Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis: BSc^2 in Econometrics and Economics

Addressing Class Imbalance in Non-Profit Donor Response Prediction Using Random Forest Quantile Classifiers: A Comparative Analysis

Kunal Rupchandani (571425)

Supervisor:	Dr. K. Gruber
Second assessor:	Dr. P.C. Schoonees
Date final version:	July 1, 2024

Abstract

Non-profit organisations frequently engage in fundraising by reaching out to potential donors through mailing campaigns. To maximise donations, they aim to predict donor responses accurately. However, the inherent class imbalance between donors and non-donors often leads to poor classification performance. Standard Machine Learning techniques fail to incorporate the class imbalance and hence are deemed unsuitable for the prediction of donor responses. By introducing donor response prediction through the lens of class imbalance, this study extends a quantile framework using Random Forest Quantile (RFQ) Classifiers. The classification performance of RFQ Classifiers is compared with XGBoost, AdaCost, and SMOTEBoost, all designed to handle class imbalance. The research focuses on predicting donor responses to a direct mailing campaign by the Paralyzed Veterans of America (PVA). Results show that RFQ Classifiers and XGBoost achieved the highest classification performance, as indicated by the G-mean and novel Positive-Weighted G-mean metric. The study also evaluates the potential improvements offered by SMOTE for the best-performing models, highlighting its limitations in addressing class overlap.

1 Introduction

Non-profit organisations (NPOs) represent a crucial sector of the economy that focuses on society's well-being. To support their activities, these organisations frequently engage in fundraising, which is considered a systematic venture primarily aimed at raising financial resources (Hommerová and Severová, 2019). One approach to fundraising involves reaching out to potential donors directly for monetary donations through direct marketing. NPOs rely heavily on these direct marketing campaigns for fundraising (Faulk et al., 2021). Moreover, given the limited resources available to these NPOs (Bromideh, 2011), it is critical to target the campaigns to the right selection of potential donors who are most likely to respond with a monetary donation, referred to as "donor response."

A common approach used to "predict" donor response by many organizations is an ad-hoc analysis of past marketing campaigns (Cacciarelli and Boresta, 2022), where they exclude potential donors from the next campaign who do not seem to respond with a donation. However, the advent of Machine Learning techniques to predict donor behaviour and more particularly, to predict if a donor is likely to donate in response to a marketing campaign has led to extensive research in current literature (Jones and Posnett, 1991), (Schetgen et al., 2021), (Farrokhvar et al., 2018).

Although Machine Learning techniques offer robust methods to classify which donors are likely to donate and which are not, a significant challenge lies in the low response rate of potential donors to marketing campaigns, creating a class imbalance problem (Cacciarelli and Boresta, 2022). This imbalance in campaign data between donors who respond with a donation (the "minority" class) and those who do not (the "majority" class) leads to skewed predictions favouring the majority class (Ramyachitra and Manikandan, 2014). Consequently, standard machine learning algorithms often struggle with class imbalances (Guo et al., 2008), resulting in higher misclassification rates for the minority class. This creates difficulties for NPOs in predicting donor response and may lead to the inadvertent exclusion of potential donors from future campaigns.

Approaches to tackle class imbalance can be broadly categorised into either preprocessing or cost-sensitive learning. Data preprocessing techniques such as the Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002), where the number of minority classes in the input dataset is synthetically increased, have been widely popular in tackling class imbalance. Preprocessing techniques such as these are a common choice due to the ease of implementation as compared to cost-sensitive learning techniques (Haixiang et al., 2017). However, SMOTE has various limitations, such as the inability to reflect the true distribution

of the minority class (Sakho et al., 2024), which leads to observations that result in unnecessary noise for the learning model.

Pre-processing and cost-sensitive learning have been widely integrated into classification algorithms to enhance model performance in imbalanced datasets. For example, SMOTEBoost combines SMOTE and Boosting, an ensemble learning technique that aggregates predictions from multiple models, or "learners" to improve overall performance (Schapire, 1990). Other ensemble techniques such as AdaCost (Fan et al., 1999) and Extreme Gradient Boosting (Chen and Guestrin, 2016)have also emerged as a popular solution to the class imbalance problem (Haixiang et al., 2017). The Random Forest (RF) framework introduced by Breiman (2001) is another ensemble technique that has been extremely successful for classification and regression problems(Biau and Scornet, 2016). In the context of regression, Meinshausen and Ridgeway (2006) have extended the RF framework to estimate conditional quantiles of the target variable through Quantile Regression Forests (QRF). QRF has been successfully applied in various domains for regression prediction problems (Völz et al., 2016), (Khan et al., 2019). In contrast, there seems to be a gap in applying a similar quantile framework for classification problems. O'Brien and Ishwaran (2019) address this gap by introducing the novel method of random forest quantile (RFQ) classifiers, especially in the efforts of solving the class imbalance problem for the binary class and multi-class classification problems. RFQ classifiers were built by introducing q^*-classifiers, which differ from the standard classification rules used by other learning techniques.

Therefore, this research aims to contribute to the literature by further enhancing the use of quantile-based methods in classification problems, with a particular focus on the RF framework. Moreover, while several methods have been introduced to tackle the class imbalance problem and applied across a wide range of application domains, the study of class imbalance in predicting donors' behaviour is relatively minimal. Hence another aim would also be to contribute to the literature by examining NPO donation behaviour through the perspective of class imbalance.

The research will be carried out by predicting donor response to a direct mailing campaign sent to existing donors who have donated at least once to the non-profit organisation Paralyzed Veterans of America (PVA) (Hettich and Bay, 1999), comparing the classification performance of different learning techniques. The performance of RFQ classifiers and other techniques such as SMOTEBoost, AdaCost and Extreme Gradient Boosting (XGBoost) will be compared against each other resulting in the following research question:

How does the classification performance of Random Forest Quantile Classifiers compare to other learning techniques for class imbalance in the context of classification of donor response?

The performance of these models will primarily be compared based on the G-mean metric, which is frequently adopted in class imbalance research (He and Garcia, 2009). However, other metrics such as the novel Positive-Weighted G-mean and the Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) score will also be employed. Therefore, a sub-question this research will investigate is:

How does the G-mean performance of Random Forest Quantile Classifiers compare to other learning techniques for class imbalance, in the context of classification of donor response?

The research finds that RFQ classifiers and Gradient Boosting methods seem to perform optimally in terms of the G-mean. However, the performance of SMOTE as a preprocessing step before implementing ensemble techniques will also be evaluated against the baseline of no preprocessing steps before making prediction from the techniques. This leads to the following sub-question:

Does SMOTE as a preprocessing step improve the classification of the minority class as compared to no preprocessing in an imbalanced dataset?

The remainder of this paper is organised as follows: Section 2 reviews the relevant literature, and Section 3 details the dataset, preprocessing and variable selection. Section 4 describes the methodology employed. The corresponding results are presented in Section 5. Finally, Section 6 provides concluding remarks on the research, evaluates the limitations of the study and recommends future suggestions for further research.

2 Literature Review

Modelling donor response in NPOs has been widely explored in the literature, primarily focusing on identifying characteristics that signal a potential donor's ability and willingness to donate. The research so far has aimed to answer two key questions: to whom should an NPO send a mailing, and who is likely to respond to it (Jonker et al., 2000)? Various characteristics can indicate donor response, but the most critical variables are captured by the Recency, Frequency, and Monetary (RFM) framework that can be applied to donors of an NPO. RFM metrics can measure the time since the last donation (Recency), the number of donations within a certain period (Frequency), and the amount of money donated (Monetary), respectively (Kaymak, 2001). Studies have demonstrated that RFM variables are crucial for selecting potential donors likely to respond with a donation (Bult, 1993). Additionally, demographic factors like household income and age are also considered important predictors of donor response (Jones and

Posnett, 1991). Based on these important predictors, research has utilised Machine Learning to forecast donor responses. For instance, Farrokhvar et al. (2018) compared Artificial Neural Networks (ANN) and Support Vector Regression (SVR) models to predict charitable giving using household income, education, and gender. Additionally, Sousa et al. (2003) employed RFM variables to develop a target selection of donors for a direct mailing campaign of an NPO, by implementing classical statistical techniques and Machine Learning.

In general, previous research does not explicitly address the class imbalance challenge in NPO datasets. However a recent study by Cacciarelli and Boresta (2022), consider the case of the NPO, World Wide Fund for Nature (WWF), where the response of donors was predicted across 16 direct marketing campaigns with a heavy imbalance such that the minority class only represented 3% of the total dataset.

The topic of class imbalance has been widely investigated in different forms. More particularly, other terms such as rare, anomaly, and abnormal are synonymous with imbalance amongst research in different domains. Haixiang et al. (2017) established at least 527 research articles related to imbalanced data and rare events, with 162 application papers across 13 domains, suggesting significant research interest in this topic. The domains where the concept of imbalance is researched vary from information technology to agriculture.

Approaches to tackling imbalanced datasets can be categorised into 2 strategies: preprocessing and cost-sensitive learning. Preprocessing involves performing steps that modify the input data and resampling the dataset before implementing a learning model. The resampling of the dataset aims to rebalance the dataset to reduce the effects of a skewed class distribution in the learning process. Resampling usually falls under the three categories of under-sampling, over-sampling, and a combination of both. Under-sampling involves manually removing observations of the majority class to reduce the skewness of the data distribution. The most common under-sampling method is Random Under-Sampling (RUS), which randomly removes observations that are labelled as the majority class. However, under-sampling has the issue of producing warped posterior probabilities (Dal Pozzolo et al., 2015). In contrast, over-sampling aims to create new observations of the minority class. A popular method of over-sampling is the Synthetic Minority Oversampling Technique (SMOTE), introduced by Chawla et al. (2002). SMOTE creates synthetic samples with a minority class that are defined by interpolation among neighbouring minority class instances (Fernández et al., 2018). The original paper introducing SMOTE written by Chawla et al. (2002) has 30,008 citations of SMOTE as of June 2024. This popularity of SMOTE stems from its straightforward interpretation and wide applicability in each domain that has to tackle the issue of class imbalance. However, SMOTE has quite a few

limitations. Most importantly, SMOTE generates synthetic samples that may not be necessarily informative and result in noise in the dataset, since the method fails to reflect the true distribution of the minority class (Sakho et al., 2024). Other disadvantages of SMOTE also include blindness of neighbour selection and sample overlapping (Jiang et al., 2021).

Additionally, cost-sensitive learning has also emerged as a promising solution to the issue of imbalanced datasets. However, according to Haixiang et al. (2017), the cost-sensitive approach is less popular than re-sampling methods. The cost-sensitive approach involves assigning different costs to the misclassification of minority and majority classes, which are typically assumed to be equal in standard classification settings. One disadvantage of cost-sensitive learning is that the specific misclassification costs are often unknown and require domain expertise to accurately estimate. However, research such as Castro and Braga (2013) and Lan et al. (2009) suggest a solution to this problem by assigning the cost of misclassifying the minority class equal to the imbalance ratio (IR) of the dataset and the cost of misclassifying the majority class equal to 1. The IR is defined as the total number of observations of the majority class divided by the number of observations of the minority class in a dataset. Another reason why cost-sensitive learning is not as widely implemented in research is the lack of machine learning expertise that is usually required to modify the learning algorithm to incorporate the different costs assigned to misclassification. Despite the drawbacks of cost-sensitive learning, the approach is usually more efficient and seems to outperform resampling methods in imbalanced data scenarios according to Pes and Lai (2021).

Resampling and cost-sensitive learning have been widely integrated with classification models that deal with imbalanced learning and datasets. These classification models can be split into the two categories of ensemble methods and algorithmic classifier modifications. The latter method involves changing existing classification models to cater to the problem of imbalanced datasets. According to Haixiang et al. (2017), 160 novel techniques based on classification techniques such as decision tree classifiers (Breiman, 2017), Neural Networks (NN), rule-based classifiers, and Näive Bayes have been modified to cater to imbalanced learning.

Ensemble classifier techniques seem to be even more popular among imbalance learning approaches Haixiang et al., 2017. Ensemble techniques combine outputs of multiple base classifiers that provide better predictive performance than a single base classifier. The idea of using ensemble techniques stems from the human nature of seeking several opinions before making a decision (Galar et al., 2011). The combination of outputs from multiple classifiers usually leads to higher predictive performance (Maclin and Opitz, 1997). However, this usually comes at a cost of increased computational complexity (Yang et al., 2010). Ensemble techniques can be

categorised into iterative and parallel-based ensembles. The latter refers to models where the base classifiers can be trained at the same time i.e. parallel. Whereas, iterative-based techniques train base classifiers, one at a time, where the next classifier draws attention to the errors made by previous ones.

Boosting as introduced by Schapire (1990), is an iterative-based ensemble technique to improve the accuracy of any given learning algorithm using a Probably Approximately Correct (PAC) learning framework. Through this technique, a set of weak learners (models which perform slightly better than random guessing) can be converted to a strong learner. AdaBoost (Freund and Schapire, 1997) was the first practical approach to boosting and has since been established as one of the top 10 data mining algorithms (Wu et al., 2008). AdaBoost utilises the complete dataset across each classifier but puts more weight on observations that were difficult to predict in the next iterations. Other popular boosting algorithms include the Gradient Boosting Machine (Friedman, 2001), which builds a model in a stage-wise fashion and optimises the loss function by iteratively adding weak learners to correct the errors of the ensemble. Extreme Gradient Boosting (XGBoost) provides a scalable and efficient method to implement gradient boosting (Chen and Guestrin, 2016). XGBoost was implemented by Priscilla and Prabha (2020) to predict credit card fraud detection, which deals with heavy class imbalance.

Several variants of the Boosting framework have been introduced that are specifically designed to tackle the imbalance of datasets. Most of these variants fall under the category of cost-sensitive boosting. These include examples such AdaCost (Fan et al., 1999), AdaC1, AdaC2 and AdaC3 (Sun et al., 2007). These variants modify the AdaBoost algorithm to handle class imbalance by incorporating cost-sensitive learning principles.

Moreover, techniques such as SMOTEBoost (Chawla et al., 2003) implement SMOTE within a boosting algorithm to address class imbalance. Cui et al. (2014) showed the effectiveness of SMOTEBoost in improving the diagnosis of power transformer insulation.

3 Data

The dataset utilised for the research is extracted from a public-domain repository[1] that contains the dataset for The Second International Knowledge Discovery and Data Mining Tools Competition (Hettich and Bay, 1999). The dataset was provided by the Paralyzed Veterans of America (PVA), which is an NPO that provides services and programs for US veterans with spinal cord injuries or diseases. PVA's in-house database consists of 13 million donors, making PVA the largest direct mailing fundraiser in the country (Hettich and Bay, 1999).

[1] https://kdd.ics.uci.edu/databases/kddcup98/kddcup98.html

The particular dataset used pertains to a mailing list sent to their donors, as part of PVA's June 1997 renewal mailing list ("97NK" mailing list). It contains 24 months of detailed PVA promotion and giving history (covering the period 12 to 36 months before this mailing) of past donors. The donors in this mailing list have donated at least once in the past, but they have been considered as lapsed donors, that made their last donation to PVA 13 to 24 months ago.

Hence, PVA would aim to recapture as many lapsed donors as possible. This requires the right identification of the donors who are most likely to donate. However, as mentioned this PVA dataset faces the problem of class imbalance for its response binary variable TARGET_B as shown in Table 1.

Table 1: Distribution of TARGET_B response variable of PVA raw dataset.

TARGET_B	**Frequency**	**Percent**
0	90569	94.9
1	4843	5.1

Table 1 displays the stark imbalance between the two classes of TARGET_B. More particularly, only 5.1 % of donors included in the mailing list responded with a donation. Therefore, we define class 1 as the minority class and class 0 as the majority class in this research. More formally, we can define this imbalance using the imbalance ratio (IR) (Ali et al., 2019) which is defined as $IR = \frac{N_0}{N_1}$, where N_0 and N_1 correspond to the number of observations of class 0 and class 1 respectively. A dataset is considered imbalanced when $IR > 1$. In the raw dataset, the IR is equal to 18.7. Therefore, the "97NK" mailing list can be considered as a highly imbalanced dataset.

The raw dataset contains 479 variables that can be used as features in our learning models. The dataset also includes the numerical counterpart of TARGET_B, TARGET_D, which provides the donation amount (in $) associated with the response of the donor.

A subset of 7 features out of the 479 variables were selected for the prediction of donor response and are summarised in Table 2, which includes the name of the feature and a short description of it.

These variables are selected since they provide the most information about the donor's past donation behaviour. More particularly, LASTDATE, TIMELAG, and AVGGIFT are variables that could be considered a good summary for the RFM variables which have a positive effect on the selection of donors who will predict (Jonker et al., 2000). Moreover, we also include an income variable which has been shown to have an impact on the responses to direct mailing

Table 2: Description of features from PVA dataset used for the prediction of donor response.

#	Name	Description	Type
1	LASTDATE	Number of days between the last donation and the day of "97NK" mailing	Numerical
2	NGIFTALL	Number of lifetime gifts to date	Numerical
3	TIMELAG	Number of months between first and second gift	Numerical
4	INCOME	Income rating which is based on a scale of 1-7 (7 being the highest rating)	Categorical
5	GENDER	Variable indicating if the donor is Female or Male	Binary
6	LASTGIFT	Dollar amount of most recent gift	Numerical
7	AVGGIFT	Average Dollar amount of gifts to date	Numerical

responses (Jones and Posnett, 1991). Finally, donor response and participation have also been shown to be sensitive to demographic variables such as AGE and GENDER (Jones and Posnett, 1991), which are also included in Table 2.

A few steps of preprocessing were required after the relevant features were extracted. More particularly, the original LASTDATE column in the raw dataset was in the format of "YYMM", where "YY" indicated the year and "MM" indicated the month of the latest donation. This variable was transformed to the number of days between the 1st day of the month of this donation and the 1st day of June 1997, which was the month when the "97NK" mailing was sent. However, the original GENDER column was a categorical variable where certain observations had "U" to indicate unknown gender and other undocumented abbreviations such as "C" and "A". The 'GENDER' variable was then converted to a binary variable where 1 indicates Female and 0 indicates Male. The unknown categories were distributed equally to 1s and 0s as this seemed to be the most suitable method to ensure the original distribution of Females to Males remained the same in the dataset. Finally, all observations with missing values were removed to deliver the final PVA dataset that will be used for the rest of this research. This dataset contains 84802 observations with $N_1 = 2382$ and hence and IR of 18.04, which is slightly less than the raw dataset

The summary statistics of these features are found below in Table 3.

Table 3: Summary statistics of the preprocessed PVA dataset used for this research.

Feature	TARGET_B	NGIFTALL	LASTDATE	AVGGIFT	LASTGIFT	GENDER	INCOME	TIMELAG
Mean	0.052	10.520	556.359	29.860	16.742	0.560	3.864	8.135
Std	0.223	8.382	124.175	39.235	13.188	0.496	1.852	8.753
Min	0.000	1.000	120.000	1.014	0.000	0.000	1.000	0.000
50%	0.000	8.000	548.000	14.714	15.000	1.000	4.000	6.000
Max	1.000	237.000	823.000	996.875	1000.000	1.000	7.000	1088.000

Table 3 shows the mean, standard deviation, median (50%), and minimum and maximum values for each feature. The summary statistics reveal a noticeable difference between the mean and the median for the target variable for certain features. More particularly, for AVGGIFT, the mean value is 29.860, however, the median value is 14.7. This indicates a positive skew in the data, where the distribution is heavily influenced by the minority class.

4 Methodology

The following sub-sections discuss the different techniques to handle class imbalance that are implemented in this research to predict donor response on our PVA dataset. Subsection 4.1 introduced the RF ensemble technique, where we introduce the quantile framework used for QRF and RFQ classifiers. Subsection 4.2 introduces the preprocessing technique of SMOTE and Subsection 4.3 introduces the concept of booking and the methods SMOTEBoost and XGBoost. Finally, Section 4.5 presents the evaluation metrics employed to compare the prediction performance of the different techniques.

4.1 Random Forest

Random Forest (RF) introduced by Breiman (2001) is an ensemble learning technique that combines multiple decision trees to improve prediction performance. In this method, each decision tree serves as a base predictor, which divides the input feature data into subsets based on certain criteria. RF introduces randomness to the decision trees through a process known as Bootstrap Aggregating, or "Bagging" (Breiman, 1996). Bagging involves creating multiple versions of the predictor by generating independent bootstrap samples, which are random subsets of the original data with replacement. Each decision tree is built using one of these bootstrap samples. The final prediction is made by aggregating the predictions from all the individual trees, typically by taking a majority vote or averaging the results. This technique is crucial because a single decision tree may not prove reliable predictions when dealing with complex data with many features. However, combining multiple trees in an ensemble tends to produce more reliable and accurate predictions (Dietterich, 2000).

More concretely, consider n independent observations of the input dataset such that an observed feature vector $X_i \in \mathbb{R}^p$, where p denotes the number of features, exists and let the observed target variable $Y_i \in \mathbb{R}$ exist for all $i = \{1, ..., n\}$ observations. This forms our existing input dataset D. Then a random forest creates randomised and independent bootstrapped samples $D_k \subset D$, where k is the number of trees. In addition to D_k, a random subset m of features from X_i will be considered at each node of the random forest. Once the bootstrapped

sample D_k with m features are picked, various thresholds of each feature are evaluated to determine the best split. The split is chosen based on a splitting criteria which differs depending on whether a regression or classification task is at hand.

To look closer, let us consider the prediction of a single tree for a new data point $X = x$, where $x \in \mathbb{R}^p$ and X is a random feature vector and $Y \in \mathbb{R}$ is the corresponding random target variable which is continuous. Hence, we consider a regression task to explain the prediction. For the new data point x, each tree in the forest determines which leaf x falls into. For each training observation Y_i, the contribution to the prediction of x is defined by a weight $w_i(x, \theta)$, where $i = 1, ..., n$ and θ is a random parameter vector that defines how a tree is grown. The prediction of a single tree for a new data point x is then the weighted average of the observations Y_i given by:

$$\text{single tree:} \quad \hat{\mu}_t(x) = \sum_{i=1}^{n} w_i(x, \theta) Y_i. \tag{1}$$

Then, by averaging the predictions of k single trees, each constructed using a vector θ_t for $t = 1, ..., k$, $w_i(x)$ is the average of $w_i(x, \theta_t)$ across all k trees:

$$w_i(x) = \frac{1}{k} \sum_{t=1}^{k} w_i(x, \theta_t). \tag{2}$$

The prediction of a random forest is then given by:

$$\text{Random Forests:} \ \hat{\mu}_{RF}(x) = \sum_{i=1}^{n} w_i(x) Y_i. \tag{3}$$

The prediction of our Random Forest for our data point $X = x$ and the corresponding random target variable Y is $m(x) = E(Y|X = x)$, which is the conditional mean of Y (Breiman, 2001).

4.1.1 Quantile Regression Forests

Quantile Regression Forests (QRFs) Meinshausen and Ridgeway (2006) take a step further from RF in a regression context. More particularly, QRFs do not only predict the conditional mean $m(x) = E(Y|X = x)$, but also the full conditional distribution function of our random target variable Y. QRFs are based on Quantile Regression (QR), which was introduced by Koenker and Bassett Jr (1978). QR attempts to estimate the α-quantile $Q_\alpha(x)$.

More particularly, the conditional distribution $F(y|X = x)$ is equal to the probability that Y is smaller than or equal to a given value $y \in \mathbb{R}$:

$$F(y \mid X = x) = P(Y \leq y \mid X = x). \tag{4}$$

Then, for a continuous distribution, $Q_\alpha(x)$ is defined such that the probability of Y being less than or equal to $Q_\alpha(x)$ is exactly equal to α when $X = x$. Therefore, we have:

$$P(Y \leq Q_\alpha(x) \mid X = x) = \alpha. \tag{5}$$

Estimating the quantiles becomes important, especially when the conditional mean fails to capture the heterogeneity in the data (Huang et al., 2017), where the effect of X varies across different points of the distribution of Y.

Meinshausen and Ridgeway (2006) extended the definition from (4) to the expectation of the indicator function for $y \leq Y$ when $X = x$:

$$F(y \mid X = x) = P(Y \leq y \mid X = x) = E(\mathbb{1}_{\{Y \leq y\}} \mid X = x). \tag{6}$$

This definition of the conditional distribution is analogous to the definition of the expectation derived for the prediction of a single tree in $\hat{\mu}_t$ from (1). Specifically, $E(Y \mid X = x)$ is estimated as the weighted mean over the observations of Y. Similarly, $E(\mathbb{1}_{\{Y \leq y\}} \mid X = x)$ represents the conditional cumulative distribution function $F(y \mid X = x)$, which is the probability that Y is less than or equal to y given $X = x$. This expectation is ordered in the sense that as y increases, $F(y \mid X = x)$ increases, reflecting the cumulative nature of the distribution function. This can be approximated using the weighted mean over the observations of the indicator function $\mathbb{1}_{\{Y \leq y\}}$. Therefore, the approximation of $E(\mathbb{1}_{\{Y \leq y\}} \mid X = x)$ using the weights $w_i(x)$ from (2) in the random forest gives us the following formula:

$$\hat{F}(y \mid X = x) = \sum_{i=1}^{n} w_i(x) \mathbb{1}_{\{Y_i \leq y\}}. \tag{7}$$

4.1.2 Random Forest Quantile Classifiers

The quantile framework for RF in a regression context explained in Subsection 4.1.1 has been thoroughly established through QRFs. RFQ classifiers introduced by O'Brien and Ishwaran (2019) extend the RF framework to classification, which is relevant to predicting donor response in our dataset. RFQ classifiers aim to mitigate class imbalance by using a quantile threshold tailored to the distribution of the minority class, thereby reducing classification errors for underrepresented classes. This was done by extending the concept of a quantile classifier (q-classifier) introduced in Mease et al. (2007).

Having a closer look at the q-classifier, let Y be a random target variable such that $Y \in \{0, 1\}$ and let X be a random feature vector such that $X \in \mathbb{R}^p$, where p is the number of features. Moreover, let x be a data point. A standard RF classification utilises the Bayes decision rule

(O'Brien and Ishwaran, 2019) to classify observations into either class label 1 or 0. Assuming class label 1 is our minority class, let $p(x) = \mathbb{P}(Y = 1 \mid X = x)$ be the classification probability for the minority class. The Bayes decision rule then classifies observations to the minority class when $p(x)$ has a probability of 1/2 or larger:

$$\delta_B(x) = \mathbb{1}\{p(x) \geq 1/2\}, \quad 0 < q < 1,$$

where $\mathbb{1}(p(x) \geq q)$ is an indicator function, which returns either a 1 or 0. The threshold of $q = 1/2$ is derived from the goal of minimising the probability of misclassification (Hastie et al., 2009). However, an assumption made when setting this threshold is that the cost of misclassifying class label 1 (minority class) as class label 0 (majority class) is the same as misclassifying class label 0 as label 1. This is problematic when $p(x)$ is small in the case of imbalanced classes. Namely, this leads to $\delta_B(x)$ classifying most observations as the majority class, leading to higher misclassification errors for the minority class label 1.

Therefore, instead of considering a threshold of a 1/2, a q-classifier can be defined as

$$\delta_q(x) = \mathbb{1}\{p(\mathbf{x}) \geq q\}, \quad 0 < q < 1, \tag{8}$$

where q is now our specified quantile. In the case of class imbalance, q would ideally be set at a lower level than a 1/2 since $p(x)$ is also a lot smaller. However, the complication of setting the optimal q arises.Hence, to circumvent this issue O'Brien and Ishwaran (2019) introduce a q^*-classifier using a density-based classifier. More particularly, they introduced the following classifier:

$$\delta_D(x) = \mathbb{1}\left\{\frac{f_{X|Y}(x|1)}{f_{X|Y}(x|0)} \geq 1\right\}, \tag{9}$$

where $f_{X|Y}(x|1)$ refers to the probability density function (pdf) of the random feature vector X given that $Y = 1$. Similarly, $f_{X|Y}(x|0)$ is the pdf of X given $Y = 0$. Therefore, when the ratio $\frac{f_{X|Y}(x|1)}{f_{X|Y}(x|0)}$ is more than 1, it means that x is more likely to come from the distribution of features for class 1.

Utilising the ratio of conditional densities ensures that the relative likelihoods of the feature distributions are considered instead of $p(x)$ which does not consider the prevalence of the minority class in the dataset. Furthermore, $\delta_D(x)$ can be re-written using the Bayes' Theorem:

$$\begin{aligned}
\delta_D(x) &= \mathbb{1}\left\{\frac{f_{X|Y}(x|1)}{f_{X|Y}(x|0)} \geq 1\right\} \\
&= \mathbb{1}\{\Delta_D(x) \geq 1\}, \\
&= \mathbb{1}\{p(x) \geq \pi\},
\end{aligned}$$

where $\Delta_D(x) = \frac{p(x)(1-\pi)}{(1-p(x))\pi}$ and $\pi = \mathbb{P}\{Y = 1\}$ which is the unconditional probability of observing class label 1, i.e. $\mathbb{P}\{Y = 1\} = \frac{N_1}{N_0+N_1}$, where N_1 refers to the number of observations with the class label of 1 (minority class) an N_0 the number of observations with the class label of 0.

Therefore, we can conclude that $\delta_D(x)$ here is actually a q-classifier with $q = \pi = \mathbb{P}\{Y = 1\}$. RFQ classifiers therefore implement the density-based q^*-classifier as the classifier rule instead of the standard $\delta_B(x)$ in RF classification.

4.2 Synthetic Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE), as introduced by Chawla et al. (2002), is an oversampling technique that is widely implemented in various imbalance class problems through data pre-processing. As opposed to random over-sampling of observations where existing observations are essentially copied to form balanced data, SMOTE facilitates the creation of synthetic observations of the minority class. It does so by the concept of interpolation amongst existing minority class observations in a defined neighborhood of the dataset's feature space. More particularly, after deciding the amount of oversampling observations N required ($N = 100$ for a 1:1 ratio amongst classes), a random sample from the minority class is selected, and then K nearest neighbours are obtained. This process is executed iteratively for each minority sample. From the K samples, N are chosen randomly to compute synthetic instances. The process is summarised in the following algorithm:

4.3 Boosting

Boosting as introduced by Schapire (1990), is an iterative-based ensemble technique to improve the accuracy of any given learning algorithm using a Probably Approximately Correct (PAC) learning framework (Galar et al., 2011). Through boosting, a set of weak learners (which are slightly better than random guessing) can be converted to a strong learner.

More specifically, the process of boosting starts by assigning equal weights to each observation n in a dataset such that $n = 1, ..., N$ where N is the total number of observations of a dataset. After each iteration of a boosting algorithm, the weights of misclassified observations are changed to provide more emphasis on the next series of weak learners. Therefore, by sequentially applying weak classifiers and reweighting the observations in a dataset, classification performance improves. AdaBoost is the most representative algorithm of Boosting (Galar et al., 2011).

Algorithm 1 SMOTE algorithm

function SMOTE(T, N, k)
  Input: T, N, k ▷ #minority class examples, oversampling, #neighbors
  Output: $(N/100) * T$ synthetic minority class samples
  Vars: `Sample[]` (original samples); `newindex` (count of synthetic samples); `Synthetic[]` (synthetic samples)
  if $N < 100$ **then**
    Randomize T samples
    $T = (N/100) * T$
    $N = 100$
  end if
  $N = (\text{int})N/100$ ▷ SMOTE amount is in integral multiples of 100
  for $i = 1$ to T **do**
    Compute k nearest neighbors for i, save indices in `nnarray`
    `POPULATE(`$N, i,$ `nnarray)`
  end for
end function

4.3.1 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a scalable and highly efficient implementation of the Gradient Boosting method, which was introduced by Chen and Guestrin (2016). Gradient Boosting methods in general use the iterative procedure of Boosting. More particularly, an aggregate classifier in Gradient Boosting can be defined as follows. Consider a data point $x \in \mathbb{R}^p$, where p refers to the number of features. Moreover, let the iteration (corresponding to a base classifier) of the boosting algorithm be $t = 1, ..., T$, where T is the total number of iterations. Then, the prediction of an aggregate classifier can be defined as $g_t(x)$. Additionally, let a loss function $L(y_i, g_t(x))$ be defined such that $i = 1, ..., N$ corresponds to observation i in the dataset. Then, this loss function penalises the difference between the actual observation y_i and the prediction from the aggregate classifier $g_t(x)$. Gradient Boosting aims to minimise the overall loss function through the iterative procedure. For every iteration, each weak learner attempts to reduce the residual loss of the weak learner in the previous iteration.

4.3.2 SMOTEBoost

SMOTEBoost is a hybrid ensemble technique that combines SMOTE and the AdaBoost algorithm. It was specifically introduced by Chawla et al. (2003) to tackle imbalanced classes. It iteratively applies the SMOTE Algorithm 1, creating synthetic samples for the minority class, followed by training a base classifier on the resampled dataset. This process is repeated for several iterations, with each base classifier focusing on the error of the previous one.

4.4 Cost-Sensitive Learning

Cost-sensitive learning aims to explicitly attach different costs to the misclassification of classes. More particularly, higher costs can be attached to misclassifying the minority class in comparison to the misclassification of the majority class. This is usually done by specifying a cost matrix C_{ij} which represents the misclassification cost of assigning an observation that belongs to class i to class j. Cost-sensitive learning can be implemented at the data level (during data preprocessing) or the algorithmic level. We implement an AdaCost method, which works at the algorithmic level.

4.4.1 AdaCost

AdaCost modifies the boosting process by incorporating the matrix C_{ij} to adjust the weights of misclassified instances. In each boosting iteration, misclassifications of the minority class are penalised more heavily, increasing their weights and forcing the model to focus more on correctly predicting these instances. This approach effectively adjusts the decision threshold, typically set at a value of 1/2 (akin to the Bayes decision rule $\delta_B(x)$). The adjusted boundary lowers the threshold for the classification of minority classes, enhancing the classification performance in imbalanced datasets.

4.5 Evaluation Metrics

The three types of evaluation metrics used to assess the performance of learning techniques can be categorised into threshold metrics, ranking methods and metrics, and probabilistic metrics (Japkowicz, 2013). This research focuses on threshold metrics as they provide the most effective means of evaluating how our methods perform in making predictions given the specific class imbalance present in the dataset.

These metrics are derived from the fundamental confusion matrix shown in Table 4, which accounts for the number of classification errors made by the prediction models:

Table 4: Confusion Matrix for TARGET_B classification

	Predicted positive	**Predicted negative**
Actual positive	True positive (TP)	False negative (FN)
Actual negative	False positive (FP)	True negative (TN)

Table 4 provides a summary of the classification output of our learning techniques. To reiterate, the positive class corresponds to TARGET_B = 1, which implies that a donor responded with a donation, and the negative class, is otherwise.

Evaluating the performance of these methods will differ from standard metrics used in machine learning classification problems. More specifically, metrics such as accuracy fail to capture the real impact of minority class observations. Accuracy, which represents the proportion of correct predictions, does not discriminate between successful predictions of the majority and minority classes. Therefore, the successful predictions of the majority class (due to the large number of observations) overshadow the poor performance of the predictions of the minority class.

4.5.1 True Positive Rate and True Negative Rate

The True Positive Rate (TPR), or sensitivity, corresponds to the proportion of positive predictions that were correctly assigned as being positive. On the other hand, the True Negative Rate (TNR), or specificity, corresponds to the proportion of negative predictions that were correctly classified as negative:

$$\text{True Positive Rate(TPR)} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{True Negative Rate(TNR)} = \frac{TN}{TN + FP}. \tag{11}$$

Unlike accuracy, these two metrics separately assess the successful predictions of each class. This is useful in the direct mailing dataset since we are most interested in the positive responses (TARGET_B = 1) that the model presents. The goal of any prediction in the context of mail campaigns would be to maximise the amount of correctly predicted positive responses. Hence, the TPR is useful to capture the effect.

4.5.2 Geometric Mean

Geometric Mean (G-mean) introduced by Kubat et al. (1998) specifically targets the problem of class imbalance. More particularly, G-mean aims to provide a balanced metric that assesses

how well a classifier classifies the respective classes:

$$G\text{-mean} = (\text{TNR} \times \text{TPR})^{1/2}. \tag{12}$$

Many learning techniques that focus on the problem of imbalanced datasets, such as RFQ classifiers use the G-mean as the default performance metric. Hence, the primary metric used to compare the different performances in this research will be the G-mean.

4.5.3 Positive Weighted G-mean

The optimal mailing list a non-profit organisation would like to send should maximise the responses they receive. Therefore, a mailing list could consist of all predicted positive responses according to a classifier. This directly corresponds to the precision of a classifier, which assesses the proportion of predicted positives that are actually positive:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{13}$$

To this end, this research proposes an adjusted and weighted G-mean (PWG-mean):

$$PWG\text{-mean} = \left(\text{Precision}^{0.7} \times \text{TPR}^{0.3}\right), \tag{14}$$

where the precision and TPR are considered. Moreover, since a non-profit organisation is more likely to focus on creating an optimal mailing list, the precision in this measure receives a higher weight than the TPR.

4.5.4 Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) score

The Receiver Operating Characteristic - Area Under the Curve (ROC) score provides a single value that summarises a model's ability to distinguish between the majority and minority classes. The score ranges between 0 and 1, with higher values indicating better classification performance. Unlike the threshold metrics G-mean and PWG-mean, the ROC-AUC score evaluates the classifier's performance across all possible decision thresholds.

5 Results

The following section is organised as follows. Firstly, in Subsection 5.1 we implement QRF on our PVA dataset to showcase the quantile framework in a regression context. In subsection 5.2 we showcase how RFQ Classifiers implement the q^*-classifier in the PVA dataset and how this

leads to G-mean optimality. In Subsection 5.3 we provide a comparative analysis of our RFQ results and compare it to XGBoost, SMOTEBoost and AdaCost. Finally, in Subsection 5.4, we implement SMOTE as a preprocessing step on the best-performing models and compare it to the baseline of no SMOTE preprocessing.

5.1 QRF

Ideally, an NPO would like to predict the monetary value of donations it receives from the donors of a particular mailing campaign. However, given the uncertainty of these predictions, a confidence interval would be more useful for the organisation. Therefore, QRFs provide an ideal way to produce the predicted response at different quantiles of the response variable. Given the dataset from Section 3, PVA can analyse what are the amounts of donation it can expect to receive based on a confidence interval. Using TARGET_D as the response variable and the features mentioned in Section 3, Figure 1 is produced. More particularly, the quantiles considered are $\alpha = \{0.025, 0.500, 0.975\}$, which corresponds to building a 95% confidence interval of QRF predictions. The QRF model is implemented for all values in the PVA dataset, however, Figure 1 only showcases a sample of 5000 observations to better visualise the predictions.

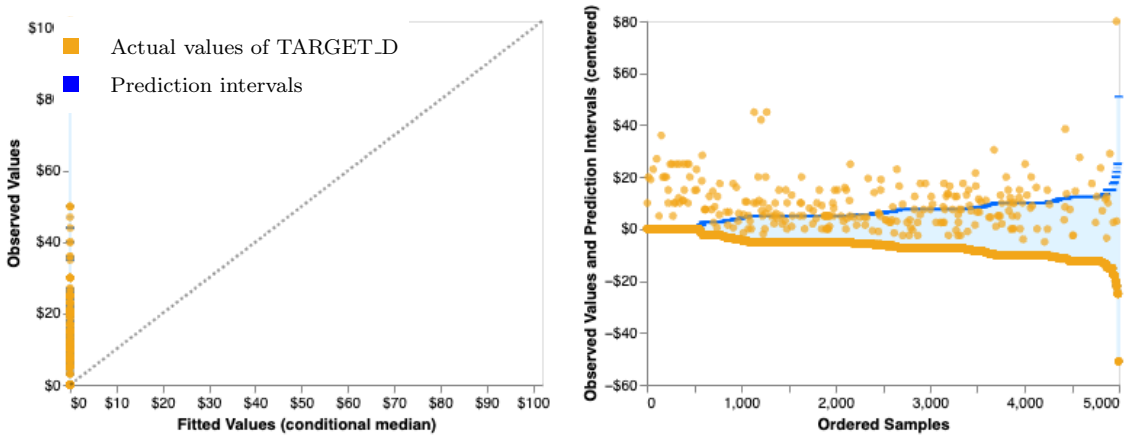

Figure 1: Estimated conditional quantiles for a sample of n=5000 observations from the PVA dataset. Left panel: Observed values plotted against predicted values. Prediction intervals are displayed by the light blue bars with the dark blue ticks representing the lower and upper bound of the prediction interval. Right panel: The same observations are ordered according to the length of the prediction interval. Note: The legend included corresponds to both panels.

The left panel of Figure 1 illustrates the observed values plotted against the fitted/predicted values. Due to the imbalance between positive and negative classes, the QRF model predicts that all values of TARGET_D in the sample are zero. This is because most donors in the PVA dataset do not donate, resulting in TARGET_D = 0. The right panel, which displays a

95% confidence interval based on quantiles α, shows that most observed values fall outside the predicted confidence interval. This indicates that standard models are ineffective in predicting donor responses in imbalanced datasets and may hinder PVA's ability to make accurate judgments based on the predictions. Consequently, specialised methods like RFQ Classifiers that tackle class imbalance through a quantile framework are necessary to address class imbalance effectively.

A similar implementation of QRF and visualisation was carried out for the Boston Housing dataset from the paper of Meinshausen and Ridgeway, 2006 and is included in Appendix A.1.

5.2 RFQ Classifier Optimality

As explained in Subsection 4.1.2, the q^*-classifier is the optimal quantile to minimise the misclassification error based on an RF classification framework. We can minimise the misclassification error by maximising the TPR and TNR since these correspond to the proportion of all predictions that were correctly classified for both the minority (positive) class and the majority (negative) class. Moreover, given that the G-mean is the geometric mean of TPR and TNR, we minimise the misclassification error by maximising the G-mean.

This subsection aims to provide empirical evidence on how $q*$ provides the optimal G-mean value across a range of quantile values when we implement RFQ Classifiers to predict the donor response of PVA donors. As mentioned in Subsection 4.1.2, $q*$ is calculated by $\frac{N_1}{N_0+N_1}$, where N_1 and N_0 are the number of observations of the minority and majority class respectively. In the PVA dataset mentioned in Section 3, $N_1 = 2382$ and $N_0 = 42980$. Hence, our $q* = 0.0525$. Figure 2 shows the predicted G-mean of an RF classifier on the PVA dataset against a range of quantile values that also includes $q*$, which corresponds to the RFQ Classifier.

To produce the RFQ Classifier, the following fine-tuning process was carried out. Firstly, the optimal hyperparameters for RFQ Classifiers were produced using a grid search cross-validation (CV) technique. The CV was 5-fold and stratified, such that the training and testing split at each iteration of the CV contained the same level of imbalance amongst the minority class (class 1) and majority class (class 0). Table 5 showcases the hyperparameters implemented, their description and the range of their corresponding values implemented in the CV.

Since we have 4 values for each `mtry`, `ntree`, and `nodesize`, the total number of iterations in the grid search total to 64. The CV results in the following optimal values of the hyperparameters: `mtry` = 2, `ntree` = 500 and `nodesize` = 20. Using these optimal hyperparameters, we train our model from scratch once again using a stratified split of our PVA dataset, where 70% of the dataset is used in training the model and the rest is used to test the model. Then,

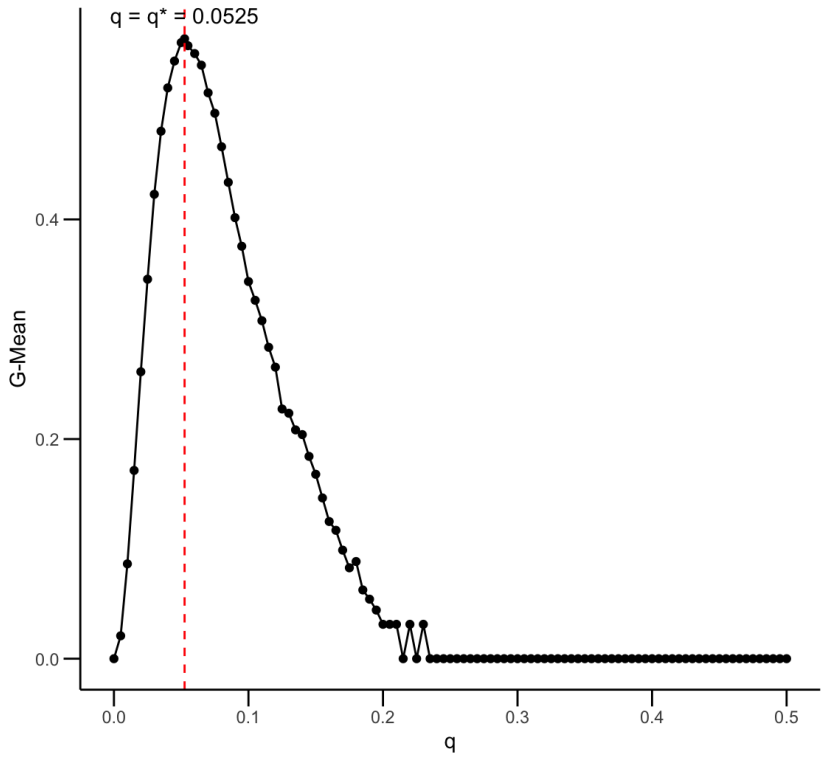

Figure 2: G-mean of the RF classifier on the PVA dataset plotted against a range of quantiles with the maximum indicated at $q*$ corresponding to an RFQ Classifier.

Table 5: Hyperparameters used in the grid search CV of RFQ Classifiers.

Parameter	**Description**	**Range of Values**
`mtry`	The number of variables randomly sampled as candidates at each split.	{2, 3, 4, 5}
`ntree`	The number of trees to grow in the forest.	{100, 500, 1000, 2000}
`nodesize`	The minimum size of terminal nodes. Smaller node size increases model complexity.	{1, 5, 10, 20}

the G-mean on our testing dataset equals 0.565. This G-mean is based on the $q*$ classifier implemented through RFQ Classifiers. We can confirm this is the maximum G-mean based on Figure 2. More particularly, Figure 2 shows the G-mean of a standard RF classification based on a range of quantiles between $[0, 0.3]$, with a step size of 0.005. The range of quantile also includes $q* = 0.0525$, which corresponds to the RFQ Classifier $\delta_D(x)$. According to 2, we conclude that the maximum G-mean obtained from the RF classification implemented at different levels of quantiles is at the quantile $q*$ as indicated by the red dashed line. Moreover, note

that the Bayes classifier $\delta_B(x)$, corresponding to $q = 0.5$ in Figure 2, is the default classifier in RF classification. This results in a G-mean of zero, highlighting the ineffectiveness of standard classifier techniques for our PVA dataset. Therefore, since G-mean is maximum at $q*$, we can conclude that RFQ Classifiers minimise the misclassification rule by maximising the G-mean.

5.3 Comparative Analysis

The structure of this comparative analysis is as follows. Firstly, the G-mean, PWG-mean, and ROC-AUC scores will be presented based on the results of a fine-tuned model on the testing dataset. Next, a box plot of G-mean CV results will be shown, highlighting the consistency of G-mean performance across models. Next, the results of SMOTE pre-processing for the best-performing models will be presented alongside feature importance.

5.3.1 Fine-Tuned Model Performance Metrics

Similar to the fine-tuning process for RFQ Classifiers explained in Subsection 5.2, the following models followed the same procedure: SMOTEBoost, AdaCost and XGBoost. More particularly, 64 iterations with different combinations of hyperparameters were selected in a grid search CV. Once, the CV produced optimal hyperparameters, the model was trained one final time and tested through a stratified split of the PVA dataset. The hyperparameter grid for XGBoost, AdaCost and SMOTEBoost can be found in Appendix A.2.

As explained in Subsection 4.4, AdaCost follows the principles of cost-sensitive learning in a Boosting algorithm. Therefore, the costs associated with the misclassification of the minority (positive) class and majority class (negative) had to be decided. The misclassification cost of classifying the majority class as the minority class was set to 1. Moreover, Table 6 showcases the range of the misclassification costs of classifying the minority class as the majority class. This range was incorporated in the grid search CV for the AdaCost model to find the optimal cost. It was concluded that `C1` $= 50$, was optimal in the CV. This implies that for the PVA dataset, the misclassification cost `C1` has to be higher than the prevalent IR to produce optimal results.

Once each model is fine-tuned and the optimal hyperparameters are extracted. The model is trained one final time with the hyperparameters using a stratified split of the PVA dataset. The performance metrics on the corresponding test set are used to evaluate and compare the models. Moreover, as a benchmark to compare all these models against, a single Decision Tree Classifier (Breiman, 2017) with the hyperparameter `class_weight` is set to "balanced", which automatically adjusts weights based on the class imbalance in the training dataset. This is

Table 6: Cost parameter range used in the grid search CV of AdaCost model. Note: IR refers to the imbalance ratio of the PVA dataset.

Cost Parameter	**Description**	**Range of Values**
`C1`	The misclassification cost of classifying the minority class as the majority class.	{IR, 35, 50, 80}

picked as the benchmark since a single classifier provides a solid baseline to compare to our ensemble techniques. If an ensemble technique performs poorer compared to the baseline, the additional complexity of the ensemble does not result in better classification performance and hence will be excluded from the rest of the comparison. The table with the final performance metrics is found below in Table 7.

Table 7: Performance metrics of all methods based on the test set from the PVA dataset. Note: Values highlighted in bold are the maximum for each metric.

Method	**G-mean Score**	**ROC-AUC Score**	**PWG-mean**
Decision Tree Classifier	0.264	0.510	0.072
RFQ Classifier	**0.565**	0.584	0.125
SMOTEBoost	0.193	0.528	0.192
AdaCost	0.280	0.507	0.059
XGBoost	0.552	**0.600**	**0.130**

The performance results in Table 7 for various models highlight significant differences in their effectiveness. Firstly, SMOTEBoost performs poorly when compared to the benchmark model of the Decision Tree Classifier. AdaCost slightly performs better than the benchmark model, with a G-mean of 0.280 as compared to 0.264 of the Decision Tree Classifier.

The RFQ Classifier, with the highest G-mean score of 0.565, demonstrates a strong balance between sensitivity and specificity. This indicates its capability to correctly identify both donors who will and will not respond, which is crucial for minimising misclassification costs in imbalanced datasets.

When focusing on the Positive Weighted G-mean (PWG-mean), XGBoost shows slightly better performance with a score of 0.130 compared to the RFQ Classifier which has a score of 0.125. PWG-mean in particular is relevant for PVA. Since precision is provided with a higher weightage in this metric, a high PWG-mean score implies that if PVA selects the list of donors

from the predicted positive responses (TARGET_B = 1), the percentage of donors who would respond with a donation would be higher if XGBoost was used instead of RFQ Classifiers. Moreover, XGBoost also does slightly better for the ROC-AUC score when compared to RFQ Classifiers.

5.3.2 Model Stability

In this section, we delve deeper into the model stability during the CV process. By examining each iteration of the CV process, we can assess the model stability in terms of the performance metrics. Figure 3 presents the G-mean result across all iterations of the CV for the RFQ Classifier, AdaCost, XGboost and SMOTEBoost models.

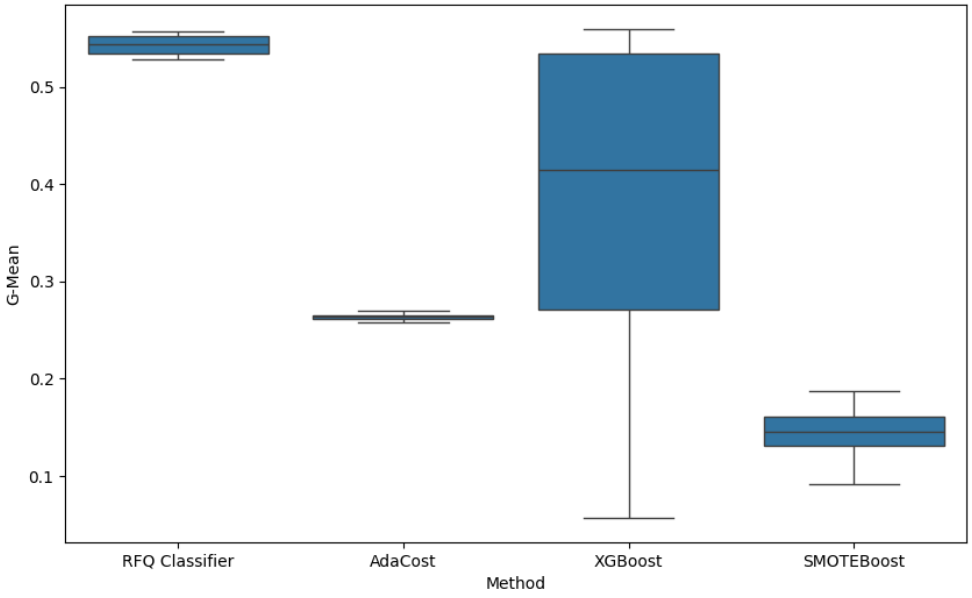

Figure 3: Boxplot of G-Mean for Different Methods (RFQ, AdaCost, XGBoost, and SMOTE-Boost) showing the variability of G-Mean performance across the CV iterations.

Figure 3 visualizes the distribution of G-Mean scores across cross-validation folds for each method implemented. The blue area represents the interquartile range (IQR), which contains the middle 50% of the data, while the line inside the box marks the median G-Mean. The whiskers extend to the minimum and maximum values within 1.5 times the IQR from the quartiles, and any points outside this range are considered outliers.

According to Figure 3, the RFQ Classifier demonstrate a consistently high G-mean across all iterations of the 5-fold cross-validation. This indicates their robustness and reliability in handling class imbalance in the PVA dataset. Moreover, the narrow IQR range and absence of significant outliers suggest that the performance of the RFQ Classifier is stable.

Conversely, the AdaCost method shows a noticeably lower G-Mean with minimal variation, signifying that it struggles to balance sensitivity and specificity effectively. XGBoost, while having a higher median G-Mean compared to AdaCost and SMOTEBoost, exhibits significant variability, as indicated by its wide IQR and presence of outliers. This variability in Figure 3 alongside the final results from Table 7 suggests that while XGBoost can achieve high performance, it may also be prone to inconsistent results depending on the specific training subset. SMOTEBoost, on the other hand, shows moderate performance with some variability, but its G-Mean values are generally lower than those of RFQ and XGBoost, indicating that the synthetic samples generated by SMOTE may not always represent the underlying data distribution accurately. These results highlight the importance of selecting a classification method that not only performs well on the final performance metric but also maintains stable performance across various folds of the data in the CV.

5.4 SMOTE Implementation

According to Table 7, XGBoost and RFQ Classifiers perform significantly better than the benchmark of the Decision Tree Classifier when compared to the AdaCost and SMOTEBoost models. Therefore, we continue with implementing SMOTE as a preprocessing step for the fine-tuned models of XGBoost and RFQ Classifiers. We utilise the same optimal hyperparameters mentioned in Subsection 5.3.1 for RFQ Classifiers and in Subsection 5.3.2 for XGBoost respectively. SMOTE is implemented for our PVA dataset as follows. Our training dataset, which is 70% of our total dataset is utilised in the Algorithm 1 in Subsection 4.2. Hence, our $T = 2381$, which is the number of minority samples and N is set to 100 to ensure a 1:1 ratio amongst the minority and majority classes in the training sample. The output of the SMOTE algorithm will be the resampled training dataset, which is then used to train our RFQ Classifier and XGBoost models separately,

Figure 4 showcases the G-mean performance of RFQ Classifiers and XGBoost from Table 7 alongside the G-mean of both models with SMOTE as a pre-processing step.

According to Figure 4, it is evident that the performance of the 2 models does not increase with the implementation of SMOTE as a pre-processing step. This highlights SMOTE's ineffectiveness in improving RFQ Classifiers' classification performance, as displayed by a significant drop in the G-mean. In contrast, while the performance of the XGBoost model remains relatively stable, SMOTE does not seem to increase the G-mean, suggesting that SMOTE might be unable to learn the correct distribution of the features pertaining to the minority class. Therefore, SMOTE applied to our PVA dataset leads to no performance gain

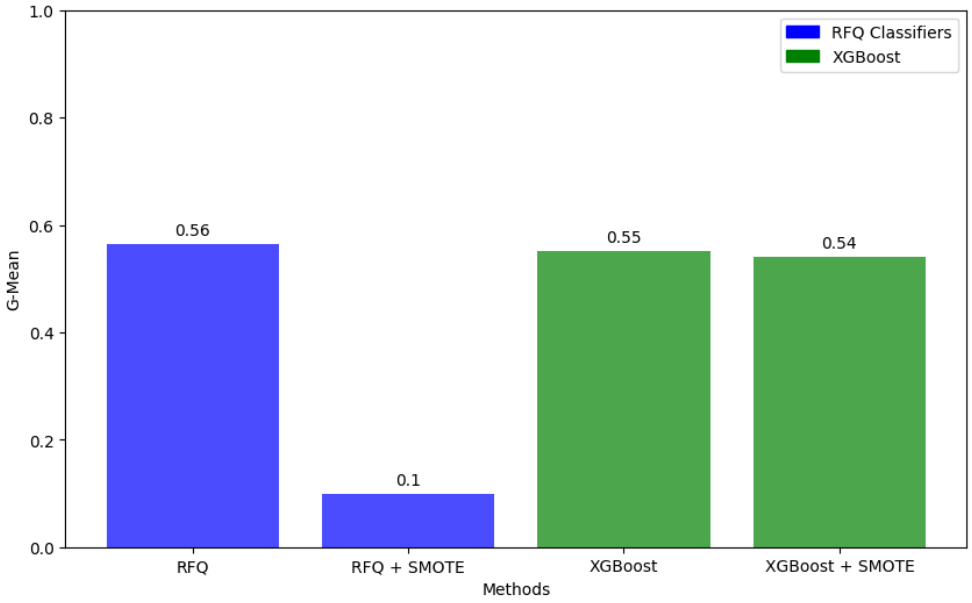

Figure 4: G-Mean performance comparison for RFQ Classifiers and XGBoost, with and without SMOTE.

for both RFQ Classifiers and XGBoost. This may imply that the synthetic examples created by SMOTE are unable to identify the correct distribution of the minority class data.

To delve deeper into the synthetic examples that SMOTE created, we consider the density plots of 2 features NGIFTALL and LASTDATE from the PVA dataset in Figure 5.

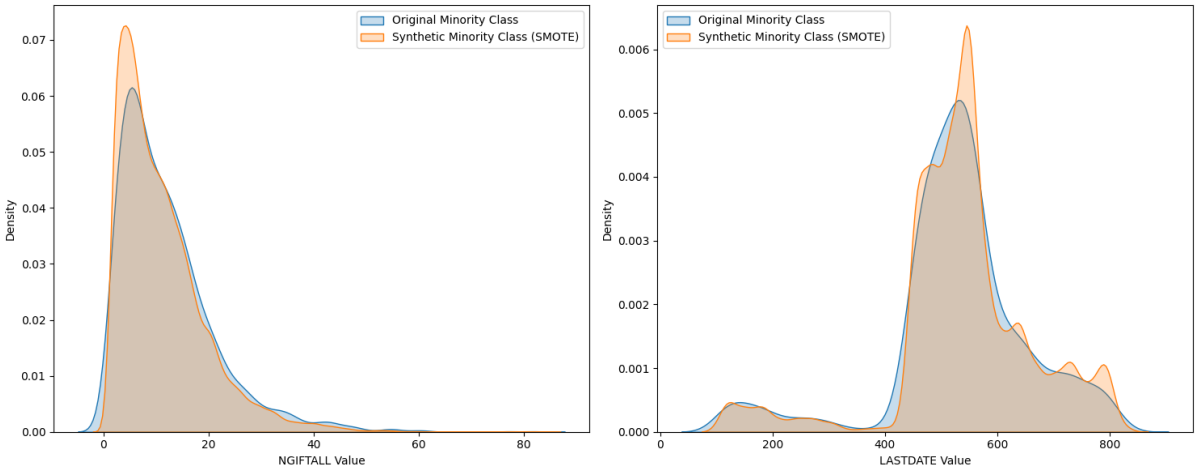

Figure 5: Density plots comparing the distributions of the original minority class, synthetic minority class (SMOTE) for the features NGIFTALL and LASTDATE.

Figure 5 represents the probability density function (PDF) of the existing data for the two features NGIFTALL and LASTDATE. More particularly, the y-axis represents the relative likelihood of data points falling in a particular value of the feature (x-axis). The plots therefore

compare the distribution of the original feature data that corresponds to the minority class (TARGET_B = 1) and the distribution of synthetic feature data for the same class. According to Figure 5, we can conclude that the synthetic examples generated by SMOTE for NGIFTALL and LASTDATE closely follow the original distribution of the original minority class, therefore providing a fairly accurate representation of the original feature space. This suggests that there might be another underlying reason why SMOTE might not be performing as expected, despite capturing the data distribution fairly accurately.

Therefore, we consider an alternate reason as to why SMOTE might now be improving the classification performance (G-mean) for the relevant models. Figure 6 again shows a density plot of the 2 features NGIFTALL and LASTDATE just as in Figure 5, although the PDF of both features corresponding to the majority class is also shown. (TARGET_B = 0).

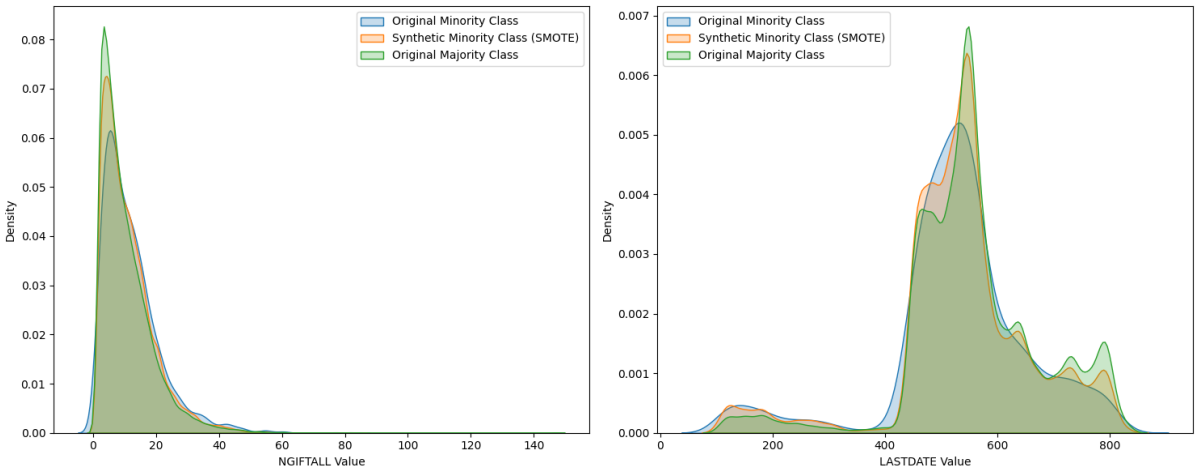

Figure 6: Density plots comparing the distributions of original minority class, synthetic minority class (SMOTE), and original majority class for the features NGIFTALL and LASTDATE.

Figure 6 reveals significant overlap in the feature densities of the minority and majority classes. The right panel, depicting the pdf of the LASTDATE feature, indicates that SMOTE has generated synthetic samples more closely aligned with the majority class than the minority class. Similarly, the left panel, illustrating NGIFTALL, shows that the density peak for the synthetic minority class data is positioned between the peaks of the majority and minority class data densities.

This alignment may occur because SMOTE interpolates between minority class instances and their nearest neighbours. When these neighbours are near the decision boundary—the threshold where the model distinguishes between the majority and minority classes—the generated samples can end up resembling the majority class, reducing their effectiveness in improving classification performance. Decision boundaries are crucial as they determine how a

classifier separates different classes, and class overlap can blur these boundaries, complicating the model's ability to differentiate between classes. Hence, resulting in a high number of misclassified instances.

6 Conclusion

This research investigated the extension of a quantile framework through Random Forest Quantile (RFQ) Classifiers to address classification problems involving imbalanced donor response classes in a Non-Profit Organisation (NPO) context. The performance of RFQ Classifiers was compared against other established techniques designed to handle class imbalance, including XGBoost, SMOTEBoost, and AdaCost. Additionally, the performance improvements offered by SMOTE for the best-performing models in the comparative study were evaluated. This study contributed to the literature by examining NPO donor response through the lens of class imbalance and exploring the implementation of quantile-based methods in classification problems.

The main research question aimed to compare the classification performance of RFQ Classifiers against other methods for tackling class imbalance. The findings concluded that RFQ Classifiers had the highest G-mean score of 0.565 on a test set of the PVA data, compared to SMOTEBoost, AdaCost, and XGBoost, which had scores of 0.193, 0.280, and 0.552, respectively, as shown in Table 7. SMOTEBoost performed the poorest, with a lower score than the benchmark Decision Tree Classifier with a G-mean = 0.263. Although XGBoost's performance on the G-mean was slightly lower than that of the RFQ Classifier, it performed slightly better on the ROC-AUC and PWG-mean scores. A higher PWG-mean score suggests that XGBoost might be optimal for predicting donor response, as PVA would be more interested in maximising the classification performance amongst the numbers of donors predicted to respond with a donation (predicted TARGET_B = 1) rather than those predicted not to donate. However, the instability of XGBoost, as presented in the CV performance in Figure 3, indicates that the model might be sensitive to the PVA data. Therefore, overall, RFQ Classifiers seemed to perform the best in performance metrics and model stability. The optimised performance of RFQ Classifiers was further established in Section 5.2. Considering a density-based approach, $q*$-classifiers minimised classification error by providing a maximum G-mean across a range of quantile values implemented for an RF classification, confirming the optimality of $q*$-classifiers.

A subquestion in this research also considered whether the popular preprocessing technique SMOTE provided any performance gains for classification compared to not applying SMOTE before implementing a learning model. The results confirmed that SMOTE seemed unable to

learn from the distribution of features corresponding to the minority class. Instead, SMOTE appeared to learn the distribution of the majority class. This could be attributed to class overlap observed in the dataset, where features across both majority and minority classes followed a similar pattern, as illustrated in Figures 6 and 5.

Class overlap can be considered a general limitation of this research. Besides being a potential cause of SMOTE's ineffectiveness, class overlap also impacts classification performance in general. According to research, class overlap has a higher negative impact on the performance of learning algorithms when compared to class imbalance (Wade and Glynn, 2020). Additionally, another limitation is the use of an NPO dataset pertaining to only one mailing campaign. A successful prediction model should be tested across various mailing campaigns, ideally using a walk-forward procedure (Cacciarelli and Boresta, 2022), where the time frame and sample size dedicated to testing depend on upcoming campaigns. This procedure would simulate a production environment, where the model can be trained on past campaign data and tested on future campaigns. This approach would have been more suitable for predicting donor response, assuming a sufficient amount of data was available.

Given these limitations, future research should primarily focus on methods to address class overlap alongside class imbalance. For instance, the novel technique of Density-Based Adaptive K-Nearest Neighbors (DBANN) introduced by Yuan et al. (2021) addresses both class imbalance and overlap.

The findings and limitations of this research have practical and theoretical implications. At a practical level, NPOs should focus on collecting donor data and features that can easily distinguish between the majority and minority classes. While this can be challenging, consistent exploratory data analysis can help aid in the decision of which features to include in a classification problem where donor response predictions are to be made.

At the theoretical level, while preprocessing steps such as SMOTE provide an easy implementation for solving class imbalance, issues like class overlap can render these methods ineffective. This extends to implementing learning techniques in general. Despite issues like class overlap, learning techniques based on cost-sensitive learning appear reliable in classification performance. Therefore, introducing novel algorithmic changes (such as RFQ Classifiers) to standard machine learning techniques is recommended for class imbalance problems. If introducing new techniques is not feasible, selecting the right values for hyperparameters and fine-tuning models can also help overcome classification performance issues in the case of class imbalance.

References

Ali, H., Salleh, M. M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: A review. *Indonesian Journal of Electrical Engineering and Computer Science*, *14*(3), 1560–1571.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*, 197–227.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*, 123–140.

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

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Bromideh, A. A. (2011). The widespread challenges of ngos in developing countries: Case studies from iran. *International NGO Journal*, *6*(9), 197–202.

Bult, J. R. (1993). Target selection for direct marketing.

Cacciarelli, D., & Boresta, M. (2022). What drives a donor? a machine learning-based approach for predicting responses of nonprofit direct marketing campaigns. *Journal of Philanthropy and Marketing*, *27*(2), e1724.

Castro, C. L., & Braga, A. P. (2013). Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *IEEE transactions on neural networks and learning systems*, *24*(6), 888–899.

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.

Chawla, N. V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). Smoteboost: Improving prediction of the minority class in boosting. *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7*, 107–119.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Cui, Y., Ma, H., & Saha, T. (2014). Improvement of power transformer insulation diagnosis using oil characteristics data preprocessed by smoteboost technique. *IEEE Transactions on Dielectrics and Electrical Insulation*, *21*(5), 2363–2373.

Dal Pozzolo, A., Caelen, O., & Bontempi, G. (2015). When is undersampling effective in unbalanced classification tasks? *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part I 15*, 200–215.

Dietterich, T. G. (2000). Ensemble methods in machine learning. *International workshop on multiple classifier systems*, 1–15.

Fan, W., Stolfo, S. J., Zhang, J., & Chan, P. K. (1999). Adacost: Misclassification cost-sensitive boosting. *Icml*, *99*, 97–105.

Farrokhvar, L., Ansari, A., & Kamali, B. (2018). Predictive models for charitable giving using machine learning techniques. *PloS one*, *13*(10), e0203928.

Faulk, L., Kim, M., Derrick-Mills, T., Boris, E., Tomasko, L., Hakizimana, N., Chen, T., Kim, M., & Nath, L. (2021). Nonprofit trends and impacts 2021. *The Urban Institute*.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, 863–905.

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.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(4), 463–484.

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. *2008 Fourth international conference on natural computation*, *4*, 192–201.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, *73*, 220–239.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (Vol. 2). Springer.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, *21*(9), 1263–1284.

Hettich, S., & Bay, S. D. (1999). The uci kdd archive [Irvine, CA: University of California, Department of Information and Computer Science].

Hommerová, D., & Severová, L. (2019). Fundraising of nonprofit organizations: Specifics and new possibilities. *Journal of social service Research*, *45*(2), 181–192.

Huang, Q., Zhang, H., Chen, J., & He, M. (2017). Quantile regression models and their applications: A review. *Journal of Biometrics & Biostatistics*, *8*(3), 1–6.

Japkowicz, N. (2013). Assessment metrics for imbalanced learning. *Imbalanced learning: Foundations, algorithms, and applications*, 187–206.

Jiang, Z., Pan, T., Zhang, C., & Yang, J. (2021). A new oversampling method based on the classification contribution degree. *Symmetry*, *13*(2), 194.

Jones, A., & Posnett, J. (1991). Charitable donations by uk households: Evidence from the family expenditure survey. *Applied Economics*, *23*(2), 343–351.

Jonker, J.-J., Paap, R., & Franses, P. H. (2000). *Modeling charity donations: Target selection, response time and gift size* (tech. rep.).

Kaymak, U. (2001). Fuzzy target selection using rfm variables. *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, *2*, 1038–1043.

Khan, N., Shahid, S., Juneng, L., Ahmed, K., Ismail, T., & Nawaz, N. (2019). Prediction of heat waves in pakistan using quantile regression forests. *Atmospheric research*, *221*, 1–11.

Koenker, R., & Bassett Jr, G. (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, 33–50.

Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine learning*, *30*, 195–215.

Lan, J.-s., Berardi, V. L., Patuwo, B. E., & Hu, M. (2009). A joint investigation of misclassification treatments and imbalanced datasets on neural network performance. *Neural Computing and Applications*, *18*, 689–706.

Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. *AAAI/IAAI*, *1997*, 546–551.

Mease, D., Wyner, A. J., & Buja, A. (2007). Boosted classification trees and class probability/quantile estimation. *Journal of Machine Learning Research*, *8*(3).

Meinshausen, N., & Ridgeway, G. (2006). Quantile regression forests. *Journal of machine learning research*, *7*(6).

O'Brien, R., & Ishwaran, H. (2019). A random forests quantile classifier for class imbalanced data. *Pattern recognition*, *90*, 232–249.

Pes, B., & Lai, G. (2021). Cost-sensitive learning strategies for high-dimensional and imbalanced data: A comparative study. *PeerJ Computer Science*, *7*, e832.

Priscilla, C. V., & Prabha, D. P. (2020). Influence of optimizing xgboost to handle class imbalance in credit card fraud detection. *2020 third international conference on smart systems and inventive technology (ICSSIT)*, 1309–1315.

Ramyachitra, D., & Manikandan, P. (2014). Imbalanced dataset classification and solutions: A review. *International Journal of Computing and Business Research (IJCBR)*, *5*(4), 1–29.

Sakho, A., Scornet, E., & Malherbe, E. (2024). Theoretical and experimental study of smote: Limitations and comparisons of rebalancing strategies. *arXiv preprint arXiv:2402.03819*.

Schapire, R. E. (1990). The strength of weak learnability. *Machine learning*, *5*, 197–227.

Schetgen, L., Bogaert, M., & Van den Poel, D. (2021). Predicting donation behavior: Acquisition modeling in the nonprofit sector using facebook data. *Decision Support Systems*, *141*, 113446.

Sousa, J., Madeira, S., & Kaymak, U. (2003). Modeling charity donations using target selection for revenue maximization. *The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ'03.*, *1*, 654–659.

Sun, Y., Kamel, M. S., Wong, A. K., & Wang, Y. (2007). Cost-sensitive boosting for classification of imbalanced data. *Pattern recognition*, *40*(12), 3358–3378.

Völz, B., Mielenz, H., Siegwart, R., & Nieto, J. (2016). Predicting pedestrian crossing using quantile regression forests. *2016 IEEE Intelligent Vehicles Symposium (IV)*, 426–432.

Wade, C., & Glynn, K. (2020). *Hands-on gradient boosting with xgboost and scikit-learn: Perform accessible machine learning and extreme gradient boosting with python*. Packt Publishing Ltd.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., et al. (2008). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*, 1–37.

Yang, P., Hwa Yang, Y., B Zhou, B., & Y Zomaya, A. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics*, *5*(4), 296–308.

Yuan, B.-W., Luo, X.-G., Zhang, Z.-L., Yu, Y., Huo, H.-W., Johannes, T., & Zou, X.-D. (2021). A novel density-based adaptive k nearest neighbor method for dealing with overlapping problem in imbalanced datasets. *Neural Computing and Applications*, *33*, 4457–4481.

A Appendix

A.1 QRF Replication from Meinshausen and Ridgeway (2006)

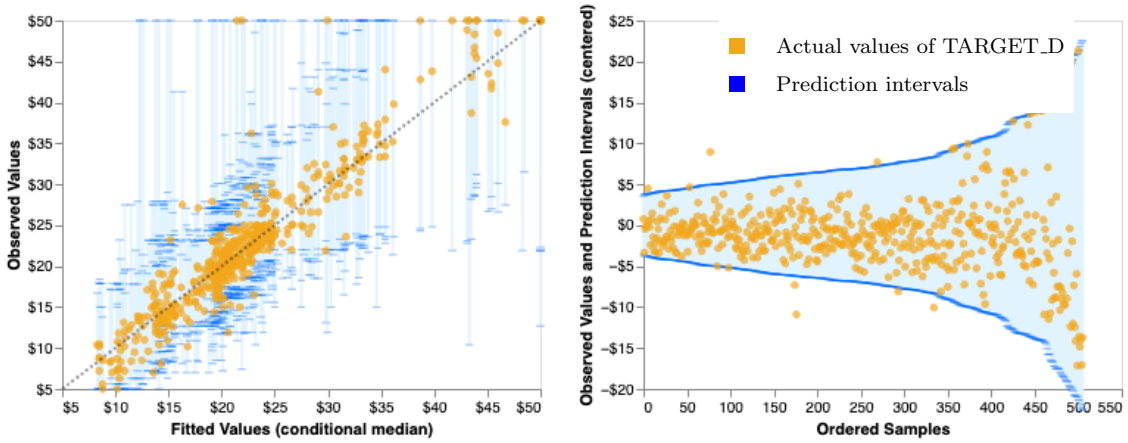

Figure A.1: Estimated conditional quantiles for a sample of n=5000 observations from the PVA dataset. Left panel: Observed values plotted against predicted values. Prediction intervals are displayed by the light blue bars with the dark blue ticks representing the lower and upper bound of the prediction interval. Right panel: The same observations are ordered according to the length of the prediction interval. Note: The legend included corresponds to both panels.

A.2 Hyperparamater Grid for CV for XGBoost, AdaCost and SMOTEBoost

Table A.1: Optimal hyperparameters used in the grid search CV of XGBoost model. Note: IR denotes the imbalance ratio of PVA Dataset.

Parameter	Description	Range of Values
`max_depth`	The maximum depth of a tree.	{4, 6}
`learning_rate`	The step size shrinkage.	{0.1, 0.2}
`scale_pos_weight`	Controls the balance of positive and negative weights.	{5, 10, IR, 25}
`colsample_bytree`	The subsample ratio of columns when constructing each tree.	{0.8, 1}
`subsample`	Fraction of training data randomly sampled at each tree	{0.8,0.9}

Table A.2: Hyperparameters used in the grid search CV of AdaCost model

Parameter	Description	Range of Values
`n_estimators`	The number of boosting rounds/iterations	{50, 100, 150, 200}
`learning_rate`	The step size shrinkage	{0.1, 0.5, 1.0, 1.5}

Table A.3: Hyperparameters used in the grid search CV of SMOTEBoost model

Parameter	Description	Range of Values
`n_estimators`	The number of boosting rounds or iterations.	{10, 50, 100, 150}
`learning_rate`	The step size shrinkage used to prevent overfitting	{0.01, 0.1, 1.0, 10.0}
`n_samples`	The number of samples to be used in the algorithm.	{50, 100, 200, 300}

A.3 Programming Code

The following description describes the programming code that was implemented for this research. The two programming languages used were Python (Version 3.11.8) and R (R version 4.4.0).

The attached file has code related to the replication which corresponds to the replication from Meinshausen and Ridgeway (2006) of the QRF method on the Boston Housing Dataset from Meinshausen and Ridgeway (2006) and from the selected PVA Dataset from this research. The QRF predictions were run using the `quantregForest` package in the `replication_qrf.R` file. The corresponding visualisations for QRF were made in `replicationQRF.ipynb` Jupyter notebook. Moreover, the raw dataset of PVA is named `cup98LRN.csv` and the subset used throughout this research is named `subset_df.csv`. The preprocessing of the raw dataset that leads to the subset is done in the `preprocessing.ipynb` notebook.

The extension code is as follows. It contains implementations of RFQ classifiers, SMOTEBoost, XGBoost and AdaCost. Once again, the raw dataset of PVA is named `cup98LRN.csv` and the subset used throughout this research is named `subset_df.csv`. RFQ classifiers and their corresponding SMOTE implementation were made using the `rfsrc` package in the `RFQ_classifier_implementation.R` file. XGBoost (and SMOTE), AdaCost and SMOTEBoost models are implemented in the `extension_models.ipynb` notebook. The grid search CV of all four methods

was run using these two files. XGBoost was implemented using the `scikit-learn` package in Python. AdaCost and SMOTEBoost were implemented using custom classes defined in `adacost.py` and `smote.py` respectively.