habits of the surveyee.

Now, the coefficients will be interpreted. The interpretation was achieved using the technique of UCLA's Institute for Digital Research & Education (s.d.). First, the coefficients are exponentiated since they currently are in logarithmic form. The exponentiated coefficients will be called the odds ratios. For each of the significant variables, the interpretation is as follows:

"If *variable*, then the odds of the surveyee being a big or mediocre saver versus being a big spender (i.e. surveyee being more likely to be a saver) is *odds ratio* times that of a survey in which *variable* is not present or zero, keeping all other variables constant". As most variables are numeric, their values are often not equal to one. To obtain the right value, the following formula can be used:

$$Total\ odds\ ratio = \begin{cases} 1 + (Odds\ ratio - 1) * value\ variable & If\ Odds\ ratio > 1 \\ 1 - (1 - Odds\ ratio) * value\ variable & If\ Odds\ ratio < 1 \end{cases}$$

This shows why changing the Likert-scale questions to -2 to 2 (from 1 to 5) eases interpretation, as such variable being 0 can now be interpreted as the person being neutral concerning that question. The interpretation of the odds ratios of these variables is thus:

"If surveyee responds positively on question concerning *variable*, then the odds of the surveyee being a big or mediocre saver versus being a big spender (i.e. surveyee being more likely to be a saver) is *odds ratio* times that of a survey in which the surveyee answers neutrally on the question concerning *variable*, keeping all other variables constant". For categorical variables, the odds ratio is comparing the specific category mentioned with the default category in the case of one-hot encoding. This coding scheme is also often referred to as treatment or dummy coding.

Chapter 4: Results

4.1 Factor Analysis

As mentioned earlier, factor analysis could not been used as a dimensionality reduction in combination with a classifier. For completeness purposes, a separate factor analysis was done to show that it can be used in conjunction with the machine learning techniques. The goal was to find some features that are linearly related to a smaller number of underlying, unobservable factors (often called hidden or latent variables).
Doing this for the full dataset resulted in 49 factors which explained 52.49% of the cumulative variance. However, it was observed that the proportion of variance explained by the factors is especially high for the one-hot encoded categorical variables and the informational ones (age, height, weight). For most Likert-scale variables, this proportion is already considerably smaller. This shows that FA is not a good idea to use when there is a combination of both Likert-scale data from closed survey questions and categorical data resulting from other survey questions. These findings are in line with what was expected ahead of this research.

In an effort to get any meaningful insight, FA was retried with only the Likert-scale variables. The KMO now has a higher value 0.816, indicating that FA can also be performed. This time, the analysis results in 37 factors. Looking at the communalities, it can also be observed that more variables now have a bigger amount of their variance explained by these 37 factors. Furthermore, the loadings now also seem to explain some common variance of several variables combined. For example, Factor 0 explains the common variance in people who enjoy going to more sophisticated/'higher class' activities like Classical Music, Opera, Reading and Art exhibitions. Meanwhile, Factor 1 focusses more on people who find their looks and social status important (Shopping, Appearance and gestures, Knowing the right people, Shopping centres, Branded clothing, Spending on looks).
As can be observed, FA can be useful to compress the total survey and find hidden relationships. This information can then be used in future surveys. Furthermore, it can also ease data interpretation. However, the interpretation can also be quite subjective, leading to different researchers coming to different conclusions. Furthermore, not each dataset will work with FA as there are some hard assumptions (no outliers, linearity, interval data). These disadvantages advocate for combining the method with machine learning techniques.

4.2 Predicting performance

The result for each feature reduction technique with all modelling algorithms (and the baseline without) can be found in Tables 3, 4, 5, 6 and 7. The values are the mean scores over the ten different folds, while the values between brackets is the standard deviation of said values. As mentioned before, it can be observed that there are two RF applications for the baseline situation. One is scaled (S), while the other is not (NS). The results show that scaling does not seem to decrease performance for the classifier. Furthermore, scaling all features was absolutely necessary for SVM and the feature extraction methods. For example, SVM considers the distances between observations. These distances will differ for non-scaled and scaled cases, therefore it was decided that the data would be standardized for each model that is performed.

It can be concluded that Extreme Gradient Boosting using the TPE optimization technique is the best performing with a weighted F1-score of 0.546, closely followed by the Random Forest algorithm (Table 3). However, it seems that the RF and XGB algorithms suffer from the dimensionality reduction techniques, while the SVM algorithm sees a (slight) improvement after using PCA (Table 4). Therefore, while PCA's output as an explanatory method was very hard to get any insight from, it has proven to have some value when it is used in combination with a Machine Learning algorithm. The reason that it could not outperform the best classifiers could be that the principal components do not necessarily have any correlation to the classification score. If the third principal component is the one that can actually separate the classes, but you only keep the two first principal components since these already cover 95% of the total variance, then you lose the variable with the classifying power. Furthermore, even keeping all variance after PCA may decrease performance, since some classifiers (e.g. RF) are sensitive to rotation of the feature axes (Rodriguez et al., 2006).

What can be observed here as well, is that PCA is slightly outperforming the supervised data reduction techniques. This might be surprising, as one could argue that using supervised techniques means that the model has more information for the optimization. However, this result is in line with previous research of Martinez & Kak (2001). They concluded that PCA can outperform LDA when the training data set is small and that PCA is less sensitive to different training data sets. This latter observation can be seen in the smaller standard deviation for the PCA models compared to the LDA ones.

Finally, the feature selection technique (RFECV) has outperformed the feature extraction techniques, although it is still performing worse than the baseline.

Removing features that are unrelated to the target label can decrease overfitting, which results in better generalization. However, it seems that overfitting is no real issue here thanks to the nested cross-validation and hyperparameter optimization applied in this research. Therefore, one could argue that these feature reduction methods would have had more success if this technique had not been used. This is also in line with the study by Clark & Provost (2019): 'Using state-of-the-art hyperparameter-selection methods, applying dimensionality reduction (DR) does not add value beyond supervised regularization, and can often diminish performance. However, if regularization is not done well (e.g., one just uses the default regularization parameter), DR does have relatively better performance—but these approaches result in lower performance overall. These latter results provide an explanation for why practitioners may be continuing to use DR without undertaking the necessary comparison to using the original features. However, this practice seems inappropriate in light of the main results, if the goal is to maximize generalization performance'.

For the hyperparameter optimization, it can be concluded that TPE has the potential to outperform both grid and randomized search, provided that it gets enough iterations to find the best parameters and that it is able to stay out of the local minima. In this research, TPE was often able to perform best for only the XGB classifier. It is believed that the optimization got stuck in a local optima while used with SVM or PCA and that the algorithm could have performed better for RF if it was allowed to do more iterations. Since TPE turned out to run faster than its counterparts during this research, it is a good recommendation to try this out.

Table 3 Performance Baseline Models

Classifier	HOT	Accuracy	Precision	Recall	F1	MCC
RF (NS)	Grid	0.531 (0.036)	0.538 (0.036)	0.532 (0.034)	0.532 (0.034)	0.294 (0.053)
	Rand.	0.543 (0.027)	0.546 (0.027)	0.543 (0.026)	0.542 (0.027)	0.312 (0.040)
	TPE	0.524 (0.040)	0.528 (0.043)	0.525 (0.042)	0.524 (0.043)	0.285 (0.063)
RF (S)	Grid	0.545 (0.021)	0.547 (0.021)	0.545 (0.021)	0.544 (0.021)	0.315 (0.032)
	Rand.	0.532 (0.025)	0.536 (0.027)	0.532 (0.027)	0.531 (0.026)	0.296 (0.040)
	TPE	0.504 (0.027)	0.511 (0.027)	0.505 (0.029)	0.505 (0.027)	0.254 (0.043)
SVM	Grid	0.514 (0.029)	0.519 (0.032)	0.514 (0.031)	0.513 (0.031)	0.269 (0.046)
	Rand.	0.511 (0.034)	0.515 (0.037)	0.511 (0.035)	0.509 (0.035)	0.265 (0.053)
	TPE	0.429 (0.058)	0.410 (0.118)	0.424 (0.066)	0.407 (0.104)	0.143 (0.087)
XGB	Grid	0.533 (0.021)	0.536 (0.023)	0.533 (0.022)	0.532 (0.022)	0.298 (0.034)
	Rand.	0.543 (0.013)	0.546 (0.014)	0.544 (0.012)	0.543 (0.014)	0.314 (0.018)
	TPE	**0.546 (0.024)**	**0.551 (0.027)**	**0.547 (0.024)**	**0.546 (0.024)**	**0.317 (0.037)**

Table 4 Performance PCA Models

Classifier	HOT	Accuracy	Precision	Recall	F1	MCC
RF	Grid	0.437 (0.025)	0.443 (0.030)	0.439 (0.027)	0.437 (0.024)	0.153 (0.042)
	Rand.	0.467 (0.030)	0.470 (0.036)	0.466 (0.031)	0.465 (0.032)	0.198 (0.046)
	TPE	0.357 (0.052)	0.360 (0.052)	0.362 (0.053)	0.360 (0.052)	0.038 (0.080)
SVM	Grid	0.515 (0.034)	0.518 (0.035)	0.514 (0.034)	0.513 (0.034)	0.270 (0.052)
	Rand.	**0.516 (0.035)**	**0.520 (0.036)**	**0.516 (0.035)**	**0.515 (0.035)**	**0.272 (0.053)**
	TPE	0.385 (0.032)	0.382 (0.029)	0.382 (0.030)	0.377 (0.028)	0.075 (0.047)
XGB	Grid	0.468 (0.036)	0.473 (0.036)	0.470 (0.037)	0.467 (0.037)	0.202 (0.056)
	Rand.	0.456 (0.051)	0.461 (0.055)	0.457 (0.051)	0.456 (0.052)	0.182 (0.078)
	TPE	0.464 (0.050)	0.467 (0.053)	0.464 (0.051)	0.464 (0.052)	0.193 (0.076)

Table 5 Performance LDA Models

Classifier	HOT	Accuracy	Precision	Recall	F1	MCC
RF	Grid	0.451 (0.042)	0.450 (0.044)	0.449 (0.042)	0.447 (0.044)	0.173 (0.063)
	Rand.	0.452 (0.048)	0.450 (0.050)	0.449 (0.049)	0.446 (0.050)	0.174 (0.072)
	TPE	0.450 (0.050)	0.447 (0.053)	0.447 (0.051)	0.444 (0.053)	0.171 (0.075)
SVM	Grid	0.454 (0.043)	0.462 (0.041)	0.453 (0.041)	0.453 (0.042)	0.178 (0.065)
	Rand.	0.462 (0.034)	**0.470 (0.040)**	0.461 (0.034)	0.460 (0.033)	0.191 (0.054)
	TPE	0.458 (0.046)	0.462 (0.047)	0.456 (0.046)	0.456 (0.047)	0.183 (0.069)
XGB	Grid	**0.464 (0.038)**	0.466 (0.041)	**0.464 (0.038)**	**0.463 (0.039)**	**0.194 (0.058)**
	Rand.	0.448 (0.053)	0.452 (0.055)	0.449 (0.053)	0.448 (0.054)	0.170 (0.080)
	TPE	0.454 (0.047)	0.460 (0.050)	0.455 (0.047)	0.456 (0.048)	0.178 (0.072)

Table 6 Performance KDA Models

Classifier	HOT	Accuracy	Precision	Recall	F1	MCC
RF	Grid	0.440 (0.031)	0.501 (0.046)	0.449 (0.029)	0.425 (0.028)	0.183 (0.056)
	Rand.	0.440 (0.033)	0.438 (0.033)	0.437 (0.033)	0.435 (0.033)	0.157 (0.050)
	TPE	0.452 (0.024)	0.493 (0.044)	0.457 (0.025)	0.443 (0.033)	0.194 (0.037)
SVM	Grid	**0.503 (0.034)**	**0.510 (0.040)**	**0.505 (0.035)**	**0.503 (0.036)**	**0.255 (0.054)**
	Rand.	**0.503 (0.034)**	**0.510 (0.040)**	**0.505 (0.035)**	**0.503 (0.036)**	**0.255 (0.054)**
	TPE	0.499 (0.039)	0.507 (0.044)	0.502 (0.040)	0.500 (0.040)	0.250 (0.062)
XGB	Grid	0.452 (0.035)	0.452 (0.039)	0.451 (0.036)	0.449 (0.036)	0.175 (0.054)
	Rand.	0.464 (0.045)	0.470 (0.048)	0.465 (0.046)	0.464 (0.047)	0.194 (0.070)
	TPE	0.465 (0.040)	0.474 (0.035)	0.467 (0.039)	0.467 (0.039)	0.196 (0.061)

Table 7 Performance RFECV Models

Classifier	HOT	Accuracy	Precision	Recall	F1	MCC
RF	Grid	0.527 (0.036)	0.529 (0.037)	0.528 (0.036)	0.527 (0.036)	0.289 (0.054)
XGB	Grid	**0.528 (0.043)**	**0.532 (0.044)**	**0.530 (0.045)**	**0.529 (0.044)**	**0.291 (0.067)**

Table 8 shows the result of all ensemble algorithms. The classifiers mentioned in the Table are the base estimators. For the multiple classifiers in the stacking case, some additional classifiers were added with their respective default parameters (cfr. supra). For RF, SVC and XGB, their respective best hyperparameters from the previous chapter were used. The meta-model for all stacking ensembles was a Logistic Regression. For the voting classifiers, only the optimized models were attempted. Overall, the soft voting ensemble using only RF and XGB performed best, even outperforming all models from the previous phase with an average weighted F1-score of 0.557. Furthermore, its results were less variate, having a smaller standard deviation compared to similar models. This comes to no surprise, as it has been mentioned before that soft voting ensembles perform well if well-calibrated classifiers are used. The results also show that adding random, unoptimized classifiers to the ensemble does not lead to better performance.

Table 8 Performance Ensembles

Method	Classifiers	Accuracy	Precision	Recall	F1	MCC
Stacking (Meta: LR)	RF+SVM	0.537 (0.026)	0.536 (0.030)	0.532 (0.025)	0.528 (0.026)	0.303 (0.040)
	RF+XGB	0.556 (0.022)	0.557 (0.022)	0.553 (0.024)	0.553 (0.024)	0.331 (0.034)
	SVM+XGB	0.546 (0.027)	0.549 (0.035)	0.544 (0.026)	0.542 (0.028)	0.318 (0.043)
	R+S+X	0.551 (0.034)	0.554 (0.040)	0.549 (0.033)	0.548 (0.034)	0.325 (0.053)
	Multiple	0.534 (0.030)	0.534 (0.034)	0.531 (0.030)	0.529 (0.031)	0.298 (0.046)
Hard Voting	R+S+X	0.543 (0.032)	0.546 (0.037)	0.541 (0.032)	0.540 (0.033)	0.312 (0.049)
Soft Voting	RF+SVM	0.544 (0.031)	0.545 (0.032)	0.543 (0.030)	0.541 (0.030)	0.314 (0.047)
	RF+XGB	**0.558 (0.010)**	**0.562 (0.012)**	**0.557 (0.010)**	**0.557 (0.010)**	**0.335 (0.016)**
	SVM+XGB	0.543 (0.020)	0.547 (0.022)	0.543 (0.020)	0.542 (0.019)	0.313 (0.032)
	R+S+X	0.549 (0.035)	0.552 (0.035)	0.549 (0.033)	0.548 (0.033)	0.321 (0.052)

4.3 Model Interpretation

The importance of the variables can then be derived from the XGB model, as it is the best performing model. Since the performance of the RF algorithm in the baseline case was close to that of extreme gradient boosting, it was decided to look at its importance plot for comparison purposes. As the best performing model of this algorithm used Gini impurity as splitting criterion, this metric was also used to measure the feature importances. Furthermore, the most important features using entropy for the same fold were very similar to what was observed with Gini impurity. This is in line with was mentioned in Chapter 3. The reader is referred to Appendix C if one wants to compare the plots. The importance of all variables according to each metric for the best fold can be found in Figures 7, 8, 9 and 10.

An important remark when computing these feature importances is that the mechanism may be biased, in the sense that it tends to increase the importance of continuous or categorical variables with a lot of distinct values. For example, Strobl et al. (2007, page 2) pointed out that 'the variable importance measures of Breiman's original Random Forest method ... are not reliable in situations where potential predictor variables vary in their scale of measurement or their number of categories'. In this research, the variables are standardized, but it is true that height and weight have a much higher amount of distinct values. Therefore, they will not be considered as important variables even though the plots suggests otherwise.

As could have been expected, some questions related to financing (entertainment spending, spending on looks) are really important for the young people's answer on whether they save all the money they can. An interesting variable that all metrics seem to agree on to be important, is the answer on the question whether the surveyee tries to do tasks as soon as possible and not leaving them until the last minute (i.e. prioritising workload). It can be argued that people that are responsible about handling their workload, will also be responsible with their money. These people will likely find achievement important as well, leading to a similar importance.

Another argumentation can be made about the question whether the surveyee has trouble getting up, as the young people who find it very difficult to get up in the morning might be those that go out more often and focus more on partying, thus spending more money. The questions about drinking alcohol being important follows the same reasoning.

To conclude the interpretation of the variable importances, it is thus observed that there are two groups of variables that obtain high importance values; those that make a distinction between both responsible young people who prioritise workload, focus on achievements, take time to make decisions, think ahead and are reliable opposed to young people who like to party, enjoy some drinks, spend money on looks and having trouble getting up in the morning. Naturally, this example is non-exhaustive, as more arguments could be made for other variables that appear in the plots below.

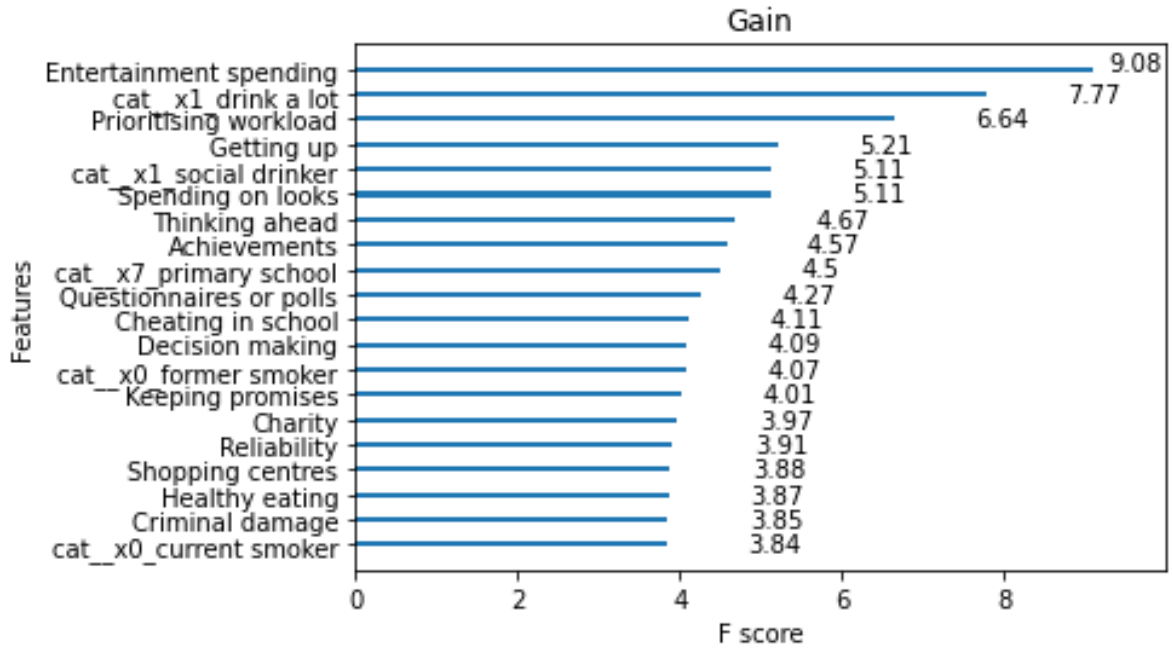

Figure 7 Extreme Gradient Boosting: Gain

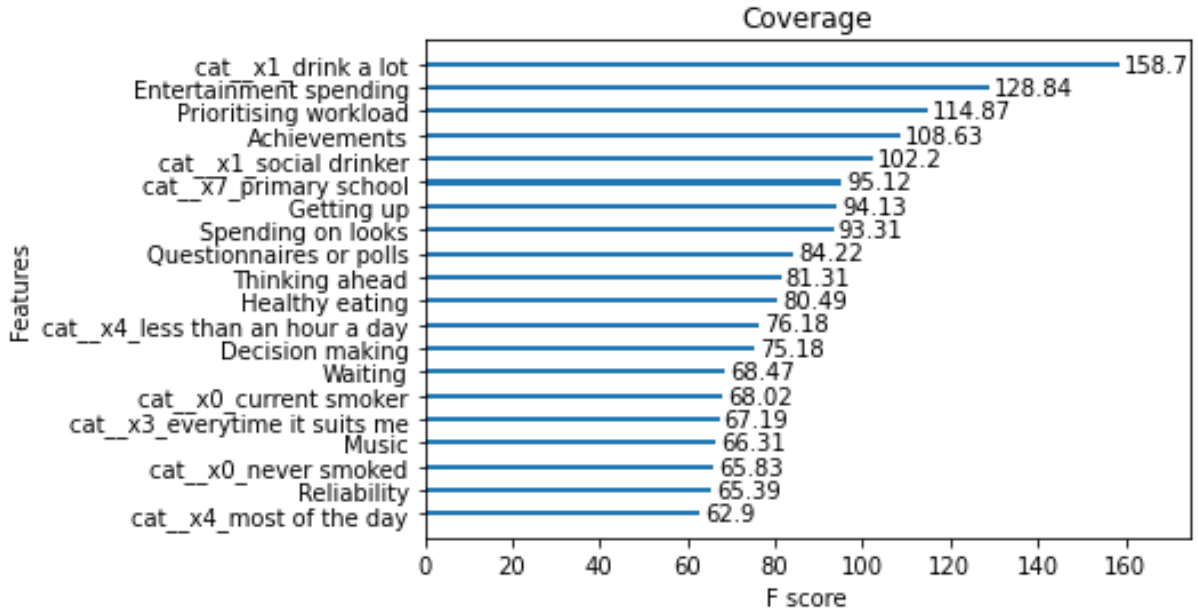

Figure 8 Extreme Gradient Boosting: Coverage

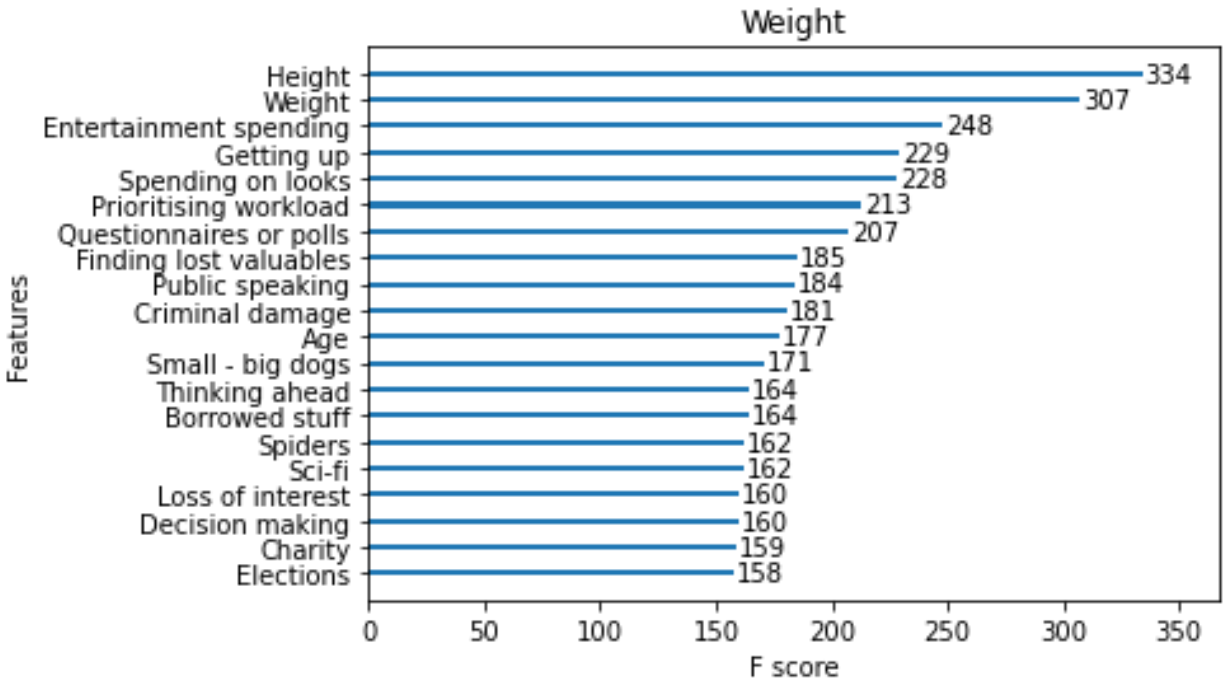

Figure 9 Extreme Gradient Boosting: Weight

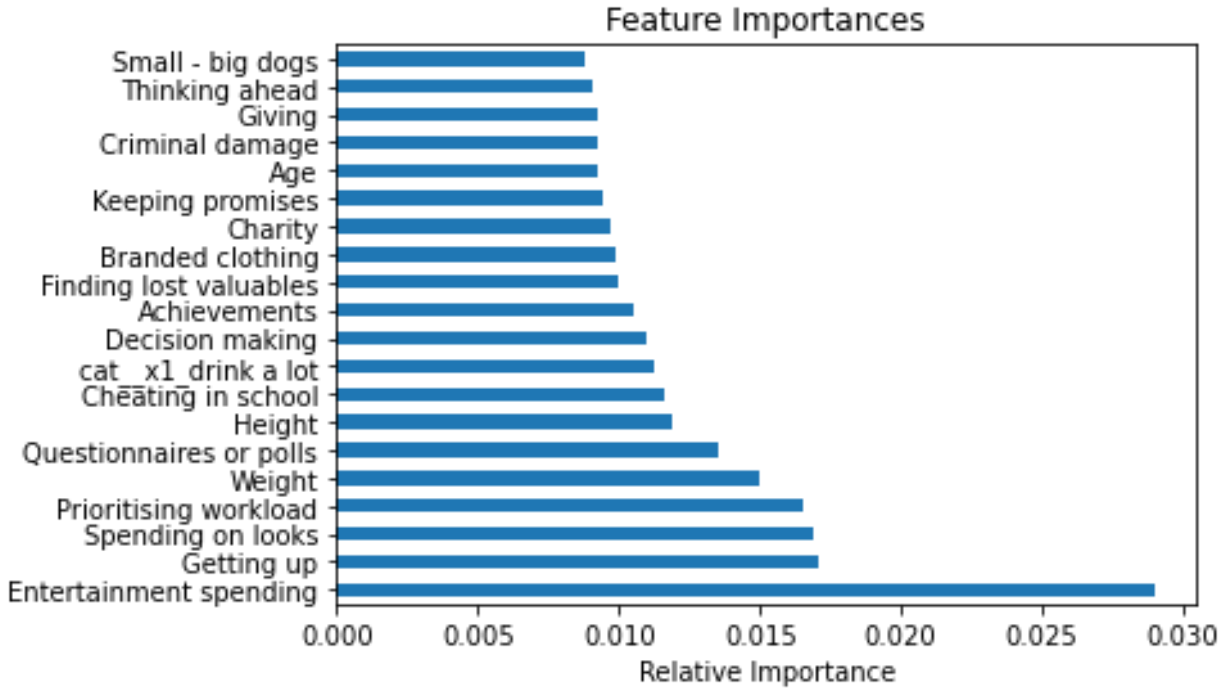

Figure 10 Random Forest: Mean Decrease in Gini

4.4 Significance and Coefficient Variables

The last part of the research consisted of performing the OLR. As mentioned before, the assumptions of ordered dependent variables and categorical or continuous have been fulfilled. As the highest VIF value was 3.20, the assumption of no multi-collinearity has been satisfied as well. However, some variables had to be deleted in order for the model to meet the proportional odds assumption.

As the model is now not violating any assumptions, one can check the summary of the model to gain insights. In particular, a closer look at the Odds Ratios of the most significant variables ($p < 0.05$) is given in Table 9. The complete summary can be found in Appendix D. Not all variables will be interpreted, as the formula is the same for all variables.

Comparing to the classification results, one can observe that many variables that were present in those plots are reappearing as significant variables in the regression model. Furthermore, the intuition about those variables was correct as well. People who prioritise their workload and take time to make their decisions are more likely to be savers. For example, the odds of someone who strongly agrees that they take their time to make decisions to be a saver is $1 + (1.2288 - 1) * 2 = 1.576$ times that of a surveyee who is neutral about this question (i.e. answering 0) and all other variables held constant. On the other hand, people who have trouble getting up in the morning, that like to dance and spend money on looks are less likely to be savers. For example, the odds of someone who is very interesting in dancing to be a saver is $1 - (1 - 0.8671) * 2 = 0.7342$ times that of a surveyee who is neutral about this question (i.e. answering 0) and all other variables held constant.

Furthermore, it is also worth mentioning the two significant categories. It seems like the odds of surveyees who have no siblings to be savers are 1.4608 times that of surveyees who do have siblings and all other variables held constant. Finally, the odds of surveyees who have a doctorate degree to be savers are 9.0358 times that of surveyees who have a college or bachelor degree (i.e. default category). It should be noted that this last value could be a bit inflated as only five respondents had a doctorate degree, but it is still an interesting result to keep in mind.

Table 9 Odds Ratios Significant Variables

Variable	Odds Ratio
Fantasy/ Fairy tales	0.8173**
Internet	1.1971**
Art exhibitions	0.8575**
Dancing	0.8671**
Prioritising workload	1.3532***
Criminal damage	0.9029**
Decision making	1.2288***
Finding lost valuables	1.1480**
Getting up	0.7786***
Questionnaires or polls	1.2056***
Spending on looks	0.7966***
Weight	0.9859**
Education_doctorate degree	9.0358**
Only child_yes	1.4608**

Chapter 5: Conclusion

Surveys have been around for a long time. Therefore, many different ways have been proposed to analyse the data of such surveys. In this research, some additional methods have been investigated. Since the data is tabular, it is argued that Machine Learning can add value to the survey analysis. The 'Young People Survey' by the Comenius University of Bratislava was used to illustrate this. In particular, it was shown that classification algorithms such as Random Forest, Support Vector Machines and Extreme Gradient Boosting can be used to predict the response of a surveyee on a particular question (which was saving habits here). Furthermore, the potential of using other techniques than PCA to summarize the data by reducing its dimensionality was investigated as well. Because the survey had only 1010 responses, it was necessary to perform nested cross-validation. While this dramatically increased the training and evaluation computation times of the models, it made certain our models did not overfit. That is why it is advised to do this for other surveys as well, since most surveys only have a few thousand responses at best.

Overall, it could be concluded that Extreme Gradient Boosting without any dimensionality reduction performed the best with a weighted F1-score of 0.546, also scoring the highest for all other metrics. While dimensionality reduction did not increase predicting performance for this dataset, it still reduced the computation time. Therefore, it is still an option if results are needed fast. It is also recommended for analysis where the data and algorithms used have not been as pre-processed and optimized as thoroughly as in this study. Finally, it is also a good idea if there are few samples, as more data is needed when there are many dimensions for the model to be able to generalize.

An important utility of these machine learning algorithms is to use its results to gain insights about the data. Therefore, the importance of the variables in these tables were investigated using the gain, coverage and frequency measures of the Extreme Gradient Boosting method and the mean decrease in Gini of the Random Forest method. Overall, it was observed that there are two groups of variables that obtain high importance values: those that make a distinction between both responsible young people who prioritise workload, focus on achievements, take time to make decisions, think ahead and are reliable as opposed to young people who like to party, enjoy some drinks, spend money on looks and having trouble getting up in the morning. Naturally, this example is non-exhaustive, as more arguments could be made for other variables.

The Ordinary Logistic Regression confirmed these conclusions and gave some additional insights concerning the significance and direction of those variables. Furthermore, it was observed that being an only child or having a doctorate degree also increases the odds of a young Slovakian person to be a saver.

This research has also shown that it is possible to complement the analysis of Machine Learning techniques with other multivariate analyses. As such, Factor Analysis was performed to obtain insights into some latent variables. However, a big downside is the inability to perform this analysis on both categorical and continuous variables.

Finally, if the user is mostly interested in improving predicting performance rather than interpreting the results, there are several ways to do so. The easiest way is to use an ensemble of the Machine Learning algorithms that have already been optimized before. In this research, both stacking and voting have been used. The latter can also be divided into hard and soft voting. As the models each have their own characteristics (and thus make different mistakes), they can learn from each other to result in a better model. This was also the case in this study, where combining Random Forest and Extreme Gradient Boosting using soft voting resulted in an improved weighted F1-score of 0.557, again also performing best in all other metrics.

.

Chapter 6: Limitations and Future Research

The biggest limitation of this study is the fact that only one dataset has been used. For comparison purposes, it could be interesting to check if the performances were similar for datasets of other sizes (both row-wise as column-wise). Moreover, using a dataset that has open questions to perform NLP techniques on could further improve the study as well.

In addition, a look into the use of Deep Neural Networks could also be beneficial for survey analysis. While the research of these kind of algorithms on tabular data is still very young, it has already proven to be able to perform very well in multiple studies, even outperforming the traditional Machine Learning models. Some of these new algorithms can be found in Appendix C. Furthermore, it can again be combined with the traditional ML algorithms in an ensemble to increase performance. This is in line with the research of Schwartz-Ziv & Armon (2021), where they show that an ensemble of the deep models and XGB performs better on these datasets than XGB alone. Therefore, the reader is advised to initially stick to the traditional methods, after which some new DNNs can be attempted (and combined) in an attempt to improve predictive performance.

Another great potential for survey data that has not been mentioned before, is to use it in recommender systems. As this dataset has many different groups of questions, one could easily compute recommendations for one group based on the answers given in another. For example, if it is known what a person listens to, it could be interesting to see if it is possible to predict which kind of movies he/she/they would like. For an interesting application, the reader is referred to the paper of He et al. (2017), where a general framework called Neural Collaborative Filtering (NCF) is introduced for building Recommender Systems using Deep Neural Networks. In these models, one replaces the matrix factorization with a Neural Network, which outperforms the former.

Finally, in the case that the reader is interested in predicting multiple variables at once, it could be intriguing to look into Multi-Task Learning. This method has two advantages. First, it could be more efficient as only one single model would need to be trained and used (compared to making a separate multi-class classification for each variable). Second, it could be more effective as the model's generalisation ability would be increased.

References

[1] Abu-Rmileh, A. (2019). The Multiple faces of 'Feature importance' in XGBoost. *Towards Data Science.* Retrieved from https://towardsdatascience.com/be-careful-when-interpreting-your-features-importance-in-xgboost-6e16132588e7

[2] Akers, D. (2018). Exploring, Analysing and interpreting data with Minitab 18. *Compass Publishing.*

[3] Arik, S.O. & Pfister, T. (2019). TabNet: Attentive Interpretable Tabular Learning. arXiv.

[4] Baboota, R., & Kaur, H. (2018). Predictive analysis and modelling football results using machine learning approach for English Premier League. *International Journal of Forecasting, 35*. doi:10.1016/j.ijforecast.2018.01.003

[5] Babu, D. (2020). Dimensionality Reduction using Factor Analysis in Python. *Analytics Vidhya.* Retrieved from https://www.analyticsvidhya.com/blog/2020/10/dimensionality-reduction-using-factor-analysis-in-python/

[6] Beernaert, B. & Goossens, D. (2020). The impact of the schedule on match outcome in football competitions.

[7] Ben-Hur, A. et al. (2008). Support Vector Machines and Kernels for Computational Biology. *PLoS Computational Biology*. doi: 0.1371/journal.pcbi.1000173

[8] Bergstra, J., Bardenet, R., Bengio, Y & Kégl, B. (2011). Algorithms for Hyper-Parameter Optimization. Advances in Neural Information Processing Systems.

[9] Bock (s.d.). Factor Analysis and Principal Component Analysis: A Simple Explanation. *DISPLAYR.* Retrieved from https://www.displayr.com/factor-analysis-and-principal-component-analysis-a-simple-explanation/

[10] Brant, R. (1990). Assessing Proportionality in the Proportional Odds Model for Ordinal Logistic Regression. *Biometrics*, 46(4), 1171–1178. doi:10.2307/2532457

[11] Brownlee, J. (2016). A Gentle Introduction to the Graident Boosting Algorithm for Machine Learning. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/

[12] Brownlee, J. (2020). Nested Cross-Validation for Machine Learning with Python. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/

[13] Brownlee, J. (2020). One-vs-Rest and One-vs-One for Multi-Class-Classification. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/

[14] Cavdaroglu, B., & Atan, T. (2020). Determining matchdays in sports league schedules to minimize rest differences. *Operations Research Letters, 48*. doi:10.1016/j.orl.2020.03.001

[15] Chicco, D. & Jurman, G. (2020) The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. BMC Genomics, 21, 6. doi:10.1186/s12864-019-6413-7

[16] Clark, J. & Provost, F. (2019). Unsupervised dimensionality reduction versus supervised regularization for classification from sparse data. Data Mining and Knowledge Discovery. 33:871–916. doi:10.1007/s10618-019-00616-4

[17] Data Science Project Management (s.d.). CRISP-DM Overview. Retrieved from http://www.datascience-pm.com/crisp-dm-2/

[18] DataCamp Team (2020). Choosing Python or R for Data Analysis? An Infographic. *DataCamp.* Retrieved from https://www.datacamp.com/community/tutorials/r-or-python-for-data-analysis#comments

[19] Davidson, R. & MacKinnon, J.G. (2004). Econometric Theory and Methods. Oxford University Press, Oxford.

[20] Han, H., Guo, X., & Yu, H. (2016). Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest. 7th IEEE International Conference on Software Engineering and Service Science (ICSESS). doi:10.1109/icsess.2016.7883053

[21] Hastie, T., Tibshirani, R. & Friedman, J. (2008). The Elements of Statistical Learning (2nd ed.). Springer. 587-588.

[22] He, X. et al. (2017). Neural Collaborative Filtering. International Conference on World Wide Web. 173-182. doi: 10.1145/3038912.3052569.

[23] Heino, O. & Sillanpää, V. (2013). Forecasting football match results - A study on modeling principles and efficiency of fixed-odds betting markets in football. (Information and Service Economy). University School of Business, Aalto.

[24] Huang, X., Khetan, A., Cvitkovic, M. & Karnin, Z. (2020). TabTransformer: Tabular Data Modeling Using Contextual Embeddings. arXiv.

[25] Husson, F., Josse, J. & Pagès, J. (2010). Principal component methods - hierarchical clustering - partitional clustering: why would we need to choose for visualizing data?. Agrocampus.

[26] Ippolito, P.P. (2019). Hyperparameters Optimization. *Towards Data Science.* Retrieved from https://towardsdatascience.com/hyperparameters-optimization-526348bb8e2d

[27] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning: with Applications in R*: Springer Publishing Company, Incorporated.

[28] Joseph, R. (2018). Grid Search for model tuning. *Towards data science.* Retrieved from https://towardsdatascience.com/grid-search-for-model-tuning-3319b259367e

[29] Knust, S. (2018). Classification of Literature on Sports Scheduling. Retrieved from http://www2.informatik.uni-osnabrueck.de/knust/sportssched/sportlit_class/

[30] Koehrsen, W. (2018). An Introductory Example of Bayesian Optimization in Python with Hyperopt. *Towards Data Science.* Retrieved from https://towardsdatascience.com/an-introductory-example-of-bayesian-optimization-in-python-with-hyperopt-aae40fff4ff0

[31] Kolassa, S. (2014). Is high AIC a bad feature of the model? Retrieved from https://stats.stackexchange.com/questions/129604/is-high-aic-a-bad-feature-of-the-model

[32] Kumar, S. (2020). Nested Cross-Validation: Hyperparameter Optimization and Model Selection. *Towards Data Science.* Retrieved from https://towardsdatascience.com/nested-cross-validation-hyperparameter-optimization-and-model-selection-5885d84acda

[33] Kutner, M. H.; Nachtsheim, C. J.; Neter, J. (2004). Applied Linear Regression Models (4th ed.). McGraw-Hill Irwin.

[34] Lanaro, G. (2016). Evaluation measures for multiclass problems. *Gabriele Lanaro.* Retrieved from http://gabrielelanaro.github.io/blog/2016/02/03/multiclass-evaluation-measures.html

[35] Lee, E. (2019). Ordinal Logistic Regression and its Assumptions - Full Analysis. *Medium.* Retrieved from https://medium.com/evangelinelee/ordinal-logistic-regression-on-world-happiness-report-221372709095

[36] Lones, M. (2021). How to avoid machine learning pitfalls: a guide for academic researchers.

[37] Martinez, A.M. & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, 228-233. doi: 10.1109/34.908974.

[38] Navlani, A. (2019). Introduction to Factor Analysis in Python. *DataCamp.* Retrieved from https://www.datacamp.com/community/tutorials/introduction-factor-analysis

[39] Nayak, P. (2019). Understand searches better than ever before. *Google: They Keyword.* Retrieved from https://blog.google/products/search/search-language-understanding-bert/

[40] Piatetsky, G. (2014). CRISP-DM, still the top methodology for analytics, data mining, or data science projects. *KDnuggets*. Retrieved from https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

[41] Popov, S., Babenko, A. & Morozov, S. (2020). Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data. ICLR.

[42] Raileanu, L.E. & Stoffel, K. (2004). Theoretical comparison between the Gini Index and Information Gain criteria. *Annals of Mathematics and Artificial Intelligence,* 77-93. doi: 10.1023/B:AMAI.0000018580.96245.c6

[43] Raschka, S. & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and Tensorflow 2 (3rd ed.), Packt Publishing.

[44] Rinker, T. (2014). On the Treatment of Likert Data. University at Buffalo.

[45] Rodriguez, J., Kuncheva, L. & Alonso, C. (2006). Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. doi: 10.1109/TPAMI.2006.211

[46] Sabo, M. (2013). Young People Survey. *Kaggle.* Retrieved from https://www.kaggle.com/miroslavsabo/young-people-survey

[47] Sadawi (s.d.). Predicting the Furture: Classification. Retrieved from http://www.saedsayad.com

[48] SAS Analytics Insights (2020) Machine Learning: What it is and why it matters. Retrieved from https://www.sas.com/en_us/insights/analytics/machine-learning.html

[49] Schwartz-Ziv, R. & Armon, A. (2021). Tabular Data: Deep Learning is Not All You Need. ICML Workshop on Automated Machine Learning (AutoML)

[50] Scikit-learn developers (2020). Tuning the hyper-parameters of an estimator. *Scikit-learn.* Retrieved from https://scikit-learn.org/stable/modules/grid_search.html#searching-for-optimal-parameters-with-successive-halving

[51] Shmueli, B. (2019). Matthews Correlation Coefficient is the best classification metric you've never heared of. *Towards Data Science.* Retrieved from https://towardsdatascience.com/the-best-classification-metric-youve-never-heard-of-the-matthews-correlation-coefficient-3bf50a2f3e9a

[52] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management, 45*, 427-437. doi:10.1016/j.ipm.2009.03.002

[53] Somepalli, G. et al. (2021). SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. arXiv.

[54] Strobl, C., Boulesteix, A., Zeileis, A. & Hothorn, T. (2007). Bias in random forest variable importance measures: illustrations, sources and a solution. BMC Bionformatics, 8:25. doi:10.1186/1471-2105-8-25

[55] The Minitab Blog (2014). How to Correctly Interpret P Values. Retrieved from https://blog.minitab.com/blog/adventures-in-statistics-2/how-to-correctly-interpret-p-values

[56] Tune, P. (2020). The Unreasonable Ineffectiveness of Deep Learning on Tabular Data. Towards Data Science. Retrieved from https://towardsdatascience.com/the-unreasonable-ineffectiveness-of-deep-learning-on-tabular-data-fd784ea29c33

[57] UCLA (s.d). How do I interpret the coefficients in an ordinal logistic regression in R? Institue for Digital Research & Education. Retrieved from https://stats.idre.ucla.edu/r/faq/ologit-coefficients/

[58] Van den Poel, D. (2018). CRISP/DM [Powerpoints slides]. Retrieved from ufora.ugent.be

[59] Vanghese, D. (2018). Comparative Study on Classic Machine learning Algorithms. *Towards Data Science.* Retrieved from https://towardsdatascience.com/comparative-study-on-classic-machine-learning-algorithms-24f9ff6ab222

[60] Wendt, A., Lechner, M. & Wuchsnig, M. (2020). Speeding up Common Hyperparameter Optimization Methods by a Two-Phase-Search. doi: 10.1109/IECON43393.2020.9254801

[61] Zhou, V. (2019). A Simple Explanation of Information Gain and Entropy. Retrieved from https://victorzhou.com/blog/information-gain/

Appendices

A: Variables in Dataset

1. Explanation Questions

Table 10 Variables in Dataset

Variable	Type	Extra Information
Music preferences		I enjoy listening to…
Music	Float	
Slow songs or fast songs	Float	I prefer slow paced (1)… fast paced (5) songs
Dance	Float	
Folk	Float	
Country	Float	
Classical Music	Float	
Musical	Float	
Pop	Float	
Rock	Float	
Metal or Hardrock	Float	
Punk	Float	
Hiphop, Rap	Float	
Reggae, Ska	Float	
Swing, Jazz	Float	
Rock n roll	Float	
Alternative	Float	
Latino	Float	
Techno, Trance	Float	
Opera	Float	
Movie preferences		I enjoy watching…
Movies	Float	
Horror	Float	
Thriller	Float	
Comedy	Float	
Romantic	Float	

Sci-fi	Float	
War	Float	
Fantasy/Fairy tales	Float	
Animated	Float	
Documentary	Float	
Western	Float	
Action	Float	
Hobbies & Interests		I am interested in…
History	Float	
Psychology	Float	
Politics	Float	
Mathematics	Float	
Physics	Float	
Internet	Float	
PC	Float	
Economy Management	Float	
Biology	Float	
Chemistry	Float	
Reading	Float	
Geography	Float	
Foreign Languages	Float	
Medicine	Float	
Law	Float	
Cars	Float	
Art exhibitions	Float	
Religion	Float	
Countryside, outdoors	Float	
Dancing	Float	
Musical instruments	Float	
Writing	Float	
Passive sport	Float	
Active sport	Float	
Gardening	Float	

Celebrities	Float	
Shopping	Float	
Science and technology	Float	
Theatre	Float	
Fun with friends	Float	
Adrenaline sports	Float	
Pets	Float	
Phobias		I am afraid of…
Flying	Float	
Storm	Float	
Darkness	Float	
Heights	Float	
Spiders	Float	
Snakes	Integer	
Rats	Float	
Ageing	Float	
Dangerous dogs	Float	
Fear of public speaking	Float	
Health habits		
Smoking	Categorical	
Alcohol	Categorical	
Healthy eating	Float	I live a very healthy lifestyle
Personality traits, views on life & opinions		
Daily events	Float	I take notice of what goes on around me
Prioritising workload	Float	I try to do tasks as soon as possible and not leave them until last minute
Writing notes	Float	I always make a list so I don't forget anything
Workaholism	Float	I often study or work even in my spare time
Thinking ahead	Float	I look at things from all different angles before I go ahead
Final judgement	Float	I believe that bad people will suffer one day and good people will be rewarded

Reliability	Float	I am reliable at work and complete my tasks
Keeping promises	Float	I always keep my promises
Loss of interest	Float	I can fall for someone very quickly and then completely lose interest
Friends versus money	Float	I would rather have lots of friends than lots of money
Funniness	Float	I always try to be the funniest one
Fake	Float	I can be two faced sometimes
Criminal damage	Float	I damaged things in the past when angry
Decision making	Float	I take my time to make decisions
Elections	Float	I always try to vote in elections
Self-criticism	Float	I often think about and regret the decisions I make
Judgement calls	Float	I can tell if people listen to me or not when I talk to them
Hypochondria	Float	I am a hypochondriac
Empathy	Float	
Eating to survive	Integer	I eat because I have to. I don't enjoy food and eat as fast as I can
Giving	Float	I try to give as much as I can to other people at Christmas
Compassion to animals	Float	I don't like seeing animals suffering
Borrowed stuff	Float	I look after things I have borrowed from others
Loneliness	Float	I feel lonely in life
Cheating in school	Float	
Health	Float	I worry about my health
Changing the past	Float	I wish I could change the past
God	Float	I believe in God
Dreams	Integer	I always have good dreams
Charity	Float	I always give to charity
Number of friends	Integer	I have lots of friends
Punctuality	Categorical	Timekeeping
Lying	Categorical	Do you lie to others?
Waiting	Float	I am very patient
New environment	Float	I can quickly adapt to a new environment

Mood swings	Float	My moods change quickly
Appearance and gestures	Float	I am well mannered and look after my appearance
Socializing	Float	I enjoy meeting new people
Achievements	Float	I always let other people know about my achievement
Responding to a serious letter	Float	I think carefully before answering any important letters
Children	Float	I enjoy children's' company
Assertiveness	Float	I am not afraid to give my opinion if I feel strongly about something
Getting angry	Float	I can get angry very easily
Knowing the right people	Float	I always make sure I connect with the right people
Public speaking	Float	I have to be well prepared before public speaking
Unpopularity	Float	I will find a fault in myself if people don't like me
Life struggles	Float	I cry when I feel down or things don't go the right way
Happiness in life	Float	I am 100% happy with my life
Energy levels	Float	I am always full of life and energy
Small – big dogs	Float	I prefer big dangerous dogs to smaller, calmer dogs
Personality	Float	I believe all my personality traits are positive
Finding lost valuables	Float	If I find something that doesn't belong to me I will hand it in
Getting up	Float	I find it very difficult to get up in the morning
Interests or hobbies	Float	I have many different hobbies and interest
Parent's advice	Float	I always listen to my parents' advice
Questionnaires or polls	Float	I enjoy taking part in surveys
Internet usage	Categorical	How much time do you spend online?
Spending habits		
Finances	Float	I save all the money I can.
Shopping centres	Float	I enjoy going to large shopping centres
Branded clothing	Float	I prefer branded clothing to non branded
Entertainment spending	Float	I spend a lot of money on partying and socializing
Spending on looks	Float	

Spending on gadgets	Integer	
Spending on healthy eating	Float	I will happily pay more money for good, quality or healthy food
Demographics		
Age	Float	
Height	Float	
Weight	Float	
Number of siblings	Float	
Gender	Categorical	
Left – right handed	Categorical	
Education	Categorical	Highest education achieved
Only child	Categorical	
Village – town	Categorical	I spent most of my childhood in a…
House – block of flats	Categorical	I lived most of my childhood in a…

2. Descriptive Statistics

Variable	N	Mean	Std. dev.	Min	Max	25%	50%	75%
Music preferences								
Music	1007	**4.73**	**0.66**	1.00	5.00	5.00	5.00	5.00
Slow songs or fast songs	1008	3.33	0.83	1.00	5.00	3.00	3.00	4.00
Dance	1006	3.11	1.17	1.00	5.00	2.00	3.00	4.00
Folk	1005	2.29	1.14	1.00	5.00	1.00	2.00	3.00
Country	1005	2.12	1.08	1.00	5.00	1.00	2.00	3.00
Classical Music	1003	2.96	1.25	1.00	5.00	2.00	3.00	4.00
Musical	1008	2.76	1.26	1.00	5.00	2.00	3.00	4.00
Pop	1007	3.47	1.16	1.00	5.00	3.00	4.00	4.00
Rock	1004	3.76	1.18	1.00	5.00	3.00	4.00	5.00
Metal or Hardrock	1007	2.36	1.37	1.00	5.00	1.00	2.00	3.00
Punk	1002	2.46	1.30	1.00	5.00	1.00	2.00	3.00
Hiphop, Rap	1006	2.91	1.38	1.00	5.00	2.00	3.00	4.00
Reggae, Ska	1003	2.77	1.21	1.00	5.00	2.00	3.00	4.00
Swing, Jazz	1004	2.76	1.26	1.00	5.00	2.00	3.00	4.00
Rock n roll	1003	3.14	1.24	1.00	5.00	2.00	3.00	4.00
Alternative	1003	2.83	1.35	1.00	5.00	2.00	3.00	4.00
Latino	1002	2.84	1.33	1.00	5.00	2.00	3.00	4.00
Techno, Trance	1003	2.34	1.32	1.00	5.00	1.00	2.00	3.00
Opera	1009	2.14	1.18	1.00	5.00	1.00	2.00	3.00
Movie preferences								
Movies	1004	4.61	0.69	1.00	5.00	4.00	5.00	5.00
Horror	1008	2.79	1.41	1.00	5.00	1.00	3.00	4.00
Thriller	1009	3.38	1.20	1.00	5.00	3.00	4.00	4.00
Comedy	1007	4.49	0.78	1.00	5.00	4.00	5.00	5.00
Romantic	1007	3.49	1.21	1.00	5.00	3.00	4.00	5.00
Sci-fi	1008	3.11	1.31	1.00	5.00	2.00	3.00	4.00
War	1008	3.16	1.35	1.00	5.00	2.00	3.00	4.00
Fantasy/Fairy tales	1007	3.75	1.18	1.00	5.00	3.00	4.00	5.00
Animated	1007	3.79	1.22	1.00	5.00	3.00	4.00	5.00

Documentary	1002	3.64	1.13	1.00	5.00	3.00	4.00	5.00
Western	1006	2.13	1.14	1.00	5.00	1.00	2.00	3.00
Action	1008	3.54	1.24	1.00	5.00	3.00	4.00	5.00
Hobbies & Interests								
History	1008	3.21	1.26	1.00	5.00	2.00	3.00	4.00
Psychology	1005	3.14	1.26	1.00	5.00	2.00	3.00	4.00
Politics	1009	2.60	1.29	1.00	5.00	1.00	2.00	4.00
Mathematics	1007	2.33	1.35	1.00	5.00	1.00	2.00	3.00
Physics	1007	2.06	1.23	1.00	5.00	1.00	2.00	3.00
Internet	1006	4.18	0.92	1.00	5.00	4.00	4.00	5.00
PC	1004	3.14	1.32	1.00	5.00	2.00	3.00	4.00
Economy Management	1005	2.64	1.35	1.00	5.00	1.00	2.00	4.00
Biology	1004	2.67	1.38	1.00	5.00	2.00	2.00	4.00
Chemistry	1000	2.17	1.38	1.00	5.00	1.00	2.00	3.00
Reading	1004	3.16	1.50	1.00	5.00	2.00	3.00	5.00
Geography	1001	3.08	1.28	1.00	5.00	2.00	3.00	4.00
Foreign Languages	1005	3.78	1.14	1.00	5.00	3.00	4.00	5.00
Medicine	1005	2.52	1.38	1.00	5.00	1.00	2.00	3.00
Law	1009	2.26	1.24	1.00	5.00	1.00	2.00	3.00
Cars	1006	2.69	1.44	1.00	5.00	1.00	3.00	4.00
Art exhibitions	1004	2.59	1.32	1.00	5.00	1.00	2.00	4.00
Religion	1007	2.27	1.32	1.00	5.00	1.00	2.00	3.00
Countryside, outdoors	1003	3.69	1.20	1.00	5.00	3.00	4.00	5.00
Dancing	1007	2.46	1.45	1.00	5.00	1.00	2.00	4.00
Musical instruments	1009	2.32	1.51	1.00	5.00	1.00	2.00	4.00
Writing	1004	1.90	1.29	1.00	5.00	1.00	1.00	3.00
Passive sport	**995**	3.39	1.41	1.00	5.00	2.00	3.00	5.00
Active sport	1006	3.29	1.50	1.00	5.00	2.00	3.00	5.00
Gardening	1003	1.91	1.18	1.00	5.00	1.00	1.00	3.00
Celebrities	1008	2.36	1.27	1.00	5.00	1.00	2.00	3.00
Shopping	1008	3.28	1.29	1.00	5.00	2.00	3.00	4.00
Science and technology	1004	3.23	1.28	1.00	5.00	2.00	3.00	4.00
Theatre	1002	3.02	1.33	1.00	5.00	2.00	3.00	4.00

Fun with friends	1006	4.56	0.74	**2.00**	5.00	4.00	5.00	5.00
Adrenaline sports	1007	2.95	1.42	1.00	5.00	2.00	3.00	4.00
Pets	1006	3.33	**1.55**	1.00	5.00	2.00	4.00	5.00
Phobias								
Flying	1007	2.06	1.21	1.00	5.00	1.00	2.00	3.00
Storm	1009	1.97	1.16	1.00	5.00	1.00	2.00	3.00
Darkness	1008	2.25	1.25	1.00	5.00	1.00	2.00	3.00
Heights	1007	2.62	1.30	1.00	5.00	2.00	2.00	4.00
Spiders	1005	2.83	1.54	1.00	5.00	1.00	3.00	4.00
Snakes	1010	3.03	1.50	1.00	5.00	2.00	3.00	4.00
Rats	1007	2.41	1.40	1.00	5.00	1.00	2.00	3.00
Ageing	1009	2.58	1.39	1.00	5.00	1.00	2.00	4.00
Dangerous dogs	1009	3.04	1.37	1.00	5.00	2.00	3.00	4.00
Fear of public speaking	1009	2.80	1.21	1.00	5.00	2.00	3.00	4.00
Health habits								
Healthy eating	1007	3.03	0.94	1.00	5.00	3.00	3.00	4.00
Personality traits, views on life & opinions								
Daily events	1003	3.07	1.12	1.00	5.00	2.00	3.00	4.00
Prioritising workload	1005	2.65	1.22	1.00	5.00	2.00	3.00	3.00
Writing notes	1007	3.08	1.41	1.00	5.00	2.00	3.00	4.00
Workaholism	1005	3.00	1.28	1.00	5.00	2.00	3.00	4.00
Thinking ahead	1007	3.41	1.14	1.00	5.00	3.00	3.00	4.00
Final judgement	1003	2.65	1.38	1.00	5.00	1.00	3.00	4.00
Reliability	1006	3.86	0.93	1.00	5.00	3.00	4.00	5.00
Keeping promises	1009	3.99	0.90	1.00	5.00	3.00	4.00	5.00
Loss of interest	1006	2.71	1.35	1.00	5.00	2.00	3.00	4.00
Friends versus money	1004	3.78	1.12	1.00	5.00	3.00	4.00	5.00
Funniness	1006	3.29	1.13	1.00	5.00	3.00	3.00	4.00
Fake	1009	2.13	1.05	1.00	5.00	1.00	2.00	3.00
Criminal damage	1003	2.6	1.5	1.0	5.0	1.0	2.0	4.0
Decision making	1006	3.2	1.2	1.0	5.0	2.0	3.0	4.0
Elections	1007	3.42	1.57	1.00	5.00	2.00	4.00	5.00

Self-criticism	1005	3.58	1.19	1.00	5.00	3.00	4.00	5.00
Judgement calls	1006	3.99	0.97	1.00	5.00	3.00	4.00	5.00
Hypochondria	1006	1.91	1.16	1.00	5.00	1.00	1.00	3.00
Empathy	1005	3.86	1.13	1.00	5.00	3.00	4.00	5.00
Eating to survive	1010	2.23	1.21	1.00	5.00	1.00	2.00	3.00
Giving	1004	2.98	1.31	1.00	5.00	2.00	3.00	4.00
Compassion to animals	1003	3.97	1.19	1.00	5.00	3.00	4.00	5.00
Borrowed stuff	1008	4.02	1.05	1.00	5.00	3.00	4.00	5.00
Loneliness	1009	2.89	1.13	1.00	5.00	2.00	3.00	4.00
Cheating in school	1006	3.74	1.25	1.00	5.00	3.00	4.00	5.00
Health	1009	3.25	1.08	1.00	5.00	3.00	3.00	4.00
Changing the past	1008	2.95	1.28	1.00	5.00	2.00	3.00	4.00
God	1008	3.30	1.48	1.00	5.00	2.00	3.00	5.00
Dreams	1010	3.30	0.68	1.00	5.00	3.00	3.00	4.00
Charity	1007	2.10	1.03	1.00	5.00	1.00	2.00	3.00
Number of friends	1010	3.34	1.06	1.00	5.00	3.00	3.00	4.00
Waiting	1007	2.67	1.00	1.00	5.00	2.00	3.00	3.00
New environment	1008	3.48	1.15	1.00	5.00	3.00	4.00	4.00
Mood swings	1006	3.26	1.04	1.00	5.00	3.00	3.00	4.00
Appearance and gestures	1007	3.60	0.94	1.00	5.00	3.00	4.00	4.00
Socializing	1005	3.16	1.09	1.00	5.00	2.00	3.00	4.00
Achievements	1008	2.96	0.94	1.00	5.00	2.00	3.00	4.00
Responding to a serious letter	1004	3.07	1.17	1.00	5.00	2.00	3.00	4.00
Children	1006	3.62	1.12	1.00	5.00	3.00	4.00	5.00
Assertiveness	1008	3.52	1.10	1.00	5.00	3.00	4.00	4.00
Getting angry	1006	3.01	1.17	1.00	5.00	2.00	3.00	4.00
Knowing the right people	1008	3.49	1.09	1.00	5.00	3.00	4.00	4.00
Public speaking	1008	3.52	1.27	1.00	5.00	3.00	4.00	5.00
Unpopularity	1007	3.46	1.12	1.00	5.00	3.00	3.00	4.00
Life struggles	1007	3.03	1.37	1.00	5.00	2.00	3.00	4.00
Happiness in life	1006	3.71	0.82	1.00	5.00	3.00	4.00	4.00
Energy levels	1005	3.63	1.00	1.00	5.00	3.00	4.00	4.00
Small – big dogs	1006	2.97	1.22	1.00	5.00	2.00	3.00	4.00

Personality	1006	3.29	0.64	1.00	5.00	3.00	3.00	4.00
Finding lost valuables	1006	2.87	1.24	1.00	5.00	2.00	3.00	4.00
Getting up	1005	3.59	1.31	1.00	5.00	3.00	4.00	5.00
Interests or hobbies	1007	3.55	1.17	1.00	5.00	3.00	4.00	5.00
Parent’s advice	1008	3.27	0.87	1.00	5.00	3.00	3.00	4.00
Questionnaires or polls	1006	2.75	1.10	1.00	5.00	2.00	3.00	3.00
Spending habits								
Finances	1007	3.02	1.14	1.00	5.00	2.00	3.00	4.00
Shopping centres	1008	3.23	1.32	1.00	5.00	2.00	3.00	4.00
Branded clothing	1008	3.05	1.31	1.00	5.00	2.00	3.00	4.00
Entertainment spending	1007	3.20	1.19	1.00	5.00	2.00	3.00	4.00
Spending on looks	1007	3.11	1.21	1.00	5.00	2.00	3.00	4.00
Spending on gadgets	1010	2.87	1.28	1.00	5.00	2.00	3.00	4.00
Spending on healthy eating	1008	3.56	1.09	1.00	5.00	3.00	4.00	4.00
Demographics								
Age	1003	20.43	2.83	15.0	30.0	19.0	20.0	22.0
Height	990	173.5	10.02	62.0	203	167	173	180
Weight	990	66.41	13.84	41.0	165	55.0	64.0	75.0
Number of siblings	1004	1.30	1.01	0	10.0	1	1	2

B: Distribution Categorical Variables

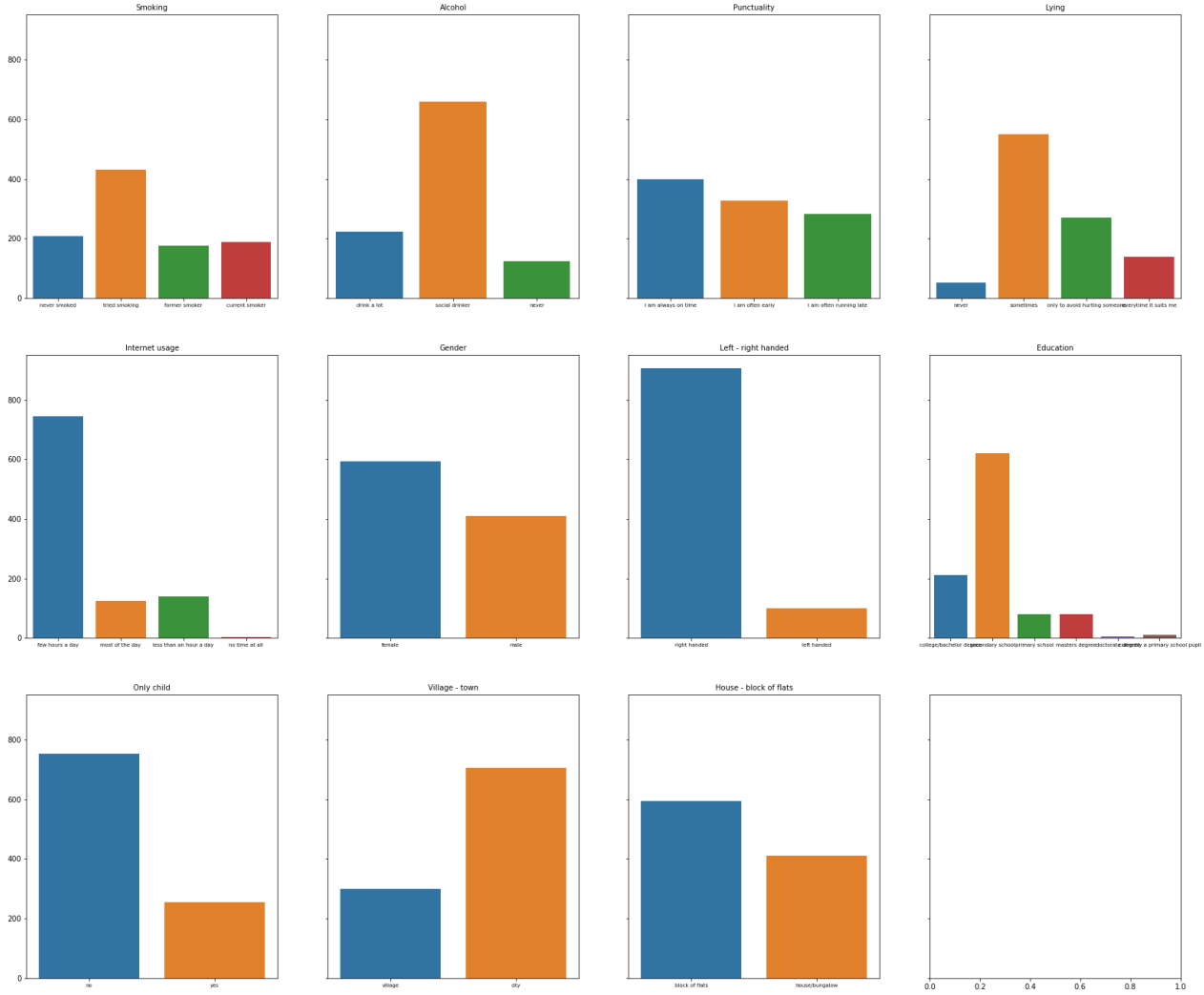

Figure 11 Distribution Categorical Variables

C: Feature Importance Plot Entropy

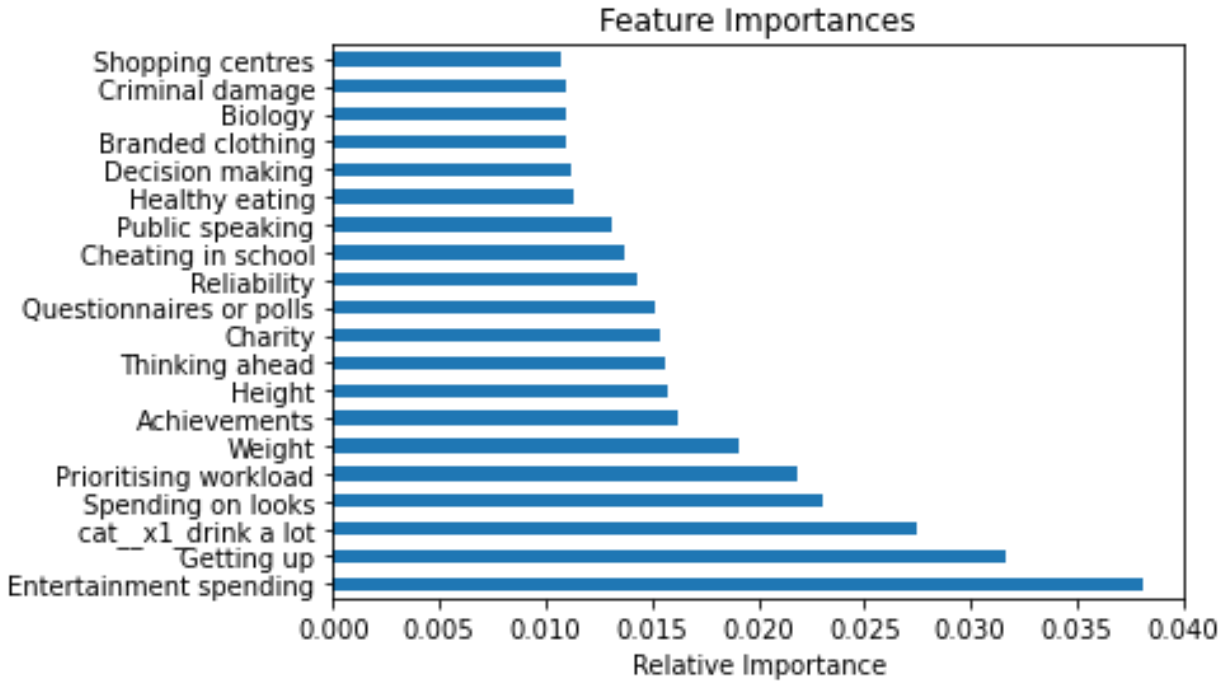

Figure 12 Random Forest: Entropy

D: Ordinal Logistic Regression

Table 11 Summary OLR Model

	Dependent variable:
	Finances
Music	0.069
	p = 0.525
Slow.songs.or.fast.songs	-0.071
	p = 0.415
Dance	0.002
	p = 0.981
Folk	0.014
	p = 0.846
Country	0.073
	p = 0.325
Classical.music	0.009
	p = 0.907
Musical	-0.017
	p = 0.806
Pop	0.066
	p = 0.372
Rock	0.061
	p = 0.453
Metal.or.Hardrock	-0.065
	p = 0.353
Punk	-0.023
	p = 0.747
Hiphop..Rap	-0.068
	p = 0.293
Reggae..Ska	0.105
	p = 0.134
Swing..Jazz	-0.059
	p = 0.416
Rock.n.roll	-0.111
	p = 0.134
Alternative	-0.041
	p = 0.526
Latino	0.070
	p = 0.296
Techno..Trance	-0.008
	p = 0.898
Opera	0.050
	p = 0.533
Movies	0.041
	p = 0.716
Horror	0.005
	p = 0.930
Thriller	-0.011
	p = 0.874

	Dependent variable:
	Finances
Comedy	0.057
	p = 0.587
Romantic	-0.125
	p = 0.102
War	0.032
	p = 0.622
Fantasy.Fairy.tales	**-0.202****
	p = 0.016
Animated	0.151*
	p = 0.056
Documentary	-0.059
	p = 0.425
Western	0.102
	p = 0.172
History	-0.017
	p = 0.808
Psychology	0.083
	p = 0.200
Politics	-0.007
	p = 0.925
Internet	**0.180****
	p = 0.048
PC	0.020
	p = 0.779
Economy.Management	0.046
	p = 0.439
Biology	0.033
	p = 0.686
Chemistry	-0.020
	p = 0.789
Reading	0.048
	p = 0.425
Geography	0.081
	p = 0.173
Medicine	0.051
	p = 0.495
Law	-0.047
	p = 0.480
Cars	0.017
	p = 0.783
Art.exhibitions	**-0.154****
	p = 0.029
Religion	-0.010
	p = 0.877
Countryside..outdoors	0.021
	p = 0.757

	Dependent variable:
	Finances
Dancing	**-0.143**** **p = 0.020**
Musical.instruments	-0.042 p = 0.433
Writing	0.045 p = 0.493
Passive.sport	-0.006 p = 0.912
Active.sport	-0.017 p = 0.754
Gardening	0.047 p = 0.481
Shopping	-0.137 p = 0.101
Science.and.technology	0.015 p = 0.829
Theatre	0.109 p = 0.130
Fun.with.friends	-0.221* p = 0.051
Adrenaline.sports	0.110* p = 0.076
Pets	-0.037 p = 0.457
Flying	-0.062 p = 0.319
Darkness	0.105 p = 0.103
Heights	-0.047 p = 0.431
Spiders	-0.073 p = 0.176
Snakes	0.025 p = 0.659
Ageing	0.054 p = 0.335
Dangerous.dogs	-0.112* p = 0.069
Fear.of.public.speaking	-0.009 p = 0.897
Smokingformer smoker	0.084 p = 0.716
Smokingnever smoked	0.392 p = 0.106
Smokingtried smoking	0.200 p = 0.317

	Dependent variable:
	Finances
Healthy.eating	0.071 p = 0.421
Daily.events	0.017 p = 0.808
Prioritising.workload	**0.302***** **p = 0.00002**
Writing.notes	-0.060 p = 0.300
Workaholism	-0.095 p = 0.165
Thinking.ahead	0.099 p = 0.167
Final.judgement	0.103* p = 0.063
Reliability	0.027 p = 0.744
Friends.versus.money	0.039 p = 0.568
Funniness	-0.061 p = 0.360
Fake	-0.097 p = 0.196
Criminal.damage	**-0.102**** **p = 0.049**
Decision.making	**0.206***** **p = 0.002**
Elections	0.013 p = 0.795
Self.criticism	0.035 p = 0.585
Judgment.calls	-0.140* p = 0.072
Hypochondria	-0.034 p = 0.619
Empathy	0.080 p = 0.264
Eating.to.survive	0.113* p = 0.061
Compassion.to.animals	-0.113* p = 0.094
Borrowed.stuff	0.106 p = 0.144
Loneliness	0.034 p = 0.646
Cheating.in.school	0.044 p = 0.492

	Dependent variable:
	Finances
Health	0.115
	p = 0.142
Dreams	-0.160
	p = 0.129
Number.of.friends	-0.088
	p = 0.279
Punctualityi am often early	-0.058
	p = 0.728
Punctualityi am often running late	-0.199
	p = 0.255
Lyingnever	-0.246
	p = 0.512
Lyingonly to avoid hurting someone	-0.261
	p = 0.287
Lyingsometimes	-0.072
	p = 0.750
Waiting	0.073
	p = 0.325
New.environment	0.104
	p = 0.132
Mood.swings	-0.115
	p = 0.145
Appearence.and.gestures	-0.038
	p = 0.651
Socializing	-0.032
	p = 0.664
Achievements	0.032
	p = 0.687
Responding.to.a.serious.letter	0.071
	p = 0.244
Children	0.067
	p = 0.330
Assertiveness	0.058
	p = 0.388
Getting.angry	0.029
	p = 0.685
Knowing.the.right.people	-0.009
	p = 0.902
Public.speaking	0.094
	p = 0.166
Unpopularity	-0.064
	p = 0.333
Life.struggles	-0.009
	p = 0.898
Energy.levels	-0.071
	p = 0.433

	Dependent variable:
	Finances
Small...big.dogs	-0.117*
	p = 0.073
Personality	-0.063
	p = 0.607
Finding.lost.valuables	**0.138****
	p = 0.025
Getting.up	**-0.250*****
	p = 0.00002
Interests.or.hobbies	-0.026
	p = 0.726
Parents..advice	0.157*
	p = 0.080
Questionnaires.or.polls	**0.187*****
	p = 0.006
Internet.usageless than an hour a day	0.222
	p = 0.291
Internet.usagemost of the day	0.002
	p = 0.992
Internet.usageno time at all	0.047
	p = 0.967
Shopping.centres	0.048
	p = 0.511
Spending.on.looks	**-0.227*****
	p = 0.005
Spending.on.gadgets	-0.038
	p = 0.578
Spending.on.healthy.eating	0.030
	p = 0.677
Age	-0.002
	p = 0.959
Weight	**-0.014****
	p = 0.047
Number.of.siblings	-0.082
	p = 0.320
Gendermale	-0.267
	p = 0.308
Left...right.handedright handed	0.038
	p = 0.871
Educationcurrently a primary school pupil	0.474
	p = 0.518
Educationdoctorate degree	**2.201****
	p = 0.046
Educationmasters degree	0.006
	p = 0.985
Educationprimary school	-0.267
	p = 0.405

	Dependent variable:
	Finances
Educationsecondary school	0.175
	p = 0.320
Only.childyes	**0.379****
	p = 0.035
Village...townvillage	0.088
	p = 0.641
House...block.of.flatshouse/bungalow	0.131
	p = 0.454
AIC	2143.3
Observations	1,010
Note:	*p<0.1; **p<0.05; ***p<0.01

E: Deep Neural Networks

1. TabNet

As the name suggests, TabNet is a deep learning architecture that is made for tabular data. It was invented by Arik & Pfister in 2019. It 'uses sequential attention to choose which features to reason from at each decision step, enabling interpretability and more efficient learning as the learning capacity is used for the most salient features'. Instead of using trees, a learnable mask on the input features is used. This allows for soft decisions, i.e. decisions can be made on a range of values instead of a single threshold. The encoder can be seen below in Figure 13. It consists of sequential decision steps that both encode and select features. In more detail, the encoder consists of feature transformers, which pre-processes the data, and an attentive transformers, where the learning happens. Prior information is backpropagated to learn and control how much a feature has already been used.

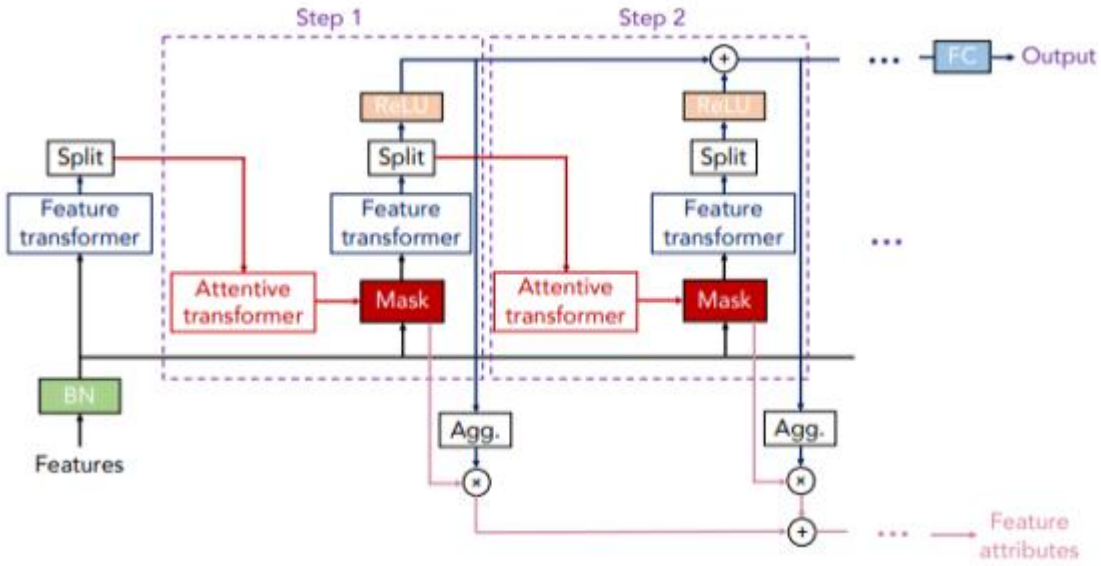

Figure 13 Encoder Architecture (Arik & Pfister, 2019)

A very useful addition to this architecture is the unsupervised pre-training, which first transforms the TabNet architecture to an encoder-decoder structure. It then deletes some cells in the tabular data, which the model then needs to predict. This model can then be used as a pre-trained model in the regular TabNet architecture, leading to the latter having more information and thus performing better.

Implementing the model is also rather straightforward now, with packages available that work with both TensorFlow and PyTorch.

2. Neural Oblivious Decision Ensembles

Neural Oblivious Decision Ensembles (NODE) are closer related to tree-based algorithms, as they are an ensemble of oblivious decision trees that all have the same depth. However, these trees are differentiable to enable backpropagation. Oblivious means that the tree is constrained to use the same splitting feature(s) and splitting threshold(s) in all internal nodes of the same depth. This makes them weaker compared to unconstrained trees, but also less prone to overfitting.

Multiple NODE layers are stacked on each other, where each layer takes both the original feature input and the preceding NODE layers as input. Similar to TabNet, the NODE layer will choose a set of features, with a range of values, to split at each level. It also has both a PyTorch and a TensorFlow implementation. The paper of Popov et al. (2020) suggests that it is able to outperform XGB on multiple datasets.

3. TabTransformer

The next modelling architecture is TabTransformer, which is proposed by Huang et al. (2020). The TabTransformer architecture comprises a column embedding layer, a stack of Transformer layers, and a multilayer perceptron. Column embedding is used to change the embeddings of categorical features into robust contextual embeddings for the sake of higher predicting power. Each Transformer layer consists of a self-attention layer followed by a position-wise feed-forward layer, with element-wise addition and layer-normalization being done after each layer. When there are only a few labelled examples and a large number of unlabelled examples, a pre-training procedure is also available to train the Transformer layers using unlabelled data.

4. Self-Attention and Intersample Attention Transformer

A final deep tabular data modelling architecture is the SAINT algorithm by Somepalli et al. (2021), which has only been proposed very recently. It projects all features, both continuous and categorical, into a combined dense vector space. These values are used as tokens into a transformer encoder. This encoder uses both self-attention (attends to individual features within each observation) and intersample attention (enhances classification of an observation by relating it to other observations). SAINT is then composed of a stack of identical stages, where each stage consists of one such encoder block.

Furthermore, SAINT also provides the option for self-supervised pe-training (contrastive learning to be more precise). Yet again, implementing SAINT is possible via PyTorch.

GHENT UNIVERSITY

FACULTY OF SCIENCES

USING MACHINE LEARNING TECHNIQUES FOR ANALYZING SURVEY DATA

Bastjaan Beernaert

Master dissertation submitted to
obtain the degree of
Master of Statistical Data Analysis

Promotor: dr. Koen Plevoets

Academic year 2020 – 2021