ERASMUS UNIVERSITEIT ROTTERDAM
ERASMUS SCHOOL OF ECONOMICS

MASTER THESIS BUSINESS ANALYTICS AND QUANTITATIVE MARKETING

Robustness and Unbalanced Data in Machine Learning: Explainable Tree-Based Ensemble Algorithms Applied in a Marketing Context

Name student:	Melanie Suijkerbuijk
Student ID number:	427460
Supervisor:	Prof. Dr. Ş. İlker Birbil
Second assessor:	Dr. M. Hakan Akyüz

July 7, 2021

Abstract

Machine Learning (ML) methods have gained popularity in recent years. One important drawback is the lack of an extensive understanding of these models. As a result, a glass-box tree-based ensemble algorithm referred to as Explainable Boosting Machines (EBM) has been developed by Nori et al. (2019). In this study we introduce another glass-box method called Explainable Random Forests (ERF), which provides an interpretation similar to EBMs. These methods are applied in a marketing context and compared to black-box tree-based ensemble methods. In this application the explainable ML methods have similar performances to the black-box models. ERF even outperforms these models in terms of balanced accuracy. Furthermore, we study the performance of the explainable ML methods on extremely unbalanced data. More specifically, we analyse the performance of three strategies to deal with unbalanced data which have proven to be effective in black-box tree-based models. Finally, we examine the robustness of the interpretations provided by the explainable ML methods in terms of the variable importance and feature functions.

Contents

1 Introduction

In the last recent years Machine Learning (ML) algorithms have gained popularity due to their strong predictive powers. These ML methods often outperform classical econometric models in terms of prediction, as econometric models often rely on the assumption of linearity. However, in general, a common shortcoming to most ML algorithms is that these methods are considered black-box and very difficult, or even impossible, to fully understand. Therefore, the predictive power of ML methods and interpretability have been considered opposites. Although current methods exist to analyse the feature importance of black-box models, this does not allow for an extensive interpretation. An extensive understanding of these models is important for purposes such as model debugging, sense checking, or answering questions of clients. Furthermore, ML methods have, in recent years, been applied to high-risk applications such as healthcare (Caruana et al., 2015) and risk modelling (Gillingham, 2016; Papouskova & Hajek, 2019). In these applications interpretability is of crucial importance.

As a result of the popularity of ML methods, promising developments have been made towards interpretation regarding ML methods. One of the developments in the field regarding this issue, is the introduction of Explainable Boosting Machines (EBMs) by Nori et al. (2019), which can be used for both regression and classification purposes. EBM is a glass-box model is designed for interpretability, which allows a thorough understanding of the model. Nori et al. (2019) have shown that this model still provides the same strong predictive power as many black-box models. This research will introduce a new explainable algorithm based on tree-based ensemble methods. This explainable algorithm will be referred to as ERF. This algorithm allows for the same extensive interpretation as EBMs, developed by Nori et al. (2019). In contrast to EBMs, this new algorithm trains classification trees for classification problems, instead of regression trees which are required for the boosting approach in EBM. As these glass-box models allow for an extensive understanding and interpretation, new research questions arise. This research will focus on the following four main questions: How good do the interpretable models perform in contrast to black-box ML models, when applied to this marketing application? Can we get a good performance with explainable ML methods when the dependent variable is slightly or even extremely unbalanced? Will this algorithm provide a similar interpretation of the model when the training data set contains noise or slightly different variables? How do we measure the sensitivity of these feature functions? To answer these questions, we focus on the interpretation provided by the EBM algorithm, applied to an imputation process in a marketing context.

Application to marketing. One of the most common challenges in marketing, is to get the right message to the right consumer at the right time. In recent years, more data is gathered to provide data-driven marketing solutions, to answer these questions. However, with the increase in data and data sources, it can be a challenge to find a common 'sense' and use different data sources, which are not always consistent, to provide insights. Nielsen Media Impact (NMI) is one of the tools to provide these insights, by providing a personalised media planning solution. This enables decision making about where and when to engage audiences with content and ad-

vertising. NMI provides insights for different channels and devices such as; TV, video streaming, magazines on devices such as desktop, mobile and tablets. This research will focus on the imputation process that is used to gain insights for video streaming, on both desktop and mobile level. More specifically, this research will focus on the modelling step in the imputation process. This process comes with of a variety of data sets extremely suitable for answering the research questions raised above. This modelling step of the imputation process uses data regarding consumer behaviour which is collected using two different approaches. The first data source is Digital Content Ratings (DCR) which measures and reports content usage at the level of age/gender demographic audiences, for the individuals subscribed to this measurement. Second, panels are used to capture the panellists online activity and application usage. Both of these data sources have their own advantages and limitations. The DCR data contains tagged content on a more general and less granular level, however this data set is known to be a good reflection of the total target population. On the other hand, the panel data contains more granular information regarding the individuals, however the insights gained from this data can be biased as the size of these panels is not sufficient. Therefore, the insights gained from the panel data is often not consistent with the DCR data. In order to maximise the use of the panel data, tree-based imputation is used to merge these data sets. The imputation of DCR-tagged video consists of two main steps. The first is the modelling step, in which the probability to be reached by a brand/sub-brand is predicted for each panellist. The second step is to use these predictions, together with the DCR data, to obtain the duration by brand/sub-brand per respondent. This research will focus on the modelling step in the imputation approach, with the goal to predict the probability for each respondent to be reached by a combination of device and (sub-)brand. In this research we will refer to these different combinations of (sub-)brand and devices as different applications or different cases of the problem. In order to predict this reach value, a model will be created to capture the relationships for each case. The natural hierarchical structure regarding this research problem is shown in Figure 1. As the objective of this application is to predict a value, this problem is suitable for ML methods. For this application an extensive understanding of the model is required for sense checking purposes, such that the imputation is based on relations that make sense. Especially since the imputed data is used to provide further marketing insights. Not only it is more safe to completely understand a model, but this understanding is crucial when it comes to answering questions of clients. Furthermore, for some of the cases the dependent variable is extremely unbalanced. However, these (sub-)brands still require the marketing insights obtained by the models.

The interpretation of ML methods is of great importance for a wide variety of applications. Therefore, a complete understanding and assessment of the robustness of glass-box ML methods is of crucial importance. Furthermore, the additional challenge of imbalanced data sets can occur in a variety of applications such as fraud detection, churn prediction or rare events such as flood events, or financial crisis. As a result, this research will not only be relevant for this particular tree-based imputation in the marketing context, but for other prediction applications with imbalanced data as well.

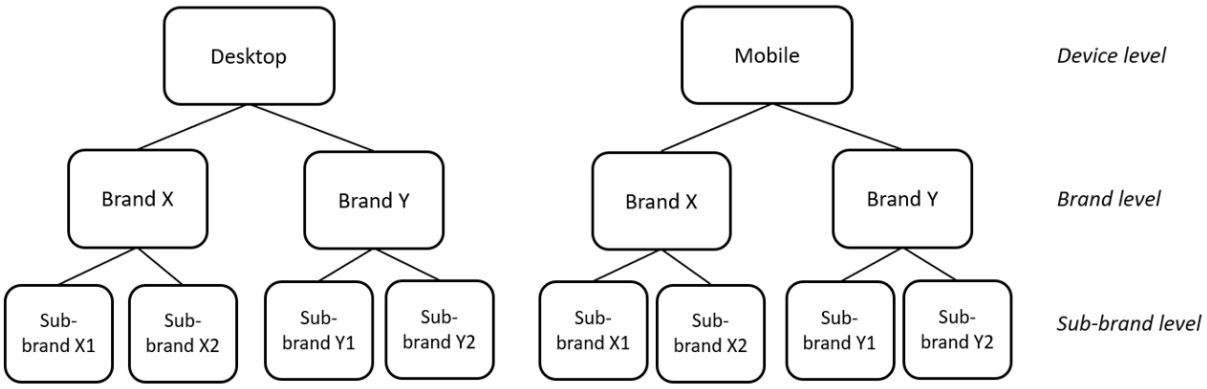


Figure 1: The natural hierarchical structure of the research problem for two brands with each two sub-brands. This research consists of more than two brands with a variety number of sub-brands. Furthermore, not all (sub-)brands necessarily exist for both desktop and mobile.

The remainder of this research is structured as follows; Section 2 starts off with a literature review which provides an overview of the relevant researches. Afterwards, Section 3 elaborates on the data used in this research. This includes an in-depth description of the data, the pre-processing steps which were applied, and an exploratory analysis. Next, Section 4 provides an elaborate description on the methods used to answer the research questions, including the introduction of the new algorithm ERF. This is followed by a presentation of the results in Section 5. Finally, based on these results, Section 6 provides a conclusion and discussion on this research, together with suggestions for further research.

2 Literature Review

This section will provide an overview of the existing literature for the problem described in Section 1. The existing research regarding interpretation and tackling imbalanced data sets with ML methods have developed separately. Therefore, we will start with a general overview of the machine learning methods in Section 2.1, after which the literature relating to the interpretability and imbalance topics separate from each other is discussed in Sections 2.2 and 2.3 respectively. This is followed by an overview of the existing literature on robustness in machine learning methods is discussed in Section 2.4. Furthermore, this section covers the relevance for this study, combined with how this research can be placed in the existing literature in Section 2.5.

2.1 Machine Learning Methods

As mentioned in Section 1, ML methods are popular for prediction as they often have a stronger prediction power in comparison to econometric models. One popular subarea of ML is based on decision trees. The two most common used approaches to build standard regression or classification trees are CART, introduced by Breiman et al. (1984) and C4.5 introduced by Quinlan (1986). The most important difference between these two methods is that CART constructs the tree on a numerical splitting criterion recursively, while C4.5 uses entropy based criteria. A single decision tree is very straightforward to interpret and explain, especially when the trees are not deep. However, the performance of a single decision tree, in terms of accuracy, is in general not as powerful as the performance of more complex ensemble models. Besides, single trees have higher variances than these ensemble models.

In order to enhance the prediction accuracy, various algorithms have been proposed in which decision trees are combined, such as random forests (RF), proposed by Breiman (2001). This algorithm uses bagging to grow separate deep trees. The prediction from random forests is based on majority voting for classification purposes, or average predictions for regression. In recent years, research has been done to improve the explainability of the random forest classifiers, such as the research by Neto & Paulovich (2020) and Petkovic et al. (2018). However, these models are still quite complex to interpret due to the large number of decision trees trained in this algorithm.

Another approach to combine trees for an enhanced prediction, is to apply boosting methods. These boosting methods combine weak classifiers to boost the collective performance. Two popular boosting methods are Ada-Boost, introduced by Freund & Schapire (1997) and Gradient Boosted Decision Trees (GBDT), introduced by Friedman (2001). Due to the sequential behaviour of these boosting algorithms, this does not allow for a parallel implementation. Therefore, these methods can be computationally intensive when dealing with large data sets. As a result Ke et al. (2017) proposed an algorithm called LightGBM, which grows trees vertically. This results in an algorithm with a higher speed, that can handle large sizes of data and requires less memory. One drawback of this algorithm is that it is more sensitive to overfitting, especially for smaller sized data.

2.2 Interpretable Machine Learning Methods

As mentioned above, the assumption of linearity in econometric models, leads to less accurate predictions than ML methods. However, due to the non-linearity, ML methods are often more difficult to interpret and explain. However, as interpretability is important in various applications, research has developed in recent years towards this aspect.

Some of these interpretable glass-box models are based on Generalised Additive Models (GAMs), developed by Hastie & Tibshirani (1986). In GAMs linearity is replaced by a continuous function. As a result these models in general have a higher accuracy than linear models. Due to the additive nature of these models, they are still transparent and easy to interpret. As a result GAMs have been applied to different research areas such as churn prediction (Coussement et al., 2010), healthcare (Caruana et al., 2015) and biology (Jowett et al., 2008). This method has been further developed by Lou et al. (2013), in which pairwise interactions are added to GAMs. This results in GAMs with Pairwise Interactions (GA^2Ms). When they compare the performance on ten different data sets, the GA^2Ms outperforms the GAMs. From these ten data sets, five are binary classification problems, with fairly balanced data as the amount of positives is between 39% and 65%. These interactions could be interpreted with the use of heat-maps. However, this complicates the interpretation of a model, and is therefore only preferred when the accuracy significantly improves.

As discussed in Lou et al. (2012) different approaches exist to obtain the non-linear functions for the features, such as using splines (Caruana et al., 2015), back-fitting (Baquero et al., 2018), (Potts, 1999; Vaughan et al., 2018) and gradient boosting. All of these functions can be visualised using graphs, and can therefore be considered interpretable. However, often these methods in general do not result in a high prediction power. Nori et al. (2019) introduces Explainable Boosting Machines (EBMs), an implementation of GA^2Ms with ML techniques such as bagging and boosting to obtain the feature functions. A similar approach is the use of Neural Additive Methods (NAM), introduced by Agarwal et al. (2020), in which the feature functions for GAMs are obtained by training Neural Networks (NN).

Besides the introduction of these glass-box models, methods have been proposed to enhance the interpretation of existing black-box models and their predictions. SHAP and LIME are two of these methods, introduced by (Lundberg & Lee, 2017) and (Ribeiro et al., 2016) respectively. These models tweak the input and measures the effect in prediction, and can therefore be applied to any complex model. Another promising development is the introduction of an algorithms by (Akyüz & Birbil, 2021), which exploit linear programming to extract a set of classification rules, which are considered to be interpretable. Furthermore, the algorithm can take into account the preferences of decision makers regarding various characteristics of these rules such as the length and weights.

2.3 Unbalanced Data in Machine Learning Methods

Data sets in which one or multiple classes are extremely rare, often provide problems in supervised ML methods. In general, these supervised ML methods favour the larger classes in prediction, while the correct prediction of rare classes can be more important. Different methods exist to handle this challenge. The most common applied method is to counter this by using sampling methods. A systematic study about the class imbalance problem has been performed by Japkowicz & Stephen (2002). Their study covers the use of different ML algorithms such as decision tree models, NNs and Support Vector Machines (SVMs) for this imbalance issue. From their study we learn that the C5.0 standard approach for decision tree generating is the most sensitive to class imbalance. They showed that NNs are in general more sensitive to imbalanced data in comparison to algorithms based on decision trees. Furthermore, they showed that over-sampling is in general effective, while under-sampling the majority class is not.

One of the more intelligent methods for re-sampling is SMOTE, introduced by Chawla et al. (2002). However, even though re-sampling is applied, splitting criteria are still skewed sensitive (Flach, 2003). Although re-sampling is the most common approach to deal with unbalanced classes, this is not the only possible approach. A new measure, later referred to as the DKM measure, introduced by Dietterich et al. (1996) has shown to have an improved performance for imbalanced data sets (Flach, 2003; Drummond & Holte, 2000; Zadrozny & Elkan, 2001). However this is measure is still slightly skew sensitive. A skew insensitive measure is the Hellinger Distance proposed by Cieslak & Chawla (2008). Both the Hellinger Distance and the DKM measure outperform the C4.5 and CART, as discussed above.

Next to re-sampling and splitting criteria in decision tree algorithms, other ML approaches exist to deal with unbalanced data sets. Sigrist & Hirnschall (2019) introduced a Grabit model in which gradient tree boosting is applied to a Tobit model. This model has the advantages that it learns nonlinear relations and that missing values can automatically be accommodated instead of being imputed. This method is applied to a case study on predicting loans. In this case study the Grabit model outperforms logit, classification trees, random forests, tree-boosted logit and tree-boosted multinomial logit. Another approach is a boosting assisted zero-inflated Tweedie model (EMTboost) introduced by Zhou et al. (2020). Tweedie is a distribution family which have a positive mass at zero, but are otherwise continuous. This is a special case of exponential dispersion models, which are commonly used for Generalised Linear Models (GLM).

2.4 Robustness and Sensitivity

As ML methods gained popularity in recent years, the literature regarding these algorithms has increased and these algorithms have been studied in terms of performance. For ML methods two types of robustness exist: robustness across similar samples, and robustness to samples with adversarial noise. This research will focus on the robustness to adversarial noise. The robustness of several ML methods have been studied in the recent literature, such as; the robustness of SVM and KNN applied to classifying the emotions in speech in Shami & Verhelst (2007);

and the analysis of global sensitivity for tree-based ensemble methods such as random forests by Jaxa-Rozen & Kwakkel (2018).

In recent years, methods have been developed to asses the robustness of models. Liu et al. (2010) developed a decision tree algorithm which is robust specifically for imbalanced data sets. Rauber et al. (2017) developed a Python package to generate perturbations and quantify the performance of ML methods. Goodfellow et al. (2018) showed that although we have effective attack algorithms against adversarial inputs, few strong countermeasures exist. Furthermore, new robust estimators have been developed, such as the estimator introduced by Lecué et al. (2020), based on the Median of Means (MoM) estimator, which is robust to outliers.

2.5 Research Context

As shown in Sections 2.1, 2.2, 2.3 and 2.4, a substantial amount of research relevant to this research problem has been performed in the past. This research will contribute to this existing literature in the following manners;

1. This research introduces a new fully explainable machine learning method, referred to as Explainable Random Forest (ERF). ERFs use the strategy from EBMs to train trees based on a single feature, such that these trees can be combined to create feature functions for Generalised Additive Models (GAM). However, this algorithm will not apply boosting methods, but only use a bagging approach. Therefore, this algorithm is similar to random forests.

2. This research will examine performance of EBM and ERF compared to other methods, in terms of variable importance, accuracy and computation time.

3. This research will examine the performance of EBM on unbalanced data. As shown in Section 2.3, a substantial amount of research has been performed to develop ML algorithms which can handle imbalanced data. This research will compare three existing strategies on EBMs: oversampling, weighted splitting criteria, and weighted loss functions. These strategies have proven to be useful for other existing boosted tree-based methodologies.

4. This research will perform a robustness analysis of EBM and ERF, which focuses on the sensitivity of the interpretation (feature functions) given by these models. Furthermore, the explanations of both of these models will be compared.

3 Data

This section provides a brief description of the data used for this research. First, the data sources which are used in this research are described in Section 3.1. Then, Section 3.2 presents the data pre-processing, which includes feature adjustments and the merging process of the different data sources. This is then followed by a description of the outlier detection in Section 3.3. The steps in Sections 3.2 and 3.3 result in the final data used in the remainder of this research. Finally, Section 3.4 provides an exploratory data analysis of this final data, which provides initial insights of the data and unbalanced dependent variable.

3.1 Data Sources

This research uses two types of data sources, which will be described below. The first data source contains yearly Digital Content Rating (DCR) data from 2020. The second source contains monthly panel data of February, March and April 2020. The panel data for this research is collected from two different sources: the mobile panel data (EMM), and desktop video panel data (Stream data). Both of these sources contain monthly data on respondent level. The EMM data captures the online activity and app usage of panellists, while the streaming data contains the actual video activity behaviour of respondents. In order to train on a sufficient amount of data, the panel data of three months is used for training purposes. Thus, the data corresponding to the predictions of April 2020 is trained on panel data from February, March and April.

Nielsen Digital Content Ratings (DCR). The DCR data contains usage at the level of $age \times gender$ demographic audiences for the set subscribed to this measurement. From this data set monthly totals of unique audiences and duration can be extracted on three device types; mobile, desktop and all devices, and on brand and sub-brand level. This data is used as the census. The target audience contains of individuals which are 18+. Therefore, only individuals with at least the age of 18 are included. For the year of 2020 this contains 2,300 observations for mobile, 2,560 observations for desktop, and 2,560 observations for all devices. As this is aggregated data on gender and age-group, this contains information for all (relevant) age-gender groups for all the (sub-)brands. Therefore, this data contains information regarding the entire population (the US).

Mobile Panel Data (EMM). The panel data from this source is gathered both on brand and sub-brand level from mobile devices. For April 2020 33 brands with a total of 83 sub-brands are included. As March and February are only included for training purposes, but not for the actual prediction, only the (sub-)brands corresponding to April 2020 are included. For each of these (sub-)brands the data contains 29 variables, of which 18 are numerical variables, two binary variable, and nine categorical variables. These variables contain information on the characteristics of respondents and their browsing behaviour. The complete list of features with the corresponding description can be found in Table 6 in Appendix A.1. The training data-set contains information of 22,448 unique panellists with a total of 2,358,861 observations as not every panellist is in the panel for all three months and all (sub-)brands.

Category	Value	Viewing behaviour
Obsessive viewer	3	In the top 25%
Frequent viewer	2	In the top 25% to 50%
Regular viewer	1	In the top 50% to 75%
Rare viewer	0	In the bottom 25%

Table 1: The rulings for total viewing categories based on the viewing behaviour

Desktop Video Panel Data (Stream data). The panel data from this source is gathered both on brand and sub-brand level from desktop devices. For April 2020, 35 brands with a total of 94 sub-brands are included. As March and February are only included for training purposes, but not for the actual prediction, only the (sub-)brands corresponding to April 2020 are included. For each of these (sub-)brands the data contains twenty-seven variables, of which sixteen are numerical variables, three binary variables, and eight categorical variables. These variables contain information on the characteristics of respondents and their video streaming behaviour. The complete list of features with the corresponding description can be found in Table 6 in Appendix A.1. The training data-set contains information of 29,165 unique panellists with a total of 3,558,707 observations as not every panellist is in the panel for all three months and all (sub-)brands.

3.2 Data Pre-processing

In order to use this data for the reach probability prediction, the data needs to be pre-processed. This pre-processing consists of two steps. First, the adjustments to the data will be described, which is followed by the merging of the various data sets. The result of this data processing is used for the outlier detection, described in Section 3.3.

Across the various features in the data set some adjustments need to be made in order to process the data correctly. The following adjustments to the features are made:

- *age,* which provides the age of the respondent. The respondents with an age below 18 are removed, as the final target audience are adults with an age of at least 18 years old.
- *age groups,* instead of using the age of the respondent as a numerical value, age groups are created. These age buckets are as follows; 18-20, 21-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-64, 65+. The choice regarding these borders are based on the age groups in the DCR data as described in Section 3.1.
- *total viewing,* This variable is added to weigh the respondents based on their total viewing behaviour. A panellist gets assigned a category based on the total viewing behaviour as described in Table 1. If the individual spends relatively much time on video the total viewing value of the individual will be relatively high and vice versa.

The data sources as described in Section 3.1, contains various data sets which need to be merged in order to get the data prepared for the modelling. The data is matched based on the respondent id. The initial starting point for this research, is to collect data for all respondents, for which the reach, the dependent variable, is known. For all these respondents the socio-demographic variables, such as *age*, *gender* and other categorical variables are known. This results in a data set with the time spent for every individual, for all combinations of $brand \times subbrand \times device$. The data contains continuous and categorical variables. The continuous features are obtained from the EMM and Stream behaviour, while the categorical variables includes socio-demographic characteristics such as; *gender*, *education*, and *marital status*.

3.3 Outlier Detection

Next to the adjustments described in Section 3.2, some adjustments need to be made to deal with outliers in the data. In this research we use the definition of an outlier from Johnson et al. (2002), which states that an outlier is *'an observation in a data set which appears to be inconsistent with the remainder of that set of data.'* These outliers can occur for various reasons such as experimental errors or a variability in measurement. In this research we correct for both uni-variate outliers and correlation outliers.

In the EMM and Stream data, these outliers are defined as viewing behaviour with extremely rare properties. Furthermore, boosted tree methods are known to be sensitive to outliers, as each tree builds on the residuals from the previous tree (Li & Bradic, 2018). As this boosting mechanism is an important part of the EBM, it is expected that EBM is sensitive to these outliers. Therefore, these outliers need to be tackled. In this research we detect outliers for all (sub-)brands separate. Due to the size of the data sets, it is not feasible to combine the brands for the (correlation) outlier detection. Furthermore, there are no grounds to assume all the (sub-)brands have similar viewing behaviours. Therefore, the definition of an outlier can differ based on the (sub-)brand the observation is corresponding to. The split between mobile and desktop is preserved as these data sets have various explanatory variables.

The exploratory variables do not follow a normal distribution, as will be shown in Section 3.4. Therefore, common outlier detection methods, such as the Modified Z-score, are not suitable. As a result, this research deals with uni-variate outliers with an uni-variate outlier screening defined by Tukey (1960), which is suitable for highly skewed distributions. This approach uses an interquartile range (IQR) defined as

$$IQR = q_{75} - q_{25}, \tag{1}$$

where q_i is the ith quantile of the data. Next, this approach makes a distinction between possible and probable outliers. The possible outliers are defined with a so-called inner fence

$$inner\ fence = \Big[\ q_{25} - 1.5IQR\ ,\ q_{75} + 1.5IQR\ \Big]. \tag{2}$$

The probable outliers are defined using a 'outer fence' as

$$\textit{outer fence} = \Big[\ q_{25} - 3\ IQR\ ,\ q_{75} + 3\ IQR\ \Big]. \tag{3}$$

In this research we follow the recommendation of Tukey (1960), to only treat the probable outliers defined by the use of the outer fence.

Next to uni-variate outliers, multivariate data should be checked for correlation outliers. This research deals with correlation outliers using the Mahalanobis distance (MD). This distance measures allows for the correlation between variables and is therefore suited for multivariate outlier detection (De Maesschalck et al., 2000). This measure determines the distance between a point and a distribution:

$$d = \sqrt{(x - \hat{\mu}_x)C^{-1}(x - \hat{\mu}_x)^T} \tag{4}$$

where x represents an observation and μ the mean of the independent variables in the data; and C^{-1} is the inverse co-variance matrix of the independent variables in the data. In this research we use the cut-off point of 7.99 for the desktop data and 8.25 for mobile, which correspond to 0.01% significance levels. These cut-off points for mobile and desktop differs as the number of explanatory variables differ for these data sets, as explained in Section 3.1.

Next to the outlier detection methods, the socio-demographic data should be in line with the general knowledge of the population. Therefore, a sense check is performed on this data as well. This sense check includes for example a check on the age of the respondent. A respondent can not have an age over 117 as this is the age of the oldest human in the world. After pre-processing the data and removing the outliers, the final data contains of 2,358,861 and 3,338,539 observations for mobile and desktop, respectively. This corresponds to 97.8% and 93.8% of the total data.

3.4 Exploratory Analysis

This exploratory data analysis allows us to gain an understanding of the features, distribution of the dependent variable across the different (sub-)brands and possible relations between the features, (sub-)brands and devices. This exploratory analysis is performed on the pre-processed data, following the process as described in Section 3.2 and the outlier detection described in Section 3.3. Note that the data sets for mobile and desktop will be considered separately from each other as the data sets contain different features, even if the (sub-)brand is the same.

Mobile. This paragraph provides the exploratory analysis of the pre-processed mobile data. First, the distribution of the dependent variable is examined across the sub-brands. After which the summary statistics of the variables is investigated on an overall level. Finally, the correlation between the variables is examined on an overall level.

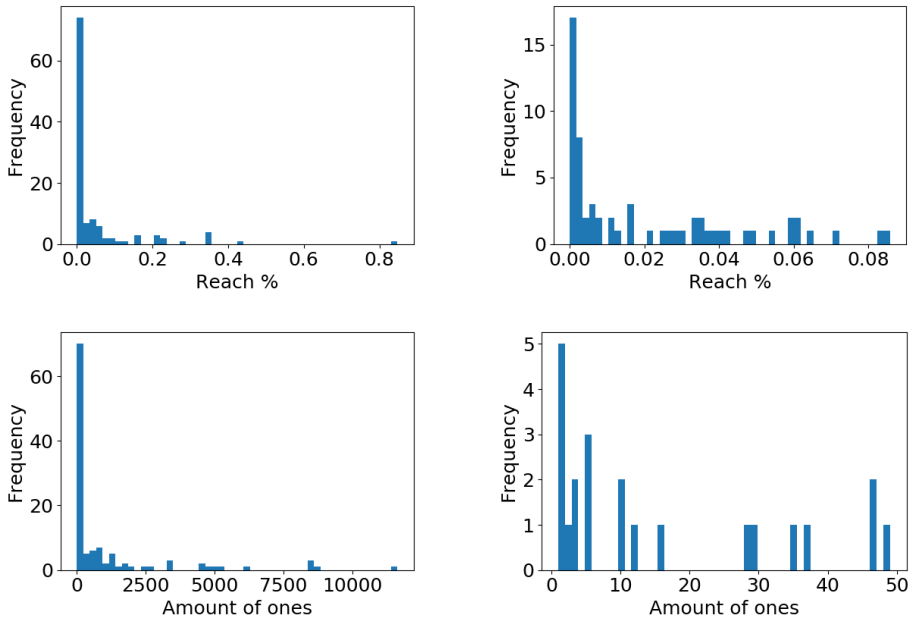


Figure 2: Distribution of the dependent variable on sub-brand level. The top two panels show the percentage of the minority value (1) for this variable, while the panels on the bottom show the amount of observations for this minority value (1). Furthermore, the panels on the left show the distribution of all the sub-brands, while the panels on the right provide the distribution of the highly unbalanced sub-brands.

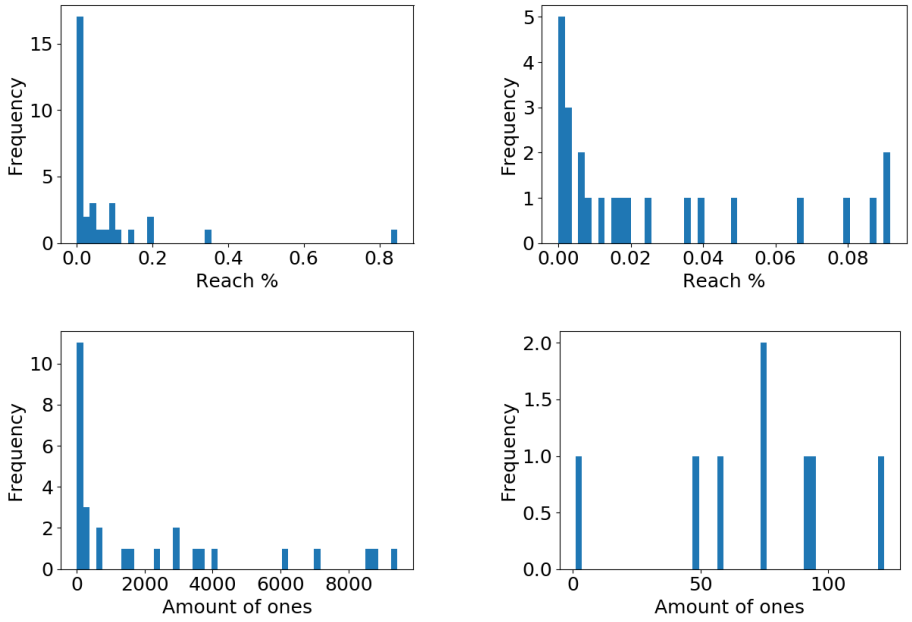


Figure 3: Distribution of the dependent variable on brand level. The top two panels show the percentage of the minority value (1) for this variable, while the panels on the bottom show the amount of observations for this minority value (1). Furthermore, the panels on the left show the distribution of all the brands, while the panels on the right provide the distribution of the highly unbalanced brands.

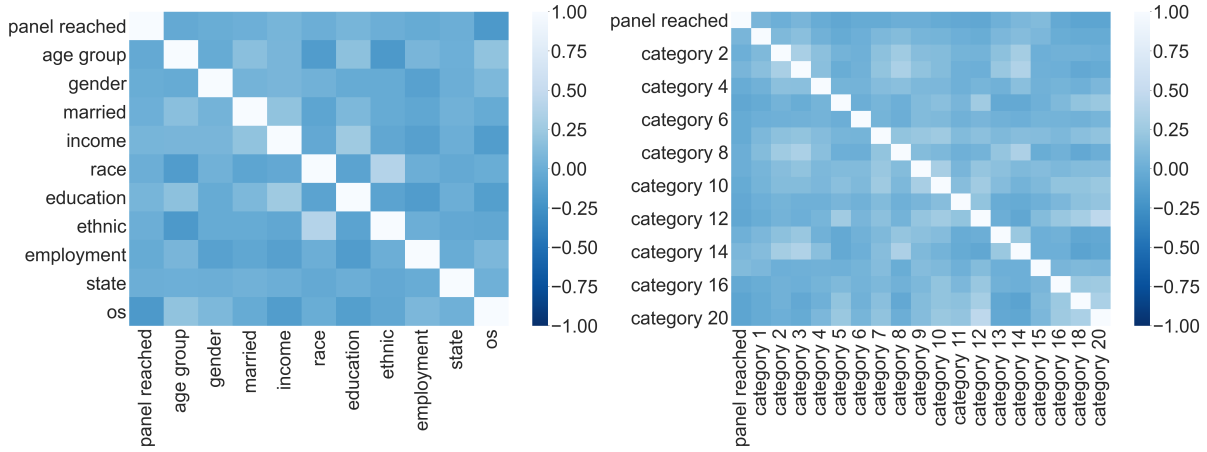


Figure 4: Correlation between the features and the dependent variable for the mobile data set. The left panel contains the correlations of the categorical variables, while the right panel displays the correlations of the continuous variables.

As described in Section 1, one of the challenges is the extremely skewed distribution of the dependent variable, *panel reached*, for some of the (sub-)brands. The distribution of the reach percentage and amount of observations with the minority variables across all sub-brands is shown in Figure 2. The panel on the top left shows that in general the dependent variable is very skewed. For most of the sub-brands the percentage of ones in the *panel reached* is less than twenty percent. The top right panel shows that quite some sub-brands have a reach percentage of below ten percent. The bottom panels, which show the corresponding amount of ones in the sub-brands data, confirm this. Furthermore, these panels even show that there are some sub-brands in which the entire data set has the *panel reached* variable set to zero. The distribution of the reach percentage and amount of observations with the minority variables across all brands is shown in Figure 3. In comparison to the reach distribution on sub-brand level, as shown in Figure 2, the data seems to be slightly less skewed on brand level. Furthermore, the bottom panels show that on brand level all the data sets contains observations with *panel reached* equal to one and zero.

The explanatory variables will be investigated more in depth. Due to the large amount of (sub-)brands, this will be done on an overall level for the mobile data set. The summary statistics for all variables are shown in Table 5 in Appendix A.2. One interesting insight is that quite some continuous variables have a large amount of zeros. These variables seem to follow an inflated zero distribution.

Next, we investigate the correlation between the variables to give further insights regarding the relationships across variables. The correlation between the continuous and categorical variables are all around zero. Therefore, the interesting correlations are shown in Figure 4. For the categorical features, shown in the left panel, none of the features have a significant correlation with the dependent variable. We observe that the variables *Ethnic* and *Spanish Language* have a strong negative correlation of -0.89. Although multicollinearity is not a concern for decision

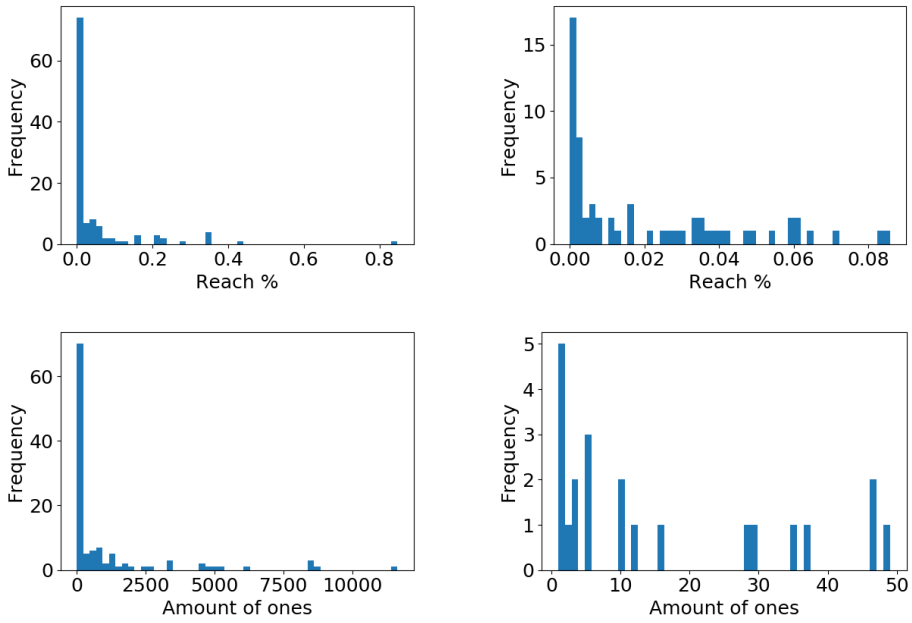


Figure 5: Distribution of the dependent variable on sub-brand level. The top two panels show the percentage of the minority value (1) for this variable, while the panels on the bottom show the amount of observations for this minority value (1). Furthermore, the panels on the left show the distribution of all the sub-brands, while the panels on the right provide the distribution of the highly unbalanced sub-brands.

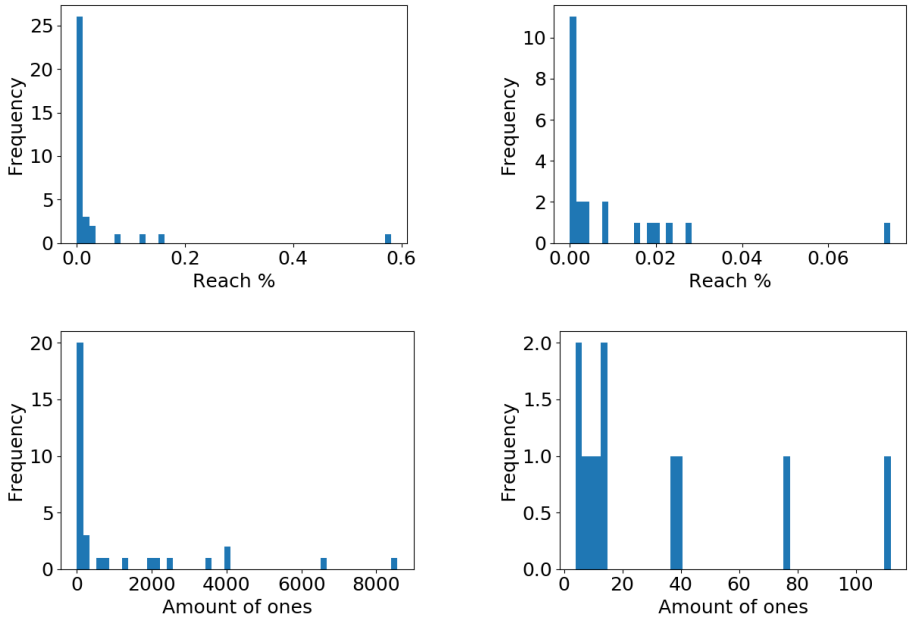


Figure 6: Distribution of the dependent variable on brand level. The top two panels show the percentage of the minority value (1) for this variable, while the panels on the bottom show the amount of observations for this minority value (1). Furthermore, the panels on the left show the distribution of all the brands, while the panels on the right provide the distribution of the highly unbalanced brands.

trees (Kotsiantis, 2013), the EBM model uses GAMs to model the relationships. Besides, these variables have a similar sense. Therefore, they are likely to describe the same effect. This makes it more difficult to interpret the model if both variables would be included. Therefore, the variable *Spanish Language* will not be used as a feature in the modelling step. The continuous features, shown in the right panel, in general have a slightly higher correlation across each other, although non of them stand out in particular. Similar to the categorical variables, these features do not have a high correlation with the dependent variable. The complete overview of all correlations across variables, including the corresponding values, can be found in Figure 17 in Appendix A.2.

Desktop. Similar to the paragraph for the device type mobile above, this paragraph will provide the exploratory data analysis of the final data. However, this section will perform this analysis on the pre-processed data corresponding to the device type desktop. Similar to the paragraph above, this paragraph will first examine the distribution of the dependent variable. This is followed by the summary statistics of the features on an overall level. Finally the correlation between the features is examined on an overall level.

As described in Section 1, one of the challenges is the extremely skewed distribution of the dependent variable, *panel reached*, for some of the (sub-)brands. The distribution of the reach percentage and amount of observations with the minority variables across all sub-brands is shown in Figure 5. The distribution of the reach percentage and amount of observations with the minority variables across all brands is shown in Figure 6. Similar to the plots for mobile in the paragraph above, the distribution of this variable is skewed. For most brands and sub-brands the reach percentage is less than twenty percent. However, in contradiction to the findings in the paragraph for the mobile data, the sub-brand data sets seem to have often a more balanced amount of observations for ones and zeros for the *panel reached* variable in comparison to the data on brand level.

The explanatory variables will be investigated more in depth. Due to the large amount of (sub-)brands, this will be done on an overall level for the desktop data set. The summary statistics for all variables are shown in Table 6 in Appendix A.2. Similar to the mobile data set, the continuous variables follow a zero-inflated distribution. Some variables, such as *Category 1* and *Category 10* contain zeros in more than 75 percent of the observations.

Next, we investigate the correlation between the features. The correlation between the continuous and categorical variables are all around zero. Therefore, the interesting correlations are shown in Figure 7. The categorical features, shown in the left panel, all have a correlation around zero with the dependent variable. Between the categorical features there is a high correlation of 0.77 between *Age group* and *Life stage*. Furthermore, a strong negative correlation exists between the variables *Occupation* and *Working status* of -0.86, and between *Hispanic* and *Spanish language* of -0.87. Using the same reasoning as for the device mobile, the features *Working Status* and *Spanish Language* will not be used in the modelling for this device type.

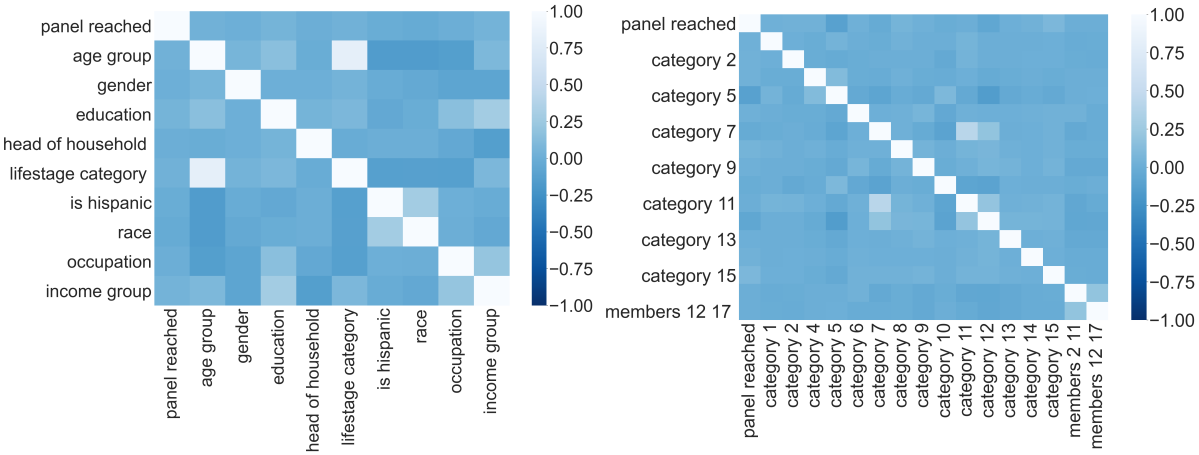


Figure 7: Correlation between the features and the dependent variable for the desktop data set. The left panel contains the correlations of the categorical variables, while the right panel displays the correlations of the continuous variables.

The correlation of the continuous features, shown in the right panel, all have a correlation around zero with the dependent variable. In contrast to the correlation between the continuous variables in the mobile data, shown in the right panel of Figure 4, the correlation between the continuous features is less strong. The complete overview of all correlations across variables, including the corresponding values, can be found in Figure 18 in Appendix A.2.

4 Methodology

This section provides an elaborate description of the methods and models used to solve the research problem as stated in Section 1. This modelling will be performed on the brand × sub-brand × device level. In this research, one application of a certain brand × sub-brand × device will be referred to as a certain *case*. Throughout the rest of this paper, when no additional comments are made, we denote k as the number of features within a case with the corresponding subscript $j \in \{1, ..., k\}$, and n as the number of observations within a specific case with the corresponding subscript $i \in \{1, ..., n\}$.

As discussed in Section 2.2, one of the recent promising developments in the field is the introduction of a glass-box model called EBM by Nori et al. (2019). To create an interpretable model for all cases, balanced and unbalanced, two models will be applied with certain variations. One of these models is EBM, the other is the new model referred to as ERF. The first interpretable ML method, EBM, is introduced in Section 4.1. Furthermore, Section 4.1 describes the modelling choices made in the standard setting with a balanced dependent variable. This is followed by the introduction of the new explainable ML algorithm called Explainable Random Forests in Section 4.2, which allows for an extensive interpretation of the model in a similar way as EBMs. Next, in Section 4.3 we present three strategies for dealing with the cases that have an unbalanced dependent variable, which includes; oversampling; adjusting the splitting criteria; and adjusting the loss function. Furthermore, Section 4.3 provides a description of relevant evaluation metrics to asses model performance for unbalanced data. Finally, Section 4.4 introduces the methodology for the robustness analysis of the interpretation from the ML methods allowing for an extensive interpretation.

4.1 Explainable Boosting Machines (EBM)

An EBM algorithm uses the generalisation of a simple linear model called GAM, where a linear combination of feature functions f_j is used. This linear combination can be written as

$$g\left(E[y]\right) = \beta_0 + \sum_{j=1}^{k} f_j(x_j), \tag{5}$$

where $g(E[y_i])$ is called the link function that links the feature functions f_j to the prediction of y_i, $\mathbf{x} = [x_1, ..., x_k]$ is a vector with explanatory variables, y is the dependent variable, β_0 is a constant, and $f = [f_1, ..., f_k]$ is a vector of flexible feature functions that links the explanatory variables to the dependent variable. An extension of the traditional GAM proposed by Lou et al. (2013) is the addition of pairwise interactions:

$$g\left(E[y]\right) = \beta_0 + \sum_{j=1}^{k} f_j(x_j) + \sum_{i \neq j} f_{i,j}(x_i, x_j), \tag{6}$$

Algorithm 1 Cyclic Boosting Procedure of EBM in pseudo-code

Data: Training set (X, Y)
Result: Estimated feature functions f_j
Initialisation;
B; the number of outer bags
k; the number of features in X
$L(y_i, F(x))$; Loss function
M; the number of iterations

Bagging step;
for $b = 1$ *to* B **do**
  Construct (X_b, Y_b)
  Initialise a first prediction: $F_{0,0}(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$
  Cycled Boosting step;
  for $m = 1$ *to* M **do**
    for $j = 1$ *to* k **do**
      if $m = 1$ *and* $j = 1$ **then**
        $residuals = Y_b - F_{0,0}(x)$
      else if $m > 1$ **then**
        $residuals = Y_b - F_{j,m-1}(x)$
      else
        $residuals = Y_b - F_{j-1,M}(x)$
      end
      Fit a decision tree $D_{j,m}$ on $(residuals, X_{b,k})$
    end
  end
  Create $f_{b,j}$ based on predictions for buckets from $D_{j,m} \forall j \in \{1, ..., k\}, m \in \{1, ..., M\}$
end
for $j = 1$ *to* k **do**
  Create the final feature functions; $f_j = \frac{1}{B} \sum_{b=1}^{B} f_{b,j}$
end

where $\mathbf{x} = [x_1, ..., x_k]$ is a vector with explanatory variables, y is the dependent variable, and $f = [f_1, ..., f_k, f_{1,2}, ..., f_{k-1,k}]$ is a vector of flexible feature functions that links the explanatory variables, including the interactions, to the dependent variable. The algorithm, as described in Lou et al. (2013), can select the relevant cross terms based on FAST, a method which measures and ranks the strength of all the interaction pairs.

What makes EBM, in sense of performance, comparable to other well-known ML methods, is the estimation of the feature functions. This estimation uses common ensemble methods such as bagging and boosting. However, this procedure differs from the traditional procedure, in the sense that a cyclic procedure is applied. As GAMs need a feature function f_j for the corresponding variables x_j, for all $j \in \{1, ..., k\}$. The structure of this algorithm, including the cycled boosting procedure, is provided in Algorithm 1. A more intuitive visualisation of this cycled procedure is shown in Figure 8. Although this procedure allows a direct parallel implementation, this procedure pays an additional training cost to keep the individual terms

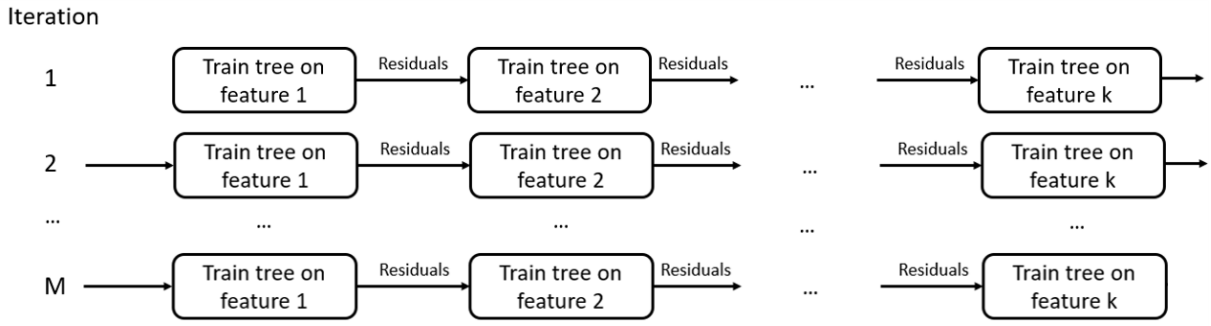


Figure 8: An intuitive visualisation of the structure of the cycled boosting step

additive. Therefore, this method is in general slower than other tree-based ensemble methods such as XGBoost (Chen et al., 2015) and LightGBM (Ke et al., 2017).

In order to translate the decision trees $[D_{j,1}, ..., D_{j,M}]$ to a feature function f_j, for all $j \in \{1, ..., k\}$, this algorithm uses so-called buckets. Before fitting the decision trees, these buckets need to be created. For categorical variables, each of the categories correspond to one single bucket. For the continuous variables, the number of buckets will be set to a certain value. The default for this, selected by Nori et al. (2019), is 256. Therefore, this value will be used for all EBMs. Then, the range of the continuous variable will be split-up in buckets with an equal amount of observations. Each of the buckets correspond to one small part of the feature function. For both the categorical and continuous features, the prediction for each bucket is based up on the fitted (boosted) decision trees. These buckets together describe the feature functions $f_j, \ \forall j \in \{1, ..., k\}$ in terms of the log odds as

$$f_j(x) = F_{0,0}(\mathbf{y}) + \sum_{m=1}^{M} \gamma \cdot Bucket(x_j) \tag{7}$$

where $F_{0,0}(y)$ is the initial prediction in terms of a log odds, γ is the learning rate, and $Bucket(x)$ is the log odds for a specific bucket. Each of the points in this function will be transformed to a probability with a logistic function, defined as

$$\hat{P}\{y_i = 1|x_i\} = \frac{e^{log(odds)}}{1 + e^{log(odds)}} = \frac{odds}{1 + odds}, \tag{8}$$

which has the range of a Sigmoid function, which makes this function suitable for calculating the probability.

Modelling Choices. In the cyclic boosting algorithm, as described in Algorithm 1, various modelling choices need to be made, such as, the loss function, the splitting criteria, and various hyper-parameters. These hyper-parameters include the number of boosting rounds; the number of bagging rounds; the size of the bagged data sets; and the learning rate. The choices and motivation for each of these decisions for the standard, fairly balanced data sets, are described below.

- As the dependent variable *panel reached* is a binary variable, the typical loss function used is binary-cross entropy loss function:

$$L\left(y, F(x)\right) = -\frac{1}{n}\sum_{i=1}^{n}\left(y_i ln\left(\hat{P}\{y_i|x_i\}\right) + (1-y_i) ln\left(1-\hat{P}\{y_i|x_i\}\right)\right), \tag{9}$$

where y_i is the true label ($y_i = 0$) when the panellist is not reached; ($y_i = 1$) when the panellist is reached); $\hat{P}\{y_i|x_i\}$ is the predicted probability of the panellist to be reached; and n is the number of observations. The initial prediction $F_{0,0}(\mathbf{y})$ of this loss function is equal to the $log(odds)$ defined as:

$$log(odds) = ln\left(\frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n}(1-y_i)}\right), \tag{10}$$

which can be converted to a probability using a logistic function as defined in Equation 8.

- As described in Algorithm 1, various decision trees $D_{j,m}$ need to be trained in the cyclic boosting procedure. In the standard setting, with relatively balanced data, the traditional boosting methods uses regression trees to estimate the residuals in a certain iteration. In these trees the *information gain* is estimated for each of the possible splitting points. The most common used metric for boosted tree algorithms to define the information gain is the *Sum of Squared Errors (SSE)*, which will be minimised:

$$SSE = \sum_{i=1}^{n}(y_i - \hat{P}\{y_i|x_i\})^2, \tag{11}$$

where y_i denotes the true label for observation i; and $\hat{P}\{y_i|x_i\}$ denotes the predicted probability for observation i.

- The hyper-parameters required in Algorithm 1, are determined based on the recommendation of Nori et al. (2019) combined with an optimisation to obtain the best accuracy and interpretability. The recommended hyper-parameters of Nori et al. (2019) are as follows; $B = 100$ outer bags for the bagging step; $M = 5000$ iterations (epochs); and a learning rate of 0.01. As a result of this small learning rate, the order in which the features are trained, becomes irrelevant (Nori et al., 2019). However, these settings will result in a computationally intensive training procedure for EBMs. More specifically, the Algorithm of EBM, as shown in Algorithm 1, requires fitting $B \times M \times k$ shallow regression trees with a limited maximum depth. As this process is computed for various data sets, B and M will be restricted to keep the total training time feasible. Therefore, an optimisation will determine the values of B, M and the maximum depth of the trees. As the learning rate is set to 0.01, the amount of iterations M to be optimised, is set to a hundred. As a result the range for the optimisation of the parameter is set to $M \in [100, 5000]$. Furthermore, the minimum number of bags in this optimisation is set to ten, resulting in a range of $B \in [10, 100]$. The optimisation for the maximum depth of the trees is restricted to $Depth \in \{3, 4, 5, 6\}$. This optimisation will be performed once on a specific combination of device $\times$ brand $\times$ $sub-brand$, to keep this computationally feasible. Furthermore, Nori

et al. (2019) recommends to use an additional hundred inner bags within the process of fitting a single tree. This process can help with the predictive performance, but adds significantly to the training time. As a result, this is not included in the default setting, In this research we will therefore not use this additional bagging step.

Evaluation Metrics. In order to compare the methods used in this research, various evaluation metrics will be used. Standard metrics to be used for this research problem are given below.

- The Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{P}\{y_i|x_i\}\right)^2}, \tag{12}$$

  where y_i denotes the true value corresponding to observation i, and $\hat{P}\{y_i|x_i\}$ denotes the predicted value for observation i.

- The binary logistic loss function ($L(y, F(x))$ as defined in Equation 9.

- The accuracy will be measured using a balanced metric defined as

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}\right), \tag{13}$$

  where TP is an abbreviation of the true positive rate; TN is equal to the true negative rate; FP is the false positive rate; and FN is equal to the false negative rate. This balanced accuracy metric can be seen as the average of sensitivity and specificity. As these measures are used on the fairly balanced data, this metric will be close to the unweighted accuracy. Furthermore, the cut-off point to determine the classification of an observation based on the predicted probability $\hat{P}\{y_i|x_i\}$ will be determined as $|TPR - FPR|$, where TPR is the true positive rate, and FPR is the false positive rate across observations.

- The *Variable Importance* is one of the most common approaches for non-interpretable methods to 'explain' the model. This variable importance is computed by sorting the predictors according to the mean decrease they achieve in the splitting criteria. In this research, this measure will be used to asses if the various models we apply, have similar variable importance rankings. As other ML methods, besides EBM, do not allow for an extensive interpretation, the variable importance is one of the few metrics to determine whether these methods find similar relationships in the data

Due to the bagging and cycled boosting procedure, EBM is known to be computationally intensive. Although many advances have been made in recent years to improve the computational powers, in practice this often still is an important aspect within model selection. Therefore, the *Computational Time* will be included to compare various methods. As this research problem deals with multiple models, as for each case a separate model is trained, these evaluation metrics will be summarised. For this aggregation the median-of-means (MoM) will be used.

Cross Validation. The primary goal of this application is to provide predictions of the probability of a panellist to be reached, and therefore this model will not need to make out-of-sample predictions. However, in order to estimate a more accurate test error rate, cross-validation will be used on all models that will be estimated. The cross-validation method used in this research is a K-fold cross validation. In order to keep the running time reasonable, a 5-fold cross-validation will be used. The estimate of the test error rate is then defined as the average of the evaluation metrics in the cross-validation rounds, estimated as follows:

$$CV_{RMSE,(5)} = \frac{1}{5}\sum_{i=1}^{5} RMSE_i, \tag{14}$$

where the RMSE is the abbreviation of the Root Mean Squared Error, as defined in Equation 12. However, this computation of the cross-validated metric holds for all evaluation metrics. Another possibility to obtain more accurate estimates of the test error rate, would be to use Model Selection or Regularisation strategies. However, the number of features does not require a subset selection. Furthermore, in this application it is important to obtain an interpretation from the EBMs for all features.

4.2 Explainable Random Forests (ERF)

In this section we introduce a new fully interpretable ML method. In this research, this method is referred to as Explainable Random Forests (ERF). In this research ERF will be applied to binary classification. However, with small adjustments, this method is suitable for multi-class classification. As explained in Section 2.1, recent advances have been made in order to enhance the interpretability of the widely applied random forest algorithm introduced by Breiman (2001). However, the explainability of these traditional Random Forests is still limited. This research introduces ERF as a new fully interpretable algorithm based on the concept of EBMs. This method is developed as an alternative for EBMs with two main purposes. First of all, this algorithm will train classification trees for classification problems, instead of regression trees which are used in EBM. Therefore, various developments specifically created for classification trees can be applied to this algorithm. In this research these developments will contain of adjustments to deal with extremely unbalanced data. Second, this algorithm will require less computational power, as no boosting procedure is applied and therefore less decision trees will be trained.

Similar to EBMs, this algorithm uses the feature function structure of GAMs, as shown in Section 4.1, and Equations 5 and 6. However, for ERF we use a substitution $f_j = v_j h_j$, where v_j are weights and h_j unweighted feature functions. As a result we can rewrite Equation 5 as

$$g(E[y]) = \sum_{j=1}^{k} v_j \cdot h_j(x_j). \tag{15}$$

In this algorithm, the feature functions are estimated using an algorithm based on the concepts of Random Forests and the weighting procedure from AdaBoost, introduced by Freund et

al. (1999). By using random forests to estimate these feature functions, this algorithm has some important differences in comparison to EBM. First of all, ERF does not use a boosting approach, but uses separately trained deeper trees. Therefore, this algorithm can be trained in parallel[1] and allows the use classification trees, as each separate tree predicts the probability instead of the residuals. As a result, this algorithm allows specific adaptions made for unbalanced data on classification trees. Note that similar to EBMs, when probabilities need to be summarised or averaged, this algorithm uses a logistic function, as defined in Equation 8, to transform the probabilities to the log of the odds. These odds are summarised, and translated back to probabilities.

This ERF algorithm uses bagging methods to estimate the feature functions of GAMs. More specifically, this algorithm produces independently grown deep trees to predict the probability of an observation corresponding to a certain class. This ensemble algorithm is well-known as Random Forests. However, in ERF each tree will be trained on a single feature. As a result these trees can be combined to create feature functions. Note that as this algorithm grows deeper trees in comparison to EBM, the number of buckets need to be optimised. In this research the number of buckets is set to 256, similar to the number of buckets in EBM, to avoid overfitting. As these trees all predict the probability in itself, rather than the predicted residuals in boosted algorithms, each of the initial predictions have a range of the total probability between zero and one. Note, that as a single tree is trained on a single feature, this algorithm trains more shallow trees in comparison to the standard Random Forests. However, as the depth of these trees are not limited, these trees will be deeper than the trees in the EBM.

In contrast to traditional Random Forests, this algorithm will assign weights to the trees in the forest. To determine the weight for a decision tree, the same concept is used as in AdaBoost. More specifically, each tree will receive a weight $v_{j,b}$ based on the performance of the tree. This weight is the importance measure in AdaBoost (Freund et al., 1999), for binary classification this is denoted as

$$v_{j,b} = \frac{1}{2} ln \left(\frac{1 - \epsilon_{j,b}}{\epsilon_{j,b}} \right) \tag{16}$$

where $\epsilon_{j,b}$ is the error rate from model $D_{j,b}$. These individual weights $v_{j,b}$ and feature functions $h_{j,b}$ are transformed to the variables needed for Equation 15. More specifically, the unweighted feature functions h_j are denoted as

$$h_j = \frac{\sum_{b=1}^{B} v_{j,b} \cdot h_{j,b}}{\sum_{b=1}^{B} v_{j,b}}. \tag{17}$$

Note that these unweighted feature functions h_j technically are weighted based on the performance of the trees in the boosting rounds. However, these functions are unweighted *between* features, in the sense that all features have an equal weight. In order to create a new prediction based on these functions a weight to reflect variable importance is required. This scaling with v_j is desired to ensure the probabilities are not scaled in the weighting process. Otherwise, when the number of variables k increases, the differences between the weights of these functions

[1] As EBMs includes a bagging step, part of the algorithm allows to be trained in parallel. However, the cycled boosting step is trained sequentially. Therefore, this step does not allow to be trained in parallel.

Algorithm 2 Explainable Random Forest (ERF) algorithm

Data: Training set (X, Y)
Result: Estimated feature functions f_j
Initialisation;
k; the number of features in X
B; the number of iterations
Bagging step;
for $b = 1$ *to* B **do**
  Construct (X_b, Y_b)
  for $j = 1$ *to* k **do**
    Fit a classification tree $D_{j,b}$ on $(Y_b, X_{j,b})$
    Predict $h_{j,b}$ based on $D_{j,b}$
    Determine weights $v_{j,b}$ as defined in Equation 16
  end
end
for $j = 1$ *to* k **do**
  Construct unweighted feature functions h_j as defined in Equation 17
  Construct weights v_j as defined in Equation 18
end

decreases. To finalise the estimation of feature functions $f_j = v_j \cdot h_j$, weights v_j defined as

$$v_j = \frac{1}{B} \sum_{b=1}^{B} v_{j,b}, \tag{18}$$

where $v_{j,b}$ are the weights of an individual tree defined in Equation 16. The exact pseudo code of this new model is shown in Algorithm 2. Note that other weighting alternatives to Equation 16 based on the performance could work. However, note that a scaling constant for these measures might be needed, due to the additive nature of GAMs to determine the final predictions. The weights $v_{j,b}$ need to be proportional to the number of features. When these weights will be relatively high this will result in extreme predictions (around 0.001 or 0.999), while relative small weights will result in predictions around 0.5.

Modelling Choices. As described in Algorithm 2, various decision trees $D_{j,m}$ need to be trained for the estimation of the feature functions from ERF. As this Algorithm does not follow a boosting procedure, in which the residuals are predicted, classification trees will be used in this algorithm for this research problem. As a result, different modelling choices can be made. More specifically, ERF will use different splitting criteria as EBM, which uses regression trees. Furthermore, this algorithm requires different hyper-parameters. The cross-validation strategy and evaluation metrics used for this method remain the same as for EBMs, as explained in Section 4.1.

- In the standard setting, with relatively balanced data, classification trees estimate the *information gain* for each of the possible splitting points. The two most common used metrics for classification problems are the *Gini impurity* and the *Cross-entropy*. This research uses the definition of these metrics as defined in James et al. (2017). In which

the Gini impurity for K classes is defined as

$$G = 1 - \sum_{k=1}^{K} p_{mk}(1 - p_{mk}), \tag{19}$$

where K is the number of classes; and p_{mk} is the proportion of observations in the mth region from the kth class (James et al., 2017). The Information Gain, introduced by Quinlan (1986), is denoted by James et al. (2017) as

$$D = -\sum_{k=1}^{K} p_{mk}\ log_2\left(p_{mk}\right), \tag{20}$$

with the same notation as used in Equation 19. As the logarithm for the information gain is computationally intensive, the use of the Gini Impurity is preferred to Information Gain.

- The hyper-parameters required in Algorithm 2, will be optimised similar to the procedure as mentioned in Section 4.1 for EBMs. Furthermore, this is based on common default values for random forests and the recommendations for EBMs by Nori et al. (2019), such the hyper-parameters align with those of EBM and common choices for non-explainable Random Forests. The specific choices and ranges for the hyper-parameters are as follows:
  - $B \in [50, 1000]$ bags, which results in B different data sets and B different classification trees per feature x_j.
  - *Maximum depth = None.* ERF will not limit the depth of the trees, a common choice for Random Forests, as they are in general based on deeper grown trees.

4.3 Strategies for Unbalanced Data

This application deals with some cases in which the dependent variable is (extremely) unbalanced. In this research, we compare three existing strategies to deal with this issue; oversampling; modifying the splitting criteria; and adjusting the loss function. As shown in Section 2.3, these strategies have proven to be valuable for other (boosted) tree-based methods.

The first strategy to deal with unbalanced data is oversampling. As shown by Japkowicz & Stephen (2002) the standard C5.0 approach of decision trees is one of the least sensitive methods to class imbalance. Furthermore, they showed that oversampling is in general effective against unbiased data, while under-sampling is not effective. In this research we will therefore use *Random Oversampling* as the sampling method. This method randomly duplicates observations in the minority class to achieve a balanced data set. More advanced sampling methods exist, such as SMOTE introduced by Chawla et al. (2002). However, we will not apply this, as this method generates synthetic samples which potentially affect the relations and explanations we discover with the explainable ML methods. As this oversampling is applied in the bagging step, this approach can be applied to both EBMs and ERFs. More specifically, this is applied in the *Bagging Step* in Algorithm 1 and Algorithm 2.

The second strategy to deal with unbalanced data, is to adjust the splitting criteria. As mentioned in Section 2.3, the metrics introduced in Section 4.1 and 4.2 are skewed sensitive splitting criteria (Flach, 2003). Therefore, these measures can not be used on the cases with an unbalanced dependent variable. As mentioned in 4.1, boosting methods use regression trees as to predict the residuals. As a result MSE is used as a splitting criteria for boosted algorithms. Random Forests contrarily combines separately grown classification trees, which utilise the Gini or Gain measure as a splitting criteria. This strategy to deal with unbalanced data therefore consists of different adaptions to these two type of models. Therefore, EBM and ERF will have different splitting criteria to deal with unbalanced data.The remainder of this section separates these two types of trees. The algorithms training regression trees, including the cycled boosted EBMs, will use a weighted SSE as a splitting criteria for the unbalanced cases, instead of the MSE. This splitting criteria is defined as

$$Weighted\ SSE = \sum_{i=1}^{n}(1 - w_i)(y_i - \hat{P}\{y_i|x_i\})^2, \tag{21}$$

where w_i is the proportion of the class corresponding to observation i. With the use of this splitting criteria, the observations corresponding to a minority class will receive more importance than the others in the general score. The algorithms training classification trees, including the ERF, will use the Hellinger distance as a splitting criteria, instead of the Gini and Gain measures. This distance was first use as a splitting criteria in introduced by Cieslak & Chawla (2008), where the decision tree trained with the Hellinger distance as a splitting criteria, is referred to as Hellinger Distance Decision Trees (HDDT). Furthermore, this splitting criteria has proven to be robust in random forests (Aler et al., 2020). This Hellinger distance, as introduced by Cieslak & Chawla (2008), is defined as

$$d_H = \sqrt{(\sqrt{P(L|+)} - \sqrt{P(L\|-)})^2 + (\sqrt{P(R|+)} - \sqrt{P(R|-)})^2}, \tag{22}$$

where $P(L)$ and $P(R)$ denote the weight of the left and right branches; $P(L|+)$ and $P(L|-)$ and $P(R|+)$ and $P(R|-)$ denote the probability of belonging to the corresponding class for the left and right branches respectively.

The third strategy to deal with unbalanced data is adjusting the loss function. More specifically, we will use a weighted cross-entropy loss function, which has been applied in XGBoost in Wang et al. (2020). This Weighted cross-entropy loss function is defined as

$$L_W(y, F(x)) = -\sum_{i=1}^{n}(\alpha \cdot y_i \cdot ln(F(x)) + (1 - y_i) \cdot ln(1 - F(x))), \tag{23}$$

where α denotes the imbalanced parameter. When $\alpha > 1$, the loss function will put more weight on the data with label 1 and vice versa. As a result, the initial value $F_{0,0}(x)$ is defined as

$$F_{0,0}(x) = ln\left(\frac{\sum_{i=1}^{N}\alpha \cdot y_i}{\sum_{i=1}^{N}(1 - y_i)}\right). \tag{24}$$

Furthermore, the imbalance parameter α will be denoted as

$$\alpha = \frac{w_0}{w_1}, \tag{25}$$

where w_1 is the proportion of the class with label 1; and w_0 the proportion of the class with label 0. As a result Equation 23 can be rewritten as

$$L_W\left(y, F(x)\right) = -\sum_{i=1}^{n}\left(\frac{1}{w_1}\cdot y_i \cdot ln(F(x)) + \frac{1}{w_0}\cdot(1-y_i)\cdot ln(1-F(x))\right), \tag{26}$$

such that a smaller proportion w_i, results in a higher weight for the observations in the minority class. Furthermore, as shown in Algorithm 1, adjusting the loss function will only change the initial prediction of $F_{0,0}$. Therefore, this adaption is always combined with weighted residuals. The weights applied to the residuals are based on the proportion of the class. These weighted residuals are denoted as

$$e_i = (\alpha y_i + (1-y_i))\cdot e_i^*, \tag{27}$$

where e_i^* denotes the unweighted residual corresponding to observation i; α is defined using Equation 25, and Y_i is the true label corresponding to observation i. As shown sections 4.1 and 4.2, the EBM algorithm uses a loss function, while ERF does not need a loss function, as no boosting is applied. Therefore, this adaption is applied only to EBM. The equivalent of this approach for ERFs is to use a weighted performance measure to determine the weights $v_{j,b}$ as denoted in Equation 16. The error rate $\epsilon_{j,b}$ will therefore be weighted. This weighted error rate is denoted as

$$\epsilon_{j,b} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{P}\{y_i|x_i\}\right|(\alpha y_i + (1-y_i)), \tag{28}$$

Evaluation Metrics. Standard classification metrics do not represent the models performance for imbalanced data. Furthermore, to fully compare a models performance it is important to take into account multiple measures (Raeder et al., 2012). Although the balanced accuracy as defined in Equation 13 takes the proportion of the classes into account, additional measures are needed to analyse the performance on unbalanced data. Therefore, other metrics to examine the model will be used in this research. These metrics are divided in two types of metrics. The first are classification metrics, which take into account the classification of an observation. The second type are probabilistic evaluation metrics. These take into account the estimated probability of an observation. The classification is made based on the optimal cut-off point as described in Section 4.1.

Common metrics, which use the predicted class of the panel reached for extremely imbalanced data, are precision and recall:

- The precision metric, also known as positive prediction rate, is defined as

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}, \tag{29}$$

where *True Positives* is the number of observations with a correct positive ($y_i = 1$) clas-

sification, and *False Positives* is the number of observations where the predicted value is equal to one ($\hat{y}_i = 1$), while the true label is equal to zero.

- The recall metric, also known as sensitivity, is defined as

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}, \tag{30}$$

where *False Negatives* is defined as the number of observations where $\hat{y}_i = 0$, while $y_i = 1$.

- Precision and Recall can be combined in a single metric using the F-score:

$$F_\beta = (1 + \beta^2)\frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}, \tag{31}$$

where the parameter β controls the trade-off of importance between precision and recall. When $\beta = 1$ precision and recall are evenly weighted. When β increases ($\beta > 1$), the F-score puts more emphasis on precision and vice versa when β is decreased.

The second type of evaluation metrics to include in this research are the metrics that use the predicted probability to evaluate a model. MSE and cross-entropy, mentioned in Section 4.1, weigh each observation equally, which are therefore not reliable for (extremely) unbalanced data. Therefore, this research will use a modified Brier score introduced by Wallace & Dahabreh (2012):

$$BS^+ = \frac{\sum_{y_i=1}(y_i - \hat{P}\{y_i|x_i\})^2}{N_{pos}} \tag{32}$$

$$BS^- = \frac{\sum_{y_i=0}(y_i - \hat{P}\{y_i|x_i\})^2}{N_{neg}}. \tag{33}$$

These measures combined provide more information on the performance of the model for unbalanced data, as both classes are taken into account.

4.4 Robustness Analysis

As mentioned in Section 1, the extensive interpretation of glass-box ML methods raises new questions such as the robustness of these feature functions. As mentioned in Section 2.4, robustness for ML models can refer to the minimum adversarial perturbation, across many similar samples, or the minimum adversarial perturbation across samples with adversarial noise. In this research we will examine the robustness of the feature functions against adversarial noise. This research will utilise the definition of robustness R as defined in Rauber et al. (2017):

$$R = [\rho(\mathbf{x})]_{\mathbf{x}} \tag{34}$$

where

$$\rho(\mathbf{x}) = \min_{\delta} d(\mathbf{x}, \mathbf{x} + \delta) \quad \text{s.t. } \mathbf{x} + \delta \text{ is adversarial}, \tag{35}$$

where $d(.)$ is a distance measure, and $\mathbf{x}+\delta$ is the data including perturbations. As this research focuses on the robustness of the explanations of glass-box models, the robustness of two aspects from these models are taken into account. The first aspect is the set of feature functions. For these feature functions, standard distance measures for values can not be applied. The distance between a single feature function $d(f_{\mathbf{x},j}, f_{\mathbf{x}+\delta,j})$ will be determined using the average of the absolute difference over the buckets in the feature functions:

$$d(f_{\mathbf{x},j}, f_{\mathbf{x}+\delta,j}) = \frac{1}{L}\sum_{l=1}^{L} \mid f_{\mathbf{x},j,l} - f_{\mathbf{x}+\delta,j,l} \mid, \tag{36}$$

where $j \in \{1, ..., k\}$ refers to the feature function of variable x_j, L denotes the number of buckets in feature function f_j, M_0 is the model trained on the data without perturbations and M_1 refers to the model trained on the data set including perturbations. These distances across features will be aggregated using a weighted average, with the weight for a feature, equal to the variable importance. The second aspect to take into account are the variable importances of the models. Although the variable importance is directly linked to the feature functions, it is important to analyse the robustness of this metric. More specifically, a change in the feature functions can occur with and without a change in the relative importance of a variable. The overall change in the variable importance will be denoted as

$$d(Imp_{\mathbf{x}}, Imp_{\mathbf{x}+\delta}) = \frac{1}{k}\sum_{j=1}^{k} |Imp_{\mathbf{x},j} - Imp_{\mathbf{x}+\delta,j}|, \tag{37}$$

where $Imp_{\mathbf{x}}$ refers to the variable importance of the model trained on the data set without perturbations, and $Imp_{\mathbf{x}+\delta}$ refers to the variable importance of the model trained on the data set including perturbations. For interpretation purposes the variable importance will be scaled such that the importances for a specific model sum up to one. This research will examine the performance of these feature functions against perturbation in terms of cell-wise contamination. The contamination will be generated using the Cell-wise contamination model introduced by Alqallaf et al. (2009), denoted as

$$\mathbf{x} + \delta = (\mathbf{I_p} - \mathbf{B}_\epsilon)\mathbf{x} + \mathbf{B}_\epsilon \mathbf{z}, \tag{38}$$

where $\mathbf{x}$ follows the model distribution, $\mathbf{z}$ follows an outlier generating distribution, $\mathbf{I_k}$ is an identity matrix, ϵ is the probability of being an outlier, and $\mathbf{B}_\epsilon = diag(B_1, ..., B_k)$ with $B_j \sim Bin(1, \epsilon)$ independently. This results in a probability that an observation has at least one outlier value of $1 - (1-\epsilon)^k$.

5 Computational Results

This section provides an overview of the results of all methods for predicting the reach across the various cases, as described in Section 4. This section starts with an overview of the results for the balanced cases in Section 5.1. Next, the results of the three different strategies on the unbalanced data are presented in 5.2. After which, the results of the robustness analysis are presented in Section 5.3. Furthermore, Section 5.4 will provide additional insights on the effect of hyper-parameters on the shape of the feature functions of ERF.

5.1 Balanced Cases

This section provides the result of the models as described in Section 4 for the fairly balanced cases (*brand* $\times$ *sub-brand* $\times$ *device*). Tree-based methods in general have a good performance on slightly imbalanced data, as mentioned in Section 2.3. Therefore, the cases with a reach between 0.4 and 0.6 are selected as balanced cases. This results in two cases, the device type desktop, and the device type mobile. These data sets have reach percentages of 41.8% and 56.4% for desktop and mobile respectively. The performance of EBMs and ERFs will be compared to other non-explainable machine learning methods, such as, RF, LightGBM, and GBDT. The hyper-parameters, resulting from the optimisation as provided in Section 4 are as follows: ERF uses $B = 100$ bags, and therefore trains $100 \cdot k$ trees; EBM uses $B = 10$ bags and $M = 200$ iterations, and therefore trains $10 \cdot 200 \cdot k$ trees; RF trains 1000 trees; GBDT trains 100 trees; the maximum depth of boosted trees is set to 3, for both GBDT and EBM.

Table 2: Performance for models on balanced cases

	RMSE		Binary logloss		Balanced Accuracy	
	In-sample	Out-of-sample	In-sample	Out-of-sample	In-sample	Out-of-sample
EBM	0.419	0.421	0.526	0.530	0.745	0.738
ERF (Gini)	0.431	0.434	0.549	0.556	0.763	**0.748**
ERF (Entropy)	0.431	0.434	0.549	0.556	0.763	**0.748**
LightGBM	0.367	**0.410**	0.419	**0.501**	0.805	0.741
Random Forest	0.150	**0.411**	0.138	**0.505**	1.000	0.712
Gradient Boost	0.431	0.434	0.556	0.561	0.744	0.736

The average in-sample and out-of-sample performance of the models on balanced cases are displayed in Table 2. The full results for data sets can be found in Tables 7 and 8 in Appendix B. In terms of the out-of-sample predictive power, the explainable ML methods (EBM and ERF) have comparable performances to the popular black-box ML models RF, GBDT and LightGBM. Although LightGBM outperforms the other models in terms of RMSE and Binary Logloss, the difference to the other models is very small for all measures. More specifically, the out-of-sample measures have a difference of at most 0.06. Furthermore, the explainable ML methods even outperform RF and GBDT in terms of accuracy. Besides, both ERF models provide similar results, in which for all measures, the difference is not greater than 0.001.

All models and its hyper-parameters are selected with the same procedure using a 5-fold cross validation to minimise the prediction error. However, the in-sample performances provide fairly large differences across the models. More specifically, RF overfits the data as the out-of-sample performance is almost three and four times as large as the in-sample metric for the RMSE and the Binary logloss respectively. Besides, the difference in in-sample and out-of-sample accuracy is almost 30%. LightGBM shows slight differences between the in-sample and out-of-sample metrics as well, although these differences are less extreme in comparison to RF. Strikingly, the other models, including the explainable ML models have similar performances on the in-sample and out-of-sample data. The difference between the in-sample and out-of-sample metrics for EBM and ERF is at most 0.07, which is smaller than a 2% difference.

Table 3: The optimal threshold for all models on balanced cases

	Desktop (41.8%)	Mobile (56.4%)
EBM	0.57	0.46
ERF (Gini)	0.62	**0.35**
ERF (Entropy)	0.62	**0.38**
LightGBM	0.49	0.50
Random Forest	0.61	0.53
Gradient Boost	0.62	0.43

As mentioned in Section 4.1, the balanced accuracy measure requires a cut-off point to label the data. These cut-off points, for both the in-sample and out-of-sample metric, are determined using the true positive rate and false positive rate for observations in the training data. The results of these optimal thresholds are provided in Table 3 for both data sets. For all models, with the exception of LightGBM, we observe that the threshold is higher for desktop than for mobile, which has a higher reach. Furthermore, we observe that for ERF the difference between the thresholds is the most extreme, specifically for mobile, as the threshold of mobile is almost two times as small as the threshold for desktop.

Variable Importance. As discussed in Section 4, the variable importance is for non-explainable ML methods the most common metrics to provide limited information of the black-box models. Therefore, it is interesting to compare the variable importance across explainable and black-box models. These variable importances across the models are provided in a heat-map in Figure 9 for mobile and desktop in the left and right panel respectively. These importances are scaled such that the most importance feature has an importance equal to one. As a result of this scaling, the importances and difference in values are easier to interpret. From both panels in Figure 9 we observe that, there is a common set of variables which are more important than other across the models. For mobile the most important features are *Category 3*, *Category 14*, *Category 2*, *Category 11*, and *Category 15*. For the desktop case the most important variables are *Total Viewing*, *Category 5*, and *Category 11*. Furthermore, we observe that Gradient Boost in general has lower importances in comparison to the other models. The effect of the top two

	ERF (Entropy)	ERF (Gini)	EBM	Light GBM	Random Forest	Gradient Boost
Category 14	1.00	0.99	1.00	0.84	1.00	0.71
Category 3	1.00	1.00	0.93	0.56	0.99	1.00
Category 15	0.65	0.65	0.83	1.00	0.82	0.31
Category 11	0.68	0.68	0.80	0.96	0.87	0.33
Category 2	0.72	0.72	0.74	0.57	0.73	0.09
Category 4	0.51	0.50	0.73	0.58	0.54	0.02
Category 8	0.59	0.59	0.71	0.47	0.58	0.00
Category 1	0.31	0.30	0.57	0.28	0.23	0.00
Category 6	0.48	0.48	0.56	0.58	0.56	0.00
Category 9	0.57	0.58	0.54	0.56	0.55	0.01
Category 7	0.52	0.52	0.49	0.48	0.52	0.00
Total Viewing	0.28	0.28	0.44	0.04	0.19	0.02
Category 16	0.25	0.24	0.44	0.53	0.34	0.00
Category 5	0.47	0.48	0.44	0.78	0.49	0.01
Category 13	0.29	0.29	0.42	0.30	0.25	0.00
Category 10	0.45	0.45	0.40	0.51	0.47	0.00
Category 12	0.41	0.41	0.36	0.70	0.49	0.00
Education	0.04	0.04	0.35	0.31	0.21	0.00
Category 18	0.22	0.22	0.26	0.50	0.46	0.00
Income	0.04	0.04	0.25	0.19	0.19	0.00
Category 20	0.22	0.22	0.20	0.62	0.46	0.00
OS	0.03	0.03	0.18	0.07	0.05	0.00
Gender	0.03	0.03	0.16	0.10	0.07	0.00
Employment State	0.04	0.04	0.11	0.12	0.19	0.00
Spanish Language	0.04	0.04	0.10	0.43	0.34	0.00
Age Group	0.04	0.04	0.08	0.09	0.09	0.00
Race	0.04	0.04	0.07	0.16	0.25	0.00
Married	0.03	0.03	0.05	0.07	0.10	0.00
Ethnic	0.03	0.03	0.04	0.03	0.07	0.00
Panel Type	0.03	0.03	0.04	0.04	0.04	0.00
	0.03	0.03	0.04	0.00	0.00	0.00

(a) Mobile

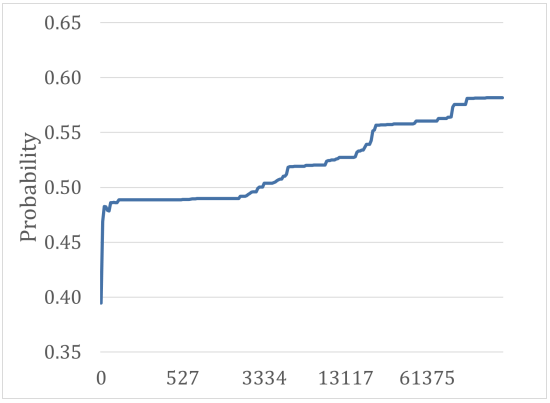
	ERF (Entropy)	ERF (Gini)	EBM	Light GBM	Random Forest	Gradient Boost
Total Viewing	1.00	1.00	0.90	0.11	0.51	1.00
Category 5	0.89	0.89	0.80	1.00	1.00	0.53
Category 11	0.87	0.87	1.00	0.58	0.76	0.92
Category 12	0.32	0.32	0.46	0.54	0.52	0.01
Category 2	0.22	0.22	0.45	0.52	0.24	0.12
Category 9	0.18	0.19	0.44	0.47	0.23	0.03
Category 6	0.18	0.18	0.43	0.50	0.21	0.03
Category 7	0.15	0.15	0.33	0.35	0.16	0.00
Category 10	0.14	0.14	0.37	0.30	0.20	0.00
Category 15	0.13	0.13	0.11	0.23	0.05	0.00
Category 8	0.09	0.09	0.04	0.13	0.03	0.00
Category 1	0.08	0.08	0.03	0.06	0.03	0.00
Head of Household	0.07	0.08	0.19	0.05	0.07	0.00
Category 4	0.07	0.07	0.02	0.22	0.09	0.00
Category 14	0.07	0.07	0.03	0.15	0.06	0.00
Category 13	0.06	0.07	0.01	0.08	0.02	0.00
Lifestage	0.06	0.06	0.15	0.23	0.24	0.00
Members 2	0.05	0.05	0.11	0.10	0.09	0.00
Race	0.05	0.05	0.17	0.19	0.12	0.00
Occupation	0.05	0.05	0.06	0.19	0.29	0.00
Education	0.05	0.05	0.12	0.19	0.25	0.00
Spanish Language	0.05	0.05	0.06	0.10	0.07	0.00
Age Group	0.05	0.05	0.13	0.35	0.29	0.00
Members 12	0.04	0.04	0.06	0.06	0.07	0.00
Working Status	0.04	0.04	0.03	0.14	0.20	0.00
Hispanic	0.04	0.04	0.05	0.00	0.04	0.00
Income Group	0.04	0.04	0.03	0.17	0.25	0.00
Gender	0.04	0.04	0.05	0.07	0.09	0.00

(b) Desktop

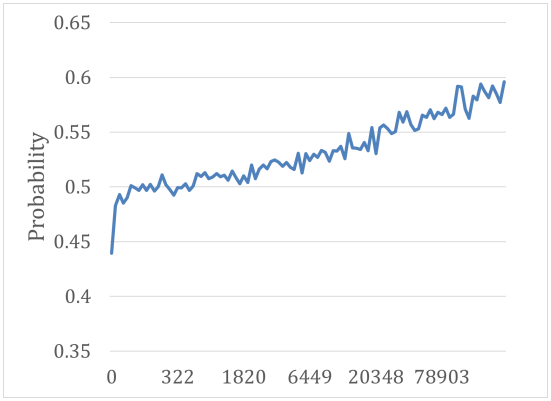
Figure 9: Variable importance across models for all features. For interpretation purposes the importances are scaled such that the most important feature has an importance equal to 1.0.

most important variables (*Category 3* and *Category 14* for mobile, and *Total viewing* and *Category 11* for desktop) therefore have a relative greater impact on the prediction in comparison to the other models. All other variables have an importance below 0.6, and over half of the variables has an importance equal to 0.00, while this does not occur in any of the other models.

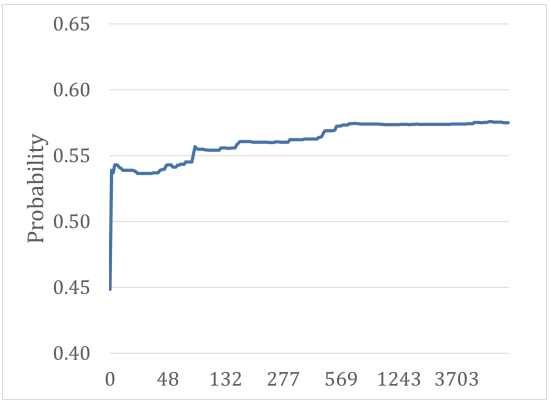
In general, we observe that the explainable ML methods have similar feature importances as RF and LightGBM. Although the values of these importances differ, there is a clear resemblance across the models on the importance across the variables. Furthermore, when we compare the importances of ERF and EBM, we observe that the values of the importances are even closer across features. As both models train separate feature functions, and in this application does not include cross-effect, this is as expected. Furthermore, the RF algorithm has issues with overfitting, as shown in Table 2, which suggests the models detects noise instead of true relations for certain variables. Therefore, this needs to be taken into account when comparing the importances across the models.



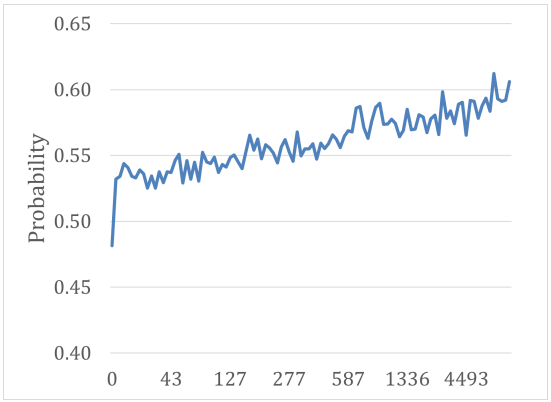
(a) EBM: *Category 5*



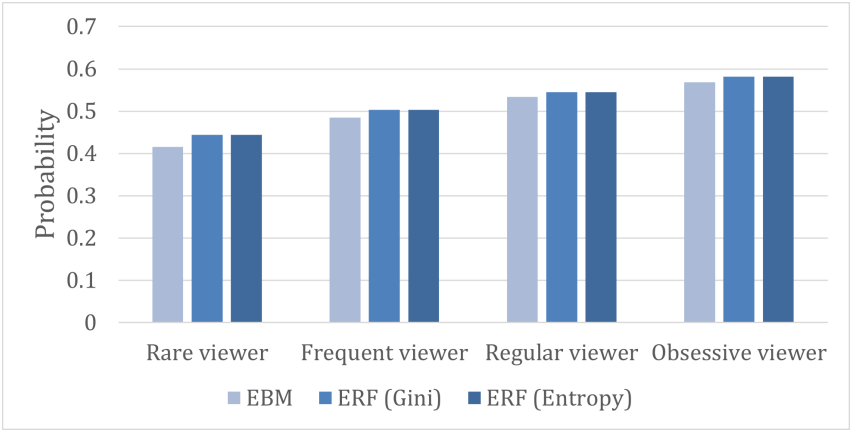
(b) ERF (Gini): *Category 5*



(c) EBM: *Category 11*



(d) ERF (Gini): *Category 11*



(e) *Total viewing* for all models

Figure 10: Feature functions of the top three most important variables for desktop

Interpretation. The explainable ML methods EBM and ERF, in comparison to black-box ML models, allow for an extensive local interpretation. The feature functions for the three most important variables of desktop, as mentioned above, are shown in Figure 10. As the feature functions of ERF with the splitting criteria Gini and Entropy are extremely similar, only the feature function of ERF (Gini) is provided. The feature functions of *Category 5* and *Category 11* for ERF (Entropy) can be found in Figure 19 in Appendix B.2. From the feature functions

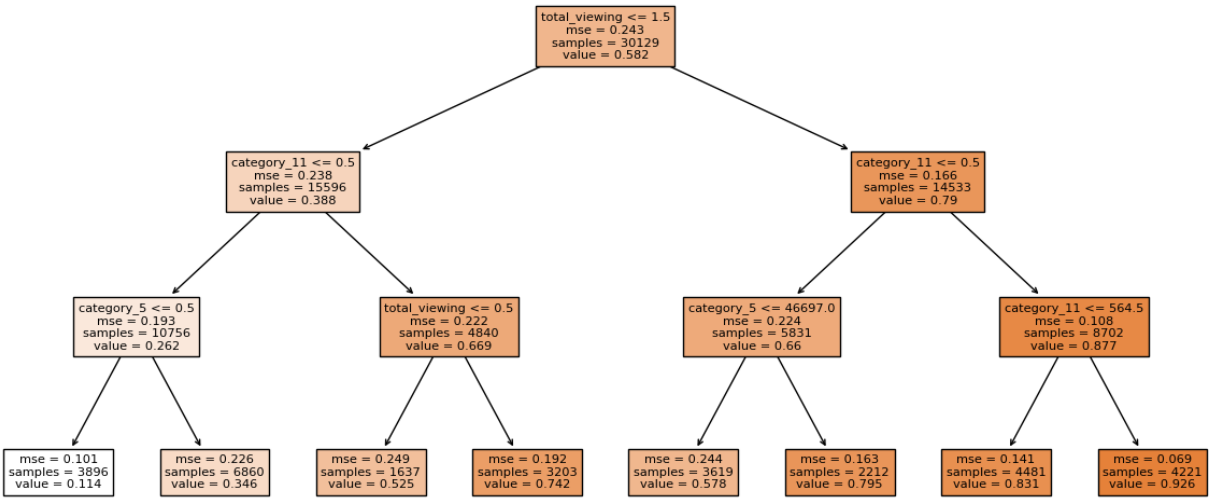


Figure 11: A single decision tree

in Figure 10, we observe that for both of the continuous variables, *Category 5* and *Category 11* the feature functions resulting from the ERF algorithm is less smooth than the feature function from EBM. Furthermore, the rough shapes of these functions are quite similar, although for both of the continuous variables the feature function of ERF has a higher starting value. This is similar for the categorical feature function of *Total Viewing*, in which all values for ERF are slightly higher than those from EBM. Although the increase across categories is similar. This clarifies the higher optimised threshold for ERFs (0.62) in comparison to EBM (0.45), as provided in Table 3. These feature functions allow for an extensive interpretation of the ML model and the contribution to forecasts from each variable. Combined with the variable importances, this provides a complete understanding of the model. The other ensemble models such as; LightGBM; Random Forest; and Gradient Boosting which are provided as a benchmark in Table 2, do not provide this extensive interpretation. However, a single decision tree provides an extensive interpretation when the tree is plotted. The plot of a single decision tree for this application, for the balanced desktop case, can be found in Figure 11. This model is limited in depth to prevent overfitting and improve the understanding of the relationships, and therefore results in only eight different probabilities. This tree has an out-of-sample balanced accuracy of 0.665. A deeper decision tree could potentially increase this performance, however, a deeper tree results in a more difficult interpretation very fast. More specifically, a fully grown decision tree with a certain *depth* results in a total of 2^{depth} different paths and a total of $2^{depth} - 1$ splits. A tree with a depth of six would therefore result in a tree with 32 paths and 31 splits. Besides, as mentioned in Section 2, these single decision trees do not have the same strong prediction power as ensemble ML methods.

Computational Considerations. As mentioned in Section 4, one of the important drawbacks of these explainable models is the computational requirement. As the explainable models estimate separate functions for each of the features, this results in additional computational requirements. The training time to fit the balanced cases for each of the models is shown in Figure 12. In which the desktop case is trained on 32,133 observations with 28 features, while the mo-

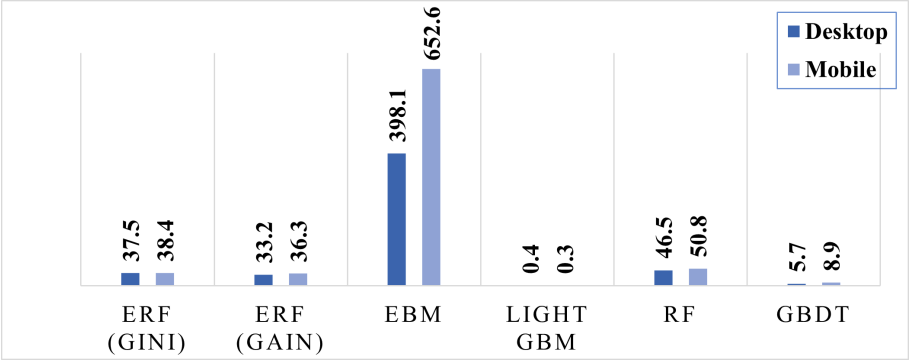


Figure 12: Training time in seconds across models

bile data includes 20,532 observations with 31 features. In Figure 12 we observe that the mobile case needs a longer training time, due to the larger amount of features. Furthermore, we observe that both of the explainable models have a significantly larger training time as the black-box ML methods. However, these explainable models are trained using a package with optimised code, which often allows for a parallel implementation. The explainable ML models however are not trained in parallel. As both EBM and ERF use a bagging approach, a further development of these models which include a parallel implementation would be desirable. Note that the computational requirements for the explainable ML models is highly dependent on the hyper-parameters such as the amount of bags B, amount of epochs M, and the number of features k. Furthermore, for both cases (mobile and desktop) ERF has a much faster training time, as this is over ten times as fast as the time for the only other explainable ML method, namely EBM.

5.2 Unbalanced Cases

This section provides the result of the three strategies to deal with (extremely) unbalanced data, as described in Section 4.3. All cases for which these strategies are applied have a reach percentage lower than twenty percent. These strategies are assessed using the evaluation metrics for unbalanced data, as provided in Section 4.3.

The two evaluation metrics which are based on the predicted probability, the balanced accuracy and Brier scores, are shown in Figure 13, for all strategies and reach percentages. From the first row of this figure, which displays the balanced accuracy, it immediately stands out that across all methods, EBM has a slightly increased performance in comparison to ERF. Furthermore, we note that in general all three strategies to deal with unbalanced data, only have a minimum impact on the accuracy. The Brier score has a score both for the minority and the majority class. From the scores of the majority class (*panel reached = 0*) in the second row of Figure 13, we observe the Brier score is close to zero for both EBM and ERF when no adjustments are made. This indicates that in general the predicted probabilities are close to zero and therefore have a high Brier score. The first plot in the third row supports this observation, as the observations in the minority class (*panel reached = 1*), have a high score resulting from a prediction close to zero. Therefore, we can conclude that, in this application, ERF predicts

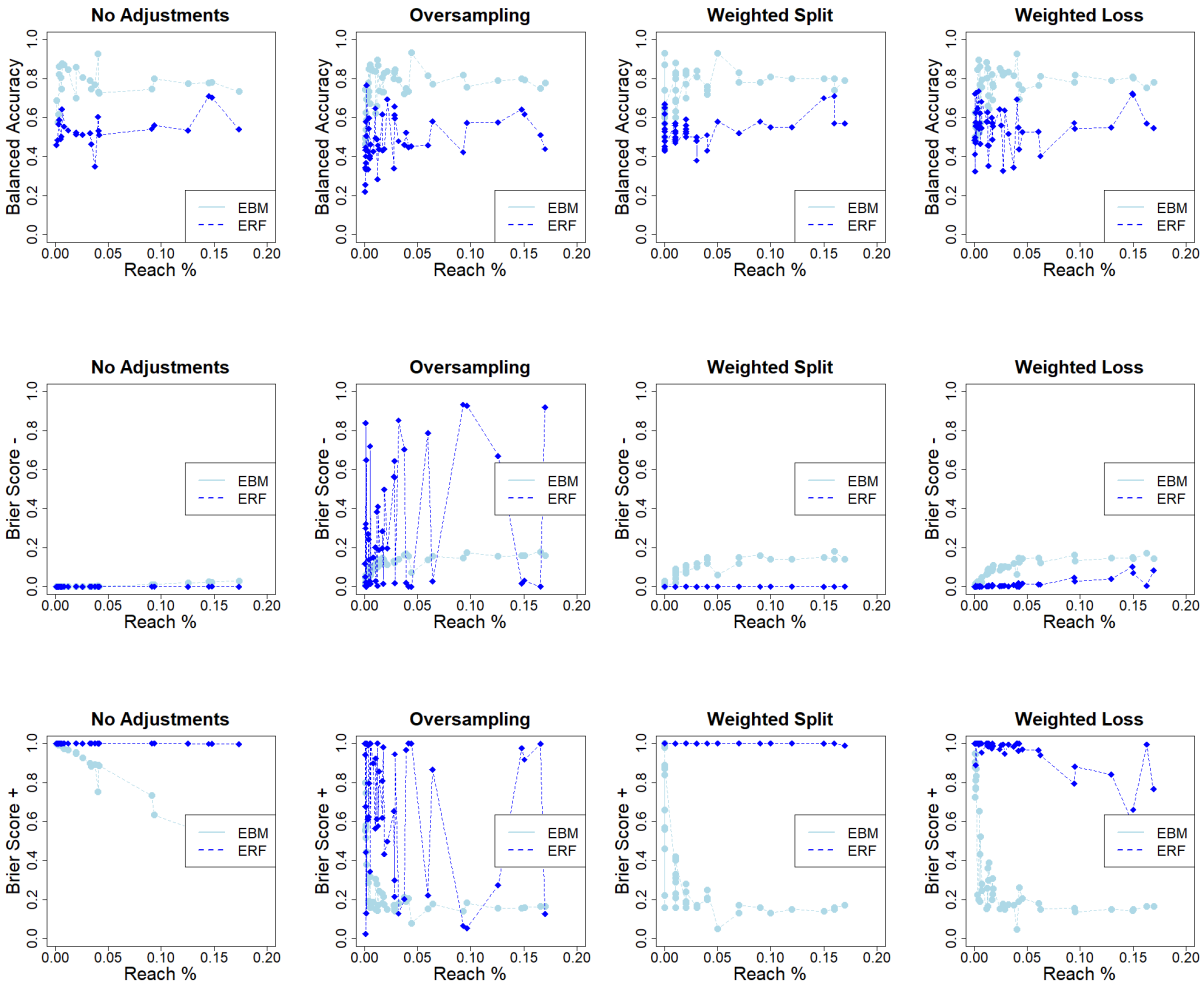


Figure 13: The balanced accuracy (first row) and Brier Score for the minority class (second row) and the majority class (third row) are displayed, across the different strategies to deal with unbalanced data for EBM (light blue) and ERF (blue). All these scores are calculated based on out-of-sample performance.

values around zero for all classes when the data is extremely unbalanced and no adjustments are made. This does not occur for EBM across all reach percentages. However, for cases with a reach below five percent, a similar effect is shown for EBMs.

Furthermore, all three strategies to deal with unbalanced data, seem to have a similar performance in terms of accuracy and Brier scores for EBM. All strategies improve the performance in comparison to a model without adjustments. However, the difference between performance of the three strategies, is minimal for EBM. For ERF on the other hand, not all strategies to deal with unbalanced data improve the performance in terms of accuracy and Brier scores. More specifically, the use of a weighted split has a similar performance to the model in which no adjustments are made. As mentioned above, this strategy does not prevent ERF to estimate a probability around zero for all classes. A similar effect is shown when the loss function is weighted for ERFs. Oversampling on the other hand, has a clear distinct effect on the perfor-

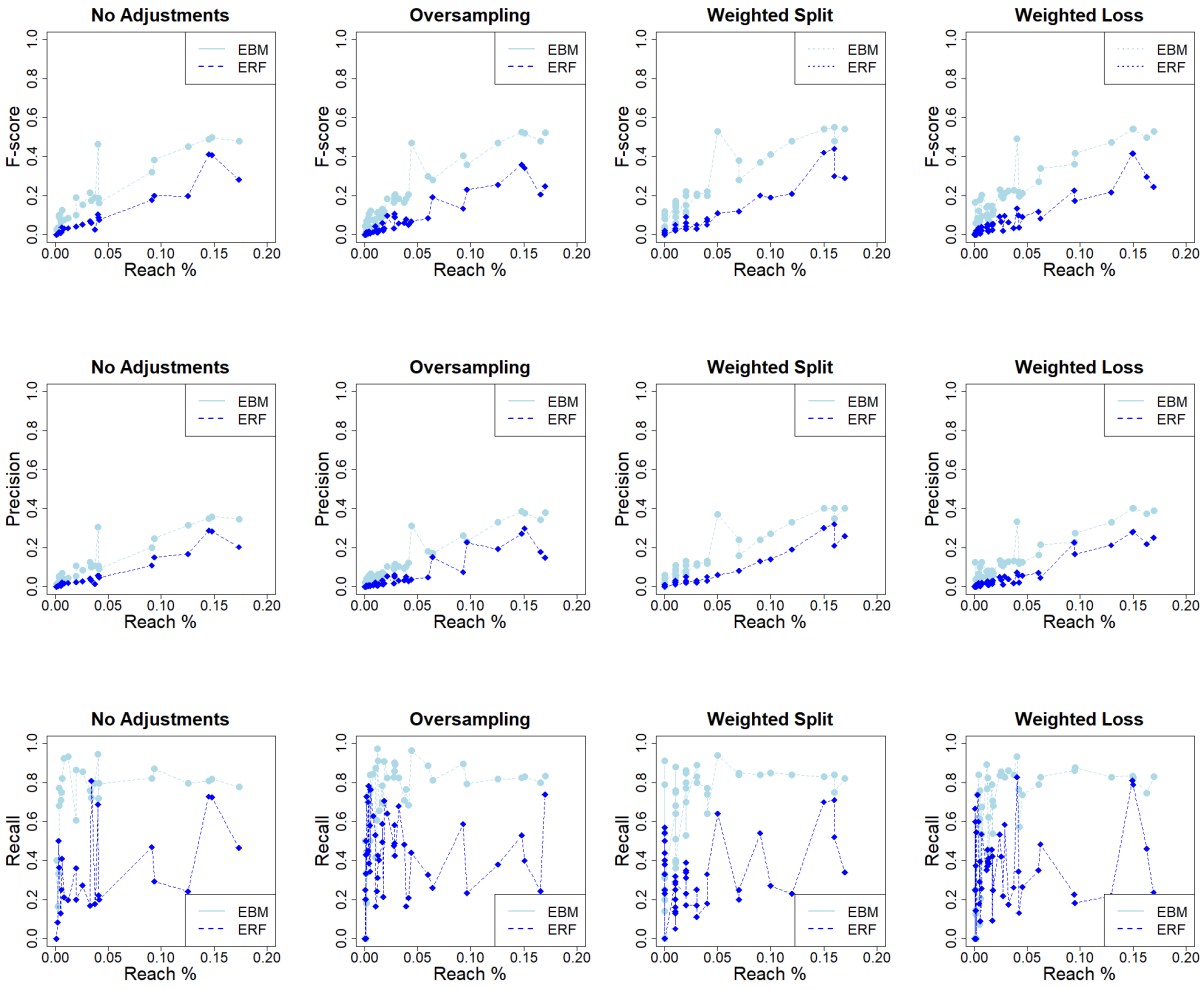
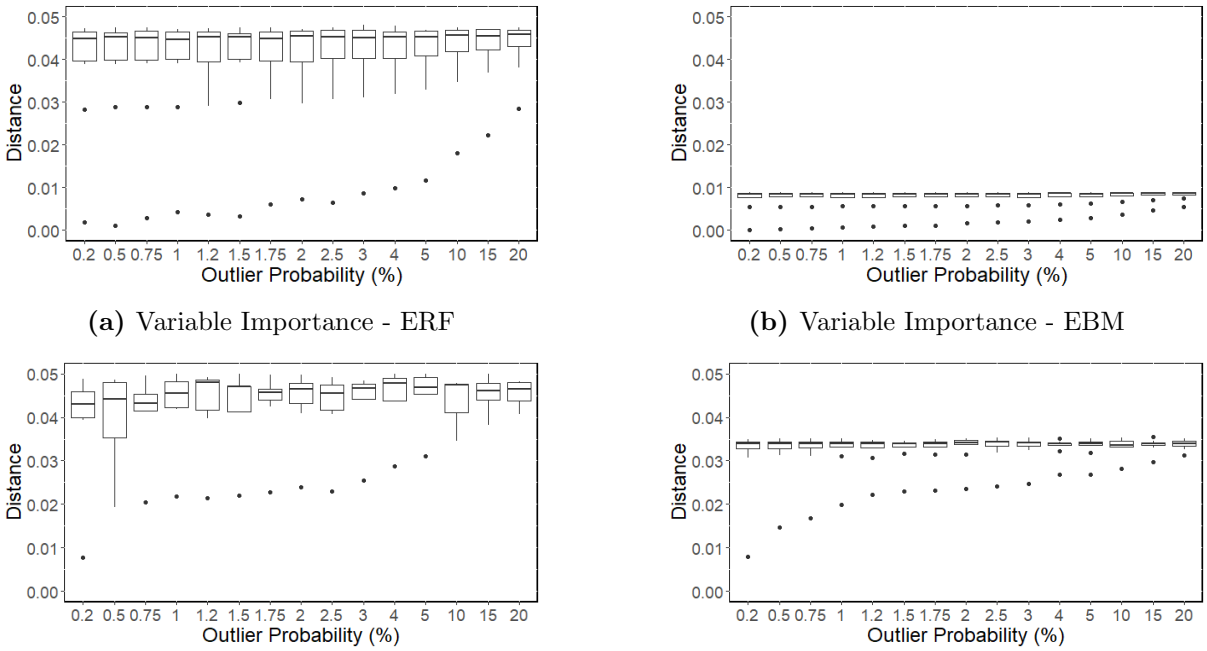


Figure 14: The F-score with $\beta = 1$ (first row), precision (second row) and recall measure (third row) are displayed for the minority class, across the different strategies to deal with unbalanced data for EBM (light blue) and ERF (blue). All these scores are calculated based on out-of-sample performance.

mance of ERF. Although the strategy does not have a positive effect for all cases. In general, the strategy does not have a positive effect on the extremely unbalanced cases with a reach lower than 0.02. However, for the less extreme unbalanced cases, the strategy only is effective about for around half of the cases.

The F-scores with $\beta = 1$, precision metric and recall metric for all strategies, are shown in Figure 14 for the minority class (*panel reached = 1*). These graphs align with the observations from Figure 13 as mentioned above. More specifically, EBM outperforms ERF across all strategies to deal with unbalanced data. Even when no adjustments are made. Furthermore, the strategies for EBM have similar performances, both in terms of precision and recall. Besides, the performances across all strategies improve when the reach percentage increases. This effect is the most visible in the plots for the F-score and Precision. The recall measure has an in general better performance across the strategies, even when no adjustments are made.



(a) Variable Importance - ERF

(b) Variable Importance - EBM

(c) Feature Functions - ERF

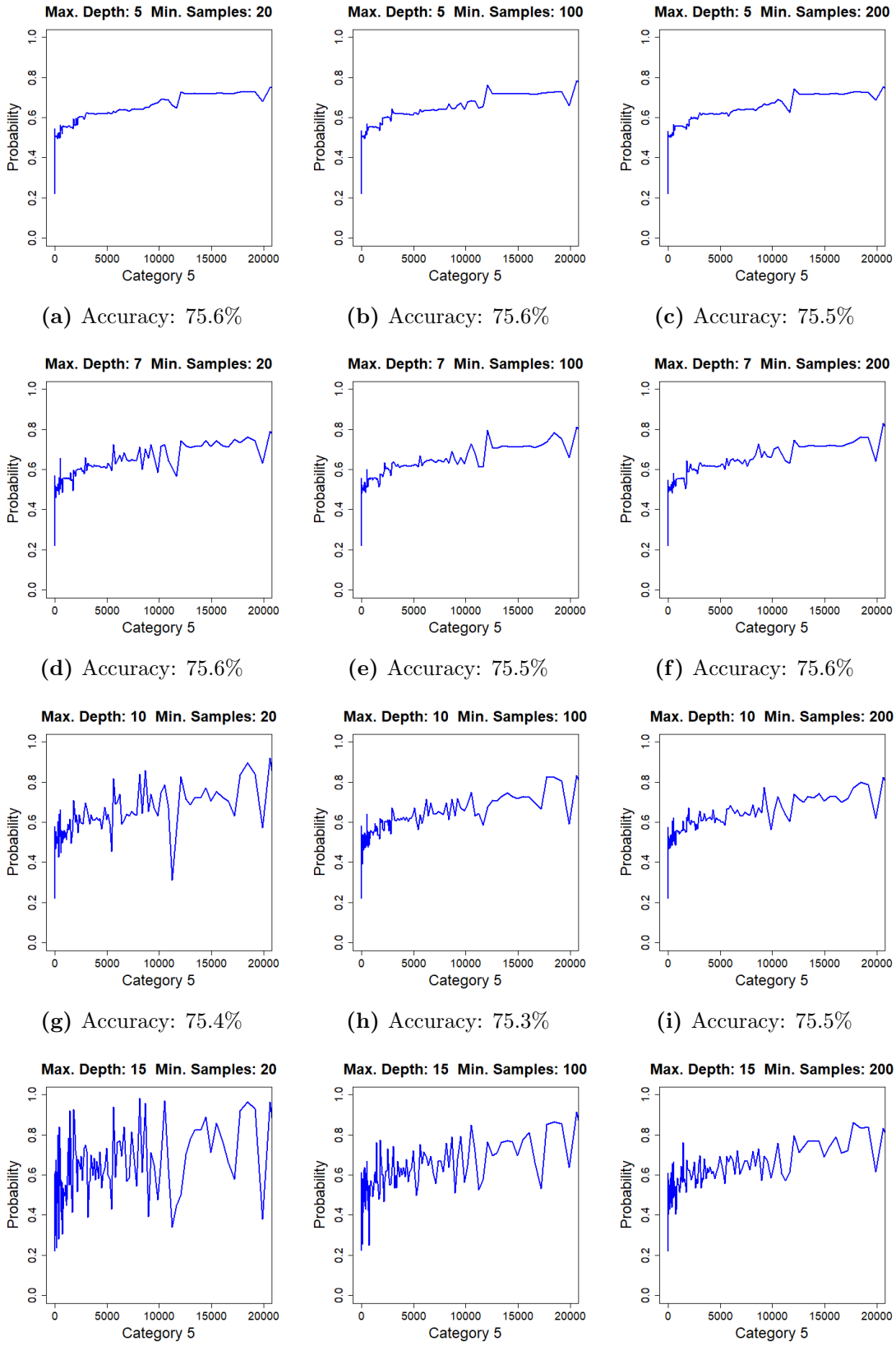
(d) Feature Functions - EBM

Figure 15: The distances for different levels of the outlier probability in percentages ($\epsilon \cdot 100$).

5.3 Sensitivity of The Explanations

This section provides the result of the sensitivity analysis, as described in Section 4.4. This sensitivity analysis is applied for the single balanced data set for device type desktop. Furthermore this sensitivity analysis is repeated ten times, as the generation of perturbation is at random.

Figure 15 provides the distances in the variable importance and feature functions for various levels of ϵ. The top two panels of Figure 15 show the average absolute distance in the variable importance. From these plots we observe that values of these distances are around 0.045 and 0.009 for ERF and EBM respectively, with a low standard deviation. Furthermore, the distances seem constant across the different values of ϵ. The large difference between these models is potentially caused by the difference in depth of the trees and amount of decision trees trained in the algorithms. However, as the individual importances are scaled between zero and one individually, with a value of one for the most important variable, this results in a minimal effect for both models. Therefore, an increase of at most 0.045 will only slightly affect the level of importance for a feature. The bottom two panels of Figure 15 show the aggregated absolute distances of the feature functions. From these plots we observe that the values of these distances are around 0.042 and 0.035 for ERF and EBM respectively. Similar to the distances for the variable importances, the distances in the feature functions do not clearly increase as the value of ϵ increases. As the level of the feature functions is between 0.40 and 0.60 for both models, the range of these feature functions have a range of about 0.20. An shift of 0.042 and 0.035 is therefore quite substantial. Furthermore, it stands out that the standard deviation of ERF is greater than the standard deviation of EBM.



(a) Accuracy: 75.6% **(b)** Accuracy: 75.6% **(c)** Accuracy: 75.5%

(d) Accuracy: 75.6% **(e)** Accuracy: 75.5% **(f)** Accuracy: 75.6%

(g) Accuracy: 75.4% **(h)** Accuracy: 75.3% **(i)** Accuracy: 75.5%

(j) Accuracy: 74.6% **(k)** Accuracy: 75.3% **(l)** Accuracy: 75.4%

Figure 16: Unweighted feature functions of *Category 5* when ERF is trained with different hyper-parameters of the *Minimum Samples* and *Maximum Depth* to train a single decision tree.

5.4 Effect of Hyper-parameters on Feature Functions: ERF

As seen in Figure 10 in Section 5.1, the feature functions of EBM are smoother in comparison to the feature functions of ERF. This section will examine the choice of specific hyper-parameters on the shape of the feature functions for ERF. More specifically, the effect of the *maximum depth* of a decision tree, and the *minimum samples* required for a split in a tree. These parameters are of interest because these parameters are different between ERF and EBM. Due to the cycling boosting procedure of EBM, the algorithm trains more trees. However, due to the restriction on the maximum depth (of three in this research) the regression trees are not fully grown like the trees trained in ERF.

This study of the parameters for ERF, is applied to a balanced case for the device type desktop. As shown in Figure 9 in Section 5.1, *Category 5* is the most important continuous variable for this data set. Therefore, the effect of the hyper-parameters will be examined based on plots from the function for this feature. In order to examine the effect of the hyper-parameters the plots of the unweighted feature functions $h_j(x_j)$, as denoted in Equation 15 and 17 will be examined. As a result these functions will have a larger range as the functions in Figure 9, as these weighted functions are scaled back towards a half.

Figure 16 shows all the plots of the feature functions for *Category 5* from the models trained with *Maximum Depth* $\in \{5, 7, 10, 15\}$ and *Minimum Samples* $\in \{20, 100, 200\}$. From this figure, we observe that a higher restriction on these two hyper-parameters will result in smoother feature functions. Note that both these measures will result in less deep trees. Furthermore, the out-of-sample balanced accuracy for each of these models, are all in a range of $[74, 6\%, 75.6\%]$. Although these values are slightly smaller than the balanced accuracy for the unrestricted model of 75.8%, as shown in Table 8, the difference is only 0.02% for the model resulting in the smoothest functions.

6 Conclusion

In this paper we studied the performance of explainable-tree based ensemble algorithms in a marketing context. Specifically, we examined the predictive power in comparison to black-box ML methods. Furthermore, we introduced a new interpretable ensemble method called ERF as an alternative to EBM. EBM applies boosting and bagging techniques on shallow trees, while ERF applies bagging techniques on full-grown decision trees. Results for balanced data show that both of the interpretable models, EBM and ERF, have similar performances in terms of accuracy as widely applied black-box models such as; Gradient Boosting; LightGBM; and Random Forests. Furthermore, ERF and EBM have small differences between in-sample and out-of-sample performances, and therefore, in this application, is not prone to overfitting. The Random Forest and LightGBM models on the other hand have larger differences between the in-sample and out-of-sample performance, and therefore over-fit the data. In contrast to the black-box models, ERF and EBM allow for an extensive interpretation of the model due to the feature functions and variable importances provided by these models. Both of these models provide a similar shape of the feature functions. However, EBM provides smoother functions than ERF, which are more bumpy. Another important finding it that the variable importances of both EBM and ERF on a high level align with the variable importances of the black-box models. Although the specific values differ for each of the models, all algorithms provide the largest importances to the same features. An important difference between EBM and ERF are the computational requirements. Both models allow for improvements by training in parallel. However, in the current implementation the training time of ERF takes only twelve percent of the time required by EBMs. Another important observation is that the optimal threshold for ERF is more extreme. This could potentially be due to the slight class imbalance.

Furthermore, this paper examined the performance of the interpretable ML algorithms on extremely imbalanced data, with a reach of at most twenty percent. More specifically, we examined the effect of three different strategies on the performance of EBM and ERF: oversampling, weighted splitting criteria, and a weighted loss function. While EBM and ERF provides similar performances on balanced data, this was not the case for the extremely unbalanced data. EBM outperforms ERF for all strategies for extremely unbalanced data. When no adjustments are made to these models, both models are biased towards the majority class. All three strategies to deal with unbalanced data, have shown a clear improvement on the performance for EBM in comparison to the model without adjustments. The differences across strategies were minimal for EBM. For ERF on the other hand, not all strategies were equally effective. The weighted splitting criteria, specifically designed for unbalanced data in classification trees, did not improve the performance in comparison to when no adjustment were made at all. Furthermore, the weighted loss functions had a slight positive effect, especially for a less extreme division of the classes. For ERF, oversampling has shown to be the most effective strategy to deal with unbalanced data. However, the strategy does not have a positive effect on all cases. More specifically, the positive effect seems to occur more often for cases which were more balanced, although this was not the case on all data sets.

Additionally, this paper examined the sensitivity of ERF and EBM against various levels of cell-wise outliers. Results show that the cell-wise outliers minimally affect the variable importance for both of these models. However, the effect of the perturbation on the feature functions is substantial with an average absolute shift of 0.04 in a range of 0.20. An important difference between ERF and EBM in these findings is that the distances of EBM have a lower standard deviation in comparison to the differences of ERF. This is potentially related to the depth of the decision trees and the number of the trees trained in the algorithms.

As mentioned above, the shape of the feature functions for continuous variables provided by ERF are more bumpy in comparison to the functions provided by EBM. In this paper we examined the effect on the shape of the continuous feature functions of two hyper-parameters required in this model: *maximum depth*, and *minimum split*. Results show that the smoothness of feature functions for continuous variables is dependent on these hyper-parameters. Specifically, a limited maximum depth or an increased number of samples required for a split will result in more smooth feature functions. Note that both of these settings will result in the training of more shallow trees.

Suggestions for further research. This research has shown that EBM and ERF provide interpretation without compromising on accuracy. Nonetheless, this research has several limitations, and potential promising advances could be made in further research. First of all, the prolonged training time in comparison to black-box models, could be troublesome for practical applications. Therefore, a study could be conducted to restrict the computational requirements. This could include a research to examine the minimum number of boosting and bagging rounds required, and a faster implementation which enables parallel training of the bagging step. Second, the proposed method referred to as ERF has been developed specifically for binary classification. This method could be further extended to be suitable for multi-class problems, with adjustments of the weighting scheme, and problems with the use of regression trees. Furthermore, this application has not included cross-effects of features. ERFs can be extended to include these cross-effects in a similar fashion as this is developed for EBMs by Nori et al. (2019). Although this addition would require additional computational costs for both EBM and ERF. As shown in this research, ERF is sensitive to (extreme) class imbalance, and is outperformed by EBMs for all strategies to deal with this imbalance. Further research would be valuable to examine this sensitivity and investigate the effect of other strategies to deal with class imbalance. For example, it would be valuable to study the effect of SMOTE on both the performance and the shape of the feature functions, as this oversampling technique generates synthetic samples. Furthermore, cross-validation has been developed to, among other things, select the hyper-parameters resulting in the best predictive performance. However, as shown in this research, the best performance can result in very bumpy feature functions for ERF. Therefore, it would be promising to do research to include this knowledge when selecting hyper-parameters. One could include the smoothness of a feature function in the cross-validation strategy to optimise the parameters. Such that the hyper-parameters for the optimal performance becomes a balance between predictive power and the smoothness of the feature functions. At last, this research has only applied these glass-box models in one context. Although EBMs have been

applied to numerous applications recently. It would be valuable to examine the performance of both EBMs and ERFs in other applications.

All in all, explainable ML methods such as EBM and ERF provides a deeper understanding of the model in comparison to black-box ML methods, while not limiting the performance. These models are not limited to the marketing context in this research, and can be valuable for various applications and sectors, such as healthcare and risk modelling. In these applications a thorough understanding of the model is of crucial importance.

References

Agarwal, R., Frosst, N., Zhang, X., Caruana, R., & Hinton, G. E. (2020, Apr). Neural additive models: Interpretable machine learning with neural nets. *arXiv.org*. Retrieved 2021-06-22, from `https://arxiv.org/abs/2004.13912`

Akyüz, M. H., & Birbil, Ş. İ. (2021). Discovering Classification Rules for Interpretable Learning with Linear Programming. *arXiv preprint arXiv:2104.10751*.

Aler, R., Valls, J. M., & Boström, H. (2020). Study of hellinger distance as a splitting metric for random forests in balanced and imbalanced classification datasets. *Expert Systems with Applications*, *149*, 113264.

Alqallaf, F., Van Aelst, S., Yohai, V. J., Zamar, R. H., et al. (2009). Propagation of outliers in multivariate data. *The Annals of Statistics*, *37*(1), 311–331.

Baquero, O. S., Santana, L. M. R., & Chiaravalloti-Neto, F. (2018). Dengue forecasting in São Paulo city with generalized additive models, artificial neural networks and seasonal autoregressive integrated moving average models. *PloS One*, *13*(4), e0195065.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and Regression Trees*. CRC press.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1721–1730).

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., et al. (2015). XGBoost: extreme gradient boosting. *R Package Version 0.4-2*. Retrieved 2021-06-22, from `https://mran.microsoft.com/web/packages/xgboost/vignettes/xgboost.pdf`

Cieslak, D. A., & Chawla, N. V. (2008). Learning decision trees for unbalanced data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (pp. 241–256).

Coussement, K., Benoit, D. F., & Van den Poel, D. (2010). Improved marketing decision making in a customer churn prediction context using generalized additive models. *Expert Systems with Applications*, *37*(3), 2132–2143.

De Maesschalck, R., Jouan-Rimbaud, D., & Massart, D. L. (2000). The mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems*, *50*(1), 1–18.

Dietterich, T., Kearns, M., & Mansour, Y. (1996). Applying the weak learning framework to understand and improve c4. 5. In *ICML* (pp. 96–104).

Drummond, C., & Holte, R. C. (2000). Exploiting the cost (in) sensitivity of decision tree splitting criteria. In *ICML* (Vol. 1).

Flach, P. A. (2003). The geometry of roc space: understanding machine learning metrics through roc isometrics. In *Proceedings of the 20th international conference on machine learning (ICML-03)* (pp. 194–201).

Freund, Y., Schapire, R., & Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, *14*(771-780), 1612.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Gillingham, P. (2016). Predictive risk modelling to prevent child maltreatment and other adverse outcomes for service users: Inside the 'black box'of machine learning. *The British Journal of Social Work*, *46*(4), 1044–1058.

Goodfellow, I., McDaniel, P., & Papernot, N. (2018). Making machine learning robust against adversarial inputs. *Communications of the ACM*, *61*(7), 56–66.

Hastie, T., & Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, *1*(3). doi: 10.1214/ss/1177013604

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning: With Applications in R*. Springer.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*(5), 429–449.

Jaxa-Rozen, M., & Kwakkel, J. (2018). Tree-based ensemble methods for sensitivity analysis of environmental models: A performance comparison with sobol and morris techniques. *Environmental Modelling & Software*, *107*, 245–266.

Johnson, R. A., Wichern, D. W., et al. (2002). *Applied Multivariate Statistical Analysis* (Vol. 5) (No. 8). Prentice hall Upper Saddle River, NJ.

Jowett, I., Parkyn, S., & Richardson, J. (2008). Habitat characteristics of crayfish (paranephrops planifrons) in new zealand streams using generalised additive models (gams). *Hydrobiologia*, *596*(1), 353–365.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (pp. 3146–3154).

Kotsiantis, S. B. (2013). Decision trees: a recent overview. *Artificial Intelligence Review*, *39*(4), 261–283.

Lecué, G., Lerasle, M., et al. (2020). Robust machine learning by median-of-means: theory and practice. *Annals of Statistics*, *48*(2), 906–931.

Li, A. H., & Bradic, J. (2018). Boosting in the presence of outliers: adaptive classification with nonconvex loss functions. *Journal of the American Statistical Association*, *113*(522), 660–674.

Liu, W., Chawla, S., Cieslak, D. A., & Chawla, N. V. (2010). A robust decision tree algorithm for imbalanced data sets. In *Proceedings of the 2010 SIAM International Conference on Data Mining* (pp. 766–777).

Lou, Y., Caruana, R., & Gehrke, J. (2012). Intelligible models for classification and regression. , 150–158.

Lou, Y., Caruana, R., Gehrke, J., & Hooker, G. (2013). Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 623–631).

Lundberg, S., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

Neto, M. P., & Paulovich, F. V. (2020). Explainable matrix—visualization for global and local interpretability of random forest classification ensembles. *IEEE Transactions on Visualization and Computer Graphics*.

Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*.

Papouskova, M., & Hajek, P. (2019). Two-stage consumer credit risk modelling using heterogeneous ensemble learning. *Decision Support Systems*, *118*, 33–45.

Petkovic, D., Altman, R. B., Wong, M., & Vigil, A. (2018). Improving the explainability of random forest classifier-user centered approach. In *PSB* (pp. 204–215).

Potts, W. J. (1999). Generalized additive neural networks. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 194–200).

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

Raeder, T., Forman, G., & Chawla, N. V. (2012). Learning from imbalanced data: Evaluation matters. In *Data mining: Foundations and Intelligent Paradigms* (pp. 315–331). Springer.

Rauber, J., Brendel, W., & Bethge, M. (2017). Foolbox: A python toolbox to benchmark the robustness of machine learning models. *arXiv preprint arXiv:1707.04131*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144).

Shami, M., & Verhelst, W. (2007). An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech. *Speech Communication*, *49*(3), 201–212.

Sigrist, F., & Hirnschall, C. (2019). Grabit: Gradient tree-boosted tobit models for default prediction. *Journal of Banking & Finance*, *102*, 177–192.

Tukey, J. W. (1960). A survey of sampling from contaminated distributions. *Contributions to Probability and Statistics*, 448–485.

Vaughan, J., Sudjianto, A., Brahimi, E., Chen, J., & Nair, V. N. (2018). Explainable neural networks based on additive index models. *arXiv preprint arXiv:1806.01933*.

Wallace, B. C., & Dahabreh, I. J. (2012). Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 IEEE 12th International Conference on Data Mining* (pp. 695–704).

Wang, C., Deng, C., & Wang, S. (2020). Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. *Pattern Recognition Letters*, *136*, 190–197.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *ICML* (Vol. 1, pp. 609–616).

Zhou, H., Qian, W., & Yang, Y. (2020). Tweedie gradient boosting for extremely unbalanced zero-inflated data. *Communications in Statistics-Simulation and Computation*, 1–23.

A Data

A.1 Data Description

Table 4: Features per device

Feature	Desktop	Mobile	Description
Continuous			
Age	✓	✓	The age of the respondent
Category 1	✓	✓	Automotive
Category 10	✓	✓	Commerce & Shopping
Category 11	✓	✓	News & Information
Category 12	✓	✓	Search Engines, Portals & Social Networking
Category 13	✓	✓	Special Occasions
Category 14	✓	✓	Telecom & Internet Services
Category 15	✓	✓	Travel
Category 16		✓	Photography
Category 18		✓	Productivity & Tools
Category 2	✓	✓	Computers & Consumer Electronics
Category 20		✓	Communication
Category 3		✓	Corporate Information
Category 4	✓	✓	Education & Careers
Category 5	✓	✓	Entertainment
Category 6	✓	✓	Family & Lifestyle
Category 7	✓	✓	Finance
Category 8	✓	✓	Government & Non-profit
Category 9	✓	✓	Home & Fashion
Members 12 17 count	✓		Number of children between age 12 and 17
Members 2 11 count	✓		Number of children between age 2 and 11
Binary			
Gender	✓	✓	Gender of the respondent
Head of household	✓		Head of household indicator
Hispanic	✓		Hispanic indicator
Categorical			
Education	✓	✓	Level of education of the respondent
Employment		✓	Employment status of the respondent
Ethnic		✓	Ethnicity of the respondent
Income	✓	✓	Income group of the respondent
Lifestage	✓		Lifestage of the respondent
Married		✓	Marital status of the respondent
Occupation	✓		Occupation type of the respondent

Continuous on next page

Feature	Desktop	Mobile	Description
OS		✓	Operating System
Race	✓	✓	Race of the respondent
Spanish language	✓	✓	Indicator whether the respondent speaks Spanish
State		✓	State in which the respondent is living
Working status	✓		Employment status of the respondent

A.2 Exploratory Analysis

Mobile

Table 5: Summary Statistics of the explanatory variables for mobile

Feature	Mean	St. Dev.	Median	Min.	Max.	25%	75%
Age Group	5.75	2.78	6	1	10	3	8
Category 1	43.81	156.33	0	0	1071	0	0
Category 10	8514.37	13136.63	2554	0	56551	231	10374
Category 11	7763.92	13094.52	1287	0	50558	102	7667
Category 12	53038.42	71023.44	23460	0	313265	4530	70020
Category 13	58.39	159.22	0	0	870	0	21
Category 14	435.05	740.06	88	0	3587	0	509
Category 15	1908.49	3303.77	372	0	19679	3	2037
Category 16	816.98	1810.24	12	0	7939	0	526
Category 18	11376.83	18315.71	2576	0	69515.5	350	11851
Category 2	338.72	635.43	42	0	2676	0	325
Category 20	35548.84	50433.07	14080	0	196490	2537	43208
Category 3	278.49	512.27	37	0	2389	0	290
Category 4	885.4	1909.62	23	0	7672	0	540
Category 5	75319.17	97979.4	34538	0	444162	7474	103350
Category 6	3682.88	6586.15	481	0	24990	9	3383
Category 7	3729.31	5784.16	1163	0	24775	86	4481
Category 8	386.36	794	17	0	3565	0	325
Category 9	1953.49	3290.68	413	0	14058.5	4	2238
Education	3.55	1.42	3	0	6	3	5
Employment	108.16	132.93	1	0	275	1	272
Ethnic	0.19	0.39	0	0	1	0	0
Gender	0.39	0.49	0	0	1	0	1
Income	11.2	24.03	2	0	75	0	3
Married	0.48	0.5	0	0	1	0	1
OS	1.73	1.98	0	0	4	0	4
Race	0.82	1.73	0	0	6	0	1
Spanish language	5.42	1.36	6	0	6	6	6
State	23.05	14.85	22	0	57	8	37

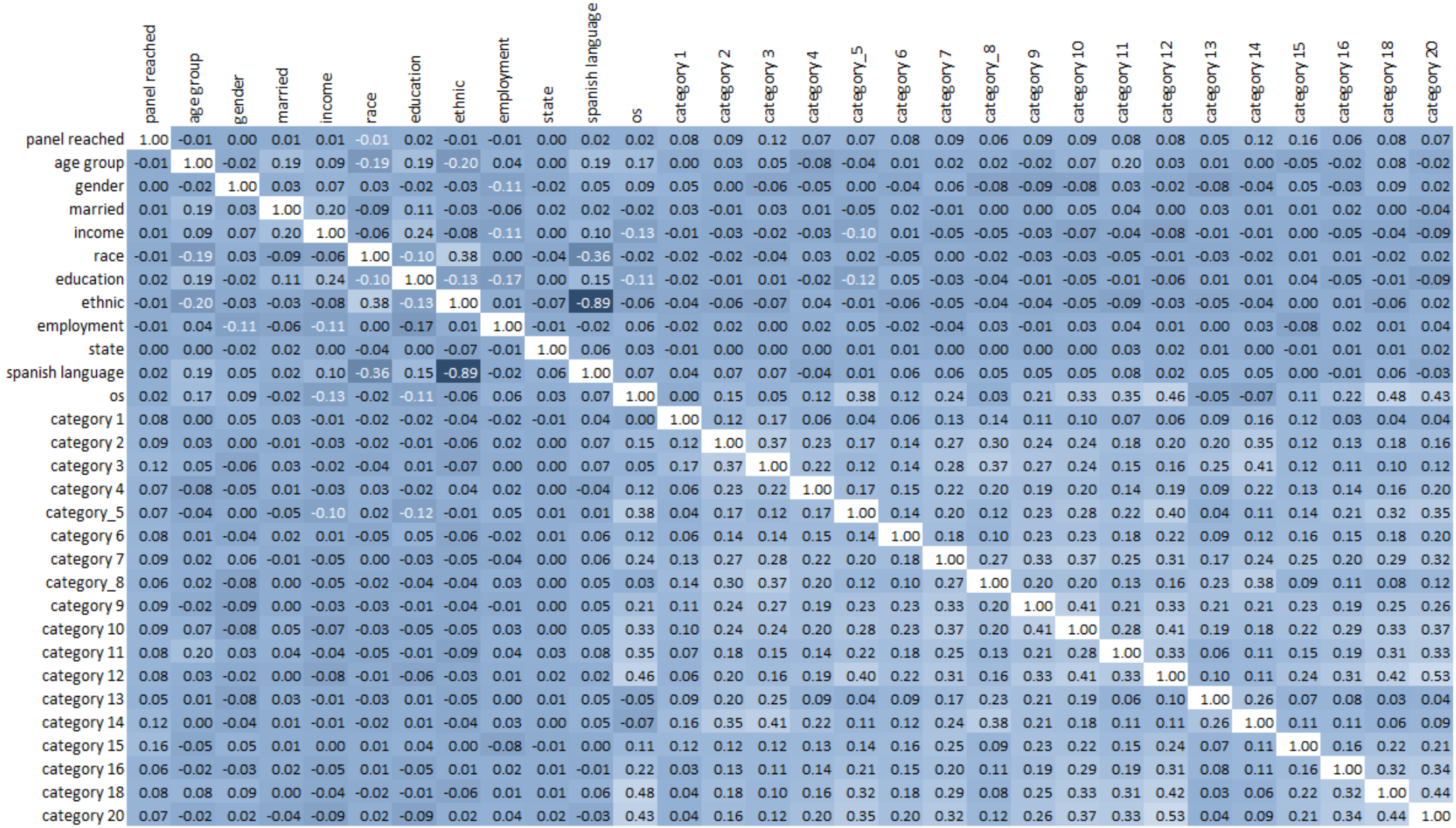


Figure 17: Correlation for mobile variables

Desktop

Table 6: Summary Statistics of the explanatory variables for desktop

Feature	Mean	St. Dev.	Median	Min.	Max.	25%	75%
Age Group	6.18	2.94	7	1	10	4	9
Category 1	2.39	30.98	0	0	1629	0	0
Category 10	116.13	461.16	0	0	5050	0	0
Category 11	353.41	998.81	0	0	5532	0	106
Category 12	3356.52	6696.49	173	0	30843	0	2688
Category 13	1.05	10.11	0	0	528	0	0
Category 14	5.3	44.82	0	0	1321	0	0
Category 15	1.29	16.6	0	0	1256	0	0
Category 2	47.4	226.65	0	0	1976	0	0
Category 4	53.42	323.96	0	0	5405.5	0	0
Category 5	15841.58	30318.08	1307	0	147579	2	12725
Category 6	60.92	277.57	0	0	2316	0	0
Category 7	66.48	281.5	0	0	1969	0	0
Category 8	2.51	26.73	0	0	1517	0	0
Category 9	80.99	327.14	0	0	2502	0	0
Education	3.72	1.41	3	0	6	3	5
Gender	0.66	0.47	1	0	1	0	1
Head of Household	0.81	0.39	1	0	1	1	1
Income	3.72	1.47	3	0	7	3	5
Hispanic	0.1	0.31	0	0	1	0	0
Lifestage	4.43	2.6	5	0	8	3	7
Members 12 17 count	0.2	0.53	0	0	4	0	0
Members 2 11 count	0.29	0.68	0	0	5	0	0
Occupation	13.18	10.25	17	1	27	2	23
Race	2.46	0.99	2	0	6	2	3
Spanish language	4.76	0.8	5	0	5	5	5
Working Status	5.92	2.42	5	1	10	4	9

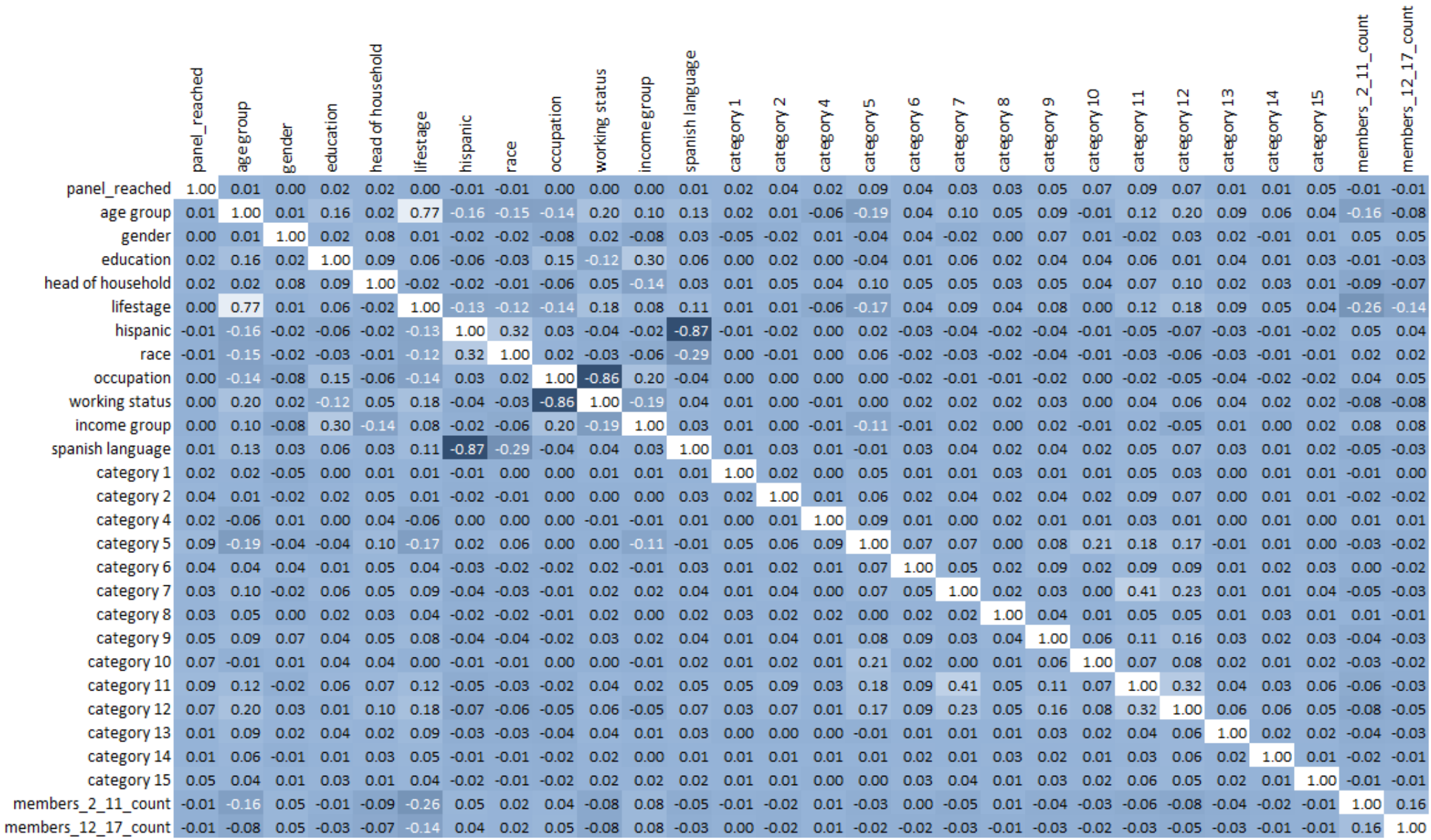


Figure 18: Correlation for desktop variables

B Results

B.1 Overall Performances

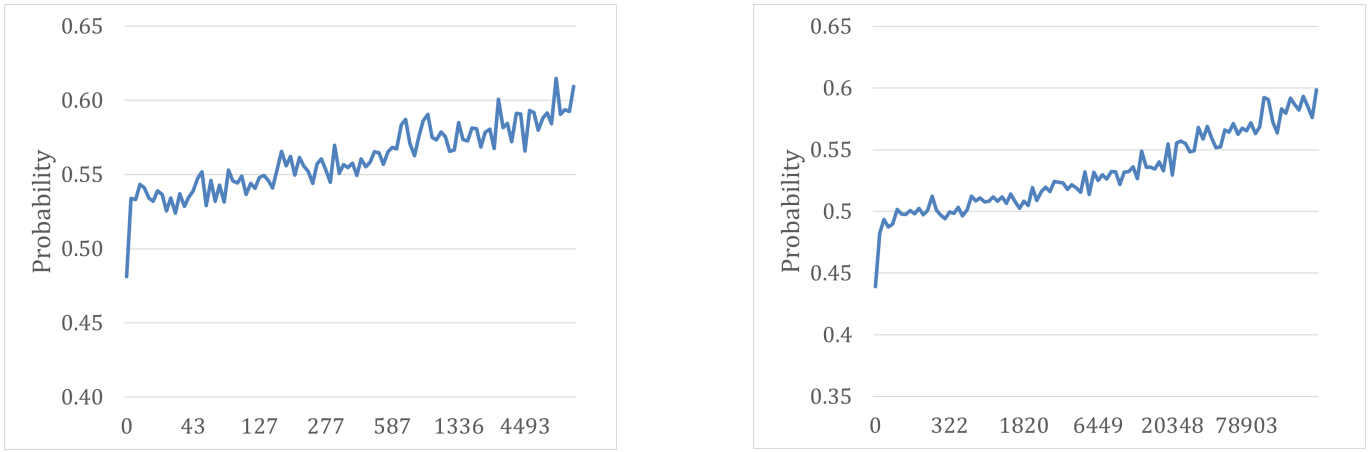
Table 7: In-sample performance on balanced data

	RMSE		Binary Logloss		Balanced Accuracy		Threshold	
	Desktop	Mobile	Desktop	Mobile	Desktop	Mobile	Desktop	Mobile
EBM	0.418	0.420	0.528	0.524	0.749	0.740	0.55	0.45
ERF (Gini)	0.437	0.425	0.563	0.535	0.772	0.758	0.62	0.31
ERF (Entropy)	0.437	0.425	0.563	0.535	0.772	0.758	0.63	0.33
LightGBM	0.373	0.362	0.426	0.412	0.793	0.817	0.61	0.49
Random Forest	0.147	0.153	0.131	0.144	1.000	1.000	0.63	0.61
Gradient Boost	0.427	0.434	0.551	0.561	0.745	0.744	0.56	0.44

Table 8: Out-of-sample performance on balanced data

	RMSE		Binary Logloss		Balanced Accuracy	
	Desktop	Mobile	Desktop	Mobile	Desktop	Mobile
EBM	0.420	0.422	0.532	0.529	0.743	0.733
ERF (Gini)	0.438	0.430	0.566	0.545	0.758	0.738
ERF (Entropy)	0.438	0.430	0.566	0.546	0.758	0.739
LightGBM	0.400	0.420	0.480	0.522	0.752	0.729
Random Forest	0.404	0.419	0.489	0.520	0.745	0.679
Gradient Boost	0.429	0.438	0.554	0.569	0.738	0.734

B.2 Feature Functions



(a) ERF (Entropy): Category 11

(b) ERF (Entropy): Category 5

Figure 19: Feature functions of ERF (Entropy) for Category 5 and Category 11