How can we optimize the performance of multiclass classification methods for minority classes in imbalanced datasets?

February 28, 2024

Master Thesis in Quantitative Marketing and Business Analytics

Author:
Sterre Spaanderman

Student number:
510941

Supervisor:
Philip Hans Franses

EUR

Second assessor:
Wendun Wang

EUR

Keywords: Imbalanced data, Undersampling, Oversampling, Multiclass Classification, Random Forest, Naive Bayes, XGBoost, Support Vector Machines

Abstract

In the marketing landscape, the popularity of data-driven decision-making has grown among both consumers and companies. For instance, predictive models are increasingly employed by companies to anticipate a customer's interest in specific car models. This enables targeted and personalized promotions, ultimately enhancing customer satisfaction and boosting company revenue. However, real-world decisions, such as recommending cars to customers, often involve imbalanced datasets where minority classes, like lesser-bought car brands or models, lack sufficient representation. Yet, accurately predicting these minority classes, such as very expensive cars, might be of great interest, as minority classes may provide the most revenue and thus are very lucrative. Using imbalanced datasets in machine learning tasks like these may result in biased models, requiring effective strategies to address this issue.

This research comprehensively explores strategies to address the challenges posed by imbalanced datasets, focusing primarily on several data adjustments. Specifically, this research exploits oversampling and undersampling techniques to rebalance class distributions in order to increase the influence of minority classes. Then, these adjusted datasets are applied to various multiclass classification models. This paper performs these steps in an attempt to increase the predictive performance of those classification models, especially for minority classes. The final sections of this paper highlight the efficacy of combining the Synthetic Minority Oversampling Technique with either the Random Forest classifier or the Extremely Gradient Boosting classifier. Both combinations demonstrate optimal results in predicting the customer's car brand, especially for the minority classes, but also overall, as they both achieve an accuracy rate of approximately 73%.

Contents

1 Introduction

The automotive industry is undergoing rapid changes. In comparison to decades ago, the market is now flooded with different cars, including those from new brands. This increase in the number of cars is not limited to brands alone; there is a growth in the diversity of cars themselves. Technological advancements, for example, the emergence of hybrid and electric cars, contribute to this expanding spectrum of choices. Consequently, the process of selecting a car has become a daunting and stressful task for consumers.

Simultaneously, this surge in variety poses a challenge for automotive businesses. There is an ongoing battle for consumer attention among different businesses and brands. The proliferation of big data in the world of marketing has accentuated the importance for businesses to deeply understand their customer base in terms of their characteristics and preferences. This understanding is crucial for targeted marketing and promotions. With an intricate knowledge of customer preferences, businesses can tailor promotions to specific customers, increasing the likelihood of customer satisfaction and, consequently, bolstering revenue.

Classification methods are the main source for businesses to analyze and predict the purchasing behavior of a customer. In particular, as we deal with cars from multiple brands and models, we apply multiclass classification methods. By using the power of machine learning, these methods empower businesses to move beyond conventional market research approaches and explore the world of predictive analytics. In doing so, they not only enhance customer satisfaction but also encourage innovation in a sector that is combining technology, mobility, and individual preferences.

One significant challenge for multiclass classification methods is the performance and reliability of them on imbalanced datasets. In a multiclass scenario, where there are more than two classes to predict, imbalanced datasets occur when the number of instances in each class is unevenly distributed. In real-life data, and thus also in data corresponding to the automotive industry, these imbalanced datasets often occur. In this research, imbalanced data is caused by some car brands and models being bought way more often than others. Specifically, the dataset that is used for this research deals with cars corresponding to three different brands: Toyota, Suzuki, and Lexus. The three brands correspond with respectively 72%, 25%, and 3% of the observations. Therefore, we can conclude that this dataset is very imbalanced, for which Toyota cars represent the majority class and Lexus cars represent the minority class.

One of the primary concerns with imbalanced datasets is that machine learning models tend to be biased toward the majority class. The model becomes overly inclined to predict the class that has more instances, often overlooking minority classes. Therefore, an imbalance may lead to a skewed understanding of the true predictive capabilities of the model, resulting in suboptimal performance when it comes to the less represented classes. Yet, precisely these minority classes might be of great interest for a company as they can include very expensive cars and thus largely influence the company's revenue. In our dataset, the minority class, which contains exclusively Lexus cars, indeed corresponds to the most costly cars. Furthermore, the traditional accuracy metrics can become deceptive when the model is performed on an imbalanced dataset. This means that a model might achieve high accuracy by predominantly predicting the majority class, even if it fails to effectively classify instances from minority classes. This can create a misleading impression of the model's effectiveness.

Concluding, imbalanced datasets can weaken the generalization ability of a particular classification model. In the presence of limited samples for certain classes, the model may struggle to learn meaningful patterns and may generalize poorly to unseen data. This phenomenon is particularly pronounced in situations where the minority classes represent critical outcomes or events. In the dataset used for this research, this might be the case with the few observations that consider the purchase of an extremely expensive car, which increases business revenue by a fair amount but barely happens.

Several techniques have been developed to address the challenges posed by imbalanced datasets in multiclass classification. Resampling methods, such as oversampling the minority class or undersampling the majority class, aim to create a more balanced training set. Balancing the dataset might increase the influence of minority classes and thus increase the predictive power of the classification models on cars that correspond to these smaller classes. Additionally, evaluation metrics such as precision, recall, F1 score, and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) will be considered in this research, to provide a more nuanced assessment of model performance in the context of imbalanced datasets. Especially, we aim to find an evaluation metric that is more robust to imbalanced datasets than accuracy and therefore gives an insight on the performance of the classifiers on the observations corresponding to minority classes.

This research seeks to assess various data-adjustment techniques to identify the most effective method for enhancing the performance of multiclass classification models, particularly for minority classes. Subsequently, we will compare these outcomes with the performance of a multiclass classification algorithm applied directly to the original imbalanced datasets. This comparison aims to ascertain which multiclass classification algorithm exhibits the greatest resilience to imbalanced datasets overall. Furthermore, we aim to establish an evaluation criterion that adequately reflects the performance of classification methods for minority classes. Hence, the paper's research question can be formulated as follows.

How can we optimize the performance of multiclass classification methods for minority classes in imbalanced datasets?

In order to answer this research question, the remainder of this paper will look as follows. The next section, Section 2, will sum up the earlier done research on the topic of imbalanced datasets, resample methods and multiclass classification methods. Then, in Section 3, I will present the dataset that was used in this research. Next, in Section 4, the actual resampling methods and multiclass classification methods that will be used in this research will be shown. Furthermore, this section will elaborate on which performance measurements will be used to determine the performance of each of the models. Subsequently, in Section 5, I will show the results of the application of these methods on both the original and the resampled datasets. Finally, in Section 6, I will draw a conclusion regarding Section 5, whereas in Section 7, I will interpret the significance of the results. On the final pages of this paper, there is an appendix, which can be used to obtain more information considering the topics touched on in this paper.

2 Literature

As mentioned above, this section will contain a review of the previously done research on this topic. We will elaborate on the results and conclusions drawn in many other papers to build a firm base of knowledge on this topic needed for further research.

Pattern recognition plays a crucial role in the realm of classification, particularly when it comes to understanding customer behavior and its connection to car purchases. In this study, our goal is to analyze the relationship between customers' interests and their choices in car ownership. Despite the seemingly straightforward nature of this task, we confront the challenge of imbalanced data.

Imbalanced data occurs when customers are not evenly distributed across different categories, leading to the existence of majority and minority classes. The majority class is the category with more customers than expected, while the minority class includes those with fewer customers than expected, which is in line with statements in Krawczyk [2016]. In the context of this research, the target variables—brand and model—contain majority and minority classes and thus exhibit imbalances. As mentioned in the previous section, certain brands and models, such as Toyota cars, may be purchased more frequently than others, creating a skew in the distribution.

This imbalance introduces complexity to our classification task, as models trained on imbalanced data may exhibit biases towards the majority classes. This is due to the fact that classifier learning algorithms assume a relatively balanced distribution between classes, as stated in Sun et al. [2009]. Consequently, accurate predictions for less-represented classes, like specific car brands or segments, become challenging. At the same time, as stated in Krawczyk [2016], minority classes represent the most important classes in terms of data mining, as they contain valuable knowledge corresponding to valuable purchases. As stated in Sun et al. [2009], recognizing and addressing these imbalances are vital steps in ensuring that our classification model provides meaningful insights into the nuanced patterns between customer characteristics and car ownership choices.

Even with many years of continuous development learning from imbalanced data, Krawczyk [2016] states that researchers still do not agree on what is the best way to deal with imbalanced datasets. Therefore, this research aims to test the most frequently used methods performed on an imbalanced dataset in order to improve the classification on minority classes. Then, we will analyze the performance of these methods and provide evidence on which methods are optimal.

As stated in Van Hulse et al. [2007], the first method we will apply in this research is to use resampling to balance the imbalanced dataset and increase the effect of minority observations. In particular, we will perform undersampling and oversampling. Oversampling refers to increasing the number of observations in minority classes in order to create a more balanced distribution across the classes. First, we will consider the most straightforward oversampling technique, which is referred to as random oversampling, as suggested by Mohammed et al. [2020]. For this method, we will randomly copy observations corresponding to the minority class. This will be done until the classes are perfectly balanced. We will consider random oversampling as the most frequently used and time-efficient oversampling technique, which occurs to perform fairly well in some cases. However, this technique does indeed balance the class distribution but does not provide any additional

information. Therefore, we will also consider another oversampling method referred to as Synthetic Minority Oversampling Technique (SMOTE), mentioned in Shelke et al. [2017]. For this method, we will add extra observations to the minority classes based on the k-nearest neighbors of each observation corresponding to the minority classes. Specifically, SMOTE considers the k-nearest neighbors of an observation in the minority class. Note that these nearest neighboring points also belong to the minority class. Then, the algorithm draws a line between the original point and each of its nearest neighbors and creates a new point at each of these lines. This method is often used in research where the aim is to increase the performance of classifying methods on minority classes, as this oversampling method not only improves the balanced distribution over classes but also adds extra information to the dataset.

Next, we will consider undersampling methods, which are the complete opposite of oversampling. Therefore, undersampling refers to decreasing the number of observations in the majority classes in order to create a more even distribution across the classes. The most straightforward method for undersampling is called random undersampling, which is mentioned in Shelke et al. [2017]. This method is rather popular due to the fact that it is easy to apply and very time-efficient. In particular, the random undersample method randomly deletes observations from the majority classes until the classes are perfectly balanced. However, as we randomly delete information from the dataset, this method can cause an enormous loss of information, especially when the classes are very imbalanced. Therefore, we aim to find a way to delete observations and thus control the distribution over classes without generating a very large loss of relevant information. Therefore, we will consider the Cluster Centroid undersampling method, which is suggested by Zhang et al. [2019]. In particular, this method decreases the number of observations in the majority class by clustering these observations using the K-means algorithm. Then, for those clusters, the method selects the observations that are near the boundaries. This causes the observations in the majority class that are near the center of the cluster to be deleted, as these points are said to be rather average and thus contain the least information. Therefore, we can conclude that the Cluster Centroids undersampling method deletes observations in a smarter way and thus controls the distribution over the classes while also keeping as much relevant information as possible.

In general, as stated by Zhang et al. [2019], undersampling methods have been proven to outperform oversampling methods. This might be due to the fact that oversampling methods, as they may add duplicate information to the dataset, increase the probability of overfitting. Overfitting, as explained in Hawkins [2004], refers to the use of models or procedures that violate parsimony. Intuitively, this means that the model uses more terms, in this case observations, than necessary. On the other side, as mentioned in Zhang et al. [2019], especially when the dataset is very imbalanced, undersampling methods suffer from high elimination of useful observations and therefore suffer from a significant loss of information.

After applying resampling methods to create a more balanced dataset, this research focuses on the application of different multiclass classification algorithms in order to predict a customer's interests and current purchases. As mentioned in Priyam et al. [2013], classification algorithms are one of the most frequently studied topics in data mining and machine learning research. In particular, classification algorithms use the values of categorical attributes to predict the values of other attributes. Classification problems occur in many different areas, such as text mining, as done by Turney [2002], neuroscience, as done by Guerra et al. [2011], and geoscience, as done by Romary et al. [2015]. As classification methods are popular among researchers, there are many different algorithms that each apply to different types of criteria and requirements

depending on the particular data. In this paper, we will consider the four most frequently used multiclass classification methods in research on an imbalanced dataset. One of the simplest classifiers, mentioned in much research, is the Naive Bayes (NB) classifier. As stated by Rish et al. [2001], this algorithm simplifies the classifying problem by assuming that the value of a feature does not provide any information about the value of any other feature once the class of the observation is known. Although independence between features in real-world data is a poor assumption, the same paper shows that in practice, the Naive Bayes classifier competes well with more advanced classifiers. Therefore, we will perform the NB classifier on the dataset used in this research, and we will analyze the results. Another frequently mentioned classification algorithm is the Random Forest (RF) classifier. As suggested by Speiser et al. [2019], this classifier is a popular developer for prediction models, as it is applicable in many different research settings. Furthermore, the same paper states that Random Forests consistently generate high prediction accuracy in comparison to other classification algorithms. In short, the Random Forest classifier relies on the combination of randomized decision trees and uses the mode of the predictions of each of these trees to create the final prediction, which is in line with the description of this model in Gall et al. [2012]. However, this classifier is mentioned in more detail in the remainder of this paper. Another classifier that has proven to be a powerful technique for classification is the Extreme Gradient Boosting (XGBoost) classifier. Again, this method is based on the combination of predictions from multiple decision trees. However, Dhaliwal et al. [2018] states that this classifier has the advantage of parallel processing. Intuitively, this means that the current decision tree and its error are used for the training of new decision trees. This means that this model is able to handle models with complex relationships and noisy data effectively. Therefore, we will apply XGBoost to this research and analyze its performance on imbalanced data. The final classification method that we will consider in this research is Support Vector Machines (SVM), as mentioned in Gold and Sollich [2003]. This classifier has been shown to perform fairly well on datasets with high-dimensional feature spaces that include a large number of features. In particular, this classifier relies on finding a hyperplane that best separates the observations corresponding to different classes in the feature space. In this paper, we will analyze the performance of SVM and compare its performance to the performance of NB, RF, and XGBoost classifiers.

To assess the effectiveness of the data adjustments and multiclass classification methods discussed earlier, we will employ multiple performance criteria. While accuracy is commonly used to interpret classification model results, as highlighted in Sokolova et al. [2006], it may not be sufficient for evaluating multiclass classification on imbalanced data, as pointed out by Menardi and Torelli [2014]. High accuracy in such cases can be misleading, as the model might simply favor the majority class. To address this limitation, we will consider two alternative performance measures, as suggested in Jeni et al. [2013]. The first measure is the F1 score, which represents the weighted average of precision and recall values. Both precision and recall will be elaborated on in this paper. In addition to the F1 score, we will calculate the AUC-ROC value for each model. This metric evaluates how well the positive cases are ranked compared to the negative cases. Both measures are known for their robustness in handling imbalanced data, providing a more comprehensive summary to determine the optimal model.

As previously discussed, this section offers a summary of existing research on addressing imbalanced datasets and employing multiclass classification methods in such scenarios. The subsequent section will delve into the details of the dataset chosen for the current research.

3 Data

In this section, I will introduce the contents and attributes of the dataset used in this research. Specifically, this dataset is a combination of two distinct datasets, both provided by the Louwman Group. In each of the sections below, I will elaborate on the content of both of these datasets. It is important to note that the examples presented in this section are entirely random and are not associated with individual customers.

3.1 Current ownership

The first dataset that is used for the combination is a dataset that shows customers, their current postal code, and the current car they are driving. This dataset contains approximately 580.000 observations and encompasses customers who have either purchased a car for private use or leased a car for work-related use. However, as customers lease a car, they will not have unlimited choice in cars. Given that the research focuses on analyzing the correlation between customer preferences and characteristics, the observations corresponding to leased cars will be excluded from the dataset. In particular, approximately 510.000 observations remain. This dataset will look as follows.

Customer ID	Postal Code	Brand	Model
164813	3061SE40A	Suzuki	Swift
1474713	6162CV6	Lexus	CT
93442	1075ST12	Toyota	Aygo

Table 1: This table shows a snapshot of the ownership dataset, which combines customers with cars and postal codes. For example, we see that customer with ID 164813 lives at 3061SE40A and drives a Suzuki Swift.

From the table above, one can note that this dataset provides information on customers owning cars from three distinct brands within the Louwman Group—specifically: Toyota, Suzuki, and Lexus. Note that those three brands will be used as target variables in the classification methods. Specifically, this means that this research aims to predict the customer's car to be either Toyota, Suzuki, or Lexus as accurately as possible. Further categorization includes different models within each of these brands, totaling 56 Toyota models, 18 Suzuki models, and 14 Lexus models, summing up to 88 distinct models. However, some of these models from Toyota, Suzuki, and Lexus are very similar. For example, the Toyota Corolla and Lexus IS are both small sedans and thus might be bought by the same kinds of people. Therefore, our classification method might struggle to distinguish the owners of both cars. Therefore, we will only consider the models corresponding to one single car brand. These models will then function as another target variable in our second classification task. As mentioned before, as over 72 percent of the observations in this dataset correspond to customers owning a Toyota car, we will focus only on predicting the Toyota model owned by a customer.

It is noteworthy that some of these Toyota models occur in less than 0.1 percent of the observations. Given the limited representation of these extreme minority classes, their patterns may not be adequately discerned. To address this, these classes are merged into a new category labeled "Others," employing a technique known as class consolidation. This process effectively reduces the total number of Toyota models to 27, facilitating

the start of a more balanced analysis of the dataset.

Besides the current car brand and model of a customer, this dataset shows the corresponding Dutch current postal codes for customers at that certain point in time, which can also be seen in Table 3.1. In this research, the term postal code contains the following items. First, we determine the Dutch postal code for a customer, which contains four numbers followed by two capital letters. Next, we add the apartment number of the customer. Finally, we add the eventual addition to the apartment number. This addition is always a letter.

3.2 Social, demographic, and geographic characteristics

The second dataset that will be used in this research is a set that combines postal codes with a large number of characteristics of customers. Again, postal codes contain not only actual Dutch postal codes but also apartment numbers and additions. A snapshot of this dataset might look as follows.

Postal Code	Age Head Household	Household Size	Time spend in the garden	Possession Credit Card
1781AB6	40-45 year	1 person	Very much	Yes
2132JG17A	60-65 year	4 people	Average	Yes
9743AK101	Older than 80	2 people	Less then average	No

Table 2: This table shows a snapshot of the Geomarkt dataset, which combines postal codes with customer characteristics. For example, we know that a customer living at postal code 1781AB6 is likely to be between 40 and 45 years old, lives alone, spends a lot of time in the garden, and owns a credit card.

The dataset described above originates from Geomarktprofiel, a company that conducts research through surveys and integrates the obtained results with publicly available information from CBS and Kadaster. By leveraging these three sources, the company constructs profiles for Dutch households represented by unique postal codes. The profile of customers includes both observations and predictions for various characteristics, and therefore the dataset contains no missing values.

Comprising a total of approximately 8.200.000 observations, each associated with a unique Dutch postal code, the dataset encompasses 149 variables. Table 11 to Table 14, shown in the appendix, provide the names and explanations of these variables. Notably, all the variables, as depicted in Table 2, are categorical in nature. Besides, the majority of these variables exhibit between 5 and 15 unique values. A comprehensive list of the exact number of values for each variable is available in the appendix.

3.3 Final dataset

The final dataset that will be used for this research will be a combination of both the ownership dataset and the characteristics dataset mentioned above. In particular, we will create a dataset that combines the current car and the current postal code of a particular customer with the corresponding social, demographic, and geographic characteristics. The final dataset for a particular customer will then look as follows:

Customer ID	Postal Code	Brand	Model	Characteristics
164813	3061SE40A	Suzuki	Swift	...
1474713	6162CV6	Lexus	CT	...
93442	1075ST12	Toyota	Aygo	...

Table 3: This table shows a snapshot of the final datasets, which combines customers with corresponding characteristic and a brand and model of a car, based on the postal code of the customer. For example, we note that the customer with ID 164813 lives at postal code 3061SE40A, owns a Suzuki Swift and has several characteristics.

This final dataset contains approximately 502.000 observations, which correspond to unique customers, and contains 153 variables. One should note that this number of variables also includes "Customer ID" and "Postal Code" which will only be used for the combination of the two datasets and will not be used for the performance of methods.

As mentioned before, this research focuses on the performance of data adjustments and multiclass classification methods on imbalanced data. Therefore, the brand and model of a car will be referred to as target variables and thus will represent the classes. Furthermore, the characteristics of the customers will be referred to as explanatory variables. As mentioned before, the explanatory variables are categorical variables. To keep the information hidden in those variables, but to increase the applicability of the dataset, we will turn these categorical variables into ordinal variables. This can be done as there is a clear order in the values for each of the variables. An example of how a character variable will be turned into an ordinal variable is shown in the table below, Table 4. This table shows how the character variable, which denotes how often customers go to a restaurant or cinema, will be transformed into an ordinal variable.

Character value	Very little	Little	Less than average	Average	More than average	Much	Very much
Ordinal value	1	2	3	4	5	6	7

Table 4: This table shows an example of how we convert a categorical value corresponding to an categorical variable into an ordinal value.

As this research focuses on the performance of methods on an imbalanced dataset, we will show that the dataset in this research is indeed imbalanced. Therefore, we provide the table below, Table 5, which shows the distribution of the observations over the car brands and Toyota models, respectively. Note that in this table, the number of classes denotes the number of classes present in the dataset after collecting all small classes into the class "Others" for the target variable *Toyota Model*.

From this table, we can see that for both target variables, *Brand* and *Toyota Model*, the difference between observations in the majority and minority classes is very large. Therefore, we can conclude that the dataset, independent of which target variable is considered, is very imbalanced.

Furthermore, we can plot the distribution of the observations over the classes in order to further analyse the degree of imbalance over the dataset. The figure below, Figure **??**, shows the distribution of the observations over the classes when target variable *Brand* is considered.

Target variable	**Brand**	**Toyota Model**
Number of classes	3	27
Total number of observations	501.764	363.679
Number of observations in largest class	363.679	84.798
Percentage of total number of observations	72.48	29.47
Number of observations in smallest class	14.405	529
Percentage of total number of observations	2.87	0.18

Table 5: This table shows the distribution over the classes of the corresponding target variable. For example, for the target variable Brand, we see that the majority class contains over 72 percent of the observations whereas the minority class contains almost 3 percent of the observations.

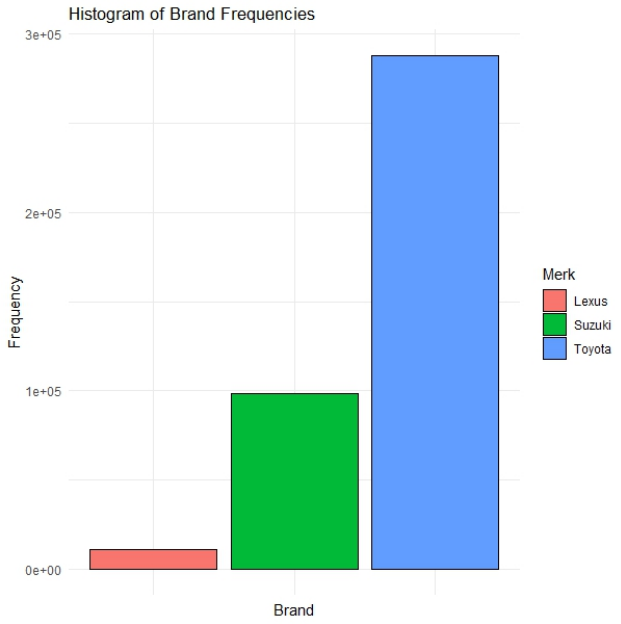

Figure 1: This figure shows the distribution of the observations over the classes which correspond to the brand of the car. One can see that this dataset is very imbalanced.

This figure, Figure 1, shows that the data is very imbalanced when *Brand* is considered to be the target variable. Again, one can see that by far the most cars in the dataset correspond to the brand Toyota, and therefore, Toyota cars present the majority class. Furthermore, one can note that the least number of cars are from Lexus, and thus the Lexus car represents the minority class.

Next, we will consider the distribution of observations over the classes when *Toyota Model* is considered to be the target variable. The plot of this distribution is presented below, in Figure 2.

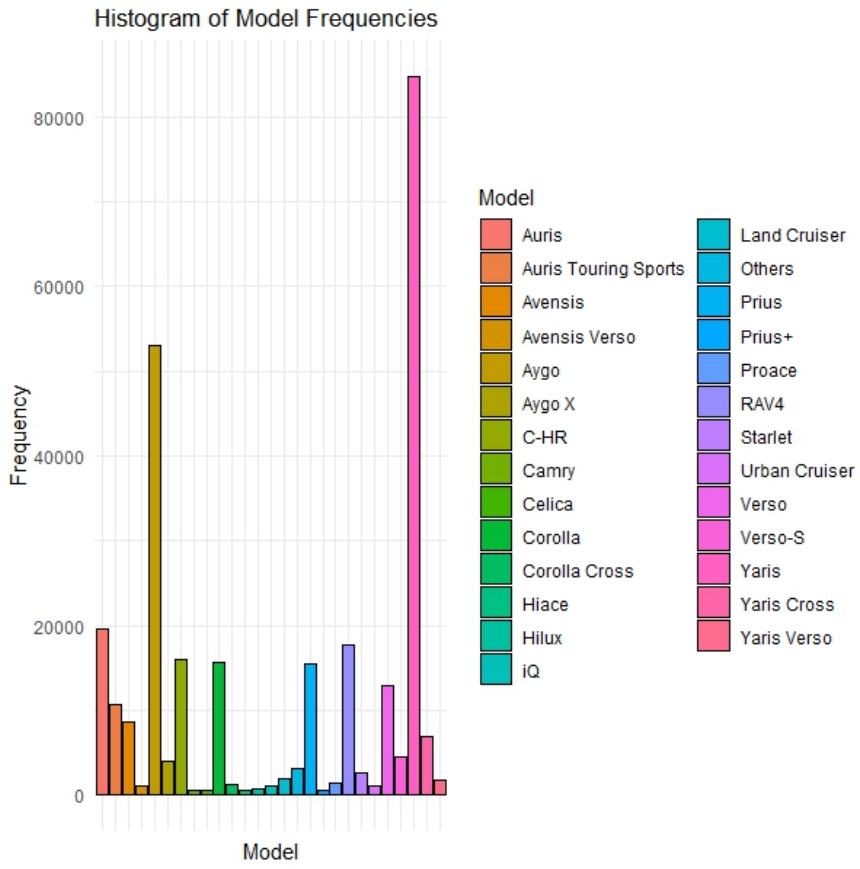

Figure 2: This figure shows the distribution of the observations over the classes which correspond to the model of the Toyota car. One can see that this dataset is very imbalanced.

This figure, Figure 2, shows that the data is very imbalanced when *Toyota Model* is considered to be the target variable. Again, one can see that by far the most cars in the dataset correspond to the Toyota Model 'Yaris', and therefore, the 'Toyota Yaris' present the majority class. Furthermore, one can note that the least number of cars corresponds to the Toyota Model 'Hiace', and thus the 'Toyota Yaris' represent the minority class.

In the next section, I will elaborate on some methods that can be used in order to handle this imbalance.

4 Methodology

In this study, our objective is to identify patterns in the car-purchasing behavior of customers based on their individual characteristics. Specifically, we aim to determine the most probable car choice for a given customer, along with the corresponding characteristics. This research involves a data adjustment task, wherein we aim to handle the imbalance in the dataset, and a classification task, wherein we categorize customers in this dataset based on their preferences. This section provides an overview of the methods employed to address imbalanced datasets and outlines the various classification techniques compared in this study. Additionally, we discuss the performance measures used to assess the effectiveness of the different models.

As discussed in the previous section, the dataset exhibits an uneven distribution across classes related to the target variable, indicating an imbalance. Imbalanced datasets are common in real-world scenarios and can adversely impact the performance of widely used classification techniques in multiclass tasks. Ghanem et al. [2010] highlighted that using imbalanced data may lead to models biased towards majority classes, resulting in inaccurate predictions for minority classes. This bias stems from the fact that classification methods optimize accuracy based on the training data they receive. Moreover, imbalanced datasets hinder the effective learning of distinguishing features for minority classes due to the lack of observations, thereby reducing classification performance. Apart from the challenges posed by imbalanced datasets, there is a risk of misleading evaluation measurements. For instance, accuracy, representing the ratio of correctly classified instances to the total, can be deceptive. Consider a classification model consistently predicting the brand "Toyota" for all cars, irrespective of customer characteristics. If 72 percent of the customers own a Toyota car in reality, the accuracy appears high at 72 percent. However, the model fails to capture the nuances of minority car brands. This illustrates that using imbalanced datasets in multiclass classification tasks may either degrade or artificially inflate model performance. Therefore, addressing imbalanced datasets in multiclass classification is a critical concern for researchers in machine learning, as emphasized in Sahare and Gupta [2012]. Specifically, one can propose modifying the data to reduce the imbalance. Furthermore, one can carefully select classification methods in order to find a method that can sufficiently handle imbalanced datasets, as suggested by Tanha et al. [2020]. Finally, one can adapt the performance measurements in order to determine which methods are optimal.

This section first introduces some frequently used data adjustments to address imbalances. Then, this section will elaborate on which classification methods will be applied and how those methods work. Specifically, we will apply all multiclass classification methods to both adjusted and original datasets in order to analyze the robustness of classification methods and explore the impact of the earlier data adjustments. Finally, all methods will be evaluated by some performance measures, also introduced in this section.

4.1 Data adjustments

As briefly mentioned in the introduction of this section, one approach to dealing with imbalanced data is to adjust the data before applying the classifier. Commonly used methods to turn an imbalanced dataset into a reasonably balanced dataset are oversampling and undersampling. As suggested by Shelke et al. [2017], these techniques are used to adjust the class distribution of a data set. This means that the distribution over classes will be more equal after applying oversampling and undersampling, and thus that the observations in

minority classes will have a greater impact on the classification methods applied later. Note that both over- and undersampling techniques are the complete opposite of each other and roughly equivalent. Furthermore, note that for this research, we will randomly split the dataset into a test and training set, which respectively correspond to 20 percent and 80 percent of the entire dataset. After this split, the test part of the dataset will be used to evaluate and compare the performance of the classification methods, whereas the training part of the dataset will be used to train the classification algorithms. Furthermore, one should keep in mind that both oversampling and undersampling techniques are only applied to the training set. Both data-adjustment techniques will be elaborated on below.

4.1.1 Undersampling

Undersampling methods work by reducing the number of observations corresponding to the majority of class samples. Undersampling can be done either randomly, which is referred to as random undersampling, or by using some statistical model, which is referred to as informed undersampling. Both of these techniques are mentioned in Shelke et al. [2017]. For random undersampling, we first determine which class(es) contain a large number of observations and, thus, which are the majority classes. Then, for each of these majority classes, we will randomly remove observations until the number of observations in these classes is almost equal to the number of observations in minority classes. Then, this dataset is used in classification methods. A disadvantage of random undersampling is that one might lose important information that could contribute to the model's understanding of the data. Furthermore, random undersampling may lead to different results each iteration as the variability of the majority classes increases. This may lead to a decrease in the performance of the model. In order to overcome the disadvantages of random undersampling, informed undersampling methods are proposed. In this research, we will particularly analyze the performance of the Cluster Centroids Undersampling method, which is mentioned in Zhang et al. [2019], as this method is frequently used to handle imbalanced datasets due to its good performance. The Cluster Centroids undersampling method works as follows. First, for the training set, we determine which class is the majority class, or, in other words, which class contains the most observations. Note that we will only assign one class to be the majority class, as with random undersampling, there could be more majority classes. Then, we will perform a clustering algorithm on a subset dataset that contains all observations corresponding to the majority class. In particular, we perform the K-means clustering algorithm as this is simple and efficient, which is in line with the algorithm in Zhang et al. [2019]. The value for K is determined to be tuned, and for this research, it results in an optimal value of 10, independent of the target variable. Then, for each of these K clusters, we determine which observations are near the borders of the clusters. Observations are considered to be border points when the euclidean distance between these points and the centroids of the corresponding cluster is maximized. We will specifically consider these instances, as the border point may contain the most information. Note that the number of border points that we consider is proportional to the size of the cluster and to the size of the minority class. The specific border point picked for each cluster will now replace the observations that correspond to the majority class in the original training set. The aim of this undersampling method is to eliminate observations in the majority class that are not near the center of a cluster. As stated in Zhang et al. [2019] these observations may contain less relevant information for a multiclass classification algorithm to learn from. This undersampling algorithm often outperforms the random undersampling algorithm, as in this case, we choose to delete observations that contain the least amount of information instead of just randomly deleting information. Furthermore, note that for each iteration, the Cluster Centroids undersampled dataset

will be approximately the same, in contrast to the dataset generated by random undersampling. After undersampling, each dataset now consists of more equally distributed observations over classes. Finally, both of these datasets are used to train a classification model.

4.1.2 Oversampling

As briefly mentioned before, oversampling is the complete opposite of undersampling. In oversampling techniques, in order to balance the dataset, new observations are added to the minority class. Again, oversampling can be done either randomly, which is referred to as random oversampling, or by using some statistical model, which is referred to as synthetic oversampling, as mentioned in Shelke et al. [2017]. For the random oversampling technique, the observations in minority classes are randomly replicated in order to increase the size of these classes. Again, the disadvantage of this random approach is that the results of each iteration of the classification models can differ as the variability of the minority classes increases. Furthermore, due to the duplication of observations, the randomly oversampled dataset might cause overfitting. This means that the classification algorithm might start memorizing specific instances rather than learning general patterns, which could result in a model that performs well on the training data but poorly on unseen data. To overcome these disadvantages of random oversampling, synthetic oversampling techniques are introduced. One of the most frequently used synthetic oversampling techniques is the Synthetic Minority Oversampling Technique (SMOTE), which is mentioned in Chawla et al. [2002]. This technique increases the number of observations in the minority class by creating "synthetic" examples rather than by duplicating observations, as is done with random oversampling. The SMOTE algorithm looks as follows. First, again, we identify the minority class in the dataset. Then, we randomly choose an observation from a class and identify its k nearest neighbors. This value for k is a pre-defined parameter equal to five, as is in line with Chawla et al. [2002]. Then, new observations are generated as follows. We start by taking a random observation from the minority class. Then, we will find its five nearest neighbors, based on the Euclidean distance between the points. Note that we look for these nearest neighbors in the subset of observations belonging to the minority class. Then, we draw a line between each of these neighbors and our original observation. We will now create five new observations by taking five random points on each of the previously created lines. These observations will be referred to as synthetic observations and will be added to the training set. Then, we will randomly take a new point from the minority class and repeat these steps until the number of observations in the minority class is equal to the mean of the number of observations in the other classes. One advantage of SMOTE is that each dataset created by SMOTE is fairly similar, whereas the datasets created by random oversampling contain much more variability. Furthermore, SMOTE decreases the risk of overfitting as new observations are added to the dataset instead of adding duplicates, which is in line with statements in Chawla et al. [2002].

One should bear in mind that all the data adjustment techniques mentioned above are implemented in order to increase the performance of classification methods for minority classes. Specifically, improving the distribution of observations over classes increases the impact of the samples in the minority classes, and therefore the classification methods are more likely to discover hidden patterns in the observations from minority classes.

In the next subsection, we will elaborate on what specific classification methods will be used in order to optimize the predictive performance on imbalanced datasets.

4.2 Classification methods

In this section, we will elaborate on the classification methods which will be used in this research to predict the class of a customer's car. As the number of classes for every target variable exceeds two, this section will contain some of the most frequently used, best performing or most time-efficient multiclass classification methods. We will introduce some notation which will be used for every classifier in mentioned in this section. Consider N customers for which customer i is a particular customer and $i = 1, ..., N$. x_i is the vector of characteristics of length K where K denotes the total number of characteristic in the dataset. Then, x_{ik} is a value for customer i and characteristics k where $k = 1, ..., K$. Furthermore, consider Y to be the actual class, and $\hat{Y}$ to be the predicted class of a customer's car. Y and $\hat{Y}$ can take values between 1 and J, where J denotes the total number of classes, depending on the target variable (*Brand* or *Toyota Model*). Furthermore, note that j determines a particular class for which $j = 1, .., J$. Finally, let N_j denote the total number of cars in the training data belonging to class j and let N denote the total number of customers in the training set.

4.2.1 Naive Bayes

The first classification method that will be considered in this research is the Naive Bayes (NB) classifier. As stated in Farid et al. [2014], this classifier is commonly used due to its efficient approach to solving classification problems in data mining. In particular, the NB approach is popular in machine learning fields due to its simplicity and relatively high classification accuracy. From Rennie [2001], we know that the NB classifier is frequently used in research areas like text classification.

The NB approach relies on the assumption that there is independence between features in the dataset when the classes are known. In this research, it means that NB assumes that given the brand or model of a car, the customer characteristics are independent of each other. In the real world, this might not be entirely true, as the characteristics of customers owning a big and luxury car might include a larger income, which might also come with a higher education level. Then, income and education might be correlated, and thus the independence assumption may not hold. However, as this assumption is barely met in real-life data, the NB classifier still works well in many cases. This might be due to the fact that the correlation between characteristics is not strong enough to affect the performance of NB. Furthermore, as briefly mentioned before, the efficiency and simplicity of the classifier of high-dimensional data, such as in this research, can outweigh the limitations of the independence assumption.

The algorithm of the NB classifier looks as follows. First, we calculate the prior probabilities for each class based on the training data. Intuitively, we calculate the probability that a customer's car belongs to a certain class, independent of all its characteristics. This prior probability is calculated as follows.

$$P(Y_i = j) = \frac{N_j}{N}$$

where $P(Y_i = j)$ denotes the prior probability that the class of the car of customer i is j. Next, for each characteristic k of customer i, x_{ik}, and each class, j, we will estimate the likelihood of observing a particular value of x_{ik} given class j. Intuitively, this is calculated by the number of times a value for x_{ik} is equal to v when customer i has a car from class j divided by the total number of customers' corresponding to class j. Mathematically, this will look as follows.

$$P(x_{ik} = v|Y_i = j) = \frac{\sum_{i=1}^{N} I[i \in j] I[x_{ik} = v]}{\sum_{i=1}^{N} I[i \in j]}$$

where $P(x_{ik} = v|Y_i = j)$ is the probability of characteristic x_{ik} being equal to v given that the car of the customer is in class j. Furthermore, $I[.]$ is an indicator function which is 1 when . is true and 0 otherwise. To handle the issue of probabilities being equal to zero when a feature has not been observed in a particular class, we will add smoothing. Then the probability $P(x_{ik} = v|Y_i = j)$ is calculated as follows.

$$P(x_{ik} = v|Y_i = j) = \frac{\sum_{i=1}^{N} I[i \in j] I[x_{ik} = v] + 1}{\sum_{i=1}^{N} I[i \in j] + V_k}$$

where V_k is the number of unique values present in the training set for feature k. Then, we will calculate the posterior probabilities of a customer's car belonging to a particular class j for customers in the test set. This posterior probability is calculated as follows.

$$P(Y_i = j|X) \propto P(Y_i = j) \prod_{j=1}^{J} P(x_{ik} = v|Y_i = j)$$

where $P(Y_i = j|X)$ denotes the probability that a new customer i, obtained from the testing set, with characteristic vector X its car belongs to class j. To provide the final prediction, we take the maximum value of this probability over all classes. This looks as follows.

$$\hat{Y_i} = argmax_j P(Y_i = j|X)$$

The class corresponding to this maximum value represents the final prediction for this customer obtained by applying the NB classifier. This prediction is compared to the actual value of the customer in the testing set.

The NB classifier has proven to be robust, as stated in Aridas et al. [2019]. In the context of this research, being robust means that the NB classifier can handle datasets in which the class imbalance is severe fairly well. In other words, the algorithm tends to provide reasonable predictions even in the presence of imbalanced class distributions. Therefore, we will apply this multiclass classifier to the imbalanced datasets used in this research. Next, we will elaborate on another popular classifying method.

4.2.2 Random Forest

One of the most frequently used methods for multiclass classification problems is the Random Forest (RF) classifier. This machine learning technique has become very popular due to its ability to run efficiently on large datasets. Besides, the classifier performs well on many different research topics and maintains effectiveness even when many observations are missing, as stated in Kulkarni and Sinha [2013].

In general, RF is an ensemble learning algorithm that combines the predictions of several independent decision trees in order to create a final prediction. The algorithm for this classification method looks as follows. The first step of the algorithm is to take a random subsample of the rows of the training set. This is referred to as bootstrapping. As noted by Kulkarni and Sinha [2013], these rows, which correspond to customers, are sampled with replacement. Then, the algorithm randomly selects a subset of characteristics, which are in the columns of the dataset, from the bootstrapped sample of the training set. The number of

variables selected is pre-set and referred to as $mtry$. In general, it holds that $mtry < K$ and therefore, in this research, we will tune this value for $mtry$ for which we obtain a value of 10. The random subsampling of the data and corresponding variables ensures that every decision tree made by the RF algorithm differs. On this particularly selected part of the training set, the algorithm trains a decision tree in which each of the nodes represents a decision that has to be made. For example, a node can represent the decision whether a particular customer in the training set has children older than 4 years old. Then, if the answer is yes, we continue down that part of the tree, whereas if the answer is no, we would have continued down the other part of the tree. The total number of decision trees created by the RF algorithm is equal to the pre-set parameter value referred to as $Ntree$, which will also be tuned. We obtained the optimal value for $Ntree$ being equal to 150. Note that, as in line with Kulkarni and Sinha [2013], we do not prune the trees, and thus each decision tree is grown as large as possible. Once all decision trees are trained, we run all customers from the testing set through all these trees. The final prediction for each testing customer is obtained by taking the class that is most frequently occurring among each of the trees in the RF.

For the application of a RF classifier on an imbalanced dataset, we can conclude the following. As Khalilia et al. [2011] suggests, the performance of an RF classifier does not seem to be largely affected by the application on imbalanced datasets. This might have to do with the fact that RF creates decision trees based on the subsamples of the training set. For these subsamples, the classes might not be as imbalanced as in the entire dataset. Furthermore, the RF algorithm classifies customers based on the majority's predictions. This allows RFs to adapt to imbalanced class distributions. Therefore, we will test this classifier both on the original and data-adjusted datasets in order to determine whether the performance of the RF classifier indeed performs fairly well on an imbalanced dataset or whether we could improve its performance by managing the imbalance of datasets.

4.2.3 Extremely Gradient Boosting

As one can note, the RF algorithm creates independent decision trees. The aim of the next classifier, referred to as Extremely Gradient Boosting (XGBoost), is to improve previously constructed decision trees by considering their mistakes. This creates a forest of dependent decision trees. Instead of creating $Ntry$ decision trees at the same time, as is done by the RF algorithm, XGBoost sequentially creates new trees. This technique is known as boosting, as is mentioned in Le et al. [2022]. Specifically, new trees are created by minimizing the loss function during the training of a particular decision tree. Due to its capability to learn from its own mistakes, XGBoost is expected to be highly efficient. Furthermore, XGBoost is applicable to large datasets due to its fast implementation and can handle datasets with missing values. Therefore, XGBoost is a frequently applied multiclass classification algorithm in modern research.

The algorithm of the XGBoost classifier in this research is in line with the notation in Le et al. [2022] and looks as follows. First, we initialize the model, creating a basic prediction. In this research, we take the mode of the target variable to be the basic prediction. The mode refers to the most occurring value over all observations in the training set. This means that the model starts by always predicting the majority class in our imbalanced dataset. This looks as follows.

$$Y_{start} = argmax_j \sum_{i=1}^{N} I[Y_i = j]$$

for which $I[Y_i = j]$ is an indicator function that is equal to 1 if class Y of customer i in the training set is equal to class j. If we have performed either random undersampling or oversampling and there is no majority class anymore, we randomly take a starting point. This starting point is then less critical as the classes are equally represented. Note that in both cases, the starting class, Y_{start}, is equal for every customer.

Next, we create a function $L(\phi)$ which is referred to as the regularized objective function. The XGBoost algorithm aims to minimize this function, which is the sum of a loss function and regularization terms in order to improve the previous predicted class of customer i. Therefore, in the first iteration, $\hat{Y}_i$ will be set equal to Y_{start}. The regularized objective function looks as follows.

$$L(\phi) = \sum_{i=1}^{N} l(Y_i, \hat{Y}_i) + \sum_{t=1}^{T} \Omega(f_t)$$

In line with the notation in this entire section, the value for N in this formula denotes the total number of observations, and thus customers, in the training set. Furthermore, $l(.)$ denotes the softmax loss function, and thus $l(Y_i, \hat{Y}_i)$ denotes the softmax loss function value for the actual class, Y_i, and the predicted class, $\hat{Y}_i$ of customer i. As stated in Baldo et al. [2023], the softmax loss function is frequently used for XGBoost algorithms in multiclass classification problems. The softmax function is applied to the raw output scores, referred to as logits, for each class. Specifically, the function converts these scores into probabilities, ensuring that they sum up to 1. Therefore, we can state that the softmax function is used to transform the problem into a probability distribution over the classes. The softmax function looks as follows

$$l(Y_i, \hat{Y}_i) = -\sum_{j=1}^{J} I[Y_i = j] log(P(Y_i = j))$$

Specifically, again, $I[.]$ is an indicator function which is equal to 1 if . is true, and $P(Y_i = j)$ is the probability of the input example belonging to class j. In this case, this probability is calculated as follows.

$$P(Y_i = j) = \frac{e^{z_j}}{\sum_{j=1}^{J} e^{z_j}}$$

where e^{z_j} denotes the exponent of the raw logit score for class j. Furthermore, the formula for the objective loss function contains a function referred to as $\Omega(f_t)$. This function denotes the regularization term for the t-th tree in the ensemble. As mentioned in Cawley and Talbot [2007], this regularization helps prevent overfitting by penalizing overly complex trees. It encourages the model to favor simpler models that generalize well to unseen data. In particular, the value for $\Omega(f_t)$ is calculated as follows.

$$\Omega(f_t) = \theta T + \frac{1}{2} \lambda \sum_{l=1}^{L_t} w_l^2$$

In this formula, the value for L_t denotes the total number of leaves in the t'th tree, and w_l denotes the weight of the l'th leaf in the t'th tree. Furthermore, θ denotes the regularization parameters for the number of leaves, and λ is the regularization parameter for the leaf weights. Note that the first term, θT penalizes the complexity of the tree by controlling the number of leaves. Intuitively, this means that a larger θ leads

to shallower trees. Besides, note that the second term, $\frac{1}{2}\lambda \sum_{l=1}^{L_t} w_l^2$ penalizes the magnitude of the scores associated with the leaves. This means that larger values of λ lead to smaller leaf scores. One should note that both regularization parameters should be tuned for optimal results. After tuning, we find that $\lambda = 0.8$ and $\theta = 1$

Now that we have constructed the objective function, $L(\phi)$, we should minimize this function by using gradient descent. For this minimization over iterations, we introduce a learning rate. This rate controls the step size during the gradient descent optimization. A lower learning rate requires more boosting rounds but may lead to better generalization. Note that for this research, this value is tuned for optimal performance of the XGBoost classifier, which results in an optimal value equal to 0.3, independent of the target variable.

Next, we note that the final prediction for observation i from the test set is generated by summing the predictions from all trees T and applying the softmax transformation. This looks as follows.

$$\hat{Y}_i = l(\sum_{T}^{t=1} \hat{Y}_{ti})$$

.

As mentioned before, the algorithm minimizes the gradient of the loss function with respect to the model's predicted probabilities in order to train a new decision tree to correct the errors of the forest containing all previous trees. In contrast to the RF algorithm, the XGBoost algorithm applies tree pruning in order to remove branches that do not contribute to the reduction of the objective loss function. This also decreases the complexity of the trees in the forest.

The XGBoost algorithm stops when a stopping criteria is met. In this research, we will tune a hyperparameter, which denotes the total number of boosting rounds. The total number of rounds is found optimal at a value equal to 50 and thus, the XGBoost algorithm will stop training once it reaches a total of 50 trees.

If we consider the performance of the XGBoost classifier on imbalanced data, we will conclude the following. As stated in Le et al. [2022], one of the major advantages of the XGBoost classifier is that the algorithm is capable of scaling imbalanced data. This might be due to the fact that, as briefly mentioned with RF classification, trees can naturally adapt to imbalanced datasets because they are able to focus on minority classes during their construction. Furthermore, the XGBoost algorithm avoids overfitting on majority classes due to the regulation terms and the ability to prune the trees. This will be beneficial when applied to both the random oversampled and random undersampled datasets, in which many duplicates can occur. For these datasets, the XGBoost classifier will be better able to capture the general patterns in the dataset.

4.2.4 Support Vector Machines

Another frequently used classification algorithm is Support Vector Machines (SVM). As mentioned in Aly [2005], basic SVM is mainly used for binary classification. This is related to the fact that the SVM algorithm involves training a model to find a hyperplane that separates the data into two classes. The primary goal is to maximize the margin between the classes while minimizing classification errors. However, as suggested by Milgram et al. [2006], a combination of several SVMs can be used to solve a multiclass classification problem.

Nonetheless, as stated in Debnath et al. [2004], this extension on the binary classification is an ongoing research issue. One of the most suitable ways to combine several basic SVMs is by means of the one-against-one method. In particular, this method trains a binary classifier for every pair of classes available in the dataset. In this research, this means that this method trains $\frac{J(J-1)}{2}$ classifiers. Then, if all these classifiers are trained, they will provide their own predictions. As with the RF classifier, the class that receives the most "votes" is the final prediction.

The algorithm of the basic binary SVM classifier, for which the notation is somewhat in line with Debnath et al. [2004], looks as follows. As we have two classes for binary classification, we assign labels to these clusters as follows, $Y_i \in -1, 1$. Then, we split the dataset into a testing and training set. For the training set, we aim to find a hyperplane that divides the customers as accurately as possible over the two labels. The function that creates the hyperplane is formulated as follows

$$f(\mathbf{x}) = \phi(\mathbf{x})\mathbf{w} + b$$

where $\mathbf{x}$ are the values for the variables for customers in the training set and $\phi(x)$ is a nonlinear mapping function of $\mathbf{x}$ into the feature space. Furthermore, $\mathbf{w}$ is a weight vector, and b is a bias term. As briefly mentioned before, the aim of a SVM is to find a hyperplane such that the distance between the hyperplane and both classes is maximized. This is equivalent to the following optimization problem

$$\begin{aligned} min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \\ s.t. f(\mathbf{x}) = \phi(\mathbf{x})\mathbf{w} + b \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, .., N \end{aligned}$$

where C is the regularization parameter to balance the maximum margin and the minimum classification error. For optimal performance of the algorithm, this hyperparameter should be tuned. The optimal value is found to be equal to 1, independent of the target variable. Furthermore, $||\mathbf{w}||^2$ denotes the norm of the weight vector $\mathbf{w}$.

If we have found the optimal hyperplane between two classes and their corresponding labels, we run all observations in the testing set through the formula of the hyperplane as the new $\mathbf{x}$. Then we classify these new customers as follows: if $f(\mathbf{x}_i) > 0$, then the label of the class of this new customer is 1. Additionally, if $f(\mathbf{x}_i) < 0$, then the label of the class of this new customer is -1.

In order to create the final prediction for the multiclass classification problem in this research, we combined the predictions of all binary SVMs. Then, we will denote for every customer in the testing set the most occurring prediction, which we will then denote as the final class prediction. To make this more intuitive, we consider a dataset that contains three classes: 1, 2, and 3. Then we run the SVM algorithm for the sets 1,2, 2,3, and 1,3, which we all turn into -1,1, respectively. Consider customer i which obtains the votes 1, -1, and 1 for the three models, respectively. We then conclude that the final prediction for this customer is that it belongs to class 2.

If we consider the performance of multiclass SVM on an imbalanced dataset, we conclude the following. As stated by Batuwita and Palade [2013], SVM often produces effective solutions for balanced datasets but

may indeed be sensitive to imbalanced datasets and may therefore produce suboptimal predictions. This can be explained as follows. Consider the hyperparameter C to represent the cost of misclassified customers. As we aim to minimize these costs, SVM might excessively often predict a customer to belong to the majority class and will therefore be biased towards the majority class. Therefore, we will perform this classifier on both the original and the data-adjusted datasets in order to see the difference in performance.

In the next subsection, we will elaborate on how to interpret the results of the data-adjustment and multiclass classification techniques mentioned above.

4.3 Performance measurements

As we now have predicted the customer's interest in particular cars, we will need to evaluate some performance measures in order to determine whether the data-adjustment and classification techniques used in this research are performing well. As mentioned in Purwandari et al. [2021], the most commonly used performance measures for this kind of research are accuracy, precision, recall, F1-score, and the Area Under the (AUC) Receiver Operating Characteristic (ROC) curve. In this section, I will elaborate on how those five performance measurements are calculated, how to interpret them, and how they behave when applied to the results of a classification algorithm on an imbalanced dataset. We will first introduce some notation, which will be used in the majority of measurements and is in line with notation in Sokolova et al. [2006].

In this section, we will use the following notation for consistency. First, we introduce TP_j, which denotes the true positives for class j. In particular, this value denotes the number of customers for which their predicted class and actual class are j. One should note that the predicted class is the class of an observation that is predicted by a particular classification method. Next, we consider TN_j, which denotes the true negatives for class j. Specifically, this value counts the number of customers for which their predicted class and actual class are not j. Besides, we introduce FP_j, which denotes the false positives for class j. This value denotes the number of customers for which their predicted class is j but their actual class is not j. Finally, we will introduce FN_j, which denotes the false negatives for class j. In particular, this value counts the number of customers for which their predicted class is not j but their actual class is j. All values mentioned above can be displayed as follows.

Actual class \Predicted class	j	not j
j	TP_j	FN_j
not j	FP_j	TN_j

Table 6: This table shows how we determine values for TP, FN, FP, and TN for a class j. For example, this table displays that the value for TP_j determines the number of observations for which the actual and predicted class are equal to j.

As we now have introduced some notation, we will mention the performance measures which will be used in this research to analyse the performance of both the data-adjustment and classification techniques.

4.3.1 Accuracy

We will start with the most generally applied performance measure in multiclass classification, due to its simplicity, as mentioned in Sokolova et al. [2006]. Intuitively, the aim of a classification method is to predict the actual class of each observation as accurately as possible. Therefore, we will simply count the number of correctly classified observations and divide this number by the total number of observations. This gives a ratio of correctly classified observations. In the case of multiclass classification, the accuracy of a model is calculated by the average accuracy of all classes. Specifically, the accuracy of a particular class j is calculated

by $\frac{TP_j+TN_j}{TP_j+TN_j+FP_j+FN_j}$. Then, the accuracy of the model is calculated by

$$accuracy = \frac{1}{J}\sum_{j=1}^{J}\frac{TP_j + TN_j}{TP_j + TN_j + FP_j + FN_j}$$

and denotes the overall correctness of the model, represented by the ratio of correctly predicted instances to the total instances. Note that optimal accuracy is equal to 1, which denotes that all observations are correctly classified. The minimum value of accuracy is 0, which denotes that all observations are incorrectly classified.

Accuracy is an intuitive measurement. However, as briefly mentioned before, the accuracy does not work well for an imbalanced dataset. This is due to the fact that accuracy does not denote the degree to which a model can understand the underlaying pattern among observations. This can be explained by the previously mentioned example, in which a model predicts all classes to be equal to the majority class. Then, we denote a fairly high level of accuracy, especially when the majority class is large, although the method has not understood any of the underlaying patterns. Therefore, we will also consider other performance measurements.

An adaptation that can be made to the previously mentioned accuracy measure is to analyze the accuracy specifically for the minority class. In particular, this means that we will only consider the observations that actually belong to the minority class and their predicted classes. This adaptation is useful as the aim of this research is to improve the performance of multiclass classification methods on minority classes. Furthermore, the accuracy of observations in the minority class can give insight into the ability of the classification method to discover underlaying patterns. The accuracy of samples in the minority class is calculated as follows. If we denote that class p is the minority class, then

$$accuracyminority = \frac{TP_p + TN_p}{TP_p + TN_p + FP_p + FN_p}$$

.

4.3.2 Precision, recall and F1 score

Besides evaluating the accuracy of the classification techniques, we will also consider some other measurements in order to create a broad view of which techniques have actually improved the performance of the classification of the original, imbalanced dataset.

Therefore, we will first examine the precision of the model. Intuitively, this measurement denotes the accuracy of the positive predictions of every class. This means that for every predicted class being equal to j, we count the number of observations actually being equal to j. In mathematical notation, the precision of a particular class j is calculated by $\frac{TP_j}{TP_j+FP_j}$. Then, the precision of the entire model is calculated by

$$precision = \frac{1}{J}\sum_{j=1}^{J}\frac{TP_j}{TP_j + FP_j}$$

. Using precision as an evaluation criteria is especially useful when the cost of false positives is high. For example, this is the case when we consider a medical diagnostic scenario where we predict whether or not

a customer has a very innocent disease based on several symptoms. However, we know that treating this disease is very expensive and causes lots of side effects for the patient. Then, false positives, which occur when we incorrectly diagnose a healthy patient as having the disease, could lead to unnecessary medical interventions. In this research, the precision may not be robust when applied to imbalanced datasets, as these measurements may prioritize observations from the majority class, as is in line with the accuracy mentioned above. Therefore, we will consider some other measurements below.

Next, we consider the recall of the model. The value of the recall denotes the effectiveness of the classifier to capture the positive predictions of observations belonging to a particular class. Specifically, the recall divides the number of true positives by the total number of observations that actually belong to the particular class. This means that for every actual class being equal to j, we count the number of observations predicted to belong to class j. This means that the recall of a particular class j is calculated by $\frac{TP_j}{TP_j+FN_j}$. Then, the recall of the entire model is calculated by

$$recall = \frac{1}{J}\sum_{j=1}^{J}\frac{TP_j}{TP_j + FN_j}$$

. This measurement is especially useful when the cost of false negatives is high. For example, this is the case when we again consider the medical diagnostic scenario mentioned above. However, the aim of this model is now to predict whether or not a customer has a very severe disease based on several symptoms. Note that when a patient is not predicted to not have the disease but has it anyway, the patient is likely to die. Then, false negatives could lead to a patient suffering, and thus the cost of false negatives is very high in this case. In this research, the recall is expected to be more robust to imbalanced datasets compared to precision. This might be due to the fact that the recall focuses on capturing all positive instances and might therefore be less affected by imbalanced distributions.

One should note from the recall and precision that both are inversely related. This means that maximizing recall may result in lower precision, and vice versa. Analyzing the trade-off between both measurements, capturing all relevant instances, and avoiding false positives can be interesting in order to determine the optimal data adjustments and classification techniques. Therefore, we will consider the following performance measurement, referred to as the F1 score.

The F1-score of the model, is a evaluation criteria which balances the values of both the precision and recall measurements. As is stated in Sokolova et al. [2006], the F1 score can be calculated as follows.

$$F1score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

. As this measurement balances between precision and recall, the F1 score will penalize the classifier for both types of errors, providing a more balanced evaluation even if one class is heavily underrepresented in the dataset. This makes that the F1 score is a good performance measurement to evaluate in this research.

One should note that all three measurements mentioned above—precision, recall, and F1 score—all range between 0 and 1. Specifically, a value of 1 for all measurements denotes perfect classification, whereas a value equal to 0 denotes very bad classification.

4.3.3 Area under the receiver operating characteristic curve

The final evaluation measurement that we analyze in this research is the area under the receiver operating characteristic (ROC) curve, which is referred to as AUC-ROC. As stated in Hameed et al. [2022], this criteria is a frequently used performance measurement in classification tasks.

In particular, the ROC curve is a graphical representation of the trade-off between the true positive rate and the false positive rate. The graph shows how well a particular classifier distinguishes between classes by plotting the true positive rate against the false positive rate at different classification thresholds. Specifically, this true positive rate (TPR) denotes the proportion of actual positive instances that are correctly identified by the classifier. This value is mentioned before and is referred to as the recall of a model. Furthermore, the false positive rate (FPR) denotes the proportion of actual negative instances that are incorrectly classified as positive by the classifier. The FPR can be calculated as follows.

$$FPR = 1 - \frac{1}{J} \sum_{j=1}^{J} \frac{TN_j}{TN_j + FP_j}$$

Then, the ROC curve shows the trade-off between the true positive rate and the false positive rate, plotted at certain thresholds. In particular, the threshold represents the point at which the classifier decides whether an instance should be classified as positive or negative based on its predicted probability.

The area under this ROC curve (AUC-ROC) is calculated in order to quantify the overall performance of the classifier. A larger AUC-ROC value indicates better performance, with 1 being the highest possible value (a perfect classifier) and 0.5 indicating a random classifier. Intuitively, a higher AUC-ROC value suggests that the classifier is better at ranking positive instances higher than negative instances, regardless of the class distribution. As stated in Jeni et al. [2013], this makes AUC-ROC a reliable performance measure, especially in scenarios with imbalanced datasets where the class distribution is skewed.

In the next section, we will show the results of the application of the data-adjustment and classification techniques mentioned above on the original, imbalanced dataset. Especially, we will focus on the most robust performance measurements, which are the accuracy of the minority class, the F1 score, and the AUC-ROC.

5 Results

In this section, I will present the outcomes of the aforementioned techniques applied to the original, imbalanced dataset. Initially, I will showcase the results of applying both undersampling and oversampling methods. Specifically, we will contrast the features of the original dataset with those of the undersampled and oversampled datasets, emphasizing alterations in the distribution across the classes corresponding to different target variables. Subsequently, our attention will be directed towards evaluating the performance of multiclass classification methods on these diverse datasets. We will analyze and compare performances using the metrics outlined in the previous section.

5.1 Data adjustments

As briefly mentioned above, in this section, we will focus on the resulting datasets after applying undersampling and oversampling methods to the original, imbalanced dataset. As the data is split into a test and training set, and the data adjustments will only be made on the training set, this section will particularly present the characteristics of the training sets. First, we will evaluate the distribution of observations over the classes in the original dataset. Note that the classes, and thus the number of classes, depend on whether we take *Brand* or *Toyota Model* to be our target variable for classification. Then, we will compare the characteristics of the original training set to the characteristics of the resulting training sets after random oversampling, SMOTE, random undersampling, and cluster centroids undersampling. The results of the data adjustments on the classes corresponding to the target variables *Brand* and *Toyota Model* are represented in Table 7 and Table 8 respectively.

	Original	Oversampling		Undersampling	
		Random	SMOTE	Random	Cluster Centroids
Target variable	Brand	Brand	Brand	Brand	Brand
Number of classes	3	3	3	3	3
Total number of observations	401575	873315	528405	34590	147293
Number of observations majority class	291105	291105	291105	11530	36823
As percentage of total number of observations	72.49	33.33	55.09	33.33	24.99
Number of observations minority class	11530	291105	138360	11530	11530
As percentage of total number of observations	2.87	33.33	26.18	33.33	7.83

Table 7: This table shows the distributional results of performing the different data adjustments on the original, imbalanced dataset when we consider *Brand* to be the target variables for classification. For example, one can denote the total number of observations, the size of the majority class, and the size of the minority class for the original and data-adjusted training sets.

The analysis of Table 7 reveals several noteworthy observations. First and foremost, one should bear in mind that the application of data adjustments does not alter the number of classes for the target variable *Brand*; the number of original classes remains intact. Additionally, it is worth noting that, in the original training set, there exists a substantial disparity of approximately 280.000 observations between the majority and minority classes. Relatively, this means that over 72 percent of the observations correspond to the majority class, whereas almost 3 percent of the observations correspond to the minority class. Therefore, we can conclude that this research, when considering the *Brand* of a car to be the target variable for classification, deals with a highly imbalanced training set.

As we deal with a highly imbalanced dataset, we can perform oversampling in order to balance this dataset. As previously mentioned, oversampling leads to an augmentation of the dataset due to the introduction of additional observations. If we consider the training set obtained by random oversampling, as can be seen in Table 7, we note that the implementation of this method increases the total number of observations by 117 percent. Besides, applying the SMOTE results in a 37 percent increase in the number of observations. Examining Table 7, it is evident that both oversampling techniques contribute to a reduction in the percentage of observations corresponding to the majority class while simultaneously increasing the percentage corresponding to the minority class. This contributes to a notable reduction in dataset imbalance. Furthermore, denote that random oversampling achieves a complete balance between the minority and majority classes, as indicated by equal percentages for both in the resulting dataset.

If we consider the undersampling methods, we denote the following. Conversely to the previously mentioned oversampling techniques, both undersampling models remove observations from the dataset, which leads to a decrease in the total number of observations. Specifically, the random undersampling method results in a 91 percent reduction, while the cluster centroid undersampling method reduces the number of observations by approximately 63 percent. Again, note that both undersampling techniques contribute to a reduction in the percentage of observations corresponding to the majority class while simultaneously increasing the percentage corresponding to the minority class. Finally, examining Table 7 it is evident that random undersampling creates perfectly balanced classes, with equal percentages for the minority and majority classes in the resultant dataset.

Next, we will consider the resulting training sets when we consider *Toyota Model* to be the target variable for classification. The characteristics of these particular training sets are represented below.

	Original	Oversampling		Undersampling	
		Random	SMOTE	Random	Cluster Centroids
Target variable	Toyota Model	Toyota Model	Toyota Model	Toyota Model	Toyota Model
Number of classes	27	27	27	27	27
Total number of observations	230129	1829466	238589	11421	168385
Number of observations majority class	42530	67758	42530	423	25661
As percentage of total number of observations	18.48	3.70	17.82	3.70	15.24
Number of observations minority class	423	67758	436	423	423
As percentage of total number of observations	0.18	3.70	0.18	3.70	0.25

Table 8: This table shows the distributional results of performing the different data adjustments on the original, imbalanced dataset when we consider *Toyota Model* to be the target variables for classification. For example, one can denote the total number of observations, the size of the majority class, and the size of the minority class for the original and data-adjusted training sets.

The examination of Table 8 also yields notable insights, mirroring findings from the preceding table (Table 7). Again, it is observed that the application of data adjustments to the original training set does not alter the number of classes for the target variable *Toyota Model*. Moreover, the initial training set exhibits a significant imbalance, as evidenced by an approximate difference of 42.000 observations between the majority and minority classes.

Furthermore, the same table indicates that oversampling methods once again contribute to the expansion of the dataset. Specifically, random oversampling results in a remarkable 695 percent increase in the number of observations, while SMOTE induces an almost 4 percent augmentation. The large difference between the sizes of the two oversampled datasets might be due to the large number of distinct classes for the target variable *Toyota Brand*. Besides, one can note that both oversampling methods increase the relative amount of observation corresponding to the minority class and similarly decrease the amount of observation corresponding to the majority class. Again, we can see that the random oversampling methods create a perfectly balanced dataset for which each of the classes is equally large.

Turning attention to undersampling methods, it is noted that both random undersampling and cluster centroids lead to a reduction in sample size, with decreases of approximately 95 percent and 27 percent, respectively. Again, it is crucial to highlight that both undersampling techniques create a more balanced training set. Furthermore, one should note that random undersampling results in a perfectly balanced train-

ing set, which is manifested by an equalization of the sizes of the majority and minority classes.

In the following subsection, we will use the previously mentioned training set for classification in order to see what data-adjustment technique improves the performance of classification techniques on imbalanced datasets.

5.2 Multiclass Classifiers

In this subsection, we will analyze and compare the performance of different multiclass classification methods on the original and previously mentioned adjusted datasets. We will first consider the datasets and classification methods when the *Brand* of a car is set to be the target variable. The performance of the models will be measured by the previously mentioned evaluation criteria. The results of the application of these methods and criteria are represented in the table below, Table 9.

		Original	Oversampling		Undersampling	
	Measurement		Random	SMOTE	Random	Cluster Centroids
NB	Accuracy	0.362	0.253	0.278	0.254	0.454
	Accuracy Minority	0.567	0.666	0.654	0.664	0.630
	F1 score	0.405	0.403	0.399	0.403	0.323
	Recall	0.452	0.452	0.442	0.451	0.315
	Precision	0.366	0.365	0.363	0.364	0.331
	AUC-ROC	0.571	0.577	0.563	0.575	0.567
RF	Accuracy	0.731	0.726	0.731	0.350	0.315
	Accuracy Minority	0.040	0.043	0.040	0.566	0.087
	F1 score	0.438	0.427	0.439	0.407	0.390
	Recall	0.363	0.369	0.363	0.457	0.335
	Precision	0.552	0.506	0.554	0.367	0.466
	AUC-ROC	0.511	0.509	0.511	0.582	0.529
XGBoost	Accuracy	0.726	0.428	0.726	0.371	0.324
	Accuracy Minority	0.004	0.453	0.006	0.550	0.081
	F1 score	0.427	0.421	0.443	0.409	0.388
	Recall	0.338	0.468	0.337	0.459	0.370
	Precision	0.579	0.383	0.645	0.369	0.407
	AUC-ROC	0.501	0.582	0.501	0.594	0.527
SVM	Accuracy	0.656	0.355	0.587	0.373	0.342
	Accuracy Minority	0.000	0.527	0.246	0.533	0.015
	F1 score	0.357	0.378	0.360	0.382	0.356
	Recall	0.347	0.411	0.367	0.413	0.358
	Precision	0.368	0.351	0.353	0.355	0.355
	AUC-ROC	0.491	0.554	0.515	0.559	0.515

Table 9: This table shows the results of the application of all multiclass classifiers when trained on the original, imbalanced training set and on the different adjusted training set when *Brand* is considered to be the target variable. For example, one can conclude from this table that when the NB classifier is trained on the original dataset, 36 percent of the observations in the test set are correctly classified when we consider the brands of the cars.

In analyzing this table, we will first focus on the performance of the NB classifier on the original, oversampled, and undersampled datasets.

5.2.1 NB classifier on target variable *Brand*

First, we consider the accuracy of the NB classifier. As mentioned before, on an unbalanced dataset, the accuracy can be biased toward the majority classes and thus might not be a helpful performance measure. However, we will consider the accuracy as it is simply the percentage of correctly classified customers and thus might be useful in particular cases. As one can see from Table 9, the accuracy obtained by applying NB to the original dataset is equal to 0.362. This means that over 36 percent of the customers are combined with the right car brand based on their demographic characteristics. One should note that if we performed this combination randomly, we would find an accuracy of approximately 33 percent. Therefore, we can conclude that performing the NB classifier on the original, unbalanced dataset to predict the brand of a customer's car increases accuracy in comparison to classifying the customers randomly. From the same table, we can see that performing NB on the random oversampled and SMOTE datasets corresponds to lower accuracies, respectively around 25 and 28 percent. We can even conclude that classifying the customers randomly over the classes is preferred over using NB on the random oversampled and SMOTE datasets. This might be due to the fact that the NB classifier relies on the assumption that all features are independent, given the class of a customer. However, in this dataset, which contains features like income, education, and the age of children, we can immediately denote that there is clear dependence between the features. If we increase the number of observations, as is done with oversampling, we might increase the dependence between features. This might cause the accuracy of the NB on these particular datasets to be low. Furthermore, we denote that the accuracy of the NB classifier on the dataset, which is undersampled by means of the cluster centroids, is fairly high. Specifically, this accuracy is equal to 0.454 which denotes that over 45 percent of the customers are correctly classified. This remarkable high accuracy, in comparison to the other accuracies, might be explained by the algorithm of the cluster centroids undersampling method. As mentioned in the previous section, cluster centroids undersampling generates centroids of clusters of observations in order to capture essential characteristics of the majority class. Intuitively, these centroids represent the "boundaries" of certain groups of observations. By taking these border points, the dependence between features might reduce, and thus the assumption of independent features might be more likely to be preserved. This makes that the NB classifiers is more suitable to this dataset than to other datasets and thus causes that the accuracy of NB on the cluster centroids undersampled datasets is this the highest. However, as mentioned before, the accuracy of the models might be biased, and therefore we will also consider different performance measurements.

Next, we will focus on assessing the accuracy specifically for observations belonging to the minority class, which in this case are cars of the brand "Lexus". It's evident that across all data-adjusted datasets, the accuracy of predicting a car's brand as "Lexus" surpasses the accuracy achieved for the minority class in the original, imbalanced training set. This indicates that utilizing a more balanced dataset resulting from the data adjustments proves advantageous for accurately classifying observations within the minority class when paired with the NB classifier.

If we consider the F1 score obtained by the performance of NB on all datasets, we conclude the following. As can be seen from Table 9, all F1 scores are fairly close to each other. Only the F1 score obtained by performing the NB classifier on the cluster centroids undersampled dataset, equal to 0.32, is fairly lower than the rest. This is remarkable, as this dataset corresponded to the highest accuracy earlier. As mentioned before, the F1 score represents the balance between recall and precision. Therefore, the low F1 score obtained by the cluster centroids in the undersampled dataset is caused by the low recall value, which is equal to

0.315. This value denotes that, on average, for all three brands, we predicted that 32 percent of all instances truly belonging to each class were predicted correctly. This value for the recall is higher for the application of NB on all other datasets. The recall might be this low due to the decrease in observations belonging to the majority class by taking the "borders" of groups of observations. By undersampling in this way, the dataset loses diversity, as we do not consider points near the centroids, in contrast to random undersampling, where diversity is preserved. Therefore, certain characteristics might only be represented by border points of clusters of the original data, which may lead to a model that cannot sufficiently capture the patterns in the data. This leads to a model that is way too severe for classifying customers. Intuitively, this means that there are more customers in the test set who can be classified into a particular class but are not in reality due to the lack of diversity in the training set. This leads to a lower recall for the cluster centroids undersampled dataset, which subsequently leads to a lower F1 score for this exact same dataset.

Finally, we will consider the AUC-ROC for the NB model. We note that all values are fairly close to each other, around 0.57. This means that the NB classifier is beneficial when performed on all datasets in comparison to random classification, which provides an AUC-ROC value of 0.5. Specifically, note that both the random oversampled and undersampled datasets perform just better than the other datasets.

If we consider all the results of the NB classifier, we note that both the random oversampled and random undersampled datasets provide the best and most consistent results. However, if we would simply like the accuracy and thus the number of correctly classified customers to be optimal, we prefer the cluster centroids undersampled dataset.

Next, we will consider the performance of the RF classifier on the original, oversampled, and undersampled datasets.

5.2.2 RF classifier on target variable *Brand*

First, considering the accuracy of the models, we note that both the application of RF on the original and oversampled datasets give an accuracy of around 0.73. This means that 73 percent of the customers are matched with the correct car brand. This is much higher than the accuracy obtained by the NB classifier. This might be caused by the fact that RF is an ensemble method that combines multiple decision trees to make predictions. Therefore, a RF classifier can capture complex relationships in datasets. As mentioned before, NB relies on the assumption of feature independence. This assumption might not hold in this dataset, and therefore, in general, RF provides a higher level of accuracy. However, as can be seen in Table 9, the accuracy of implementing RF on the undersampled datasets is much lower. This might be due to the fact that, with undersampling, we delete information from the dataset. This is caused by either randomly deleting observations from observations in the not-minority classes, which is done by random undersampling, or by only taking the border points of clusters in the observations in the majority class, which is done by cluster centroids undersampling. In both cases, the loss of information might lead to a model that is less able to capture important patterns in the dataset. This may cause lower accuracy. Furthermore, by deleting observations, but mainly by taking the border points to be the new observations, we reduce the variation between observations and, in general, the total number of observations in the dataset. This reduced diversity may also limit the ability of the RF classifier to learn the full range of patterns present in the undersampled

datasets, as each tree is tuned on a bootstrapped sample of the training set. Therefore, the cluster centroids undersampled dataset corresponds to an even lower accuracy than the random undersampled dataset.

Then, we will consider the accuracy for just the minority class. For this measurement, we note that, again, the accuracy provided by the original, imbalanced dataset is the lowest of all accuracies corresponding to the minority class. This means that for an imbalanced dataset, the RF classifier is not able to reveal much of the hidden patterns among observations that correspond to underrepresented classes. Furthermore, we note that the random oversampled and random undersampled datasets, which are both perfectly balanced, provide optimal accuracy for the minority instances when combined with the RF classifier.

Next, we will consider the F1 score of the RF classifier on the datasets. Again, we see that the F1 scores of applying RF to the original and oversampled datasets are close to each other. Besides, the F1 scores obtained by the undersampled datasets are again fairly lower. In the case of the random undersampled dataset, the low F1 score is caused by the low value for precision. This means that the RF classifier for this dataset, on average, suggests that a significant portion of instances predicted as positive for a particular class are actually negative. This might be caused by the immense loss of information when performing random undersampling; 91 percent of the observations are deleted. As the cluster centroids undersampled dataset also decreases the number of observations, but only of the majority class, the precision of this model is somewhat higher.

Finally, we consider the AUC-ROC performance measure for the RF classifier. One should note that Table 9 shows that the performance of RF on a random and cluster centroids undersampled datasets provide the highest AUC-ROC. These high values might be caused by the deletion of many instances, which makes the amount of noise decrease, which leads to the highest AUC-ROC.

For the application of the RF classifier on imbalanced data, we can conclude the following. First, we note that oversampling the imbalanced dataset by means of the SMOTE algorithm and then applying an RF classifier provides the best and most consistent results when the aim is to optimize the overall accuracy and F1 score. However, one can note from Table 9 that using the random undersampled dataset provides optimal accuracy for the minority class and an optimal AUC-ROC value. This means that the random undersampled dataset, when combined with the RF classifier, can classify the samples of the minority class fairly accurately and is therefore able to discriminate between classes better than other datasets.

Third, we will consider the performance of XGBoost on the particular datasets.

5.2.3 XGBoost classifier on target variable *Brand*

Again, as we consider Table 9, we note that the accuracy of applying the XGBoost classifier to the original and SMOTE datasets is optimal. Furthermore, we notice that now not only the accuracy of the undersampled datasets is low, but also the accuracy of the performance of XGBoost on the random oversampled dataset is much lower. This might be due to the fact that random oversampling involves randomly duplicating instances from the minority classes to balance the class distribution. This process may lead to information redundancy as it simply replicates existing instances. The model may struggle to generalize well to new, unseen instances, resulting in lower accuracy. SMOTE, on the other hand, generates synthetic instances for

the minority class, introducing new data points that are not exact duplicates of existing instances. This can contribute to a more diverse and representative dataset, potentially leading to better model generalization and higher accuracy. The fact that the random oversampled dataset provides high accuracy with the RF classifier but fairly lower accuracy with the XGBoost classifier can be explained as follows. As mentioned before, for XGBoost, each tree corrects the errors of the previous ones. If there are duplicates in the data, it might emphasize certain patterns excessively, potentially leading to overfitting. The RF algorithm constructs the decision trees independently which means that, due to bootstrapping, the model naturally introduces diversity into the trees. Therefore, we can conclude that XGBoost is more adversely affected by duplicate observations compared to RF. Next, the low accuracy provided by both undersampled datasets can, again, be explained by the loss of information for both datasets, which is in line with the low accuracy for both models when the RF classifier is applied.

Then, we will consider the accuracy of instances belonging to the minority class. By doing so, we can denote that, as in line with the results for the RF classifier, the accuracy for minority classes is optimal when the XGBoost classifier is trained on a perfectly balanced dataset. This might be due to the fact that, in this case, the model has enough observations to reveal hidden patterns in the data, even for the minority class.

Next, we will consider the F1 score of the XGBoost classifier when performed on all datasets. We note that the SMOTE dataset provides the highest F1 score, equal to a value of 0.443. This is due to the fact that the precision of this model, 0.645 is very high. A high precision means that out of all instances predicted to correspond to a particular brand, on average for all brands, 65 percent actually belong to that brand. Besides, we note that the application of the XGBoost classifier on the cluster centroids undersampled dataset provides the lowest F1 score, equal to a value of 0.388. This is due to the low values for both recall and precision. Both are probably caused by the decrease in information, but mainly by the decrease in diversity in the undersampled dataset, as the cluster centroids algorithm deletes the most average observations from the dataset.

Finally, we will consider the AUC-ROC provided by the XGBoost classifier. For this performance measure, we note that both the values for the random oversampled and undersampled datasets, equal to 0.582 and 0.594 respectively, are optimal. This means that both datasets, in combination with the XGBoost multiclass classifier, can discriminate between brands of cars pretty well. This might be caused by the fact that random data adjustments create a perfectly imbalanced dataset. This means that the number of observations in each sample is equal. Subsequently, this improved balance might contribute to better discrimination between classes and thus result in a higher AUC-ROC.

For the application of the XGBoost classifier on imbalanced data, we can conclude the following. We note that the SMOTE dataset provides the most optimal and consistent results if we consider the overall accuracy and F1 score. This means that performing XGBoost on a SMOTE dataset results in most customers being classified correctly and a model being optimal when there are high costs for false positives and false negatives. However, the optimal accuracy for the minority class and the optimal AUC-ROC value correspond to the random undersampled dataset. This means that performing the XGBoost classifier on the random undersampled dataset is optimal when the aim is to reveal hidden patterns among the observations and classify minority instances in an optimal way.

Finally, we will consider the performance of SVM classifier on the original, oversampled, and undersampled datasets.

5.2.4 SVM classifier on target variable *Brand*

In this subsection, we will analyze the results of performing the SVM classifier on the different training sets. As is in line with the result of the XGBoost classifier, both the original dataset and the SMOTE dataset provide much higher accuracy than the random oversampled, random undersampled, and cluster centroids undersampled datasets. As previously mentioned, we know that SVM can be sensitive to class imbalance, and therefore it may prioritize the majority class in the decision boundaries. This means that the accuracy of the model may be fairly high, as is the case in Table 9. The high accuracy provided by the SMOTE dataset might be explained by analyzing the algorithm of the SMOTE method. As mentioned before, this method generates synthetic samples for the minority class and therefore helps SVM better capture the decision boundaries for both the majority and minority classes. Therefore, the accuracy of the SVM classifier performed on the SMOTE dataset is fairly high. Next, we consider the results of random oversampling, which, as mentioned before, may lead to overfitting, as duplicating instances might result in SVM placing much weight on certain points in the feature space. This can lead to decision boundaries that do not generalize well to unseen data, reducing accuracy. The randomly undersampled dataset provides low accuracy due to the loss of information caused by the deletion of observations. In this case, SVM may struggle to find an optimal decision boundary, leading to reduced accuracy. The lowest accuracy is provided by the cluster centroids in the undersampled dataset. In this case, the cluster centroids method changed the distribution of majority class instances, as we only kept border observations, which subsequently negatively affected the SVM's ability to generalize. Therefore, low accuracy is generated.

Furthermore, we will analyze the accuracy of classifying the instances belonging to the minority class. First, we note that the original, highly imbalanced dataset misclassified all observations belonging to the minority class. Besides, we see that this accuracy can be largely increased by performing the SVM classifier on both the random oversampled and random undersampled datasets, which are both perfectly balanced.

Next, we will consider the F1 scores of the datasets. As one can note from Table 9, these scores are fairly close to each other. The optimal F1 score, equal to 0.382, is provided by the random, undersampled dataset. This is due to the high recall, which contributes to the F1 score. This means that this model, which corresponds to a recall of 0.413, is correctly identifying approximately 41 percent of the relevant instances for all classes. This might be due to the fact that the random undersample method removes instances from the majority of classes randomly and thus may reduce noise in the training data. This can help the SVM classifier to focus on the distinguishing features of each class and improve its ability to identify relevant instances. Furthermore, we note that the lowest F1 score is provided by the cluster centroids in the undersampled dataset. This is due to the low recall of the SVM classifier on this dataset. The value of the recall, which is equal to 0.358, denotes that only 36 percent of the relevant instances are correctly classified for every brand. This might be due to the fact that this data adjustment method changes the underlying distribution of the majority class by only considering the border points of the clusters in the dataset. Therefore, the generated centroids might not capture the diversity of the majority of class instances, which leads to lower performance and, thus, a lower recall of the model.

Finally, we will consider the AUC-ROC values of the application of SVM on the different datasets. First, we note that the AUC-ROC value provided by the original dataset is equal to 0.491, which is below 0.5. This means that this model's performance is worse than random chance. This might be due to the fact that the original dataset is very imbalanced and, thus, the model does not generalize well. Furthermore, we note that the AUC-ROC value for the randomly undersampled dataset is optimal. This might be caused by the fact that this dataset is perfectly balanced, which helps the model discover patterns in the data. This is supported by the fact that the random oversampled dataset, which is also perfectly imbalanced, also provides a fairly high AUC-ROC value.

From the application of the SVM classifier, we can conclude the following. If the aim of the research is to classify as many customers as possible correctly, we can perform the classifier on the original, imbalanced dataset. However, if the aim of the research is to really discover underlying patterns in the dataset, and thus, if not accuracy, accuracy of the minority class, F1 score, and AUC-ROC values are more relevant, one should use random undersampling data adjustments in order to optimize the results.

Overall, in order to predict the brand of a model driven by a customer, we can note the following. If we consider the consistency of all performance measures, we can conclude that the performing classifiers on both the random oversampled and random undersampled datasets gives good results. Especially, when the aim is to correctly classify instances beloninging to the minority classes, the random oversampled and undersampled datasets are optimal.

Next, we will consider the results obtained when predicting the Toyota model that a customer owns. In order to do so, we will analyze the results shown in Table 10, presented below.

		Original	Oversampling		Undersampling	
	Measurement		Random	SMOTE	Random	Cluster Centroids
NB	Accuracy	0.0804	0.0513	0.0761	0.0475	0.1025
	Accuracy Minority	0.000	0.019	0.019	0.019	0.000
	F1 score	0.0765	0.0722	0.0783	0.0764	0.0723
	Recall	0.0990	0.0997	0.0995	0.1061	0.0943
	Precision	0.0623	0.0566	0.0646	0.0597	0.0587
	AUC-ROC	0.5755	0.5589	0.5702	0.5673	0.5755
RF	Accuracy	0.2833	0.2342	0.2808	0.0538	0.1770
	Accuracy Minority	0.038	0.038	0.038	0.094	0.038
	F1 score	0.0665	0.0699	0.0667	0.0737	0.0724
	Recall	0.0521	0.0602	0.0518	0.0993	0.0574
	Precision	0.0921	0.0833	0.0935	0.0586	0.0982
	AUC-ROC	0.5197	0.5403	0.5185	0.5359	0.5099
XGBoost	Accuracy	0.2974	0.2342	0.2999	0.0548	0.1909
	Accuracy Minority	0.019	0.066	0.009	0.123	0.028
	F1 score	0.0853	0.0699	0.0743	0.0700	0.0758
	Recall	0.0594	0.0602	0.0533	0.0887	0.0588
	Precision	0.1513	0.0833	0.1225	0.0578	0.1065
	AUC-ROC	0.5163	0.541	0.5181	0.5351	0.5144
SVM	Accuracy	0.2714	0.0582	0.2712	0.0508	0.1679
	Accuracy Minority	0.000	0.028	0.009	0.038	0.009
	F1 score	0.0605	0.0587	0.0517	0.0550	0.0511
	Recall	0.0557	0.0686	0.0472	0.0619	0.0507
	Precision	0.0662	0.0513	0.0571	0.0494	0.0515
	AUC-ROC	0.5140	0.5307	0.5102	0.5147	0.5023

Table 10: This table shows the results of the application of all multiclass classifiers when trained on the original, imbalanced training set and on the different adjusted training set when *Toyota Model* is considered to be the target variable. For example, one can conclude from this table that when the NB classifier is trained on the original dataset, 8 percent of the observations in the test set are correctly classified when we consider the models of the Toyota cars.

In order to analyse the results from the table above, one should first note the following. The number of distinct Toyota models in the dataset is equal to 27. This means that random assigning the models to the customers would provide an accuracy of 0.037, which is equal to an accuracy of almost 4 percent. Keeping this in mind, we will now consider the results in Table 10. In analyzing this table, we will first focus on the performance of the NB classifier on the original, oversampled, and undersampled datasets.

5.2.5 NB classifier on target variable *Toyota Model*

While analysing the results if the NB classifier on the datasets, we note that, as in line with the results of the NB classifier in Table 9, the accuracy of this model is highest when performed on the cluster centroids undersampled dataset. Specifically, this accuracy is equal to 0.1025, which means that over 10 percent of the customers are combined with the correct Toyota model. This high accuracy might be caused by preserving the assumption of independence between features as the border points of clusters in the observations are added to the dataset. Furthermore, we see that the lowest accuracies are provided by both random data-adjustments. The random oversampled dataset and the random undersampled dataset both classify approximately 5 percent of the customers correctly. However, one should note that these models are still more optimal than classifying randomly, which is correctly done for 4 percent of the observations. The fairly low accuracy provided by the random oversampled dataset might be due to the fact that this method adds duplicates to the original dataset. In this way we increase the dependency between features. Therefore, we can state that, in this case, the independency assumption used in the NB classifier is less likely to hold which may cause a worse performance of the classifier on this dataset. The low accuracy provided by the random undersampled dataset might be caused by the loss of information which occurs when this method deletes random observations from the majority class in the dataset. With this loss of information, the classifier is less likely to discover underlaying patterns among the dataset.

Second, we will consider the accuracy for the minority class only. In this case, this value determines the percentage of customers who actually drive a 'Toyota Hiace' and are classified as such. First, we note that the original, imbalanced dataset corresponds to an accuracy for the minority class equal to zero percent. This means that of all 'Hiace' drivers, none are classified as such. This might be due to the fact that there are only 423 'Hiace' drivers in this dataset, which corresponds to only 0.18 percent of the total number of observations in the original dataset, which can be seen in Table 8. Therefore, we may conclude that this number is too small in order for the NB classifier to identify the characteristics of 'Hiace' drivers. However, as can be seen in Table 10, the random oversampled dataset, the SMOTE dataset, and the random undersampled dataset all increase the accuracy of the minority class. As again can be seen from Table 8, both the random oversampled and random undersampled datasets are perfectly balanced. This means that the NB classifier is able to analyze the characteristics of each class equally well, as the number of observations is equal. Furthermore, we note that the SMOTE dataset provides optimal accuracy for the minority class due to the fact that, again, the independence assumption, which is needed for optimal performance of the NB classifier, is more likely to hold due to the introduction of new data points for the minority class.

Next, we will consider the F1 scores provided by the different datasets. One can note that the values of the F1 score are fairly close to each other. However, the highest F1 score, equal to 0.0783, is provided by the SMOTE dataset. This is caused by the fairly high precision, 0.0646, corresponding to the same dataset. Intuitively, this precision shows that approximately 6.5 percent of the instances predicted as positive were actually true positives, on average for all classes. This high value may again be due to the fact that the SMOTE, besides making the original dataset more balanced, also makes the data more likely to preserve the independent assumption which is used to perform the NB classifier.

Finally, we will consider the AUC-ROC values for this classifier. One can note that these values are fairly high, in comparison to the AUC-ROC values for other classifiers, which means that all models have a good

ability to discriminate between the models of Toyota cars. Specifically, Table 10 shows that the performance of NB on the original and cluster centroids undersampled dataset provide the highest AUC-ROC values. Both values are equal to 0.5755, which shows that that the model has some discriminatory power.

Overall, we can conclude that, in order to find the optimal results while using NB as the classifying method, we should adjust the dataset using the cluster centroids undersample method in order to obtain the most optimal results for both the general accuracy and AUC-ROC value. However, if we want to obtain a model that increases the ability to discover underlaying patterns in the characteristics of observations corresponding to the minority class, combining the NB classifier with the SMOTE oversampled dataset is optimal.

Next, we will consider the performance of the RF classifier on the different datasets.

5.2.6 RF classifier on target variable *Toyota Model*

First, we will analyze the values of the accuracies obtained by the datasets when combined with the RF classifier. In particular, we can note that there is a fairly large difference between all values for accuracy. The minimal accuracy is provided by the random undersampled dataset. This accuracy is equal to 0.054, which means that almost 5.5 percent of the customers are correctly classified. Note that the RF classifier on the random undersampled dataset improves the random classification of customers in the random undersampled dataset. The relatively low value of this accuracy may be caused by the fact that this data adjustment method, random undersampling, randomly deletes information from the dataset. As this method equals the size of every class, and thus the number of observations in every class will be equal to the number of observations in the minority class, this method deletes 95 percent of the observations. This huge loss of information decreases the accuracy of the model significantly. The optimal accuracy, equal to 0.2833, is provided by the application of RF on the original dataset. Note that this means that over 28 percent of the customers are correctly combined with the model of Toyota car they are driving. However, as this dataset is very imbalanced, the accuracy of the model does not tell us much about the capability of the model to discover patterns in the data. Therefore, we will also consider the accuracy of the minority class and the F1 scores provided by the datasets.

First, we consider the accuracy of the minority class provided by all datasets when used for RF classification. We note that most accuracies for the minority class are equal to 0.038, which means that almost 4 percent of the customers driving a 'Toyota Hiace' are classified as such. However, note that the random undersampled dataset corresponds to an accuracy for the minority class equal to 0.094, which means that in this case, over 9 percent of these customers are classified correctly. This twice-as-large accuracy of the minority class can be explained as follows. First note that, as is the case with the random oversampled dataset, the random undersampled dataset is perfectly balanced, and thus every class occurs equally often in the dataset. Therefore, it might be easier for the RF classifier to reveal the patterns in the observations corresponding to the minority class. Furthermore, we note that using random undersampling reduces this bias by deleting many observations from the majority class. Again, this allows the model to learn the patterns present in the minority class more effectively.

Next, we will consider the F1 scores provided by the combination of the RF classifier with the different datasets. As in line with the results from the NB classifier, we note that the F1 scores are again fairly close

to eachother. However, the largest value for the F1 score is equal to 0.0737 and corresponds to the random undersampled dataset. In particular, this F1 score is optimal due to the high recall for this model, which is equal to 0.0993. This means that this model, on average for every class, correctly identifies almost 10 percent of the instances belonging to that class. The relatively high value for the recall of the RF classifier performed on this dataset may be caused by the fact that the random undersampling method creates a perfectly balanced dataset without adding duplicates which can cause overfitting, as is done by random oversampling. Furthermore, we note that the lowest F1 score, which is equal to 0.0665, is provided by the original dataset. The low F1 score is caused by the low value for the recall, equal to 0.0521, which corresponds to this F1 score. The low recall for the original dataset might be caused by the imbalance of the original dataset. For imbalanced datasets, there are fewer instances of the minority class. For these minority classes, RF classifier might not be able to discover the patterns in those classes well. Therefore, the classifier might miss a substantial number of positive instances resulting in a higher number of false negatives. This will result in a lower recall value and therefore in a lower F1 score.

Finally, we will consider the values for AUC-ROC. From the table above, Table 10, we can see that the optimal value is equal to 0.5403 and corresponds to the randomly oversampled dataset. Furthermore, one can see that also the randomly undersampled dataset generates a fairly high AUC-ROC value. As both datasets represent a perfectly balanced dataset, this might be an incentive for these relatively high values. Furthermore, we note that the lowest value for the AUC-ROC is obtained by the cluster centroids undersampled dataset. As this value, equal to 0.5099, is almost equal to 0.5 (which is the AUC-ROC value for random classification), we note that the model's ability to discriminate between the positive and negative classes is close to random chance. This might be due to the fact that, by replacing the observations in the majority class with border points, the distribution of the features may shift, and therefore, the RF classification algorithm does not perform very well.

Overall, we can conclude that the RF classifier can handle imbalanced datasets fairly well, as the accuracy of the classifier on the original dataset is optimal. However, if we aim to find a model that is able to discover underlaying patterns among the observations corresponding to the minority class, we prefer using the random undersampled dataset in combination with the RF classifier.

Third, we will consider the results of the XGBoost classifier in order to predict the model of a Toyota car on the different datasets.

5.2.7 XGBoost classifier on target variable *Toyota Model*

Again, we will first consider the accuracies which are obtained by performing the XGBoost classifier on the original and data-adjusted datasets. For the accuracy, we denote that the optimal value is provided by the SMOTE oversampled dataset, which is able to classify almost 30 percent of the customers correctly. This value is nearly followed by the accuracy provided by original dataset, which classifies 29.7 percent of the customers correctly. The original dataset generates a relatively high accuracy as XGBoost handles imbalanced datasets to some extent by adjusting the weights of different classes during training. However, this high accuracy may also be caused by the predicting all instances as the majority class. Therefore, we will soon also consider other performance measures below. The high accuracy provided by the SMOTE dataset

might be caused by the addition of synthetic sample. Those samples can provide additional information to the model, potentially improving its ability to generalize. The lowest accuracy is provided by the random undersampled dataset, which is in line with the results of both the NB and RF classifier. Again, this is caused by the huge loss of information caused by the random undersampling over many (27) classes.

As an alternative performance measurement, we will also consider the accuracy provided by the minority class only. By analyzing this value, we note that, again, the randomly undersampled dataset performs optimally when analyzing only the minority class. From Table 10, we can see that over 12 percent of the actual 'Hiace' drivers is classified as such. Again, this might be to the fact that this dataset is perfectly balanced and that it contains less bias than the original dataset. Furthermore, this dataset doubles the accuracy of the minority class in the randomly oversampled dataset, which is also perfectly balanced, due to the following fact. As mentioned before, in order to obtain the oversampled dataset, we add duplicate observations for the minority class in the original dataset. By doing so, as the number of observations increases but we add no extra information, the risk of overfitting increases, and thus the accuracy for the minority class might be decreasing.

Next, we will consider the F1 score provided by the datasets. Here, we note that the original dataset corresponds to the most optimal F1 score, which is equal to 0.0853. This is mainly due to the high precision of this dataset, which is equal to 0.1513. Intuitively, this value shows that approximately 15 percent, on average for every class, of the instances predicted to belong to the specific class, actually belong to this class. As mentioned before, this high value might be caused by the quality of the XGBoost classifier to handle imbalanced datasets to some extent. During training, it adjusts the weights of different classes, placing more emphasis on the minority class which can lead to higher precision for the minority class.

Finally, we will consider the AUC-ROC values for the XGBoost classifier. As can be seen in Table 10, the highest values for this measurement are provided by both the random data adjustments. This means that the perfect balance of classes in the dataset, which is the case for both data adjustments, is optimal when performing XGBoost classification. As previously mentioned, the XGBoost classifier adjusts the weights of different classes during training. When the dataset is perfectly balanced, each class is represented more equally, and the model is forced to learn features that distinguish between all classes, including the minority class. This can lead to a more robust model that generalizes well to unseen data, especially for minority class instances.t

Overall, we can conclude that XGBoost works fairly well for imbalanced datasets, if we consider the general accuracy. However, the SMOTE oversampled dataset does provide an even more optimal accuracy for all observations. Nevertheless, if we aim to classify the customers in the minority classes well, we prefer to use the randomly undersampled dataset in combination with the XGBoost classifier.

Finally, we will analyse the results in Table 10 which correspond to application of the SVM classifier on the different datasets.

5.2.8 SVM classifier on target variable *Toyota Model*

First, we will focus on the accuracies obtained by the different datasets in combination with the SVM classifier. We note that the accuracies of the original dataset and of the SMOTE dataset, which both correspond to correctly classifying 27 percent of the observations, are optimal. Furthermore, we note that both the random oversampled and random undersampled datasets correspond to the lowest accuracies. This might be due to the fact that the random oversampled dataset introduces duplicates, which can also introduce extra noise into the dataset. Furthermore, the low accuracy of the random undersampled dataset might be caused by the loss of information that the dataset suffers when random undersampling is applied.

Second, we will consider the accuracy provided by only considering the minority class, which considers customers driving a 'Toyota Hiace' car. As is in line with the results of the NB classifier, the original, imbalanced dataset classifies none of the observations in the minority class correctly. Furthermore, we note that both the random oversampled and random undersampled datasets increase the accuracy of the minority class to approximately 3 and 4 percent, respectively. Again, this might be due to their perfectly balanced nature.

Next, we will analyze the F1 scores obtained by the different datasets. We note that all F1 scores are fairly close to each other. However, the most optimal F1 score, equal to 0.061, is provided from the original dataset. This optimal performance of SVM on the imbalanced dataset can be explained as follows. As mentioned before, the SVM classifier aims to find a hyperplane (decision boundary) that maximizes the margin between two different classes. Therefore, the algorithm is not much bothered by the number of instances available for every class.

However, if we consider the AUC-ROC values for the SVM classifier, we note that, again, both the random oversampled and undersampled datasets provide the highest value. This would indicate that, in line with the AUC-ROC values for RF and XGBoost, it is beneficial for this performance measure to be calculated for a perfectly balanced dataset. Overall, we can state that the SVM is not much influenced by imbalanced datasets, and thus the results are optimal when the SVM is directly applied to the original dataset.

Overall, if we aim to predict the model of a Toyota car owned by a particular customer by using the SVM classifier, we can note the following. If the aim of the research is to simply achieve the most correctly classifier observations, we suggest the performance of SVM on the original imbalanced dataset or on the SMOTE oversampled dataset. However, as the aim of this research is to identify the patterns in the data that correspond to the minority class, we prefer using the random oversampled and random undersampled datasets in order to obtain the best results.

In the next section, I will draw a brief conclusion regarding the results in this section.

6 Conclusion

In this section, I will summarize the key findings discussed earlier and present conclusions based on these results. At the end of this section, we will answer the research question *How can we optimize the performance of multiclass classification methods for minority classes in imbalanced datasets?*

In the preceding section, our focus was on evaluating the impact of data adjustment techniques on the original training set and the subsequent performance of the minority class of these datasets when subjected to classifier methods. Analyzing the original dataset and its corresponding training set reveals the following insights. Multiclass classification methods applied to the original training set yield pretty good results, particularly when aiming for optimal accuracy in predicting a customer's car, regardless of a model's ability to recognize underlying patterns in the dataset. Especially when combined with both the RF and XGBoost classifiers, the original dataset exhibits optimal accuracy. However, when considering accuracy for the minority class, F1 score, and AUC-ROC as performance measures, the original training set does not fare optimally across various classification methods and thus might not be very robust to imbalanced datasets. Consequently, using an imbalanced dataset for multiclass classification in scenarios where the costs of false negatives and false positives are high is not recommended. Furthermore, using this imbalanced dataset in order to classify customers who own cars that are rare and thus do not occur frequently is not optimal.

Therefore, in this research, we considered the data adjustment. The first takeaway from the application of the data adjustment techniques is that random oversampling will greatly increase the number of observations. Especially when the total number of classes in the dataset is large, for example, when analyzing the model of a customer's car, the adjusted training set becomes almost eight times as large as the original training set. This means that these adjustments may be very computationally expensive. However, as mentioned before, the random oversampling adjustment increases the performance of classification methods for the minority class. Especially in combination with the RF classifier, the XGBoost classifier and the SVM classifier, random oversampling provides good results. This is probably due to the perfectly balanced nature of the random oversampled dataset.

Second, we will consider the results corresponding to the SMOTE dataset. From the results in the previous section, we can conclude that SMOTE provides the best results when applied together with the XGBoost classifier. Especially when this combination is used to predict the model of a customer's Toyota car, when there are many classes in the dataset, we increase the performance of the original, imbalanced dataset. Furthermore, we denoted that the SMOTE increases the number of observations, but not drastically. This means that this training set performs better than the random oversampled training set with every classifier on each dataset and is less computationally expensive. This means that SMOTE can be the solution to many real-world multiclass classification problems. However, we also note that the SMOTE dataset increases the performance of multiclass classification models on imbalanced datasets but is not optimal in comparison to other data adjustment techniques.

Turning to the results of the random undersampling method, we have observed that this technique significantly reduces the number of observations, making the dataset less computationally expensive. Besides, this method provides optimal accuracy if we only consider the minority class of the original dataset. This means that this dataset is able to classify cars that are not frequently bought but might be yielding a lot

for the business. Therefore, this data-adjustment technique is suggested when we aim to optimize the performance of multiclass classification methods for minority classes in imbalanced datasets. Furthermore, the random undersampled method consistently yields optimal AUC-ROC values across different target variables and multiclass classification methods. Both aspects can be attributed to the removal of outliers and noise.

Next, we have noted that the examination of classification methods on the cluster centroids undersampled training set reveals somewhat promising outcomes, especially when combined with the NB classifier. Specifically in these cases, the general accuracy of the cluster centroids undersampled dataset is optimal. This might be due to the fact that the replacement of original observations with border points of the clusters enhances the likelihood of meeting the independence between features assumption, a crucial condition assumed by the NB classifier. This holds true for datasets related to both target variables, *Brand* and *Toyota Model*. In combination with other classifiers, we note that the cluster centroids undersampled training set oftentimes provides low accuracy, which is unbeneficial when the aim of the research is to classify as accurately as possible. However, note that in these cases, the values, especially the F1 scores, are fairly high. This means that the underlaying patterns between the observations are more likely to be explored by the use of the cluster centroids undersampling technique.

We will finalize this section with an answer to the research question mentioned before, namely, textitHow can we optimize the performance of multiclass classification methods for minority classes in imbalanced datasets?. As elaborated earlier, the aim of this research is to optimize the ability of methods to reveal underlaying patterns among observations, especially those corresponding to the minority class in the dataset. Therefore, this research suggests that the use of random undersampling, but also random oversampling techniques, performs optimally when classifying minority instances. We can even conclude that this is the case for all classification methods. Therefore, especially the random undersampling technique is recommended when the aim is to correctly predict rarely bought cars owned by customers based on their characteristics.

In the next section, we will provide a brief discussion regarding this research. In this discussion, I will examine the quality and suitability of the different steps taken in this research.

7 Discussion

In this section, we'll start by comparing the results from the previous section with those of other studies. We'll highlight similarities and differences between our findings and those in existing research, emphasizing the contributions of our study to the field. Additionally, we'll acknowledge the limitations of our research to understand its scope, and we'll identify areas for future investigation.

As mentioned in the literature section, previous research on imbalanced datasets has produced varying results due to differences in datasets, data adjustment techniques, and multiclass classification tasks. Our study focuses on customer car data from the Louwman Group, revealing that certain algorithms perform better than others. For instance, while past research suggested that the Naive Bayes (NB) algorithm is suboptimal for imbalanced datasets, our study shows that, when combined with cluster centroid undersampling, NB can perform well. Similar to other research, we find that RF and XGBoost classifiers are generally robust to imbalanced datasets when we consider the general accuracy of the models. However, this research adds that this accuracy can be further optimized with the SMOTE.

Furthermore, this research indicates that, in an attempt to classify minority classes, the randomly undersampled dataset outperforms all other datasets, independent of the mutliclass classification algorithm used. This is in line with results in other papers, which state that random undersampled techniques can both increase accuracy for minority classes. However, in this research, the random undersampling technique outperforms the cluster centroid undersampling technique, which is a contradiction to the conclusions in some other papers published earlier.

Concluding, the findings in this research serve as a practical guideline for real-world, imbalanced datasets, bridging the gap between suggested data adjustment methods, multiclass classification models, and their actual application.

A notable limitation of our study is the proximity of the outcomes. While our analysis provides valuable insights, the closeness of the results requires careful interpretation. Additionally, in the dataset with the target variable *Toyota Model*, we grouped classes with minimal observations into an "Others" category, leading to a loss of detailed information. Combining these classes obscures meaningful distinctions, making it challenging to identify specific contributions from these small classes to observed patterns.

For future research, we recommend using even larger datasets to retain information in small classes without the need to combine them into one class. Furthermore, one can try to apply a weighted loss function to address imbalanced datasets by giving more weight to important minority classes during model training. This ensures a heavier penalty for misclassifications in these minority classes. Moreover, exploring additional multiclass classification models like Neural Networks, known for handling complex problems, could provide further insights.

In conclusion, our study contributes valuable insights and practical guidance for many other studies on this topic. However, acknowledging its limitations opens avenues for future research to delve deeper into specific aspects and refine the methodologies.

8 References

M. Aly. Survey on multiclass classification methods. *Neural Netw*, 19(1-9):2, 2005.

C. K. Aridas, S. Karlos, V. G. Kanas, N. Fazakis, and S. B. Kotsiantis. Uncertainty based under-sampling for learning naive bayes classifiers under imbalanced data sets. *IEEE Access*, 8:2122–2133, 2019.

F. Baldo, J. Grando, Y. Y. Correa, and D. A. Policarpo. Adaptive fast xgboost for multiclass classification. *Journal of Information and Data Management*, 14(1), 2023.

R. Batuwita and V. Palade. Class imbalance learning methods for support vector machines. *Imbalanced learning: Foundations, algorithms, and applications*, pages 83–99, 2013.

G. C. Cawley and N. L. Talbot. Preventing over-fitting during model selection via bayesian regularisation of the hyper-parameters. *Journal of Machine Learning Research*, 8(4), 2007.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

R. Debnath, N. Takahide, and H. Takahashi. A decision based one-against-one method for multi-class support vector machine. *Pattern Analysis and Applications*, 7:164–175, 2004.

S. S. Dhaliwal, A.-A. Nahid, and R. Abbas. Effective intrusion detection system using xgboost. *Information*, 9(7):149, 2018.

D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan. Hybrid decision tree and naïve bayes classifiers for multi-class classification tasks. *Expert systems with applications*, 41(4):1937–1946, 2014.

J. Gall, N. Razavi, and L. Van Gool. An introduction to random forests for multi-class object detection. In *Outdoor and Large-Scale Real-World Scene Analysis: 15th International Workshop on Theoretical Foundations of Computer Vision, Dagstuhl Castle, Germany, June 26-July 1, 2011. Revised Selected Papers*, pages 243–263. Springer, 2012.

A. S. Ghanem, S. Venkatesh, and G. West. Multi-class pattern classification in imbalanced data. In *2010 20th International Conference on Pattern Recognition*, pages 2881–2884. IEEE, 2010.

C. Gold and P. Sollich. Model selection for support vector machine classification. *Neurocomputing*, 55(1-2): 221–249, 2003.

L. Guerra, L. M. McGarry, V. Robles, C. Bielza, P. Larranaga, and R. Yuste. Comparison between supervised and unsupervised classifications of neuronal cell types: a case study. *Developmental neurobiology*, 71(1): 71–82, 2011.

Z. Hameed, B. Garcia-Zapirain, J. J. Aguirre, and M. A. Isaza-Ruget. Multiclass classification of breast cancer histopathology images using multilevel features of deep convolutional neural network. *Scientific Reports*, 12(1):15600, 2022.

D. M. Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1): 1–12, 2004.

L. A. Jeni, J. F. Cohn, and F. De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*, pages 245–251. IEEE, 2013.

M. Khalilia, S. Chakraborty, and M. Popescu. Predicting disease risks from highly imbalanced data using random forest. *BMC medical informatics and decision making*, 11:1–13, 2011.

B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

V. Y. Kulkarni and P. K. Sinha. Random forest classifiers: a survey and future research directions. *Int. J. Adv. Comput*, 36(1):1144–1153, 2013.

T.-T.-H. Le, Y. E. Oktian, and H. Kim. Xgboost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems. *Sustainability*, 14(14):8707, 2022.

G. Menardi and N. Torelli. Training and assessing classification rules with imbalanced data. *Data mining and knowledge discovery*, 28:92–122, 2014.

J. Milgram, M. Cheriet, and R. Sabourin. "one against one" or "one against all": Which one is better for handwriting recognition with svms? In *tenth international workshop on Frontiers in handwriting recognition*. Suvisoft, 2006.

R. Mohammed, J. Rawashdeh, and M. Abdullah. Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (ICICS)*, pages 243–248. IEEE, 2020.

A. Priyam, G. R. Abhijeeta, A. Rathee, and S. Srivastava. Comparative analysis of decision tree classification algorithms. *International Journal of current engineering and technology*, 3(2):334–337, 2013.

K. Purwandari, J. W. Sigalingging, T. W. Cenggoro, and B. Pardamean. Multi-class weather forecasting from twitter using machine learning aprroaches. *Procedia Computer Science*, 179:47–54, 2021.

J. D. Rennie. Improving multi-class text classification with naive bayes. 2001.

I. Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

T. Romary, F. Ors, J. Rivoirard, and J. Deraisme. Unsupervised classification of multivariate geostatistical data: Two algorithms. *Computers & geosciences*, 85:96–103, 2015.

M. Sahare and H. Gupta. A review of multi-class classification for imbalanced data. *International Journal of Advanced Computer Research*, 2(3):160, 2012.

M. S. Shelke, P. R. Deshmukh, and V. K. Shandilya. A review on imbalanced data handling using undersampling and oversampling technique. *Int. J. Recent Trends Eng. Res*, 3(4):444–449, 2017.

M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *Australasian joint conference on artificial intelligence*, pages 1015–1021. Springer, 2006.

J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip. A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications*, 134:93–101, 2019.

Y. Sun, A. K. Wong, and M. S. Kamel. Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence*, 23(04):687–719, 2009.

J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour. Boosting methods for multi-class imbalanced data classification: an experimental review. *Journal of Big Data*, 7:1–47, 2020.

P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032*, 2002.

J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, 2007.

J. Zhang, L. Chen, F. Abid, et al. Prediction of breast cancer from imbalance respect using cluster-based undersampling method. *Journal of healthcare engineering*, 2019, 2019.

9 Appendix

In this section, one can find the variables, their explanation and the corresponding number of unique values, used in this research.

Variable Name	**Explanation Variable**	**Number of unique values**
HH_4G1	Intellectual Culture Lovers (degree)	10
HH_4G10	Average Income Sports Enthusiasts (degree)	10
HH_4G11	Traditional Villagers (degree)	10
HH_4G12	Family-oriented Deal Seekers (degree)	10
HH_4G13	Quirky Computer Enthusiasts (degree)	10
HH_4G14	Budget-driven Tenants (degree)	10
HH_4G2	Ambitious Trend Followers (degree)	10
HH_4G3	Online Networkers (degree)	10
HH_4G4	Active Internet Families (degree)	10
HH_4G5	Homely Families (degree)	10
HH_4G6	Conservative Philanthropists (degree)	10
HH_4G7	Creative Environmentalists (degree)	10
HH_4G8	Committed Seniors (degree)	10
HH_4G9	Active Over-fifties (degree)	10
HH_4GEOTYP	GeoType	15
HH_A_ZKLK2	Business car	2
HH_AANT_DL	Number of charities	6
HH_AFS_BNK	Distance to bank office	8
HH_AFS_BS	Distance to elementary school	8
HH_AFS_NSS	Distance to railway station	8
HH_AFS_OVH	Distance to public transport	8
HH_AFS_SUP	Distance to supermarket	8
HH_AFS_WCG	Distance to large shopping centre	8
HH_AFS_WCK	Distance to small shopping centre	8
HH_APERSHH	Number of people	7
HH_B_BELEG	Investment	2
HH_B_LENEN	Loan	2
HH_B_SPAAR	Saving	2
HH_BET_BER	Willingness to pay	10
HH_BIO	Buying organic products	4
HH_BS_DIV	Diversity of schools in the area	6
HH_BS_MAX	Most common type of elementary school in the area	7
HH_BS_SRT	Closest type of elementary school	7
HH_BWJ_WON	Home construction year (class)	13

Table 11: This table shows the variables present in the final dataset by their name, explanation and number of unique values

Variable Name	Explanation Variable	Number of unique values
HH_CREDCRD	Credit card ownership	2
HH_DAG_ONL	Online shopping daily groceries	4
HH_DAGBLAD	Reading newspaper	2
HH_DB_TYPE	Preferred type of newspaper	4
HH_DH_EETD	Density of eating and drinking facilities (1 km)	8
HH_DH_UITG	Density of entertainment venues (5 km)	8
HH_DOELMED	Purpose of using media (radio and TV)	4
HH_DONAT	Donor	2
HH_E_BSPR	Energy-saving measures	4
HH_E_LABEL	Energy efficiency rating	8
HH_E_TYPE	EnergyType	9
HH_FINTYP1	Financial Professionals (degree)	8
HH_FINTYP2	Advisory Sensitive Families (degree)	8
HH_FINTYP3	Interested Conservatives (degree)	8
HH_FINTYP4	Active Borrowers (degree)	8
HH_FINTYP5	Young Potentials (degree)	8
HH_FINTYP6	Passive Laymen (degree)	8
HH_FINTYPE	FinType	7
HH_FUNCTIE	Jobtype main breadwinner	5
HH_GAS_VBR	Gas consumption	15
HH_GEBR_OV	Use of public transport	4
HH_HERBW	Reconstruction cost value	13
HH_INBDLW	Household effects value	13
HH_INKOM3	Income	8
HH_INTTIJD	Time spent on the internet	4
HH_INTTYPE	Purpose of using internet	4
HH_KLSEGM	Preferred clothing segment	6
HH_KOOPKR	Purchasing power	7
HH_KS_BUDG	Clothing segment budget	6
HH_KS_EXCL	Clothing segment exclusive	6
HH_KS_MIDK	Clothing segment middle-class	6
HH_KS_MODI	Clothing segment fashionable	6
HH_KS_TWEE	Clothing segment second-hand	6
HH_KWH_VBR	Electricity consumption	15
HH_LFT_KND	Age of the eldest child	5
HH_LFTIJD	Age	14
HH_LOS_ART	Reading individual (news) articles	2
HH_LVNSFS2	Household composition	9
HH_MBEWUST	Environmental awareness	10

Table 12: This table shows the variables present in the final dataset by their name, explanation and number of unique values

Variable Name	Explanation Variable	Number of unique values
HH_MENTMIL	Mentality	9
HH_MM_GG	Convenience oriented (degree)	10
HH_MM_KP	Cosmopolitans (degree)	10
HH_MM_MB	Modern mainstream (degree)	10
HH_MM_NC	New conservatives (degree)	10
HH_MM_OM	Social climbers (degree)	10
HH_MM_PH	Postmodern hedonists (degree)	10
HH_MM_PM	Post-materialists (degree)	10
HH_MM_TB	Traditional villagers (degree)	10
HH_ONLINE	Online shopping (excl. groceries)	6
HH_OPLEID2	Education	8
HH_RISAVRS	Risk aversion (insurances)	4
HH_SOC_MED	Use social media	4
HH_SOCKLAS	Social class	6
HH_SPECZKN	Visit specialty store	2
HH_SPORT	Sport	3
HH_SRT_SUP	Preference for price-oriented supermarket	2
HH_SRTDOEL	Preferred type of charity	4
HH_STADSVW	District heating	3
HH_STV	Association or foundation present	2
HH_SWITCH	Switch sensitivity	4
HH_TECHNOL	Rapid acquisition of new technology	2
HH_TIJDSCH	Reading magazines	3
HH_TV_COMM	Commercial television channels	6
HH_TV_PUBL	Public television channels	6
HH_TV_REG	Regional television channels	6
HH_TV_STRM	Streaming services	6
HH_TV_YTUB	YouTube	6
HH_TVVOORK	Preferred type of TV channel	4
HH_TWEEVD	Dual income	3
HH_VAK_ACC	Holiday accommodation	3
HH_VAKBEST	Holiday destination	4
HH_VB_CULT	Culture (e.g. classical concert, museum, playing an instrument)	6
HH_VB_GRP	Group activity (e.g. amusement park, board game, cinema)	6
HH_VB_KLUS	Outdoors and DIY (e.g. nature, gardening, do-it-yourself)	6
HH_VB_PERS	Personal development (e.g. museum, reading, course)	6
HH_VB_STAP	Going out (e.g. disco, cafe, popconcert)	6

Table 13: This table shows the variables present in the final dataset by their name, explanation and number of unique values

Variable Name	Explanation Variable	Number of unique values
HA_CREDCRD	Credit card ownership (class)	10
HA_VZ_INBD	Household effects insurance (class)	10
HA_VZ_RCHT	Legal assistance insurance (class)	10
HA_VZ_REIS	Travel insurance (class)	10
HA_VZ_UITV	Funeral insurance (class)	10
HH_VHPC1JR	Relocations in the Postal Code last year	5
HH_VHPC5JR	Relocations in the Postal Code last 5 years	7
HH_VR_RUIM	Free space on the parcel	8
HH_VRIJWIL	Volunteer work	2
HH_VZ_INBD	Household effects insurance	2
HH_VZ_RCHT	Legal assistance insurance	2
HH_VZ_REIS	Travel insurance	2
HH_VZ_UITV	Funeral insurance	2
HH_WON_EIG	Home ownership	3
HH_WON_INH	Home content	6
HH_WON_LST	Living costs	6
HH_WON_OPP	Home surface area	8
HH_WONDOEL	Home target group	5
HH_WONHPR2	House rental price	15
HH_WONTYP2	Type of housing	12
HH_WONWOZW	WOZ value of property	17
HH_WOZONTW	WOZ value development of property	8
HH_WRKUREN	Working hours	3
HH_ZZP	Self-employed present	2
URB	Urbanisation	8
HH_VB_THS	Home hobby (e.g. puzzling, reading, needlework)	6
HH_VB_UIT	Visiting theaters and restaurants	6
HH_VEGETA	Eating vegetarian meals	4
HH_VERD_SB	Distribution between savings and investments	4
HA_B_BELEG	Investment (class)	10
HA_B_LENEN	Loan (class)	10
HA_B_SPAAR	Saving (class)	10
HH_CHARI_J	Annual spending on charities	7
HH_CONSFRQ	Consumption frequency	4

Table 14: This table shows the variables present in the final dataset by their name, explanation and number of unique values