LUNDS UNIVERSITET

Individual revenue forecasting in the banking sector

Authors: Ricardo Brandão, Simona Šulžickytė

Supervisor: Jonas Wallin

Lund University School of Economics and Management

Master's programme in Data Analytics and Business Economics

2023

Contents

List of Figures

List of Tables

Abstract

This paper analyses data from a Swedish bank combined with macroeconomic indicators to forecast revenues for individual customers over the course of four years. Separate models are created for recurring customers and customers who have just joined the bank. XGBoost is shown to outperform linear regression, random forest, neural network and support vector regression when comparing both mean absolute error and mean squared error. Macroeconomic variables reveal little to no significance for such forecasts. Finally, a cluster-based method is proposed where customers are first assigned a cluster and then different models are trained for each cluster. We conclude that such a method is only effective if the classification of customers into their respective cluster is sufficiently accurate.

Keywords: Revenue forecasting; Banking; Machine Learning; XGBoost; Customer segmentation

1 Introduction

Technological developments and advancements in machine learning have been changing practices and decision-making across virtually every industry. In that regard, the banking sector has been at the forefront of such transformations. Data has been used to improve relationships with customers and marketing tactics (e.g., Sun, Morris, Xu, Zhu & Xie, 2014; Hung, He & Shen, 2020), to inform risk management decisions (e.g., Rahman & Iverson, 2015; Martins, Mamede & Correia, 2022), to reinforce security (e.g., Srivastava & Gopalkrishnan, 2015; Rao & Lakshmanan, 2022) and to detect cases of fraud (e.g., Moreira, Junior, Silva, Junior, Costa, Gomes & Santos, 2022; Aftabi, Ahmadi & Farzi, 2023).

Among all of these different applications of data to the banking business, the task of revenue prediction emerges as both important and challenging. In particular, the prediction of revenues at the level of individual customers has the potential to help banks identify the real value of each, allowing the institutions to focus their efforts on those with the highest potential to generate revenue. The positive effects that such information can have for a business are obvious but so is the complexity of such an assignment.

The field of research for prediction of revenues in the financial industry at a disaggregate level is still underdeveloped and this paper aims at contributing to this discussion. To accomplish this, we will be focusing on three main research questions:

1. **What is the best method to forecast individual revenues?** We compare five models to predict revenues: linear regression, random forest, XGBoost, neural network and support vector regression.

2. **Can macroeconomic variables be used to improve revenue forecasts?** We will study the impact of seven macroeconomic indicators and analyse the significance of their impact on predictions.

3. **Can the application of different models for different clusters of customers improve forecasts?** We will study the efficacy of the method of dividing customers into revenue segments and afterwards training different models for each.

The paper starts by reviewing some of the most relevant works pertaining to this problem in Section 2. In Section 3 we describe the data set used for our analysis. Section 4 exhibits the theory behind the methods used over the course of the paper. Section 5 details the methodology used. Finally, Section 6 describes the results obtained in our analysis.

2 Literature review

2.1 Revenue forecasting in the financial industry

Forecasting revenue or profitability at a disaggregated level is a task that has been previously studied by several authors, both within the field of financial services and other industries. These works differ in terms of the statistical and machine learning tools used for prediction.

Larivière and Van den Poel (2005) is an early example of this, having studied both customer retention and customer profitability prediction using data from a Belgian financial services provider. Regarding profitability, the authors forecast not only the profit evolution but also a binary "profit drop" variable indicating whether profit went up or down for a particular customer in the period of analysis. In the study, regression forests are used to forecast profit evolution and random forests are used to forecast the categorical variable of profit drop. Past customer behaviour (e.g., type of products owned, total number of products owned), customer demographics (e.g., age, gender, geographic information) and variables related to intermediaries (e.g., selling tendency of the salesperson contacting the customer) were used as inputs for the models, with the authors finding that behaviour variables were the most relevant ones for profitability. Regression forests and random forests outperformed linear regression and logistic regression, respectively, when predicting profit evolution and profit drop.

A later study conducted by Fang, Jiang and Song (2016) found similar results, concluding that random forests outperform four other methods: linear regression, decision trees, support vector machines (SVM) and generalised boosted models. The authors faced a similar challenge of predicting customer profitability, this time in the insurance industry, working with data provided by a Taiwanese company. Faced with this problem, Fang, Jiang and Song (2016) found a random forest with 200 trees and 8 candidate variables at each split to be the most adequate method for predicting profits. After further analysis, the region of residence, age, sex, insurance status (valid or invalid contract) and customer source (traditional channels or others, such as online) were identified as the most important variables in determining profitability.

Several other authors have worked on revenue or profit prediction including an additional step in the process: clustering customers into different segments and applying different models for each. This approach is based on the assumption that different factors have uneven effects across customer segments and, this way, one can better capture the intricacies of the data. We highlight in this subsection two works: Ekinci and Duman (2015) and Rogić, Kašćelan, Kašćelan and Đurišić (2021).

Ekinci and Duman (2015) worked on data from a Turkish bank and were interested in predicting the profit of each customer for the year of 2009 using information of the previous year, including but not limited to types and number of products, number of new products, total assets and value of the services (e.g., credit cards, loans) used by the customer. Before fitting a linear regression to predict the profit value for 2009, each customer was classified into different profitability segments. To do so, the authors made use of three techniques: classification and regression trees, logistic regression and chi-squared automatic interaction detection. Furthermore, the authors performed a cost sensitive classification analysis as the misclassification of certain classes were deemed to be more damaging than others. In the end, Ekinci and Duman (2015) found that the methods used outperformed previously used methods.

Rogić, Kašćelan, Kašćelan and Đurišić (2021) worked with data from a Montenegrin insurance company, using information on customer attributes, policies subscribed, the brand (premium, middle or budget) and purchasing behaviour to predict profits on an individual level. The authors start by identifying the issue of asymmetry in customer profitability prediction tasks. In general, only a small minority of customers are very profitable and this is the case for virtually every

business. In spite of its relatively small size, this customer segment can actually include valuable information and removing it from the analysis can lead to a significant loss for decision makers. To overcome this issue, the authors propose a three-step procedure. First, cluster customers based on purchasing behaviour, namely the Recency-Frequency-Monetary (RFM) approach, using the k-means algorithm. Second, classify customers into each cluster using a SVM model. Finally, use a support vector regression (SVR) model to predict the profit value of the customer. To validate the accuracy of this method, the authors compared its results with the ones obtained using a more usual methodology: generalised linear regression, gradient boosted trees and random forests. The proposed method outperformed the other three used as a benchmark, with the biggest gain in accuracy concentrating in the most profitable customers. For this segment, the method achieved an average error of ±6% while other models get a value greater than ±48%.

2.2 Customer segmentation

Given the importance of customer segmentation for some of the methods used in the past when predicting revenue or profitability within the financial services sector, we dedicate this subsection to such methods.

Khalili-Damghani, Abdi and Abolmakarem (2018) explore the problem of segmenting customers based on the value generated, an important topic for companies when trying to identify the best customers to attract. The authors work with two cases: one in the insurance industry and another in the telecommunication sector. The customers were clustered into 3 groups: high value, potential value and low value. These groups were created using the k-means clustering algorithm and, afterwards, customers were assigned to each cluster using classification trees.

Sivasankar and Vijaya (2017) tackled a somewhat related problem: churn prediction in the telecommunication industry. The paper is a comparative study of different unsupervised clustering techniques: fuzzy c-means, possibilistic fuzzy c-means and k-means. Out of the three algorithms, k-means was found to be the best at improving classification accuracy when used in combination with decision trees.

2.3 Impact of macroeconomic variables in the financial performance of banks

The final subsection of this paper's literature review is dedicated to the incorporation of macroeconomic variables in the analysis of banks' financial performance. Even though past literature has not typically included these variables when forecasting individual revenues or profits, there is extensive literature studying the relation between macroeconomic indicators and the overall profitability of banks. These articles cover a wide range of geographies and periods of time. Importantly, the variables studied also vary considerably, although one can identify Gross Domestic Product (GDP), either in absolute value or as its growth rate, and inflation as the most studied variables by a large margin.

Looking only at works from this century, we highlight 9 papers, presented next in chronological order of publication.

Bikker and Hu (2002) studied the impact of the business cycle on profits, provisioning and lending of banks for 26 OECD (Organisation for Economic Co-operation and Development) countries from 1979 to 1999. The authors conclude that, as expected, profitability is impacted by GDP growth, as "[p]rofits, at a GDP growth level of over 2%, turn out to be almost 2½ times those at GDP growth levels bellow 2%" (Bikker & Hu, 2002).

Clair (2004) studied the impact of 29 different variables on the financial performance and resilience of banks in Singapore using data from 1990 through 2003. The different variables can

be grouped into 7 categories: data on bankruptcies, exchange rates, GDP, inflation, interest rates, stock market and unemployment rate. Out of these, exchange rates, GDP, interest rates and unemployment rate were found to be the most significant. In fact, Clair (2004) concluded that "[o]n average, roughly two-thirds of the changes in the local banks' aggregate financial performance can be explained by changes in the macro environment".

Athanasoglou, Delis and Staikouras (2006) researched the impact of real GDP per capita and inflation on the profitability of banks across 7 South Eastern European countries during a 5 year period, from 1998 to 2002. According to the study, inflation has a positive impact on banks' profits while per capita income had no significant effect, although the authors point out that the latter could be a result of the small sample period used.

Flamini, McDonald and Schumacher (2009) also study the impact of these two variables in one of the largest studies in terms of geographical scope. The paper analyses the profitability of 389 banks in 41 Sub-Saharan African countries. The authors used a sample from 1998 to 2006 and found that "macroeconomic policies that promote low inflation and stable output growth [do] boost credit expansion" (Flamini, McDonald & Schumacher, 2009) and impact bank profits.

Investigating the determinants of banking profits in Jamaica between 2000 and 2010, Moulton (2011) found a positive relation between profits and improvements in GDP, inflation and the stock market.

More recently, Almaqtari, Al-Homaidi, Tabash and Farhan (2019) worked on panel data from 69 Indian banks in a period spanning from 2008 to 2017. Using return on assets (ROA) and return on equity (ROE) as proxies for profitability, the authors studied the effects of GDP, inflation, interest rates and exchange rates on these variables. Furthermore, the authors researched the effects of phenomena like the financial crisis and the 2016 Indian banknote demonetisation. The results showed that inflation rate, exchange rate, interest rate and demonetisation had a significant impact on ROA, while all variables except demonetisation affected ROE.

Later in the same year, Batten and Vo (2019) conducted a study focusing on Vietnamese banks and determined a positive relation between inflation and profitability. Regarding GDP growth, the authors failed to discover any relation with banks' profitability, noting that "[t]his finding is not in line with a number of previous articles in the established literature" (Batten & Vo, 2019).

Finally, Jigeer and Koroleva (2023) performed a panel data study with 16 Chinese banks for the period of 2008 until 2020, studying the impact of province GDP and inflation alongside other explanatory variables related to the banks. The authors concluded that both of the macroeconomic variables were significant in explaining the profitability of the banks analysed.

3 Data

This paper works with data provided by a Swedish bank and complements it with macroeconomic data for Sweden. This section introduces the reader to the data set used in our analysis and is divided into two subsections: the first regarding customers' data provided by the bank and the second with respect to the macroeconomic variables.

Due to confidentiality concerns, we will not present detailed statistics of the variables and some of the products, namely the credit cards, were described in vague terms.

3.1 Customers' data

The bank provided fully disaggregated data at the account level. This included tables with account information, customer data, balance changes, types of bank products, consumer loan sales and applications, payment solution applications, insurance information, history of transactions and, more importantly, monthly revenues. Extensive preprocessing was required in order to arrive at the final data set.

Before explaining this preprocessing, it is important to define the scope of our analysis. Firstly, all of the customers included in the data set provided are based in Sweden. Secondly, we had access to revenues from 2019 to 2022, so this will be the time frame of our forecasts. Finally, we will limit our analysis to bank accounts that were active by the end of March 2023. The reason for this last limitation comes down to the fact that accounts that were inactive are unlikely to generate any revenue in the period of the analysis. Looking at the date of closing of such accounts is also ambiguous as, due to the nature of some of banking products provided, accounts may remain as active without use for a long period of time before being closed by the bank.

The preprocessing stage involved the creation of all of the variables used in modeling. Given that the data was stored, for the most part, at the account level but we are interested in predicting revenue at the level of individual customers, aggregation by customer was needed. Furthermore, revenue and other variables were also aggregated by quarter. Two main reasons are behind this decision: first, some of the macroeconomic variables we are including in our analysis are reported by quarters and, second, doing a more granular analysis would increase the computational requirements for running models.

Given our objective of forecasting, revenues for a quarter are predicted based on inputs of the previous quarter. For example, the forecast of revenue of a customer for the 2nd quarter of 2020 takes into account the number of transactions a customer has done in the 1st quarter of that year. This raises an immediate problem: how can we forecast revenues of a customer that has no previous history with the company? To address this issue, we will divide our data set into two: one for recurring customers (for which we would have such information) and another for new customers (only in their first quarter as customers). Different models are trained for each subset of customers. The model for new customers only uses variables that would be available at the time when the customer opens an account and allows the prediction of revenues for the period in which they enter the bank. Afterwards, the customer becomes a recurring one as the lack of history ceases to be an issue.

The variables in common for these two models are the following:

- **Age** (numerical): age of the customer in years; to retrace this variable to previous periods it was assumed that all customers' birthdays are on January 1st.
- **Gender** (categorical): binary variable with "1" indicating male and "0" female.

- **Quarter** (categorical): quarter of the year; variable that accounts for seasonality in the data; transformed into 3 indicator variables.
- **Invoice accounts** (numerical): number of invoice accounts held by the customer; these accounts are created when a consumer finances a specific purchase at a retailer (e.g., buying a washing machine at a store with credit).
- **Consumer loans** (numerical): number of consumer loans held by the customer; consumer loans are credit products offered by the bank without a specific purpose required (e.g., a customer asks for a loan and is free to spend it on a vacation or refurbishing their house).
- **Credit cards A**, **Credit cards B**, **Credit cards C** (numerical): 3 variables each representing the number of credit cards of a different type held by the customer.
- **Buy-now-pay-later** (numerical): number of buy-now-pay-later accounts held by the customer; these are occurrences when the customer buys something in a store but decides to pay for it at a later date.
- **Insurance** (categorical): binary variable indicating whether a customer subscribed to a payment protection insurance (PPI) policy.
- **Co-applicant** (categorical): binary variable indicating whether a customer had a co-applicant in one of their accounts.
- **Default probability** (numerical): probability of a customer defaulting on a payment; this is calculated by the bank at the time of a customer's application for a consumer loan or payment solution.
- **Total revenue** (numerical): outcome variable we are studying; revenue generated by a customer during one quarter across all accounts.

The variables only available for recurring customers are the following:

- **Number of accounts** (numerical): total number of accounts held by the customer.
- **Longevity** (numerical): period of time that the customer has been with the bank, measured in quarters.
- **Number of transactions** (numerical): number of transactions in the customer's accounts.
- **Loan extensions** (numerical): number of times a customer was approved for increasing the amount of an existing loan; this is a cumulative variable, meaning that it counts the total number of extensions up until the quarter of the analysis.
- **Maximum limit** (numerical): maximum value that the credit limit reached during the quarter, across all accounts.
- **Minimum limit** (numerical): minimum value that the credit limit reached during the quarter, across all accounts.
- **Maximum balance** (numerical): maximum value that the account balance reached during the quarter, across all accounts.

- **Minimum balance** (numerical): minimum value that the account balance reached during the quarter, across all accounts.
- **Late payment** (categorical): binary variable indicating whether a customer was late for a payment on at least one account during the quarter.

Additionally, some observations were removed from the data. Two of such cases were all the instances where the bank product was a savings account or the customer had an unassigned gender. In both situations, the decision of removal was made due to the infrequency of such observations in the data set. Moreover, due to artefacts in the data, a few customers had negative longevity, which were also removed. At last, there were some missing values in the variables *age*, *default* and the ones relating to balance and credit limit, which due to their infrequency were eliminated. In total, 22 684 observations were removed, a very small number compared to the dimension of the final data set.

Similarly, the variable *default probability* also included missing values. However, unlike in the previous case, such missing observations in the final data set ascended to 1 371 018 and removing them would lead to a more significant loss of information. As such, these missing values were imputed according to the method exposed in Subsection 5.1.

3.2 Macroeconomic variables

Regarding the macroeconomic data included in the forecasts, the choice of variables was mostly informed by the literature analysis in Subsection 2.3. GDP, either in absolute value or as a growth rate, was unanimously included in the reviewed papers. In our analysis, we will include GDP growth as this reflects better the impact of the fluctuations of the business cycle on the bank's revenue. Inflation was also nearly universally included, with the only exception being Bikker and Hu (2002). As such, inflation rate will also be included. Furthermore, Clair (2004) and Almaqtari et al. (2019) study the effects of interest rates and exchange rates, which we will also do in this paper. Clair (2004) includes one more variable of interest that will enter our analysis: the unemployment rate.

Besides the variables already researched by previous literature, we will look into the effect of the consumer confidence index and the consumption of durable goods in individual revenues. Considering the bank's focus around consumer loans and retail finance, there is a good reason to suspect its revenues may be connected with consumption of such goods and that this variable, in turn, is influenced by the consumer confidence index.

In the end, the following variables will be included in both models:

- **GDP growth** (numerical): growth rate of real GDP over the previous period, non-seasonally adjusted; source: Euromonitor International's "Passport" database.
- **Inflation** (numerical): growth rate of the price of goods and services over the previous period, non-seasonally adjusted; source: Euromonitor International's "Passport" database.
- **Interest rate** (numerical): lending rates to households on new and renegotiated agreements; average rate for the period; source: Statistics Sweden.
- **Exchange rate** (numerical): effective exchange rate of the Swedish Krona; average rate for the period; source: Riksbank's "the krona index" (KIX).
- **Unemployment rate** (numerical): percentage of active population that is unemployed, non-seasonally adjusted; average rate for the period; source: Euromonitor International's "Passport" database.

- **Consumer confidence index** (numerical): index measuring consumer optimism or pessimism based on surveys; source: National Institute of Economic Research's "Consumer Micro index" with questions relating to the household's economy and plans for future major purchases.
- **Consumption of durables** (numerical): index for the total volume of retail sales of durable goods, non-seasonally adjusted; average index for the period; source: Statistics Sweden.

The frequency for all the variables is quarterly. Taking into account that all accounts in the data set are based in Sweden, the variables described above pertain to this country. Finally, it is important to mention that in our study we have used historical data for the period in which the revenues occur but in a real world scenario these variables would inevitably correspond to forecasts.

3.3 Preliminary analysis

To have a closer look at the data and better understand the characteristics of different features, the data is examined by calculating measures of central tendency and variability. Due to the confidential nature of the data, these statistics are not exhibited but some of the general conclusions are presented in this subsection. Furthermore, scatter plots, histograms and correlation coefficients are analysed, in this case presented without reference to the scale of the values. Due to the large number of variables, these graphs are spread across five plots in the Appendix (see Figures 10, 11, 12, 13, 14) and one in this subsection with some of the highlights (see Figure 1) for recurring customers.

Looking at the statistical information for recurring customers, we can notice that our variable of interest, *total revenue*, has a wide range and several outliers although most of its values are centred around the mean as we can observe in the histogram in Figure 1.

In general, most variables have a distribution that is tightly packed around the mean but including some, possibly extreme, outliers. Their distribution is skewed to the right indicating the presence of observations with very high values. The features *age* and *longevity* are two exceptions to this rule. *Age* is the only variable that follows a distribution close to being normal. *Longevity*'s distribution shows us that most customers are either recent or have been with the bank for many years.

From the histograms of the indicator variables (see Figure 14) it is apparent that the values of *total revenue* for females are more spread out around the mean compared to males. Furthermore, there is a shift towards higher *total revenues* for customers that subscribe to an insurance policy. One can also conclude that the distribution in *gender* is balanced, but only a small fraction of customers have a co-applicant, are late with their payment or subscribe to an insurance product.

Looking at the scatter plots, one can notice that for higher values of *longevity*, values of *loan extensions* are more dispersed. The opposite effect is evident for *loan extensions* and *number of accounts* as there is a noticeable decrease in the variance of *loan extensions* for higher values of *number of accounts*. Furthermore, a slight shift towards higher *total revenues* could be seen with the increasing number of *consumer loans* and *credit cards A*.

Upon examining the correlation coefficients one can notice that the highest positive correlation reaching 0.965 is between *maximum limit* and *maximum balance* which may imply a risk of multicollinearity in the linear regression. There are also strong positive correlations observed between *total revenue* and *maximum limit*, as well as between *total revenue* and *maximum balance*. The coefficients indicate that customers with higher credit limits and balances tend to generate more revenue. Not as strong but still substantial positive correlation between *total revenue* and *consumer loans* reveals a relation between the two variables. In addition, *consumer loans* are significantly

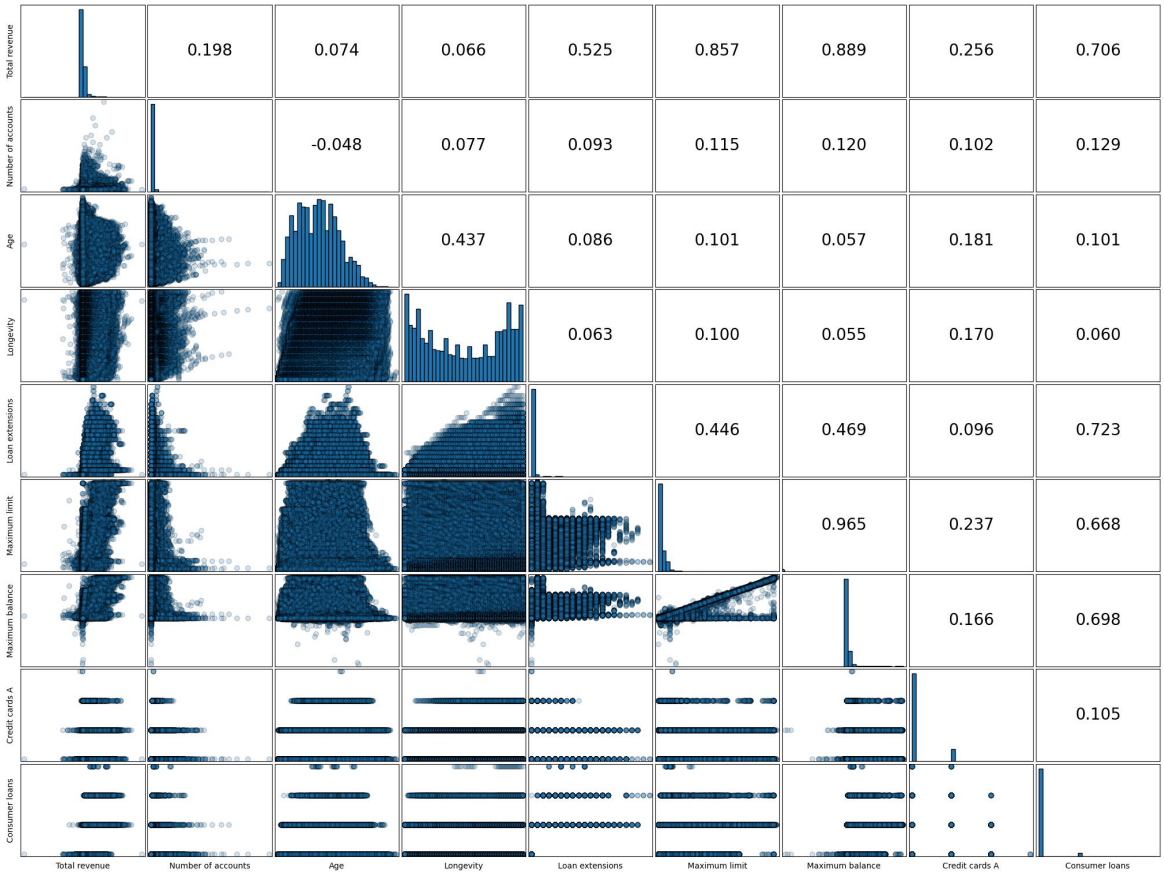


Figure 1: Scatter plots, correlation coefficients and histograms of continuous variables for recurring customers

correlated with several variables such as *maximum balance*, *maximum limit* and *loan extensions*. All correlation coefficients with the exception being the one between *age* and *number of accounts* are positive.

For new customers, the same analysis was conducted to gain new insights into the data set. Furthermore, the figures for this type of customer are spread across three plots in the Appendix (see Figures 15, 16 and 17) and one in this subsection (see Figure 2).

Analysing the summary statistics for new customers, we were able to see that the range of *total revenue* has diminished but the presence of outliers remains. Furthermore, one should point out the drop in the percentage of customers subscribing to an insurance policy compared to recurring customers.

The general conclusions regarding the distribution of the data (right skewness for most variables, with the exception of *age*) remain true for new customers. The variable *gender* continues to be balanced. Moreover, from the histograms of the indicator variables (see Figure 17) it is evident that the distributions of *total revenue* for both genders are almost identical.

Observing the scatter plots in Figure 2, one can observe that a slight shift towards higher *total revenues* persists if a client possesses *consumer loans*. In addition, the variance of *age* diminishes with increasing values of *total revenue*. From the correlation coefficients, it is apparent that the highest correlation is between *total revenue* and *consumer loans*.

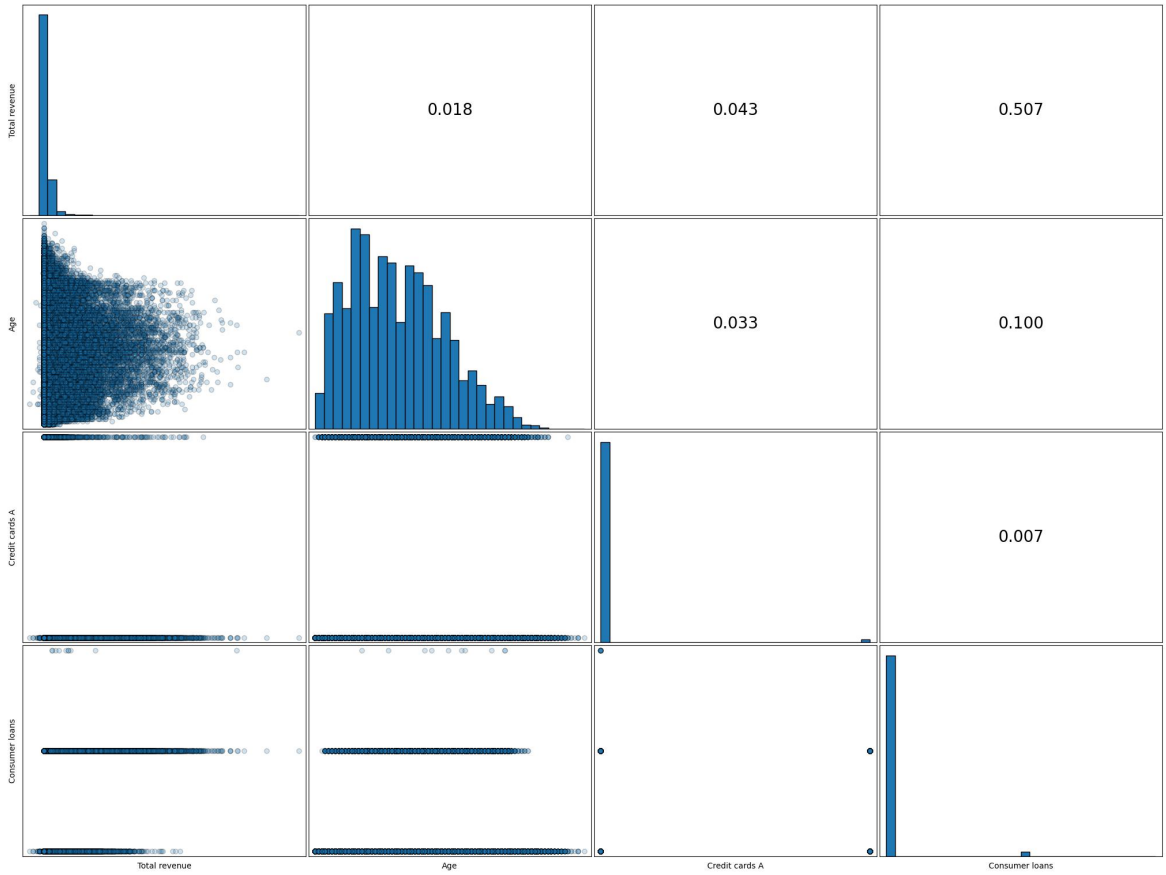


Figure 2: Scatter plots, correlation coefficients and histograms of continuous variables for new customers

4 Theoretical framework

4.1 Linear regression

Linear regression is a supervised machine learning model which captures a linear relationship between a continuous dependent variable denoted by y and independent variables denoted by $\boldsymbol{x} = (x_1, x_2, \ldots, x_p)^T$ (Hastie, Tibshirani & Friedman, 2008). The model is of the following form:

$$f(\boldsymbol{x}) = \beta_0 + \sum_{j=1}^{p} x_j \beta_j. \tag{4.1}$$

Linear regression is chosen in order to determine which variables have a significant impact on the dependent variable. While the model might not produce the most accurate predictions compared to other machine learning methods, its interpretability remains a beneficial feature.

The coefficients β_j's are estimated by the least squares method (Hastie, Tibshirani & Friedman, 2008) which minimises the residual sum of squares.

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{N} (y_i - f(\boldsymbol{x_i}))^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 \tag{4.2}$$

The significance of variables in the linear regression model is evaluated by applying the t-test to individual regressors. For this reason, each β_j coefficient is tested under the hypothesis (Hastie, Tibshirani & Friedman, 2008):

$$\begin{cases} H_0: & \beta_j = 0 \\ H_1: & \beta_j \neq 0. \end{cases} \tag{4.3}$$

For the purpose of using a linear regression model to identify the most significant variables, p-values are adjusted using a Bonferroni correction (Mundfrom et al., 2006). The Bonferroni-adjusted p-values are calculated by using $\frac{\alpha}{n}$ instead of α for each variable to determine the significance where α is a p-value and n is the number of hypothesis tests. The adjusted p-value is then compared to the level of significance and, in case it is bigger, the variable is deemed to be insignificant and removed from the model (Haynes, 2013).

The linear regression model is based on 6 main assumptions (Casson & Farmer, 2014):

- Dependent variable is continuous.
- Linear relationship between dependent and independent variables.
- Absence of correlation between the independent variables and the error term.
- Errors have zero mean conditional on the independent variables.
- Homoscedasticity.
- Uncorrelated errors.

The first assumption is guaranteed depending on the problem at hand. The second assumption can be verified by plotting the dependent variable against the independent ones. The third assumption can be checked by calculating the Pearson correlation coefficient between the independent variables and the error terms. For the fourth assumption to hold, the plot of residuals should show them scattered randomly around the value of zero.

To ensure the robustness of the model, one should verify the assumption of homoscedasticity, In order words, verify whether the variance of the residuals does not depend on the values of the regressors. A model constructed for strongly heteroscedastic data is not reliable because residuals are not equally distributed. The Breusch-Pagan test can be used to check homoscedasticity (Breusch & Pagan, 1979):

$$\begin{cases} H_0: & \text{residuals are homoscedastic} \\ H_1: & \text{residuals are heteroscedastic.} \end{cases} \tag{4.4}$$

Finally, the last assumption requires checking if the residuals of different observations are correlated. The Durbin-Watson test is used to check autocorrelation (Schreiber-Gregory & Bader, 2018; Salehnia, Salehnia, Torshizi, & Kolsoumi, 2020):

$$\begin{cases} H_0: & \text{residuals are not autocorrelated} \\ H_1: & \text{residuals are autocorrelated.} \end{cases} \tag{4.5}$$

Finally, one should be aware of the issue of multicollinearity as it increases the variance of the ordinary least squares estimators (Wißmann, Toutenburg & Shalabh, 2007). Multicollinearity may arise when two or more independent variables are strongly correlated with each other. The variance inflation factor (VIF) can be used to check whether this issue occurs. Different sources suggest different VIF values indicating when variables should be removed, ranging from 5 (Shrestha, 2020) and 10 (Schreiber-Gregory & Bader, 2018) to 100 (Dodge, 2008).

4.2 Decision tree and ensemble methods

4.2.1 Decision tree

Decision trees are rule-based models designed with the purpose of creating subsets of the data that effectively divide the observations into groups as homogeneous as possible in terms of the outcome y (Ekinci & Duman, 2015). To do so, a decision tree applies splits based on rules such as $x_1 < 3$, with x_1 being one of the independent variables in the data set (Lindholm, Wahlström, Lindsten & Schön, 2022). After this split, the observations have been divided into two nodes and the algorithm once again calculates the optimal split, further subdividing the data set. For that reason, decision trees are a recursive partitioning method (Ekinci & Duman, 2015).

If applied to numeric problems, the algorithm is said to be a regression tree (Lindholm et al., 2022). In such a case, the tree evaluates all possible split rules and chooses the best as the one that minimises some loss function. The resulting prediction $\hat{y}(\boldsymbol{x}_*)$ can be written as:

$$\hat{y}(\boldsymbol{x}_*) = \sum_{l=1}^{L} \hat{y}_l \, I(\boldsymbol{x}_* \in R_l) \,, \tag{4.6}$$

where L is the total number of nodes, R_l represents the l-th node after the split is done and $I(\boldsymbol{x}_* \in R_l)$ is an indicator function of whether a point $\boldsymbol{x}_*$ belongs to node R_l. $\hat{y}_l$ is the prediction for the l-th node and is calculated as the average value of the dependent variable for all points in that node.

For classification problems, the procedure is the same with two main differences (Lindholm et al., 2022). First, one must adapt the loss function to the type of problem. Defining the average loss in node l as Q_l and considering a problem with M different classes, some of the most common criteria are:

$$\text{misclassification rate: } Q_l = 1 - \max_m \hat{\pi}_{lm}\,, \tag{4.7}$$

$$\text{Gini index: } Q_l = \sum_{m=1}^{M} \hat{\pi}_{lm}\,(1 - \hat{\pi}_{lm})\,, \tag{4.8}$$

$$\text{entropy criterion: } Q_l = -\sum_{m=1}^{M} \hat{\pi}_{lm}\, ln(\hat{\pi}_{lm})\,, \tag{4.9}$$

where $\hat{\pi}_{lm}$ is the proportion of training data points of class m within the l-th node. Secondly, the predictions are obtained by a majority vote instead of taking the average value of the dependent variable in the node.

After generating a split, a decision tree never takes into consideration changing that split (Lindholm et al., 2022). For each successive split, the previous splits are kept intact and the algorithm is "greedy" as it does not take into account the effect of a split for potential future splits.

Theoretically, a decision tree can be grown to the point where all training data points are predicted without error (Lindholm et al., 2022). However, in practice, this situation would not be ideal as one would be overfitting the data. There are some ways of preventing this, such as defining the maximum depth of the tree beforehand or imposing a minimum number of points in an internal node in order to split it.

4.2.2 Random forest

The predictive power of decision trees can be improved by using ensemble methods, that is, techniques that combine the results of different trees (Lindholm et al., 2022). One of such ensemble methods is bagging. Bagging is an abbreviation for "bootstrap aggregating" and it focuses on reducing variance in low-bias models. In essence, a tree that is grown deep will adapt very well to the training data and, as such, have low bias. However, for the same reason, the predictions of such a tree will be very dependent on the training data set fed into it. Bagging reduces this variance by creating different training data sets by sampling with replacement from the original one, fitting a decision tree for each new data set (called "bootstrap") and, finally, taking the average prediction across all the bootstrapped models. It is this averaging of predictions that allows for a reduction in variance. This can be seen in expression 4.10, which represents the average of B predictions, each with variance σ^2 and positive pairwise correlation ρ (Hastie, Tibshirani & Friedman, 2017). As the number of trees B increases, the second term goes to zero but the same does not occur for $\rho\sigma^2$.

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{4.10}$$

As such, the reduction in the variance achieved by bagging is bounded by the correlation between each tree, ρ. Random forests show up as a response to this problem (Lindholm et al., 2022). The main idea consists of reducing the correlation between the models by introducing more variability in the decision trees fitted, which in turn makes them more distinct from each other. This is achieved by not considering all the variables when deciding on a split but instead only making some independent variables available for the algorithm to choose from when deciding a split. Such method forces the decision trees to take even more distinct paths from the usual, as some of the most popular independent variables may not even be available when making a split.

Several parameters can be chosen by the user to fine tune a random forest including, but not limited to, the number of trees to include and the number of candidate variables to consider at each split, alongside any parameters to tune the individual trees (Lindholm et al., 2022).

Finally, one of the biggest merits of random forests is the fact that it allows for some interpretability of how predictions came to be, namely by calculating metrics that show the importance of each variable. Many methods have been proposed to ascertain the importance of covariates with mean decrease impurity (MDI) being one of the most popular ones (Nembrini, König & Wright, 2018). In the past years, another framework, Shapley values, has been gaining traction in the field of machine learning (Jullum, Redelmeier & Aas, 2021). Unlike MDI, Shapley values allow the user to observe how each feature impacts the dependent variable. Shapley values represent the average marginal contribution of a variable across all different possible combinations of features and are calculated according to the following formula:

$$\phi_j^i = \sum_{S \subseteq X \setminus \{j\}} \frac{|S|!(|X| - |S| - 1)!}{|X|!} [f_i(S \cup \{j\}) - f_i(S)], \tag{4.11}$$

where ϕ_j^i is the Shapley value of the j-th variable for the i-th observation, S is a subset of the features, $|X|$ is the number of features, $|S|$ is the size of the subset and $f_i(S)$ is a function (in this case, a random forest) trained using the variables in subset S and evaluated at observation i.

4.2.3 XGBoost

Another ensemble method is called boosting, which in its turn is focused on bias reduction (Lindholm et al., 2022). The key concept behind boosting is the use of high-bias and low-depth models, called weak learners. Low-depth trees have a small variance but high bias. The combination of several of these high-bias trees can reduce the overall bias of the model, resulting in a strong low-variance and low-bias algorithm. Unlike bagging, boosting constructs the models in a sequential manner (Lindholm et al., 2022). Each model is built to account for the mistakes of the previous model. This is achieved through the adjustment of the training data set by assigning higher weights to miscalculated data points. The final model is achieved by taking the weighted average of all models.

One of the algorithms of boosting is XGBoost. XGBoost is based on the gradient boosting framework which is an additive model represented in the following form (Lindholm et al., 2022):

$$f^{(B)}(\boldsymbol{X}) = \sum_{b=1}^{B} \boldsymbol{\alpha^{(b)}} f^{(b)}(\boldsymbol{X}), \tag{4.12}$$

where $\boldsymbol{\alpha^{(b)}}$ is a vector of coefficients and $f^{(b)}(\boldsymbol{X})$ is a shallow decision tree. The objective at each iteration is to select $\{\boldsymbol{\alpha^{(b)}}, f^{(b)}(\boldsymbol{X})\}_{b=1}^{B}$ which decreases the cost function $J(\boldsymbol{X})$'s value. Therefore, the b-th decision tree is supposed to satisfy the following expression:

$$J\big(f^{(b-1)}(\boldsymbol{X}) + \boldsymbol{\alpha^{(b)}} f^{(b)}(\boldsymbol{X})\big) < J\big(f^{(b-1)}(\boldsymbol{X})\big). \tag{4.13}$$

This is achieved by taking a step in the negative direction of the gradient of the cost function (Lindholm et al., 2022). Given that the cost function is the average loss over the training data, we can calculate the negative gradient of the loss function. This is done for an observation i using the formula:

$$d_i^{(b)} = -\frac{1}{n} \left[\frac{\partial L(y_i, c)}{\partial c} \right]_{c = f^{(b-1)}(\boldsymbol{x_i})}, \tag{4.14}$$

where n is the number of observations, c is the prediction of tree $f^{(b-1)}$ for the i-th observation and $L(y_i, c)$ is the loss function. Afterwards, the model $f^{(b)}(\boldsymbol{X})$ is fitted on the $\{\boldsymbol{x_i}, d_i^{(b)}\}_{i=1}^{n}$.

XGBoost is constructed on the same principles, however, its computational speed is improved through systems optimisation and its performance is boosted due to algorithmic enhancements (Morde, 2019). The algorithm is computationally enhanced by implementing parallelisation, a more efficient tree pruning and optimising available disk space. From the algorithmic point of view, the model is improved due to the inclusion of regularisation that penalises more complex models and built-in cross-validation. Moreover, the method is adapted to data with missing values by assigning these instances to the default direction which is learnt from the data (Chen & Guestrin, 2016). This is done by first splitting the data set between observations without and with missing inputs. Then, the algorithm calculates the reduction in loss of adding the points with a missing value to the left and to the right of the first quantile. The process is repeated for all quantiles. In the end, the points are assigned to the position that resulted in the biggest reduction in loss. This will be the default direction of the model. Furthermore, the model uses the Weighted Quantile Sketch algorithm (Chen & Guestrin, 2016) which speeds up the process of calculating the optimal feature split by dividing the data set into multiple subsets and calculating the quantiles for each data set in parallel. Afterwards, the quantiles of different subsets are combined together to get an approximation of the quantiles for the whole data set which are then checked for the split. Checking splits only at the quantiles is another way of speeding up the process, used by XGBoost as well as other algorithms.

The loss function that can be used in XGBoost for regression problems is the mean squared error (MSE) while for classification problems the softmax function may be chosen (Lindholm et al., 2022):

$$\text{softmax function: } g_m(z_m) = \frac{e^{z_m}}{\sum_{j=1}^{M} e^{z_m}}\,, \tag{4.15}$$

where g_m is the probability of class m and z_m is the logit.

To measure the importance of each feature, one can use Shapley values similarly to random forest.

4.3 Neural network

Neural network is a nonlinear model that could be viewed as the extension of a linear regression as it uses multiple layers of generalised linear regression models stacked on top of each other to capture complex relationships between inputs and outputs (Lindholm et al., 2022). The first hidden layer contains inputs $\mathbf{x}$ multiplied by a weight matrix $\mathbf{W}^{(1)}$ and summed with an offset vector $\mathbf{b}^{(1)}$ (Lindholm et al., 2022). The following hidden layers use the output of its previous layer multiplied by a weight matrix $\mathbf{W}^{(l)}$ and summed with an offset vector $\mathbf{b}^{(l)}$, where $l \in \{2, \ldots, L\}$ is an index that enumerates layers and L stands for the number of layers. Given that there are no cycles or loops in the network, this algorithm is called a feed forward neural network, alluding to the fact that each layer feeds its inputs to the next layer and never the other way around (Kumar & Gayathri, 2021).

$$\begin{aligned}
\mathbf{q}^{(1)} &= h(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}),\\
\mathbf{q}^{(2)} &= h(\mathbf{W}^{(2)}\mathbf{q}^{(1)} + \mathbf{b}^{(2)}),\\
&\vdots\\
\mathbf{q}^{(L-1)} &= h(\mathbf{W}^{(L-1)}\mathbf{q}^{(L-2)} + \mathbf{b}^{(L-1)}),\\
\widehat{y} &= g(\mathbf{W}^{(L)}\mathbf{q}^{(L-1)} + \mathbf{b}^{(L)})
\end{aligned} \tag{4.16}$$

There are U_l hidden units $\mathbf{q}^{(l)} = [q_1^{(l)} \dots q_{U_l}^{(l)}]$ in each hidden layer. Moreover, the non-linearity in the model is introduced by using a non-linear activation function $h : \mathbb{R} \to \mathbb{R}$. One of the most popular activation functions is called a rectified linear unit (ReLU) (Lindholm et al., 2022). For a regression problem, the activation function $g : \mathbb{R} \to \mathbb{R}$ used in the output layer is tipically a linear function. After the initialisation of the model, the best parameters are found by minimising the cost function $J(\boldsymbol{\theta})$ (Lindholm et al., 2022):

$$\widehat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}), \text{ where } J(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} L(\boldsymbol{x_i}, y_i, \boldsymbol{\theta}). \tag{4.17}$$

The most commonly used loss function $L(\boldsymbol{x_i}, y_i, \boldsymbol{\theta})$ for regression problems is the squared error loss. Numerical optimisation algorithms are used to find the best parameters through iterative updates such as Adam or RMSprop (Lindholm et al., 2022). Both methods use gradient based search to update weights during training. The Adam method relies on the usage of gradients from previous steps by giving higher weights to recent gradients and lower weights to older gradients. The search direction d_t and the learning rate γ_t are updated using the formulas:

$$\begin{aligned} d_t &= (1-\beta_1)\sum_{i=1}^{t}\beta_1^{t-i}\nabla J_i, \\ \gamma_t &= \frac{\eta}{\sqrt{t}}\left((1-\beta_2)\ \text{diag}\left(\sum_{i=1}^{t}\beta_2^{t-i}\|\nabla J_i\|^2\right)\right)^{1/2}, \end{aligned} \tag{4.18}$$

where ∇J_i is the gradient, t is the iteration number and the rest (β_1, β_2 and η) are the tuning parameters. β_1 is the exponential decay rate for the first moment estimate, β_2 is the exponential decay rate for the second moment estimate and η is the initial learning rate.

Neural networks are universal functional approximators which may lead to overfitting. One of the ways to avoid overfitting is to use regularisation techniques. However, modifying the cost function is not the only way to achieve this. Two other popular methods are early stopping and dropout (Lindholm et al., 2022). The main idea of the early stopping is to suspend the optimisation procedure once a certain condition is satisfied. Usually the method is implemented by choosing a validation data set and calculating validation loss after each epoch. Once the validation loss starts increasing, the procedure is stopped even though the training loss keeps decreasing. This method helps to prevent overfitting because it focuses on the validation data set instead of the training data set. Dropout aims to diminish the risk of overfitting by randomly deactivating a proportion of the hidden units in a certain layer (Lindholm et al., 2022). This way a subset of the units is obtained at each epoch (a full cycle through the training data) as different neurons are dropped each time. Generated sub-networks share only some parameters with each other which introduces some level of randomness and can improve the model's ability to generalise to new data.

4.4 Support vector regression

As the name suggests, support vector regression is an adaptation of support vector machines to problems with numeric outcomes. In order to describe the theory behind this method, we will begin by exploring non-linear transformations of inputs and kernel ridge regression, following the explanation outlined in Lindholm et al. (2022).

A linear regression model can be extended by including non-linear transformations of the input x, for example by including powers of that variable in the model (Lindholm et al., 2022). We can represent by $\boldsymbol{\phi(x)}$ the set of all inputs (original and transformed) and by d the number of inputs in this new model. As such, we have $\boldsymbol{\phi(x)} = [1\ x\ x^2 \dots x^{d-1}]^T$.

Kernel ridge regression originates from a linear regression model with L^2-regularisation, where the number of features d is allowed to be very large, possibly approaching infinity (Lindholm et al., 2022). The general solution for the learned parameters $\hat{\boldsymbol{\theta}}$ of such a ridge regression is given by:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x_i}) - y_i)^2 + \lambda ||\boldsymbol{\theta}||_2^2 \,, \tag{4.19}$$

where λ is the regularisation parameter. In matrix notation, formula 4.19 can be written as:

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{\Phi}(\boldsymbol{X})^T \boldsymbol{\Phi}(\boldsymbol{X}) + n\lambda \boldsymbol{I})^{-1} \boldsymbol{\Phi}(\boldsymbol{X})^T \boldsymbol{y} \,. \tag{4.20}$$

From this, predictions can be obtained simply by multiplying the learned coefficients by the vector of transformed inputs (Lindholm et al., 2022).

$$\hat{y}(x_*) = \hat{\boldsymbol{\theta}}^T \boldsymbol{\phi}(\boldsymbol{x_*}) \tag{4.21}$$

Equation 4.21 can be rewritten as:

$$\hat{y}(x_*) = \boldsymbol{y}^T \, (\boldsymbol{\Phi}(\boldsymbol{X}) \boldsymbol{\Phi}(\boldsymbol{X})^T + n\lambda \boldsymbol{I})^{-1} \, \boldsymbol{\Phi}(\boldsymbol{X}) \boldsymbol{\phi}(\boldsymbol{x_*}) \,, \tag{4.22}$$

where:

$$\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\Phi}(\boldsymbol{X})^{\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x_1})^T \boldsymbol{\phi}(\boldsymbol{x_1}) & \boldsymbol{\phi}(\boldsymbol{x_1})^T \boldsymbol{\phi}(\boldsymbol{x_2}) & \dots & \boldsymbol{\phi}(\boldsymbol{x_1})^T \boldsymbol{\phi}(\boldsymbol{x_n}) \\ \boldsymbol{\phi}(\boldsymbol{x_2})^T \boldsymbol{\phi}(\boldsymbol{x_1}) & \boldsymbol{\phi}(\boldsymbol{x_2})^T \boldsymbol{\phi}(\boldsymbol{x_2}) & \dots & \boldsymbol{\phi}(\boldsymbol{x_2})^T \boldsymbol{\phi}(\boldsymbol{x_n}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{\phi}(\boldsymbol{x_n})^T \boldsymbol{\phi}(\boldsymbol{x_1}) & \boldsymbol{\phi}(\boldsymbol{x_n})^T \boldsymbol{\phi}(\boldsymbol{x_2}) & \dots & \boldsymbol{\phi}(\boldsymbol{x_n})^T \boldsymbol{\phi}(\boldsymbol{x_n}) \end{bmatrix} \text{ and} \tag{4.23}$$

$$\boldsymbol{\Phi}(\boldsymbol{X})\boldsymbol{\phi}(\boldsymbol{x_*}) = \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{x_1})^T \boldsymbol{\phi}(\boldsymbol{x_*}) \\ \boldsymbol{\phi}(\boldsymbol{x_2})^T \boldsymbol{\phi}(\boldsymbol{x_*}) \\ \vdots \\ \boldsymbol{\phi}(\boldsymbol{x_n})^T \boldsymbol{\phi}(\boldsymbol{x_*}) \end{bmatrix} \,. \tag{4.24}$$

One should notice that all the elements in the matrices 4.23 and 4.24 are scalars. This opens the door to computational savings, as we may only compute the product $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x'})$ instead of explicitly calculating the d-dimensional vectors $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\phi}(\boldsymbol{x'})$ (Lindholm et al., 2022).

This technique leads us to the concept of kernels. A limited definition proposed by Lindholm et al. (2022) describes a kernel as "any function that takes two arguments $\boldsymbol{x}$ and $\boldsymbol{x'}$ from the same space and returns a scalar". Following this concept, we can rewrite equation 4.22 as:

$$\hat{y}(x_*) = \boldsymbol{y}^T \, (\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + n\lambda \boldsymbol{I})^{-1} \, \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x_*}) \,, \tag{4.25}$$

where $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$ is known as the Gram matrix and is obtained by evaluating the kernel at all training points. That is:

$$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{x_1}, \boldsymbol{x_1}) & \boldsymbol{k}(\boldsymbol{x_1}, \boldsymbol{x_2}) & \dots & \boldsymbol{k}(\boldsymbol{x_1}, \boldsymbol{x_n}) \\ \boldsymbol{k}(\boldsymbol{x_2}, \boldsymbol{x_1}) & \boldsymbol{k}(\boldsymbol{x_2}, \boldsymbol{x_2}) & \dots & \boldsymbol{k}(\boldsymbol{x_2}, \boldsymbol{x_n}) \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{k}(\boldsymbol{x_n}, \boldsymbol{x_1}) & \boldsymbol{k}(\boldsymbol{x_n}, \boldsymbol{x_2}) & \dots & \boldsymbol{k}(\boldsymbol{x_n}, \boldsymbol{x_n}) \end{bmatrix} \,, \tag{4.26}$$

where $\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x})$ is any generic kernel. Likewise, the matrix $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x_*})$ is obtained in the following way:

$$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_*) = \begin{bmatrix} \boldsymbol{k}(\boldsymbol{x_1}, \boldsymbol{x}_*) \\ \boldsymbol{k}(\boldsymbol{x_2}, \boldsymbol{x}_*) \\ \vdots \\ \boldsymbol{k}(\boldsymbol{x_n}, \boldsymbol{x}_*) \end{bmatrix} . \tag{4.27}$$

The choice of kernel is left for the user and is, in most part, informed by the type of problem at hand. $\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x'})$ is an example of a kernel, but many others exist (Lindholm et al., 2022). A popular kernel choice is the radial basis function (RBF) kernel, defined as:

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x'}) = e^{-\frac{||\boldsymbol{x}-\boldsymbol{x'}||_2^2}{2\gamma^2}} , \tag{4.28}$$

where γ is a hyperparameter defined by the user or found using cross validation methods.

Looking back at equation 4.25, one can see that only $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}_*)$ depends on the test data. As such, we only need to calculate $\boldsymbol{y}^T\,(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + n\lambda\boldsymbol{I})^{-1}$ once, which can be stored in a $1 \times n$ vector denominated henceforth as $\hat{\boldsymbol{\alpha}}$.

At last, we arrive at support vector regression. SVR can be interpreted as an extension of kernel ridge regression, as described previously (Lindholm et al., 2022). The main distinguishing factor of SVR is its loss function, which can be defined as:

$$L(y, \hat{y}) = \max(0, |y - \hat{y}| - \epsilon) , \tag{4.29}$$

where ϵ is chosen by the user.

Similar as before (see equation 4.19), the optimal parameters $\hat{\boldsymbol{\theta}}$ are found by solving the following problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} \max(0, |y_i - \boldsymbol{\theta}^T\boldsymbol{\phi}(\boldsymbol{x_i})| - \epsilon) + \lambda||\boldsymbol{\theta}||_2^2 \tag{4.30}$$

Following the rationale applied to kernel ridge regression, we can get predictions for the model with:

$$\hat{y}(x_*) = \hat{\boldsymbol{\alpha}}^T\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x_*}) , \tag{4.31}$$

where $\hat{\alpha}$ this time does not have a closed-form solution, but instead needs to be found through numerical methods. These numerical methods attempt to solve the following optimisation problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} (\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{y} + \epsilon||\boldsymbol{\alpha}||_1) \tag{4.32}$$

This new formulation lets us interpret the parameter ϵ in a different way: it is the parameter that controls the level of regularisation in a lasso regression (Lindholm et al., 2022). The implication of this fact becomes clearer once we note that the loss function is only non-zero for points that fulfil the condition $|\hat{y}(x_i) - y_i| \geq \epsilon$. This makes the vector $\hat{\alpha}$ sparse. As such, the predictions obtained from equation 4.31 only depend on a fraction of the points in the training data set. These points are called the "support vectors".

Additionally, this zone where the loss function is zero can be extended with the help of a new hyperparameter: C (Rogić et al., 2021). This parameter controls the elasticity of this zone, with smaller values of C reducing the number of support vectors, and vice-versa.

In summary, a SVR is trained on all data but is only affected by data points with non-zero loss for making predictions. The parameters ϵ and C control how many of these support vectors we

consider when making predictions: the larger the ϵ and the smaller the C, the fewer the support vectors. Predictions are obtained by calculating equation 4.31 and, in order to do so, one must numerically solve the problem in equation 4.32.

4.5 K-means clustering

Clustering algorithms are methods to divide a data set into smaller groups in such a way that points within the same cluster are as similar to each other as possible and as dissimilar as possible from points in different clusters (Hastie, Tibshirani & Friedman, 2017). The most popular of these methods is the k-means algorithm, privileged for its simplicity (Rogić et al., 2021).

K-means requires all the variables to be quantitative, that all points get assigned to only one cluster and that no cluster is left without members (Hastie, Tibshirani & Friedman, 2017; Sivasankar & Vijaya, 2017). Under this method, the level of dissimilarity between two points is measured by the squared Euclidean distance (Hastie, Tibshirani & Friedman, 2017). The problem underlying the learning of the clusters in the k-means algorithm can be summarised in expression 4.33.

$$\min_{\boldsymbol{C},\{m_k\}_1^K} \sum_{k=1}^{K} \sum_{i=1}^{N} I(C(i)=k) \sum_{C(i)=k} ||\boldsymbol{x_i} - \boldsymbol{m_k}||^2 , \tag{4.33}$$

where C corresponds to the cluster assignment, K is the number of clusters, $\{m_k\}_1^K$ is the set of means of the clusters and $I(C(i) = k)$ is an indicator function of whether a point i belongs to cluster k. Consequently, we are looking for the cluster assignments C, to which corresponds a set of means $\{m_k\}_1^K$, that minimises for all clusters the average distance between points in a cluster and the cluster mean.

Solving the minimisation problem in 4.33 is done by resorting to an alternating optimisation procedure (Hastie, Tibshirani & Friedman, 2017). The first step is to determine the number of clusters. This is a decision left for the user but techniques like the elbow method can help inform the choice. The elbow method consists in learning K clusters using k-means and tracking a cost function along the way, starting with $k = 1$ and increasing that number progressively (Liu & Deng, 2020). As K increases, the clustering becomes more accurate and the cost decreases. As K increases past its optimal point, these gains become increasingly smaller. We can then plot the evolution of this cost function. We should expect it to take the shape of an elbow and choose the value K that stands at the inflexion point of this graph. Having decided on K, the algorithm runs as follows (Rogić et al., 2021):

- Step 1: Sample K training data points to be the initial cluster centroids $\{m_k\}_1^K$.
- Step 2: For each vector $\boldsymbol{x_i}$, calculate the distance to the K centroids.
- Step 3: Assign each vector $\boldsymbol{x_i}$ to the cluster whose centroid is at the shortest distance.
- Step 4: Calculate the new cluster centroids $\{m_k\}_1^K$.
- Step 5: Repeat steps 2-4 until the recalculation of cluster centroids remains unchanged.

K-means is a simple algorithm that can be used for diverse clustering problems. Nevertheless, it contains some drawbacks, namely its dependence on the initial step of sampling the initial clustering centroids (Rogić et al., 2021). One way of controlling for this issue is by running the model several times, initialising it with different centroids each time.

4.6 ROC and AUC

A common way to evaluate the performance of a classification model is to look at the AUC - the area under the ROC (receiver opearating characteristic) curve (Rogers & Girolami, 2016). The ROC curve is created by plotting sensitivity against the false positive rate for all thresholds ranging from 0 to 1. Threshold is a value used as a cutoff point to classify an observation into one of the classes. Sensitivity and false positive rate are calculated as:

$$\text{Sensitivity: } \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}\,, \tag{4.34}$$

$$\text{False positive rate: } 1 - \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}\,. \tag{4.35}$$

The maximum possible value of AUC is 1, indicating a perfect classification of the data, while a classifier with a random class allocation will have an AUC equal to 0.5.

4.7 Class-imbalance problem and SMOTE+Tomek

Classification problems in machine learning are often plagued with the problem of class-imbalance. This has been identified as an issue by many authors and it refers to the tendency of classification algorithms to ignore instances of a class that only represents a small minority of the data set (Chawla, Japkowicz & Kołcz, 2004). Many methods have been proposed to overcome this issue and improve the performance of classifiers in predicting a minority class, a class which in spite of its small size is not less important. In fact, in many circumstances, it is the outcome we are most interested in predicting. One of such methods corresponds to oversampling the minority class or classes.

A simple way of oversampling the minority class is to randomly duplicate points in the training set that belong to that class but such a method can lead to overfitting (Batista, Prati & Monard, 2004). A method that surpasses this issue is called "synthetic minority over-sampling technique" (SMOTE), which achieves this by generating new points that are interpolated from the original data but don't replicate it completely (Batista, Prati & Monard, 2004). In essence, SMOTE generates a new point $\boldsymbol{x_{new}}$ by first selecting a random point from the original data set belonging to the minority class $\boldsymbol{x_i}$ (Zeng, Zou, Wei, Liu & Wang, 2016). In a second step, the algorithm finds the K-nearest neighbours of $\boldsymbol{x_i}$ belonging to the same class and randomly chooses one, $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$. The new sample is thus generated by adding to the original point $\boldsymbol{x_i}$ a fraction δ of the distance between $\boldsymbol{x_i}$ an its neighbour $\hat{\boldsymbol{x}}_{\boldsymbol{i}}$ using the following equation:

$$\boldsymbol{x_{new}} = \boldsymbol{x_i} + \delta(\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x_i})\,. \tag{4.36}$$

The number of neighbours K and the proportion of each class after the resampling are choices the user needs to make. The parameter δ is a random value between zero and one.

The algorithm can be adapted to handle categorical variables by using SMOTE-NC (Chawla, Bowyer, Hall & Kegelmeyer, 2002). This method introduces a change in the way of finding the K-nearest neighbours by using a different Euclidean distance computation. For continuous variables, the calculation is the same but for numerical variables a penalty is introduced if the value of that variable is different between the two points. This penalty corresponds to the median of the standard deviations of all continuous variables belonging to the minority class. Let's take the example where we have two continuous variables, X_1 and X_2, and two categorical ones, X_3 and X_4, two points $A = (5.0, 6.5, 1, 0)$ and $B = (3.5, 2.0, 1, 1)$ and the median of the standard deviation of X_1 and

X_2 for all points belonging to the minority class corresponds to Med. The Euclidean distance between points A and B is equal to: $\sqrt{(5.0-3.5)^2+(6.5-2.0)^2+0^2+Med^2}$. Points A and B have the variable X_3 in common so it contributes with a zero to the Euclidean distance, while there is a difference in variable X_4 leading to the penalty. Additionally, the method assigns values of the categorical variables for the new sample in a different way than the one described in equation 4.36. For categorical variables, the new sample gets assigned the most frequent value among the K neighbours of the original point.

SMOTE can be combined with Tomek links with the latter being a way of cleaning the data. Two points are said to form a Tomek link if one belongs to the minority class and the other to the majority class and each corresponds to the closest neighbour of the other point (Zeng et al., 2016). In this context, a point $\boldsymbol{x_i}$ is the closest neighbour to another point $\boldsymbol{x_j}$ if there is no other third point $\boldsymbol{x_k}$ that has a smaller Euclidian distance to the point $\boldsymbol{x_j}$. After identifying the points that form a Tomek link, either both points or just the point belonging to the majority class are removed from the data set depending on the objective (Batista, Prati & Monard, 2004). If used as a technique to undersample the majority class, only the point belonging to that category is removed. If used for cleaning the data, both points are removed. The combination of SMOTE with Tomek links can be beneficial as the oversampling of points in the minority class can lead to these points invading the space of the majority class. By removing points from different classes that are close together one can achieve better-defined class boundaries.

5 Methodology

5.1 Imputation of missing data in *default probability*

Given that after the preprocessing we were left with 1 371 018 missing values for the variable *default probability*, we complemented our data set by performing imputation, thus saving a considerable proportion of it. There are many ways to address this problem, ranging from the most simple ones like imputing missing values using a mean, median or mode (Shah et al., 2022) to more sophisticated approaches such as multiple imputation by chained equations (MICE) (Jaramillo et al., 2021), KNN (Pazhoohesh et al., 2021) or random forest (Sahoo & Ghose, 2022) .

Data imputation models commonly face limitations such as the inability to handle mixed data types or capture nonlinear relations between variables (Tang & Ishwaran, 2017), resulting in less accurate predictions. However, the aforementioned problems could be overcome by using a random forest which not only accounts for the interactions between variables but is also easily scalable (Tang & Ishwaran, 2017). In particular, the missforest algorithm, a variant of random forest, has proven to have the lowest imputation error compared to other methods such as MICE and KNN (Waljee et al., 2013). When examining other available random forest algorithms for data imputation such as the proximity imputation and on-the-fly-imputation, research indicates that all three methods exhibit similar performance in settings with low to medium correlations between variables (Tang & Ishwaran, 2017). Given that *default probability* isn't highly correlated with other variables and missforest has displayed superior performance compared to other methods (Stekhoven & Bühlmann, 2012), it is chosen as a preferable method.

Missforest was introduced as a non-parametric model that can handle different types of data by Stekhoven and Bühlmann (2011). The main idea behind the method (Tang & Ishwaran, 2017) is to fill missing values by iteratively regressing each variable against other variables and calculating predictions for the response variable until a stopping criterion is met. In our case, we are interested in only one variable, therefore the fitting is applied only once by treating *default probability* as the dependent variable and calculating predictions for its missing values. The same procedure is applied separately for new and recurring customers since different variables are present in the data sets.

5.2 Forecasting models

Linear regression, random forest and support vector regression were among the most used methods for revenue or profit prediction in the studies referenced in Section 2. As such, we will be including these models in our analysis. Furthermore, Fang, Jiang and Song (2016) have used generalised boosted models, but failed to outperform random forests. In this paper, we will try a different boosting model, XGBoost, which has become very popular among these type of algorithms. Finally, we will also experiment with using a neural network as this method can be of great use to capture non-linear relationships between inputs and output but seems to be unexplored in the available literature.

The five models - linear regression, random forest, support vector regression, XGBoost and neural network - will be trained and tested on predicting revenues from the 1st quarter of 2019 until the 3rd quarter of 2022. Furthermore, as explained before in Subsection 3.1, each model will be fit on the data for recurring and new customers. In order to compare them, three metrics will be analysed: mean absolute error, mean squared error and root mean squared error. While mean absolute error penalises errors in a linear way, mean squared error and root mean squared error penalise errors in a quadratic way. In turn, the main difference between squared error and root

mean squared error resides in the fact that the former is expressed in squared units while the latter is expressed in the same units as the dependent variable.

All models with the exception of linear regression, require the choice of hyperparameters. Given that, for the most part, there are no rules to obtain these best parameters, we will run random searches through some of them. Random searches consist of randomly sampling combinations of parameters, fitting models on all or a sample of the data and deciding on the best combination with recourse to cross-validation. The details of these searches alongside other approaches we took in this paper can be found in Subsections 9.1 to 9.7 in the Appendix.

Finally, having determined the two best models for predicting revenues from the 1st quarter of 2019 until the 3rd quarter of 2022, one for recurring and another for new customers, they will be retrained on all data from that period and used to forecast revenues in the 4th quarter of 2022. This experiment puts to the test the forecasting performance of these models in a situation as similar as possible to the real world.

5.2.1 Linear regression

One of the main reasons for the use of a linear regression is the model's interpretability. To preserve such interpretability, no transformations of the variables such as taking the logarithm or square root were used.

In a first stage, the model is fit on all variables using the ordinary least squares method. Then, the variables are tested for their significance at a 5% level using the Bonferroni correction to control the family-wise error rate. In case there are any insignificant variables, the one with the highest p-value is removed and the model is retrained on the remaining explanatory features. This cycle of learning the model, testing variables' significance and removing the least significant one repeats until only significant variables remain.

Subsequently, the VIF of the variables is calculated and all the ones with a value greater than 5 are removed. As seen previously in Subsection 4.1, there is no consensus about the value to choose as a threshold to remove variables. The choice of 5 corresponds to the most conservative value suggested by literature. This process is also conducted by removing one variable at a time, starting with the one with the highest VIF. If, along this process, some variable becomes insignificant, it is also removed.

Afterwards, we verify if the assumptions of linear regression hold in order to ensure the validity of our results.

At last, we arrive at the final model and we are free to interpret the coefficients of each independent variable and gain some understanding of the impact of each variable on individual revenues.

5.2.2 Random forest

Before fitting two random forests, one for recurring customers and another for new ones, we will run a random search through 5 hyperparameters. A description of hyperparameters with the available values and results can be found in the Appendix (see Subsection 9.1). The search was conducted with just 50 thousand points due to computational costs and went through 50 different combinations.

After fitting the best models, random forests allow us to check for variable importance by calculating Shapley values for each variable. Given the computational costs of this task, Shapley values were calculated on a subsample of 20 thousand points collected from the test data.

5.2.3 XGBoost

Due to its faster computations, XGBoost's random search was conducted with 1 million points for the model with recurring customers and all the data for new customers, both times evaluating 50 different combinations on 7 parameters (see Subsection 9.2). Furthermore, the loss function used to evaluate the splits was the squared error loss. Absolute error loss was not tested given its unavailability in the package used.

Finished with the random search, two models were fitted, for new and recurring customers, and Shapley values were calculated for both, once again using a subsample of 20 thousand points.

5.2.4 Neural network

Defining the architecture of a neural network is a somewhat arbitrary process. From trial and error, we have found that a network with 4 hidden layers, with respectively 2000, 1000, 500 and 250 hidden units, works well on the data. Furthermore, the inclusion of dropout layers after each hidden layer proved to be beneficial. As such, 4 dropout layers were added with, respectively, a 50%, 30%, 30% and 20% dropout rate. Ultimately, the activation function for the hidden layers is the ReLU function, which has become the most popular one for this kind of networks in recent years (Lindholm et al., 2022). In addition, the output layer has a linear activation function due to the output variable being numerical.

Before running a neural network, one should "standardize all inputs to have mean zero and standard deviation one" (Hastie, Tibshirani & Friedman, 2017).

After rescaling the data, a random search is carried out for two parameters (see Subsection 9.4). The random search was carried out with 1 million points for recurring customers and all data for new customers, checking 10 combinations for each model. In each trial, the model ran for a maximum of 30 epochs, with early stopping (Hastie, Tibshirani & Friedman, 2017) being implemented. That is, the model ran for 30 epochs unless it failed to improve validation loss for 5 consecutive epochs, in which case learning is aborted.

After finding the best hyperparameters, two networks are learned for the two types of customers with the architecture described before, again for 30 epochs with early stopping being implemented.

5.2.5 Support vector regression

Similarly to neural networks, data needs to be rescaled before working with SVR. After some experimentation, using a scaling method more robust to outliers proved to be more successful than the transformation described for neural networks. Therefore, features were scaled by subtracting the median, instead of the mean, and dividing by the interquartile range, instead of the standard deviation.

Possibly the most important decision in SVR concerns the choice of kernel. The polynomial kernel and the RBF kernel are two of the most popular choices in literature. Liu, Li and Tan (2005) found the polynomial kernel to be more robust than RBF, however we have found in our analysis that using the polynomial kernel is too computationally costly for this problem. Hence, the RBF kernel will be used in the analysis.

Once again, we look for adequate hyperparameters using a random search (see Subsection 9.5). Given the high computational costs of running a SVR, a sample is used for both the random search and the final model. For the random search, the sample includes 40 thousand observations for both new and recurring customers, running for 10 trials. For the final model, a larger sample of 100 thousand data points was used.

5.2.6 Forecasting test on the 4th quarter of 2022

At last, as explained previously, the models are compared based on mean absolute error, mean squared error and root mean squared error, allowing us to identify the best model for recurring customers and for new customers. The models are retrained on all data from the 1st quarter of 2019 to the 3rd quarter of 2022 and generate predictions for the customers in the 4th quarter of 2022.

5.3 Cluster-based method

As mentioned in Section 2, a common approach when forecasting revenues or profits is to first cluster the customers based on some criteria and then fit separate models for each cluster. We will test the effectiveness of such a method by conducting an experiment on forecasting revenues of the 4th quarter of 2022. To do so, we will focus our analysis on the model which showed the best performance when forecasting for the period starting with the 1st quarter of 2019 up until the 3rd quarter of 2022, both in terms of recurring and new customers. These two separate models will be trained for different clusters of customers on all data spanning from the 1st quarter of 2019 to the 3rd quarter of 2022 and, afterwards, we will forecast revenues for the 4th quarter of 2022. From these forecasts, we will once again calculate the mean absolute error, mean squared error and root mean squared error, which will be used to compare the cluster-based method with the one in Subsection 5.2.6.

5.3.1 Clustering into revenue segments

The cluster-based method is built on, first, clustering customers based on their revenue without looking into any other characteristics. This allows us to split the customers into revenue segments. The assumption for such a method to improve forecasts is that the revenues of customers in different segments are impacted by different variables or by the same ones but in a different manner.

Rogić et al. (2021), Khalili-Damghani, Abdi and Abolmakarem (2018) and Sivasankar and Vijaya (2017) have used the k-means algorithm to successfully divide customers into segments. Furthermore, considering it is a simple algorithm (Rogić et al., 2021) and we are clustering based on only one variable, k-means was the method chosen to find the revenue segments. In order to overcome the algorithm's dependence on the initial cluster centroids, we will run it 10 times with different initialisations. Moreover, the number of clusters will be determined using the elbow method.

The clusters will be discovered on data from the 1st quarter of 2019 to the 3rd quarter of 2022. Having defined the clusters, one can train the best model in the data from the same period, this time doing it for each cluster.

Lastly, the forecasts for the 4th quarter of 2022 are obtained in a two-step procedure. First, assign a customer to a revenue segment using a classification model (see Subsection 5.3.2). Second, run that observation through the model for the corresponding cluster and generate a prediction of revenue for that individual (see Subsection 5.3.3).

5.3.2 Classification of customers into clusters

The optimal choice of a classification model varies based on the task at hand. In a paper with a similar task of bank customer segmentation (Abedin et al., 2023) a random forest yielded the highest accuracy on the primary data while XGBoost achieved better results on the data sets which were preprocessed with feature selection methods. Among the 10 compared models,

including naive bayes, logistic regression and support vector machine, random forest and XGBoost were concluded to be the top-performing models. Another research focusing on the classification of social media posts demonstrated that XGBoost slightly outperformed a random forest (Arora, Srivastava & Bansal, 2019). Chaubey et al. (2022) studied the classification of customer purchasing behaviour and found random forest and XGBoost to have the same accuracy. Considering the similar performance of both models and their varying superiority in different studies, both random forest and XGBoost are tested for customer segmentation.

Both models are fine-tuned using a random search (see Subsections 9.6 and 9.7) with the same arguments as in the Subsections 5.2.2 and 5.2.3 with the exception of criterion for the random forest and objective for the XGBoost. Given that we are faced with a classification problem, the criterion for the random forest is changed from a squared error and absolute error to the entropy and Gini impurity while the objective for the XGBoost is switched from the squared error loss to the softmax.

One important aspect of the classification problem is the fact that the number of customers in each cluster differs significantly, leading to the issue of class-imbalance which is mostly significant for new customers. Batista, Prati and Monard (2004) have found that "the problem seems to be related to learning with too few minority class examples in the presence of other complicating factors, such as class overlapping". The authors found a combination of over-sampling and under-sampling techniques to be the most effective. One of the two best performing methods was the use of SMOTE for generating new samples of the minority class and the removal of points forming Tomek links for cleaning the data. A more recent study (Sharma & Gosain, 2023) supported the previous authors' conclusion, having found SMOTE+Tomek to outperform four other techniques. As such, it is no surprise to find that this method has been used in several papers where authors had to deal with unbalanced data sets (e.g., Zeng et al., 2016; Liu, Wu, Mirador, Song & Hou, 2018; Hairani, Anggrawan & Priyanto, 2023).

Therefore, the random forest and XGBoost models will be fitted without any corrections and also using SMOTE+Tomek to account for the imbalance in the data set for new customers. For recurring customers, our analysis showed such a correction to not be necessary. When implementing SMOTE, we will first resample the data so that all classes have an equal distribution in the data set. The number of neighbours used in the SMOTE process is equal to 5. Afterwards, points from both the majority and minority classes forming a Tomek link are removed in order to clean the resampled data set.

The resulting models are compared with respect to the score of the AUC in order to determine the best one. There are multiple reasons to choose AUC over other evaluation metrics. To begin with, standard classifiers are centred around the overall prediction accuracy which might have an impact on the misclassification of smaller classes as the evaluation is biased towards larger ones in unbalanced data sets (Xu et al., 2023). To address this issue, two other measures, AUC and F1 scores, are commonly implemented in the literature (e.g., Mansourifar & Shi, 2021; Xu et al., 2022; Xu et al., 2023; Sadreddin & Sadaoui, 2022). In our case AUC score is preferred to F1 because we are equally interested in the accurate prediction of all classes and F1 score focuses on the classification of minority classes (Wang & Liu, 2023).

5.3.3 Forecasting test on the 4th quarter of 2022

To reiterate, in Subsection 5.3.1 we divided the customers into different revenue segments for the period spanning the 1st quarter of 2019 up until the 3rd quarter of 2022. The two best models identified previously, one for recurring and another for new customers, are then trained separately for each cluster. Subsequently, using the best classification model identified in Subsection 5.3.2 the

customers in the 4th quarter of 2022 are assigned to a cluster and, afterwards, the model trained specifically for that cluster is used to get a final revenue prediction.

Finally, the performance of this method will be compared to the one without clustering (see Subsection 5.2.6) with the objective of evaluating if running separate models for each revenue segment produces any significant improvement in forecasting capabilities.

6 Analysis of results

6.1 Linear regression

6.1.1 Estimation results

Our analysis started with running a linear regression on all the variables for recurring customers (see Subsection 9.8.1). This resulted in an R-squared of 0.84 with all but two variables being considered significant at a 5% level.

At first, these results seem to be exceptional, but a further analysis into the values of the VIF of the variables discloses an underlying issue of multicollinearity. This issue is evident in Table 10 and it undermines the results obtained before. Such a problem is expected given that, for example, the variable *number of accounts* is the sum of the different disaggregate variables for bank products, such as *invoice accounts* and *consumer loans*, resulting in a VIF greater than one million.

Consequently, some variables were removed in an iterative process as described in Subsection 5.2.1. In that process, the variables *age*, *number of accounts*, *maximum limit*, *minimum limit*, *GDP growth*, *interest rate*, *exchange rate*, *unemployment rate*, *consumer confidence index*, *consumption of durables*, *quarter two* and *quarter four* were removed from the model either due to their high multicollinearity with other variables or due to having become insignificant in subsequent models. As a result, the remaining variables all have VIF values smaller than 5 (see Table 10).

The final regression produced the same R-squared of 0.85 but this time one can be more confident of the results encountered. The final estimates can be found in Table 1.

Variable	**Coefficient**	**Standard error**	**P-value**
Constant	- 88.91	0.55	0.00
Gender	- 12.99	0.32	0.00
Longevity	- 0.38	0.01	0.00
Insurance	295.30	0.70	0.00
Loan extensions	197.81	2.06	0.00
Co-applicant	- 43.91	2.79	0.00
Invoice accounts	47.83	0.33	0.00
Buy-now-pay-later	77.07	0.58	0.00
Credit cards A	402.23	0.96	0.00
Credit cards B	92.25	0.42	0.00
Credit cards C	449.58	2.64	0.00
Consumer loans	628.24	5.25	0.00
Default probability	1 608.56	13.06	0.00
Minimum balance	<0.01	<0.01	0.00
Maximum balance	0.02	<0.01	0.00
Late payment	115.44	0.71	0.00
Number of transactions	- 1.12	0.02	0.00
Inflation	- 0.21	0.15	0.00
Quarter three	14.82	0.37	0.00

Table 1: Summary of the final linear regression for recurring customers

The same approach was taken for new customers (see Subsection 9.8.2). The first regression with all variables produced an R-squared of 0.39 and all but two variables were deemed significant.

Both insignificant variables were macroeconomic indicators: *GDP growth* and *exchange rate*.

Once more, the VIF for all the variables was calculated (see Table 11), allowing us to detect and eliminate multicollinearity from the model. After repeating the iterative process of removing variables due to insignificance or multicollinearity, we arrived at the final model for new customers. The estimation results of the final model can be found in Table 2.

Variable	**Coefficient**	**Standard error**	**P-value**
Constant	38.57	3.09	0.00
Gender	30.32	1.04	0.00
Insurance	1 721.07	31.86	0.00
Co-applicant	312.89	17.12	0.00
Invoice accounts	- 14.25	2.79	0.00
Buy-now-pay-later	199.26	3.14	0.00
Credit cards A	179.51	6.39	0.00
Credit cards B	15.82	3.00	0.00
Credit cards C	42.61	6.56	0.00
Consumer loans	1 256.15	19.21	0.00
Default probability	- 95.33	25.67	0.00
Quarter four	- 21.45	1.12	0.00

Table 2: Summary of the final linear regression for new customers

6.1.2 Checking assumptions of linear regression

To have confidence in the results presented before, one must check that the assumptions of linear regression (see Subsection 4.1) have not been violated.

The first assumption of the dependent variable being continuous is ensured as one can interpret *total revenue* as taking on any real value even if in practice its value cannot be subdivided beyond 1 cent (for Swedish krona, öre).

The second assumption of a linear relationship between dependent and independent variables can be checked by observing the plots in Subsection 9.10 of the Appendix. Such examination makes it apparent that the assumption may be deemed somewhat feeble. However, it is also evident that there is an absence of any polynomial or exponential relationships. As such, it can be concluded that the second assumption remains unviolated.

The third assumption calls for no correlation between the independent variables and the error term. To verify its observance, we have calculated the Pearson correlation coefficient between each independent variable and the residuals (see Table 12), which reveals that the assumption holds.

The fourth assumption requires errors to have zero mean conditional on the independent variables, which can be verified with Figure 7 in the Appendix. The mean of the residuals is effectively zero for both regressions. For recurring customers, the residuals take the form of a cloud distributed along the value of zero. For new customers, a pattern of heteroscedasticity is evident but this does not impact the conclusions for the fourth assumption.

The fifth and sixth assumptions require the use of two tests: Breusch-Pagan and Durbin-Watson. Their results are summarised in Table 3. The conclusion is the same for both models: we reject the null hypothesis in the Breusch-Pagan test but not in the Durbin-Watson test.

Considering the results from the Bresch-Pagan test on heteroscedasticity, the standard errors calculated in Subsection 6.1.1 are robust. The heteroscedasticity robust standard errors used are

Test	Purpose	Recurring customers	New customers
Breusch-Pagan	Heteroscedasticity	<0.01	<0.01
Durbin-Watson	Correlation of residuals	>0.99	>0.99

Table 3: Results from statistical tests on the linear regressions

the ones defined according to the HC3 specification, whose covariance matrix is calculated with the formula (Long & Ervin, 2000):

$$HC3 = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\,\boldsymbol{X}^T\,diag\left(\frac{e_i^2}{(1-\boldsymbol{x_i}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{x_i}^T)^2}\right)\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\,, \tag{6.1}$$

where e_i represents the residual of the i-th observation.

The final assumption of correlation holds given the result of the Durbin-Watson test.

6.1.3 Importance and effect of variables

For recurring customers, an analysis of Table 1 shows that both *default probability* and *late payment* have a positive impact on revenues. This is a direct result of how banks operate: customers with higher risk are charged a higher interest rate, resulting in higher revenues. In addition, if a customer misses a payment, they are also expected to generate more revenue as a result of fees and penalties. For new customers, Table 2 indicates the opposite relation between *default probability* and revenues, although with a much smaller impact. This seems to indicate that this variable is less important in determining revenues of new customers compared to recurring ones.

Another important factor in determining bank revenues is the number of products a customer subscribes to. Consumer loans have the highest impact but all the bank product variables have a significant positive impact on revenues, with the notable exception of *invoice accounts* for new customers.

For the variables only available for recurring customers, some expected conclusions followed from the results. For example, there is evidence of a positive relation between *maximum balance* and revenues. However, some others were somewhat surprising such as *longevity* and *number of transactions* having a negative impact on revenues.

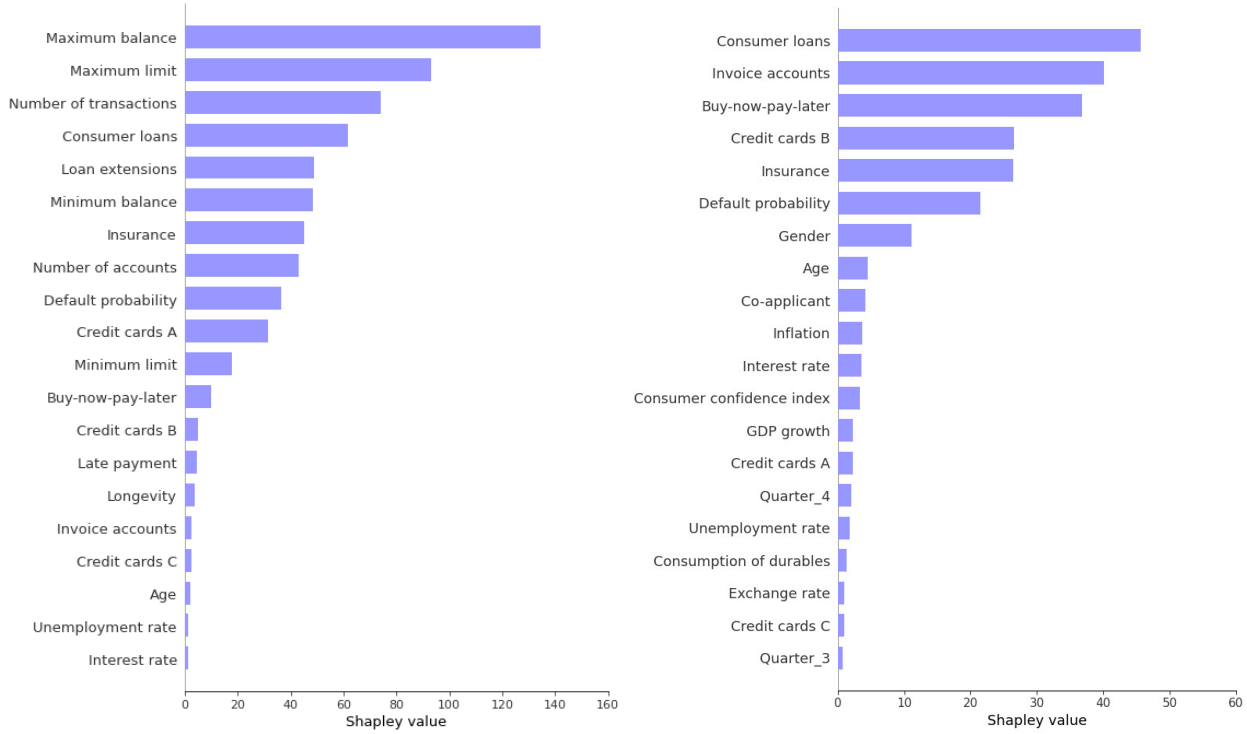
In the model for new customers, the *co-applicant* feature gains more importance and has a positive rather than slightly negative impact on revenues.

Regarding the impact of macroeconomic variables, most were even removed from the final models, indicating no significant impact of economic conditions on individual revenues.

6.2 Random forest: Importance and effect of variables

An examination of the average absolute Shapley values for each variable (see Figure 3) gives us an indication of the features that the model deemed more important when predicting revenue.

For recurring customers, variables related to the customer's financial position in the previous quarter seem to be the most influential (*maximum balance* and *maximum limit*) together with the customers' behaviour regarding *number of transactions*. Furthermore, *consumer loans* and *insurance* are the two most influential bank product variables, but others also have a relevant impact on revenues. In addition, the random forest also gave significance to the indicator variable *loan extensions* and the variable *default probability*.



(a) Recurring customers

(b) New customers

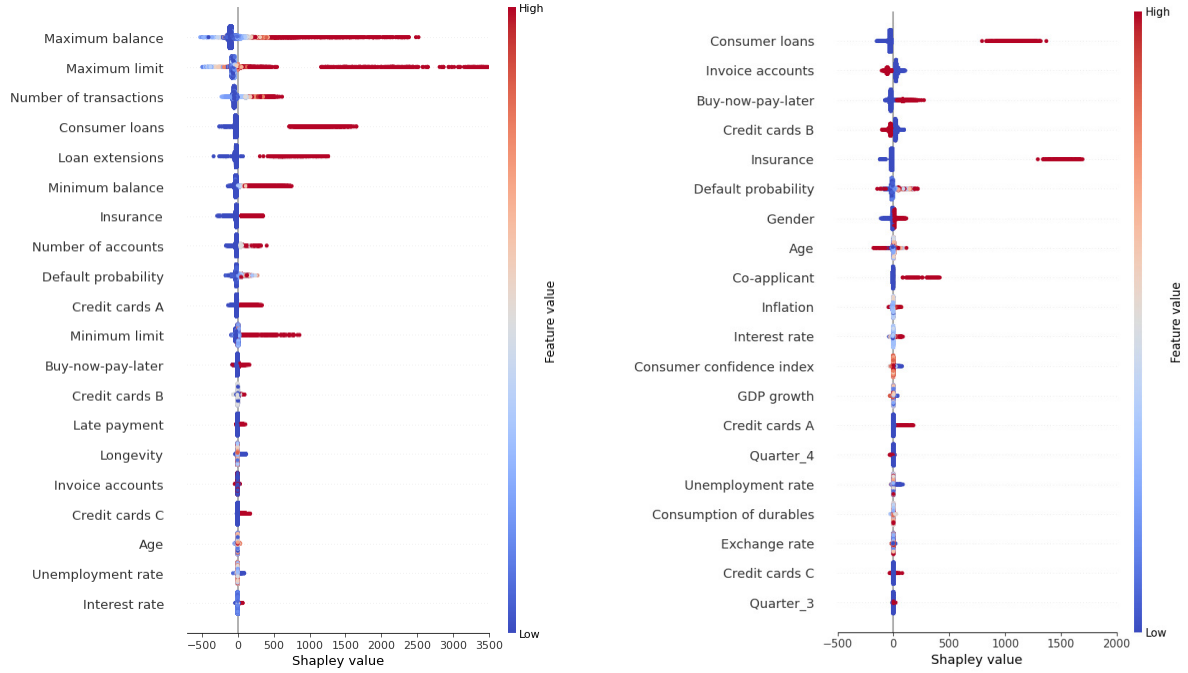
Figure 3: Average absolute Shapley values of the 20 most important variables in random forest

For new customers, most of the features mentioned in the previous paragraph are simply not available for the model. As such, the random forest places a higher emphasis on the variables related to bank products, with *consumer loans*, *invoice accounts* and *buy-now-pay-later* taking the leading places in the ranking. The variable *default probability* is also considered important.

Common to both models, little importance is given to variables related to macroeconomic data, repeating the result found in linear regression.

Besides analysing the average absolute values, one can plot the Shapley values for different observations in a beeswarm plot (see Figure 4). This plot allows us to identify the relation between the variables and revenue. That is, if high values of a feature lead to a positive or negative impact in revenues.

The general conclusions are not surprising: a customer with more products, higher balance and higher limit is expected to generate more revenue. This analysis also allows us to confirm some conclusions from the linear regression for new customers: the negative relation between having an invoice account and revenues and the positive relation with having a co-applicant. In addition, we also observe as in the linear regression a positive relation between *default probability* and revenue for recurring customers. For new customers, we now get a better picture of the effects of this variable: higher values of *default probability* either correspond to a more positive or negative impact in revenues while lower values of *default probability* have a smaller impact in revenues altogether. Finally, macroeconomic variables seem to once again not have a significant impact on predictions.



(a) Recurring customers

(b) New customers

Figure 4: Shapley values of the 20 most important variables in random forest

6.3 XGBoost: Importance and effect of variables

In a similar fashion to random forests, we can also calculate Shapley values for XGBoost. The average absolute values can be found in Figure 8 and more details can be found in the beeswarm plot in Figure 9, both present in the Appendix.

In general, there are only small differences between the Shapley values calculated for XGBoost compared to the ones for random forest. For recurring customers, the balance of a customer is the most important factor determining revenues with *maximum balance* having by far the highest Shapley values in absolute value and *minimum balance* having the fourth highest. Furthermore, XGBoost seems to place a lower emphasis on the variable *loan extensions* and a higher emphasis on *number of accounts*. For new customers, the model produces very similar results compared to random forest. Overall, all of the main conclusions stated in Subsection 6.2 hold and the consensus between linear regression, random forest and XGBoost when interpreting the importance and effect of variables gives us more confidence in the findings obtained.

6.4 Comparison of forecasting performance of the models

Having fitted all the models for recurring and new customers, it was time to compare them based on the three previously mentioned metrics: mean absolute error (MAE), mean squared error (MSE) and root mean squared error (RMSE).

The results for recurring customers can be found in Table 4. Linear regression and SVR are the models with the worst performance, while neural network and XGBoost distinguish themselves for the opposite reason. XGBoost is, in fact, the indisputable best model out of the five as it achieves a lower error across all three metrics. As such, XGBoost is used for modeling the behaviour of recurring customers in the second phase of our analysis when we test the effect of having different

models for distinct clusters of customers.

	MAE	MSE	RMSE
Linear regression	210	193 281	440
Random forest	156	142 906	378
XGBoost	136	109 334	331
Neural network	136	114 377	338
Support vector regression	193	381 362	618

Table 4: Model performance for recurring customers

For new customers, the results are presented in Table 5. The performance of the five models becomes less disperse when it comes to modeling new customers. This time, SVR actually outperforms all other models by comparing MAE. However, the model also has the highest MSE/RMSE indicating its poor accuracy at predicting outliers. This leaves us with neural network (2nd best MAE and MSE/RMSE) and XGBoost (3rd best MAE and the best MSE/RMSE) as the alternatives for modeling new customers' revenue. Considering that XGBoost is less computationally costly and has already been chosen as the best model for recurring customers, this was our choice for the second phase of our analysis.

	MAE	MSE	RMSE
Linear regression	153	137 843	371
Random forest	150	135 003	367
XGBoost	146	124 509	353
Neural network	141	127 418	357
Support vector regression	129	213 498	462

Table 5: Model performance for new customers

6.5 Classification algorithms

When it comes to the algorithms for predicting which revenue segment a customer belongs to, random forest and XGBoost performed very similarly when trained and tested on data from the 1st quarter of 2019 until the 3rd quarter of 2022.

For recurring customers, XGBoost achieved a slightly higher accuracy and AUC (see Table 6).

	Accuracy	AUC
Random forest	0.969	0.984
XGBoost	0.973	0.987

Table 6: Performance of classification algorithms when predicting clusters for recurring customers

An analysis of the confusion matrix (see Figure 5) gives us a breakdown of the accuracy of the model for each class.

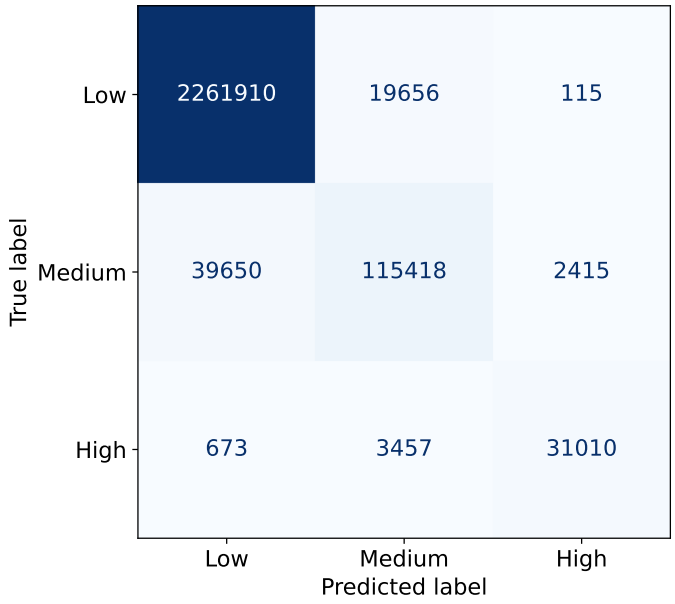


Figure 5: Confusion matrix of XGBoost for recurring customers

XGBoost achieves a very high AUC and the confusion matrix reveals that the class-imbalance does not seem to excessively affect the prediction capabilities for classes "medium" and "high".

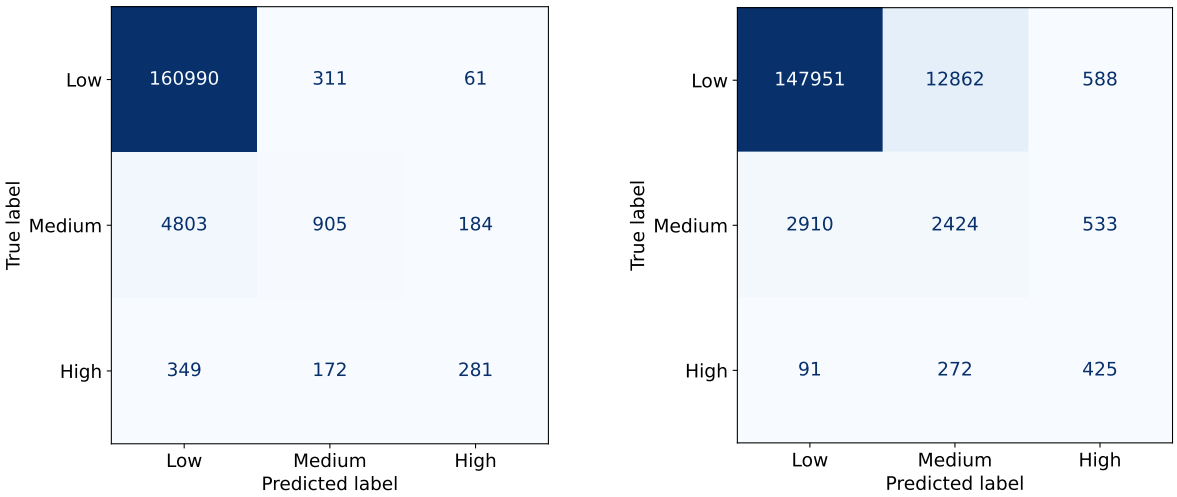
The results for new customers are presented in table 7. This time, random forest achieves a better accuracy but XGBoost has a higher AUC, the metric used to choose the best model. However, an analysis of the confusion matrix of this model (see Figure 6a) reveals a very poor prediction capability for classes "medium" and "high". As such, the XGBoost model was also fitted on a resampled training data set using the SMOTE+Tomek method.

	Accuracy	**AUC**
Random forest	0.966	0.903
XGBoost	0.965	0.917
XGBoost with SMOTE+Tomek resampling	0.897	0.875

Table 7: Performance of classification algorithms when predicting clusters for new customers

The resampling strategy resulted in an increased accuracy at predicting revenue segments "medium" and "high" at the expense of accurately predicting the majority class (see Figure 6b). Such a trade-off is inevitable when trying to alleviate the problem of class-imbalance in a classification task.

In summary, XGBoost will be used to predict the cluster that a customer belongs to during our forecasting test on the 4th quarter of 2022, both for recurring and new customers. For new customers, given the effects of class-imbalance in the prediction of the minority classes, we will also experiment with the XGBoost model fitted on a resampled data set using the SMOTE+Tomek method.



(a) XGBoost

(b) XGBoost with SMOTE+Tomek resampling

Figure 6: Confusion matrices for new customers

6.6 Forecasting test on the 4th quarter of 2022

Having clustered customers into revenue segments, XGBoost was trained on each cluster for both recurring and new clients. Parameters for each model were fine-tuned using a random search. The examined parameters and results can be found in Subsection 9.3. When comparing the method without clustering to the approach with client segmentation for recurring customers, it is evident that the adaptation of different models for each category resulted in a slightly better performance across all three metrics - MAE, MSE and RMSE (see Table 8). Looking at the results for new customers, the opposite effect is apparent as XGBoost trained on data without clustering yielded the best results (see Table 9). Furthermore, clustering on a resampled training data set produced the least precise predictions. One potential explanation for this outcome is that the resampling process resulted in a higher incidence of misclassifications for clients with low revenues, which constitutes the majority of the data set. Therefore, the improved precision for the classification of medium and high revenues did not outweigh the increased number of errors within the low revenue segment.

	MAE	**MSE**	**RMSE**
XGBoost	154	217 096	466
Clusters + XGBoost	149	209 039	457

Table 8: Model performance for recurring customers on the 4th quarter of 2022

From the results one can notice that customer segmentation may lead to better performance in terms of revenue forecasting on the individual level. However, having in mind the higher score of AUC for the classification of recurring customers compared to new customers, it becomes apparent that achieving a distinct separation between clusters is crucial for the effectiveness of the segmentation method.

	MAE	MSE	RMSE
XGBoost	287	527 709	726
Clusters + XGBoost	291	682 369	826
Clusters with SMOTE+Tomek resampling + XGBoost	356	789 446	889

Table 9: Model performance for new customers on the 4th quarter of 2022

7 Conclusion

This paper set out to answer three main research questions (see Section 1) and the findings are hereby summarised.

Regarding the first question of discovering the best model for predicting revenues of individual bank customers, we compared five different methods: linear regression, random forest, XGBoost, neural network and support vector regression. Our analysis found XGBoost to have the best performance both in terms of MAE and MSE/RMSE for recurring customers. For new customers, support vector regression achieved the best MAE but proved to be unreliable at predicting outliers as it simultaneously obtained the highest MSE/RMSE. XGBoost and neural network, on the other hand, proved to be successful at forecasting revenues even for outliers.

The success of XGBoost is not surprising as past literature (Fang, Jiang & Song, 2016; Larivière & Van den Poel, 2005) also found a tree-based ensemble method, namely random forest, to be the best at predicting revenues. XGBoost had not been tested in past literature, potentially due to being a relatively new method, but it has demonstrated to outperform the other four models.

The second research question regarding the importance of macroeconomic variables in revenue prediction was answered by the models that allow for some interpretability, specifically by analysing the coefficients of linear regression and the Shapley values of random forest and XGBoost. There is a consensus between the three methods indicating little to no impact of these variables on the revenues of individual customers. Previous literature refrained from including any macroeconomic variables and such a decision appears to be justified. The extensive existing literature covers the impact of macroeconomic variables on banks' total revenues (see Subsection 2.3), seemingly indicating that such variables affect the aggregate results of banks but not the revenues of individual customers.

The final question explored the feasibility of fitting different models for each revenue segment based on a two-step procedure: first, assign customers into a cluster based on a classification model and, next, get a prediction for that customer's revenue based on the model for that respective cluster. Our analysis found a small improvement for recurring customers but a significant increase in forecasting error for new customers. This difference in results is caused by the distinct classification accuracy when assigning the clusters. For recurring customers, the classification model is very precise and, as such, the method improves forecasts by making predictions on models that are better fit for that customer segment. For new customers, the classification model makes excessive errors when assigning clusters, making prediction errors more exacerbated for customers that, for example, belong to the cluster of "high" revenue but are incorrectly assigned the "low" label. In summary, we can conclude that the cluster-based method is viable if the classification of customers into said clusters is accurate enough.

Finally, we conclude this paper by indicating avenues for potential further research which were not pursued either due to the unavailability of data or to time constraints. First, it would be interesting to confirm the conclusions for the second research question by extending the period of analysis beyond the four years analysed in the paper. Second, we used SMOTE+Tomek to deal with the unbalanced data set for new customers but other methods can be explored to attempt to improve cluster predictions for each new customer. At last, the model for predicting revenues for recurring customers could be complemented by a separate machine learning model predicting the probability of churn, allowing the bank to calculate the expected revenue of a customer for a future period.

8 References

Abedin, M., Hajek, P., Sharif, T., Satu, S. & Khan, I. (2023). Modelling Bank Customer Behaviour Using Feature Engineering and Classification Techniques, Research in International Business and Finance, [e-journal] vol. 65, p.101913, Available Online: https://www.sciencedirect.com/science/article/pii/S0275531923000399 [Accessed 8 May 2023]

Aftabi, S., Ahmadi, A. & Farzi, S. (2023). Fraud Detection in Financial Statements Using Data Mining and GAN Models, Expert Systems with Applications, [e-journal] vol. 227, p.120144, Available Online: https://www.sciencedirect.com/science/article/pii/S0957417423006462 [Accessed 18 May 2023]

Almaqtari, F., Al-Homaidi, E., Tabash, M. & Farhan, N. (2018). The Determinants of Profitability of Indian Commercial Banks: A panel data approach, International Journal of Finance & Economics, [e-journal] vol. 24, no. 1, pp.168–185, Available Online: https://onlinelibrary.wiley.com/doi/epdf/10.1002/ijfe.1655?src=getftr [Accessed 17 May 2023]

Arora, A., Srivastava, A. & Bansal, S. (2019). Business Competitive Analysis Using Promoted Post Detection on Social Media, Journal of Retailing and Consumer Services, vol. 54, p.101941, Available Online: https://www.sciencedirect.com/science/article/pii/S0969698919306708 [Accessed 18 May 2023]

Athanasoglou, P., Delis, M. & Staikouras, C. (2006). Determinants of Bank Profitability in the South Eastern European Region, Athens: Bank of Greece, Available Online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4163741 [Accessed 17 May 2023]

Batista, G., Prati, R. & Monard, M. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data, ACM SIGKDD Explorations Newsletter, [e-journal] vol. 6, no. 1, pp.20–29, Available Online: https://dl.acm.org/doi/pdf/10.1145/1007730.1007735 [Accessed 17 May 2023]

Batten, J. & Vo, X. V. (2019). Determinants of Bank Profitability: Evidence from Vietnam, Emerging Markets Finance and Trade, [e-journal] vol. 55, no. 6, pp.1417–1428, Available Online: https://www.tandfonline.com/doi/epdf/10.1080/1540496X.2018.1524326?src=getftr [Accessed 17 May 2023]

Bikker, J. & Hu, H. (2002). Cyclical Patterns in Profits, Provisioning and Lending of Banks, Amsterdam: DNB Staff Reports, Available Online: https://www.researchgate.net/publication/4799455_Cyclical_Patterns_in_Profits_Provisioning_and_Lending_of_Banks [Accessed 17 May 2023]

Breusch, T. & Pagan, A. (1979). A Simple Test for Heteroscedasticity and Random Coefficient Variation, Econometrica, [e-journal] vol. 47, no. 5, pp.1287–1294, Available Online: https://www.jstor.org/stable/1911963?seq=1 [Accessed 24 May 2023]

Casson, R. & Farmer, L. (2014). Understanding and Checking the Assumptions of Linear Regression: A primer for medical researchers, Clinical & Experimental Ophthalmology, [e-journal] vol. 42, no. 6, pp.590–596, Available Online: https://onlinelibrary.wiley.com/doi/epdf/10.1111/ceo.12358?src=getftr [Accessed 18 May 2023]

Chaubey, G., Gavhane, P., Bisen, D. & Arjaria, S. (2022). Customer Purchasing Behavior Prediction Using Machine Learning Classification Techniques, Journal of Ambient Intelligence and Humanized Computing, Available Online: https://link.springer.com/article/10.1007/s12652-022-03837-6 [Accessed 18 May 2023]

Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. (2002). SMOTE: Synthetic minority over-sampling technique, Journal of Artificial Intelligence Research, [e-journal] vol. 16,

pp.321–357, Available Online: https://arxiv.org/pdf/1106.1813.pdf [Accessed 23 May 2023]

Chawla, N., Japkowicz, N. & Kolcz, A. (2004). Editorial: Special issue on learning from imbalanced data sets, ACM SIGKDD Explorations Newsletter, [e-journal] vol. 6, no. 1, p.1, Available Online: https://dl.acm.org/doi/pdf/10.1145/1007730.1007733 [Accessed 17 May 2023]

Chen, T. & Guestrin, C. (2016). XGBoost: A scalable tree boosting system, in 2016 KDD Conference, 2016, San Francisco: Knowledge Discovery and Data Mining, Available Online: https://arxiv.org/pdf/1603.02754.pdf [Accessed 17 May 2023]

Clair, R. (2004). Macroeconomic Determinants of Banking Financial Performance and Resilience in Singapore, Singapore: Monetary Authority of Singapore, Available Online: https://www.mas.gov.sg/~/media/MAS/Monetary%20Policy%20and%20Economics/Education%20and%20Research/Research/Economic%20Staff%20Papers/2004/MAS_Staff_Paper_No_38_RSTC_V3.pdf [Accessed 17 May 2023]

Dodge, Y. (2008). The Concise Encyclopedia of Statistics, 1st edn, [e-book] New York: Springer, pp.96–98, Available Online: https://link.springer.com/referenceworkentry/10.1007/978-0-387-32833-1_66 [Accessed 19 May 2023]

Ekinci, Y. & Duman, E. (2015). Intelligent Classification-Based Methods in Customer Profitability Modeling, Intelligent Techniques in Engineering Management, [e-journal] vol. 87, pp.503–527, Available Online: https://link.springer.com/chapter/10.1007/978-3-319-17906-3_20?utm_source=getftr&utm_medium=getftr&utm_campaign=getftr_pilot [Accessed 17 May 2023]

Euromonitor International. (2023). Passport Database, Available Online: https://www.euromonitor.com/our-expertise/passport [Accessed 10 April 2023]

Fang, K., Jiang, Y. & Song, M. (2016). Customer Profitability Forecasting Using Big Data Analytics: A case study of the insurance industry, Computers & Industrial Engineering, [e-journal] vol. 101, pp.554–564, Available Online: https://www.sciencedirect.com/science/article/pii/S0360835216303515 [Accessed 17 May 2023]

Flamini, V., McDonald, C. & Schumacher, L. (2009). The Determinants of Commercial Bank Profitability in Sub-Saharan Africa, Washington, DC: International Monetary Fund, Available Online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1356442 [Accessed 17 May 2023]

Hairani, H., Anggrawan, A. & Priyanto, D. (2023). Improvement Performance of the Random Forest Method on Unbalanced Diabetes Data Classification Using Smote-Tomek Link, JOIV : International Journal on Informatics Visualization, [e-journal] vol. 7, no. 1, pp.258–264, Available Online: https://joiv.org/index.php/joiv/article/view/1069 [Accessed 17 May 2023]

Hastie, T., Tibshirani, R. & Friedman, J. (2008). Springer Series in Statistics the Elements of Statistical Learning Data Mining, Inference, and Prediction Second Edition, 2nd edn, [e-book] Stanford, California: Springer, pp.43–48, 392–401, 501-512, 587–597, Available Online: https://hastie.su.domains/Papers/ESLII.pdf [Accessed 17 May 2023]

Haynes, W. (2013). Bonferroni Correction, Encyclopedia of Systems Biology, [e-book] New York, NY: Springer, pp.154–154, Available Online: https://link.springer.com/referenceworkentry/10.1007/978-1-4419-9863-7_1213 [Accessed 23 September 2022]

Hung, J., He, W. & Shen, J. (2019). Big Data Analytics for Supply Chain Relationship in Banking, Industrial Marketing Management, [e-journal] vol. 86, Available Online: https://www.sciencedirect.com/science/article/pii/S0019850118304681#bb0255 [Accessed 18 May 2023]

Jaramillo, J., Formigari, G., Vale, D., Ursini, E. & Martins, P. (2021). Missing Data: Comparison of multiple-imputation algorithms for social determinants of health in cervical cancer stage detection, in IEEE Xplore, 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, 2021, IEEE, pp.509–514, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9623097 [Accessed 18 May 2023]

Jigeer, S. & Koroleva, E. (2023). The Determinants of Profitability in the City Commercial Banks: Case of China, Risks, [e-journal] vol. 11, no. 3, p.53, Available Online: https://www.mdpi.com/2227-9091/11/3/53 [Accessed 17 May 2023]

Jullum, M., Redelmeier, A. & Aas, K. (2021). GroupShapley: Explaining predictive models using Shapley values and non-parametric vine copulas, Dependence Modeling, [e-journal] vol. 9, no. 1, pp.62–81, Available Online: https://martinjullum.com/publication/preprint-jullum-2021-groupshapley/ [Accessed 17 May 2023]

Khalili-Damghani, K., Abdi, F. & Abolmakarem, S. (2018). Hybrid Soft Computing Approach Based on Clustering, Rule Mining, and Decision Tree Analysis for Customer Segmentation Problem: Real case of customer-centric industries, Applied Soft Computing, [e-journal] vol. 73, pp.816–828, Available Online: https://www.sciencedirect.com/science/article/pii/S1568494618305052 [Accessed 17 May 2023]

Kumar, D. & Gayathri, A. (2021). Accuracy Analysis of Data Fraud Detection for Company Transactions Using Two Layered Feed Forward Neural Network Approach Compared with Random Forest, in ECS Transactions, Vol. 107, 1st International Conference on Technologies for Smart Green Connected Society 2021, Online, 2021, Institute of Physics, pp.13481–13490, Available Online: https://www.scopus.com/record/display.uri?eid=2-s2.0-85130576320&origin=resultslist&sort=r-f&src=s&st1=feed+forward+neural+network&nlo=&nlr=&nls=&sid=f3177341bfe712a80d1467688f315b13&sot=b&sdt=b&sl=42&s=TITLE-ABS-KEY%28feed+forward+neural+network%29&relpos=0&citeCnt=0&searchTerm [Accessed 17 May 2023]

Larivière, B. & Van den Poel, D. (2005). Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques, Expert Systems with Applications, [e-journal] vol. 29, no. 2, pp.472–484, Available Online: https://www.sciencedirect.com/science/article/pii/S0957417405000965 [Accessed 17 May 2023]

Lindholm, A., Wahlström, N., Lindsten, F. & Schön, T. (2022). Machine Learning: A first course for engineers and scientists, Cambridge: Cambridge University Press, pp.25–36, 112, 133–161, 163–187, 189–202, Available Online: http://smlbook.org/book/sml-book-draft-latest.pdf [Accessed 17 May 2023]

Liu, C., Wu, J., Mirador, L., Song, Y. & Hou, W. (2018). Classifying DNA Methylation Imbalance Data in Cancer Risk Prediction Using SMOTE and Tomek Link Methods, in Communications in Computer and Information Science, Vol. 902, International Conference of Pioneering Computer Scientists, Engineers and Educators, Singapore, 2018, Springer, Available Online: https://link.springer.com/chapter/10.1007/978-981-13-2206-8_1 [Accessed 17 May 2023]

Liu, F. & Deng, Y. (2021). Determine the Number of Unknown Targets in Open World Based on Elbow Method, IEEE Transactions on Fuzzy Systems, [e-journal] vol. 29, no. 5, pp.986–995, Available Online: https://ieeexplore.ieee.org/document/8957623 [Accessed 24 May 2023]

Liu, J., Li, J. & Tan, Y. (2005). An Empirical Assessment on the Robustness of Support Vector Regression with Different Kernels, in Fourth International Conference on Machine Learning and Cybernetics, 2005, Guangzhou: IEEE, pp.4289–4294, Available Online: ht

tps://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1527691&tag=1 [Accessed 17 May 2023]

Long, J. & Ervin, L. (2000). Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model, The American Statistician, [e-journal] vol. 54, no. 3, pp.217–224, Available Online: https://www.jstor.org/stable/2685594?seq=1 [Accessed 17 May 2023]

Mansourifar, H. & Shi, W. (2021). Cross-Concatenation: Tackling Uncertainty in Imbalanced Big Data Classification, in IEEE Xplore, 2021 IEEE International Conference on Big Data (Big Data), Orlando, 1 December 2021, IEEE, pp.867–875, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9671763 [Accessed 18 May 2023]

Martins, J., Mamede, H. & Correia, J. (2022). Risk Compliance and Master Data Management in Banking: A novel BCBS 239 compliance action- plan proposal, Heliyon, [e-journal] vol. 8, no. 6, p.e09627, Available Online: https://www.sciencedirect.com/science/article/pii/S240584402200915X [Accessed 18 May 2023]

Morde, V. (2019). XGBoost Algorithm: Long May She Reign!, Medium, Available Online: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d [Accessed 17 May 2023]

Moreira, M., Junior, C., Silva, D., Junior, M., Costa, I., Gomes, C. & Santos, M. (2022). Exploratory Analysis and Implementation of Machine Learning Techniques for Predictive Assessment of Fraud in Banking Systems, Procedia Computer Science, [e-journal] vol. 214, pp.117–124, Available Online: https://www.sciencedirect.com/science/article/pii/S1877050922018634 [Accessed 18 May 2023]

Moulton, A. (2011). An Investigation of the Determinants & Forecast Performance of Bank Profits: The case of Jamaican banks, Kingston: Bank of Jamaica, Available Online: https://boj.org.jm/uploads/pdf/papers_pamphlets/papers_pamphlets_An_Investigation_of_the_Determinants_&_Forecast_Performance_of_Bank_Profits__The_Case_of_Jamaican_Banks.pdf [Accessed 17 May 2023]

Mundfrom, D., Perrett, J., Schaffer, J., Piccone, A. & Roozeboom, M. (2006). Bonferroni Adjustments in Tests for Regression Coefficients, Multiple Linear Regression Viewpoints, [e-journal] vol. 32, no. 1, pp.1–6, Available Online: https://www.academia.edu/80122835/Bonferroni_Adjustments_in_Tests_for_Regression_Coefficients [Accessed 17 May 2023]

National Institute of Economic Research. (2023). Household Indicators, Monthly, Available Online: http://statistik.konj.se/PxWeb/pxweb/en/KonjBar/KonjBar__hushall/Indikatorhus.px/ [Accessed 10 May 2023]

Nembrini, S., König, I. & Wright, M. (2018). The Revival of the Gini Importance?, Bioinformatics, [e-journal] vol. 34, no. 21, pp.3711–3718, Available Online: https://repository.publisso.de/resource/frl:6411640/data [Accessed 17 May 2023]

Pazhoohesh, M., Javadi, M., Gheisari, M., Aziz, S. & Villa, R. (2021). Dealing with Missing Data in the Smart Buildings Using Innovative Imputation Techniques, in IEEE Xplore, IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, 2021, IEEE, pp.1–7, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9612650 [Accessed 18 May 2023]

Rahman, N. & Iverson, S. (2015). Big Data Business Intelligence in Bank Risk Analysis, International Journal of Business Intelligence Research, [e-journal] vol. 6, no. 2, pp.55–77, Available Online: https://www.igi-global.com/article/big-data-business-intelligence-in-bank-risk-analysis/149262 [Accessed 18 May 2023]

Rao, S. & Lakshmanan, L. (2022). Map-Reduce Based Ensemble Intrusion Detection System with Security in Big Data, Procedia Computer Science, [e-journal] vol. 215, pp.888–896, Available Online: https://www.sciencedirect.com/science/article/pii/S1877050922021627 [Accessed 18 May 2023]

Rogers, S. & Girolami, M. (2016). A First Course in Machine Learning, Second Edition, 2nd edn, London: CRC Press, pp.196–199

Rogić, S., Kašćelan, L., Kašćelan, V. & Đurišić, V. (2022). Automatic Customer Targeting: A data mining solution to the problem of asymmetric profitability distribution, Information Technology and Management, [e-journal] vol. 23, no. 4, pp.315–333, Available Online: https://link.springer.com/article/10.1007/s10799-021-00353-5?utm_source=getftr&utm_medium=getftr&utm_campaign=getftr_pilot [Accessed 17 May 2023]

Sadreddin, A. & Sadaoui, S. (2022). Training and Testing Cascades for Imbalanced Data Classification, in IEEE Xplore, 2022 IEEE Symposium Series on Computational Intelligence (SSCI), Singapore, 1 December 2022, IEEE, pp.261–268, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10022190 [Accessed 18 May 2023]

Sahoo, A. & Ghose, D. (2022). Imputation of Missing Precipitation Data Using KNN, SOM, RF, and FNN, Soft Computing, vol. 26, pp.5919–5936, Available Online: https://link.springer.com/article/10.1007/s00500-022-07029-4 [Accessed 18 May 2023]

Salehnia, N., Salehnia, N., Torshizi, A. & Kolsoumi, S. (2020). Rainfed Wheat (Triticum Aestivum L.) Yield Prediction Using Economical, Meteorological, and Drought Indicators through Pooled Panel Data and Statistical Downscaling, Ecological Indicators, [e-journal] vol. 111, p.105991, Available Online: https://www.sciencedirect.com/science/article/pii/S1470160X19309860 [Accessed 19 May 2023]

Schreiber-Gregory, D. & Bader, K. (2018). (PDF) Logistic and Linear Regression Assumptions: Violation recognition and control, in SESUG, 2018, Available Online: https://www.researchgate.net/publication/341354759_Logistic_and_Linear_Regression_Assumptions_Violation_Recognition_and_Control [Accessed 17 May 2023]

Shah, S., Telrandhe, M., Waghmode, P. & Ghane, S. (2022). Imputing Missing Values for Dataset of Used Cars, in IEEE Xplore, 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, 2022, IEEE, pp.1–5, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9908600&tag=1 [Accessed 18 May 2023]

Sharma, H. & Gosain, A. (2023). Oversampling Methods to Handle the Class Imbalance Problem: A review, in Soft Computing and Its Engineering Applications, Vol. 1788, 4th International Conference IcSoftComp 2022, Changa, 2023, Springer, pp.96-110, Available Online: https://link.springer.com/content/pdf/10.1007/978-3-031-27609-5.pdf?pdf=button [Accessed 17 May 2023]

Shrestha, N. (2020). Detecting Multicollinearity in Regression Analysis, American Journal of Applied Mathematics and Statistics, [e-journal] vol. 8, no. 2, pp.39–42, Available Online: http://pubs.sciepub.com/ajams/8/2/1/index.html [Accessed 19 May 2023]

Sivasankar, E. & Vijaya, J. (2017). Customer Segmentation by Various Clustering Approaches and Building an Effective Hybrid Learning System on Churn Prediction Dataset, Advances in Intelligent Systems and Computing, [e-journal] vol. 556, pp.181–191, Available Online: https://link.springer.com/chapter/10.1007/978-981-10-3874-7_18 [Accessed 17 May 2023]

Srivastava, U. & Gopalkrishnan, S. (2015). Impact of Big Data Analytics on Banking Sector: Learning for Indian banks, Procedia Computer Science, [e-journal] vol. 50, pp.643–652,

Available Online: https://www.sciencedirect.com/science/article/pii/S1877050915005992 [Accessed 23 May 2023]

Statistics Sweden. (2023a). Lending Rates to Households and Non-Financial Corporations, Breakdown by Fixation Periods. Month 1987M03 - 2023M03, Available Online: https://www.statistikdatabasen.scb.se/pxweb/en/ssd/START__FM__FM5001__FM5001C/RantaT01N/ [Accessed 10 May 2023]

Statistics Sweden. (2023b). Retail Trade, Sales Volume (Seasonally Adjusted): Turnover in Retail Trade, March 2023, Available Online: https://www.scb.se/en/finding-statistics/statistics-by-subject-area/trade-in-goods-and-services/domestic-trade/turnover-in-the-service-sector/pong/tables-and-graphs/retail-trade/retail-trade-sales-volume-seasonally-adjusted.-turnover-in-retail-trade-march-2023/ [Accessed 10 May 2023]

Stekhoven, D. & Bühlmann, P. (2012). MissForest: Non-parametric missing value imputation for mixed-type data, Bioinformatics, [e-journal] vol. 28, no. 1, pp.112–118, Available Online: https://academic.oup.com/bioinformatics/article/28/1/112/219101 [Accessed 18 May 2023]

Sun, N., Morris, J., Xu, J., Zhu, X. & Xie, M. (2014). ICARE: A framework for big data-based banking customer analytics, IBM Journal of Research and Development, [e-journal] vol. 58, no. 5/6, pp.4:1–4:9, Available Online: https://ieeexplore.ieee.org/document/6964895?denied= [Accessed 18 May 2023]

Sveriges Riksbank. (2023). Search Interest & Exchange Rates, Available Online: https://www.riksbank.se/en-gb/statistics/search-interest--exchange-rates/?g151-SEKKIX92=on&from=01%2F10%2F2018&to=30%2F12%2F2022&f=Quarter&c=cAverage&s=Dot [Accessed 10 May 2023]

Tang, F. & Ishwaran, H. (2017). Random Forest Missing Data Algorithms, Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 10, no. 6, pp.363–377, Available Online: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5796790/ [Accessed 18 May 2023]

Waljee, A., Mukherjee, A., Singal, A., Zhang, Y., Warren, J., Balis, U., Marrero, J., Zhu, J. & Higgins, P. (2013). Comparison of Imputation Methods for Missing Laboratory Data in Medicine, BMJ Open, vol. 3, no. 8, p.e002847, Available Online: https://bmjopen.bmj.com/content/3/8/e002847 [Accessed 18 May 2023]

Wang, Z. & Liu, Q. (2023). Imbalanced Data Classification Method Based on LSSASMOTE, IEEE Access, [e-journal] vol. 11, pp.32252–32260, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10082915 [Accessed 18 May 2023]

Wißmann, M., Toutenburg, H. & Shalabh. (2007). Role of Categorical Variables in Multicollinearity in the Linear Regression Model, Journal of Applied Statistical Science, Munich: University of Munich, Available Online: https://www.researchgate.net/profile/Shalabh-Shalabh/publication/33028294_Role_of_Categorical_Variables_in_Multicollinearity_in_the_Linear_Regression_Model/links/00b495294c45f5f59d000000/Role-of-Categorical-Variables-in-Multicollinearity-in-the-Linear-Regression-Model.pdf?origin=publication_detail [Accessed 17 May 2023]

Xu, Y., Ye, H., Zhang, N. & Du, G. (2022). Leveraging Autoencoder and Focal Loss for Imbalanced Data Classification, in IEEE Xplore, 2022 12th International Conference on Information Technology in Medicine and Education (ITME)V, Xiamen, 1 November 2022, IEEE, pp.502–506, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10086367 [Accessed 18 May 2023]

Xu, Y., Yu, Z., Chen, C. L. P. & Liu, Z. (2023). Adaptive Subspace Optimization Ensem-

ble Method for High-Dimensional Imbalanced Data Classification, IEEE Transactions on Neural Networks and Learning Systems, [e-journal] vol. 34, no. 5, pp.2284–2297, Available Online: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9526880 [Accessed 18 May 2023]

Zeng, M., Zou, B., Wei, F., Liu, X. & Wang, L. (2016). Effective Prediction of Three Common Diseases by Combining SMOTE with Tomek Links Technique for Imbalanced Medical Data, in 2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS), 2016, Chongqing: IEEE, pp.225–228, Available Online: https://ieeexplore.ieee.org/document/7563084 [Accessed 17 May 2023]

9 Appendix

9.1 Random search for random forest regression

9.1.1 Parameters

- n_estimators: number of decision trees in the forest (values: 40, 50, 60, 70, 80).
- criterion: loss function to evaluate splits (values: squared error, absolute error).
- max_features: maximum number of variables available for each split (values: log2 of the total number of inputs, square root of the total number of inputs).
- max_depth: maximum number of levels in each decision tree (values: 3, 5, 10, 15, 20).
- min_samples_split: minimum number of observations required in a node in order to split it, calculated as a percentage of all observations (values: 0.05%, 0.1%, 0.25%, 0.5%).

9.1.2 Results

- n_estimators: 60 for recurring customers, 80 for new customers.
- criterion: squared error for both models.
- max_features: square root for recurring customers, log2 for new customers.
- max_depth: 15 for recurring customers, 20 for new customers.
- min_samples_split: 0.05% for recurring customers, 0.1% for new customers.

9.2 Random search for XGBoost regression

9.2.1 Parameters

- n_estimators: number of decision trees boosted (values: 70, 100, 200).
- min_child_weight: minimum weight in a leaf node in order to make a split (values: 1, 2, 3).
- gamma: minimum reduction in loss necessary for a split to happen (values: 0, 0.1, 0.2).
- subsample: portion of obsevations used by each tree when training (values: 60%, 70%, 80%, 90%, 100%).
- colsample_bytree: portion of features used when learning each tree (values: 60%, 70%, 80%, 90%, 100%).
- max_depth: maximum number of levels in each decision tree (values: 2, 3, 4, 5, 6, 7).
- eta: boosting learning rate (values: 0.3, 0.4, 0.5).

9.2.2 Results for all customers

- n_estimators: 200 for recurring customers, 70 for new customers.
- min_child_weight: 2 for recurring customers, 3 for new customers.
- gamma: 0.1 for recurring customers, 0.2 for new customers.
- subsample: 90% for recurring customers, 100% (no subsampling) for new customers.
- colsample_bytree: 80% for both models.
- max_depth: 6 for both models.
- eta: 0.3 for both models.

9.3 Random search for XGBoost regression by cluster

9.3.1 Parameters

- n_estimators: number of decision trees boosted (values: 70, 100, 200).
- min_child_weight: minimum weight in a leaf node in order to make a split (values: 1, 2, 3).
- gamma: minimum reduction in loss necessary for a split to happen (values: 0, 0.1, 0.2).
- subsample: portion of obsevations used by each tree when training (values: 60%, 70%, 80%, 90%, 100%).
- colsample_bytree: portion of features used when learning each tree (values: 60%, 70%, 80%, 90%, 100%).
- max_depth: maximum number of levels in each decision tree (values: 2, 3, 4, 5, 6, 7).
- eta: boosting learning rate (values: 0.3, 0.4, 0.5).

9.3.2 Results for customers with low revenues

- n_estimators: 200 for both models.
- min_child_weight: 1 for recurring customers, 3 for new customers.
- gamma: 0 for both models.
- subsample: 90% for recurring customers, 80% for new customers.
- colsample_bytree: 100% (no subsampling) for both models.
- max_depth: 7 for recurring customers, 5 for new customers.
- eta: 0.3 for both models.

9.3.3 Results for customers with medium revenues

- n_estimators: 200 for recurring customers, 100 for new customers.
- min_child_weight: 3 for both models.
- gamma: 0.1 for recurring customers, 0 for new customers.
- subsample: 90% for recurring customers, 100% (no subsampling) for new customers.
- colsample_bytree: 70% for recurring customers, 100% (no subsampling) for new customers.
- max_depth: 7 for recurring customers, 3 for new customers.
- eta: 0.3 for both models.

9.3.4 Results for customers with high revenues

- n_estimators: 200 for recurring customers, 70 for new customers.
- min_child_weight: 2 for recurring customers, 3 for new customers.
- gamma: 0.2 for recurring customers, 0.1 for new customers.
- subsample: 100% (no subsampling) for recurring customers, 90% for new customers.
- colsample_bytree: 80% for both models.
- max_depth: 7 for recurring customers, 2 for new customers.
- eta: 0.3 for both models.

9.4 Random search for neural network

9.4.1 Parameters

- optimiser: algorithm to modify the attributes of the network (values: adam, rmsprop).
- learning_rate: parameter that determines the step size the algorithm takes at each iteration when updating the weights of the model (values: 10 equally spaced values in a log scale ranging from 1×10^{-6} to 1×10^{-3}).

9.4.2 Results

- optimiser: Adam for both models.
- learning_rate: 0.0001 for recurring customers, 0.00046 for new customers.

9.5 Random search for support vector regression

9.5.1 Parameters

- C: regularisation parameter, as described in Subsection 4.4 (values: 10 equally spaced values in a log scale ranging from 1×10^{-5} to 1×10^{5}).
- gamma: coefficient in the kernel function (values: 10 equally spaced values in a log scale ranging from 1×10^{-5} to 1×10^{5}).

9.5.2 Results

- C: 7743 for recurring customers, 599 for new customers.
- gamma: 0.00001 for recurring customers, 0.00013 for new customers.

9.6 Random search for random forest classification

9.6.1 Parameters

- n_estimators: number of decision trees in the forest (values: 40, 50, 60, 70, 80).
- criterion: loss function to evaluate splits (values: gini, entropy).
- max_features: maximum number of variables available for each split (values: log2 of the total number of inputs, square root of the total number of inputs).
- max_depth: maximum number of levels in each decision tree (values: 3, 5, 10, 15, 20).
- min_samples_split: minimum number of observations required in a node in order to split it, calculated as a percentage of all observations (values: 0.05%, 0.1%, 0.25%, 0.5%).

9.6.2 Results

- n_estimators: 80 for both models.
- criterion: entropy for both models.
- max_features: log2 for recurring customers, square root for new customers.
- max_depth: 20 for recurring customers, 10 for new customers.
- min_samples_split: 0.05% for both models.

9.7 Random search for XGBoost classification

9.7.1 Parameters

- n_estimators: number of decision trees boosted (values: 70, 100, 200).
- min_child_weight: minimum weight in a leaf node in order to make a split (values: 1, 2, 3).
- gamma: minimum reduction in loss necessary for a split to happen (values: 0, 0.1, 0.2).
- subsample: portion of obsevations used by each tree when training (values: 60%, 70%, 80%, 90%, 100%).
- colsample_bytree: portion of features used when learning each tree (values: 60%, 70%, 80%, 90%, 100%).
- max_depth: maximum number of levels in each decision tree (values: 2, 3, 4, 5, 6, 7).
- eta: boosting learning rate (values: 0.3, 0.4, 0.5).

9.7.2 Results

- n_estimators: 200 for both models.
- min_child_weight: 3 for recurring customers, 2 for new customers.
- gamma: 0.2 for both models.
- subsample: 90% for recurring customers, 100% (no subsampling) for new customers.
- colsample_bytree: 60% for recurring customers, 90% for new customers.
- max_depth: 6 for recurring customers, 5 for new customers.
- eta: 0.3 for recurring customers, 0.5 for new customers.

9.8 Linear regression

9.8.1 Formula for recurring customers

$$\begin{aligned}\text{total revenue}_{it} = \; & \beta_0 + \beta_1\,\text{age}_{it} + \beta_2\,\text{gender}_{it} + \beta_3\,\text{longevity}_{it} + \\ & \beta_4\,\text{number of accounts}_{it} + \beta_5\,\text{insurance}_{it} + \beta_6\,\text{loan extensions}_{it} + \\ & \beta_7\,\text{co-applicant}_{it} + \beta_8\,\text{invoice accounts}_{it} + \beta_9\,\text{buy-now-pay-later}_{it} + \\ & \beta_{10}\,\text{credit cards A}_{it} + \beta_{11}\,\text{credit cards B}_{it} + \beta_{12}\,\text{credit cards C}_{it} + \\ & \beta_{13}\,\text{consumer loans}_{it} + \beta_{14}\,\text{default probability}_{it} + \beta_{15}\,\text{minimum limit}_{it} + \\ & \beta_{16}\,\text{maximum limit}_{it} + \beta_{17}\,\text{minimum balance}_{it} + \\ & \beta_{18}\,\text{maximum balance}_{it} + \beta_{19}\,\text{late payment}_{it} + \\ & \beta_{20}\,\text{number of transactions}_{it} + \beta_{21}\,\text{GDP growth}_t + \beta_{22}\,\text{inflation}_t + \\ & \beta_{23}\,\text{interest rate}_t + \beta_{24}\,\text{exchange rate}_t + \beta_{25}\,\text{unemployment rate}_t + \\ & \beta_{26}\,\text{consumer confidence index}_t + \beta_{27}\,\text{consumption of durables}_t + \\ & \beta_{28}\,\text{quarter two}_t + \beta_{29}\,\text{quarter three}_t + \beta_{30}\,\text{quarter four}_t\end{aligned} \tag{9.1}$$

where i represents a customer and t a quarter.

9.8.2 Formula for new customers

$$\begin{aligned}\text{total revenue}_{it} = \; & \beta_0 + \beta_1\,\text{age}_{it} + \beta_2\,\text{gender}_{it} + \beta_3\,\text{insurance}_{it} + \beta_4\,\text{co-applicant}_{it} + \\ & \beta_5\,\text{invoice accounts}_{it} + \beta_6\,\text{buy-now-pay-later}_{it} + \beta_7\,\text{credit cards A}_{it} + \\ & \beta_8\,\text{credit cards B}_{it} + \beta_9\,\text{credit cards C}_{it} + \beta_{10}\,\text{consumer loans}_{it} + \\ & \beta_{11}\,\text{default probability}_{it} + \beta_{12}\,\text{GDP growth}_t + \beta_{13}\,\text{inflation}_t + \\ & \beta_{14}\,\text{interest rate}_t + \beta_{15}\,\text{exchange rate}_t + \beta_{16}\,\text{unemployment rate}_t + \\ & \beta_{17}\,\text{consumer confidence index}_t + \beta_{18}\,\text{consumption of durables}_t + \\ & \beta_{19}\,\text{quarter two}_t + \beta_{20}\,\text{quarter three}_t + \beta_{21}\,\text{quarter four}_t\end{aligned} \tag{9.2}$$

where i represents a customer and t a quarter.

9.8.3 Results

Variable	VIF for first regression	VIF for final regression
Age	14.66	-
Gender	2.23	1.93
Longevity	4.20	2.84
Number of accounts	1 927 846.84	-
Insurance	1.44	1.40
Loan extensions	2.16	2.15
Co-applicant	1.04	1.04
Invoice accounts	400 609.77	1.37
Buy-now-pay-later	286 625.84	1.21
Credit cards A	82 596.94	1.62
Credit cards B	586 349.78	2.11
Credit cards C	10 187.77	1.08
Consumer loans	22 779.71	3.39
Default probability	1.21	1.17
Minimum limit	7.83	-
Maximum limit	32.77	-
Minimum balance	5.47	1.56
Maximum balance	27.57	2.77
Late payment	1.18	1.16
Number of transactions	1.59	1.52
GDP growth	21.36	-
Inflation	23.54	1.93
Interest rate	711.52	-
Exchange rate	1 264.96	-
Unemployment rate	255.41	-
Consumer confidence index	1 114.58	-
Consumption of durables	2 121.08	-
Quarter two	17.06	-
Quarter three	9.41	1.43
Quarter four	50.73	-

Table 10: Variance inflation factor (VIF) for recurring customers after the first regression and for the final model

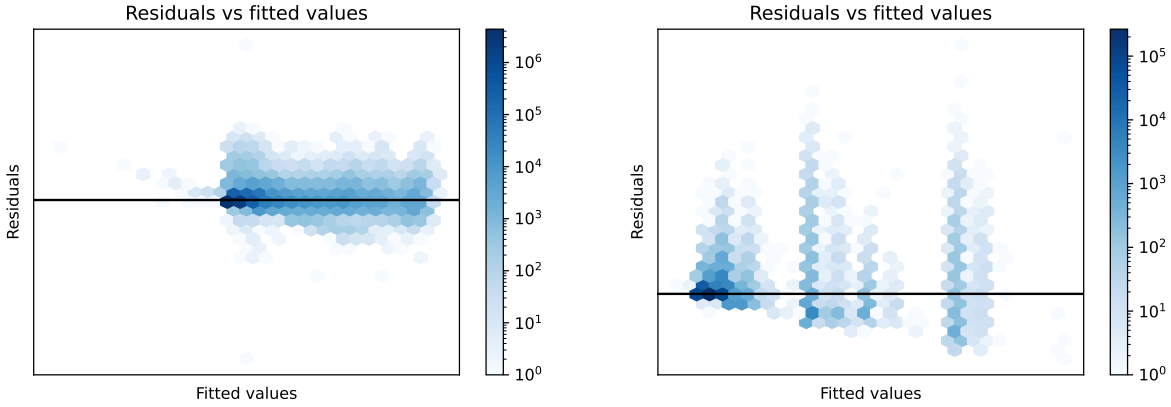
Variable	VIF for first regression	VIF for final regression
Age	10.27	-
Gender	2.10	1.99
Insurance	1.23	1.23
Co-applicant	1.02	1.02
Invoice accounts	7.80	1.45

Buy-now-pay-later	4.55	1.33
Credit cards A	1.27	1.03
Credit cards B	9.12	1.53
Credit cards C	1.12	1.02
Consumer loans	1.66	1.29
Default probability	1.20	1.18
GDP growth	21.21	-
Inflation	16.14	-
Interest rate	538.81	-
Exchange rate	1 059.35	-
Unemployment rate	236.62	-
Consumer confidence index	949.95	-
Consumption of durables	1 944.77	-
Quarter two	13.40	-
Quarter three	8.45	-
Quarter four	55.49	1.30

Table 11: Variance inflation factor (VIF) for new customers after the first regression and for the final model

Variable	**Recurring customers**	**New customers**
Gender	-6.2×10^{-14}	-4.8×10^{-15}
Longevity	-7.6×10^{-14}	-
Insurance	-3.3×10^{-13}	7.3×10^{-14}
Loan extensions	-7.5×10^{-13}	-
Co-applicant	-2.3×10^{-13}	3.9×10^{-15}
Invoice accounts	1.1×10^{-13}	-5.5×10^{-15}
Buy-now-pay-later	-2.7×10^{-14}	-2.0×10^{-14}
Credit cards A	-2.5×10^{-13}	6.6×10^{-15}
Credit cards B	8.8×10^{-14}	8.8×10^{-15}
Credit cards C	-7.3×10^{-14}	8.1×10^{-16}
Consumer loans	-1.1×10^{-12}	3.6×10^{-14}
Default probability	-1.0×10^{-13}	4.2×10^{-15}
Minimum balance	-8.1×10^{-13}	-
Maximum balance	-1.5×10^{-12}	-
Late payment	-4.6×10^{-14}	-
Number of transactions	-3.2×10^{-13}	-
Inflation	-5.1×10^{-15}	-
Quarter three	-2.5×10^{-15}	-
Quarter four	-	-1.6×10^{-15}

Table 12: Pearson correlation coefficients between independent variables and error terms

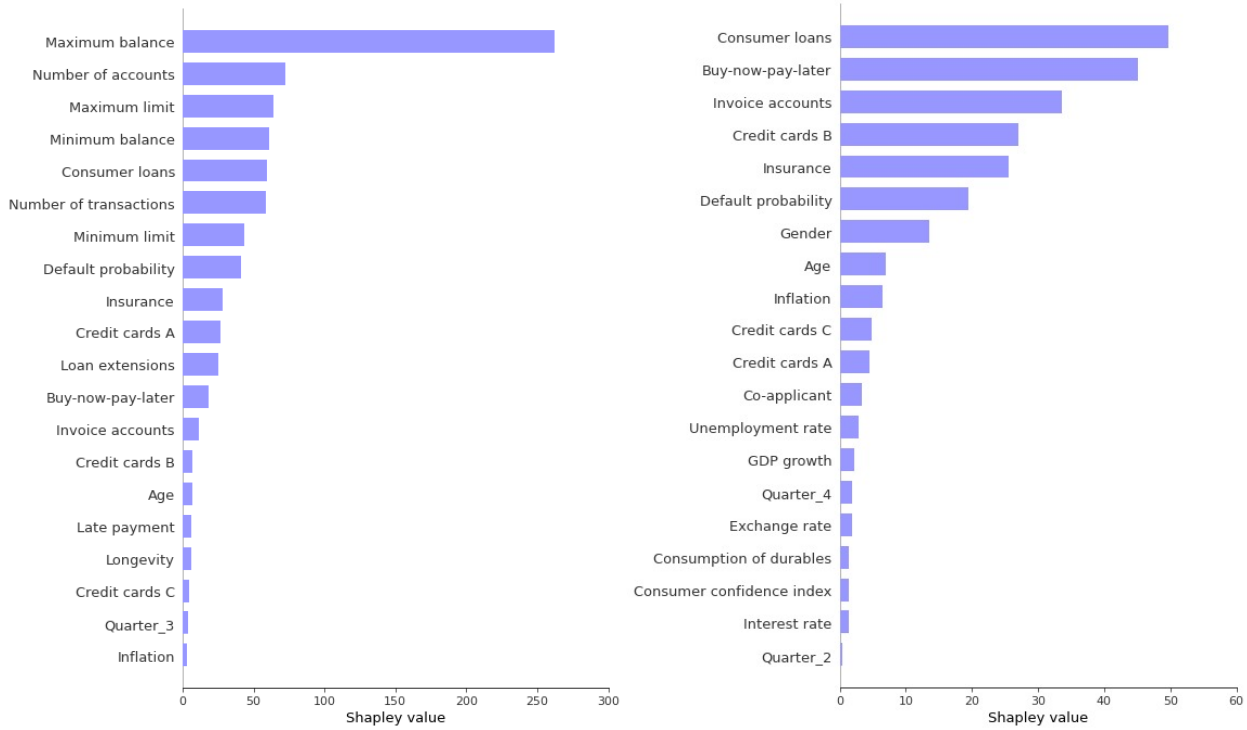


(a) Recurring customers

(b) New customers

Figure 7: Residuals vs fitted values (horizontal black line represents residuals equal to zero)

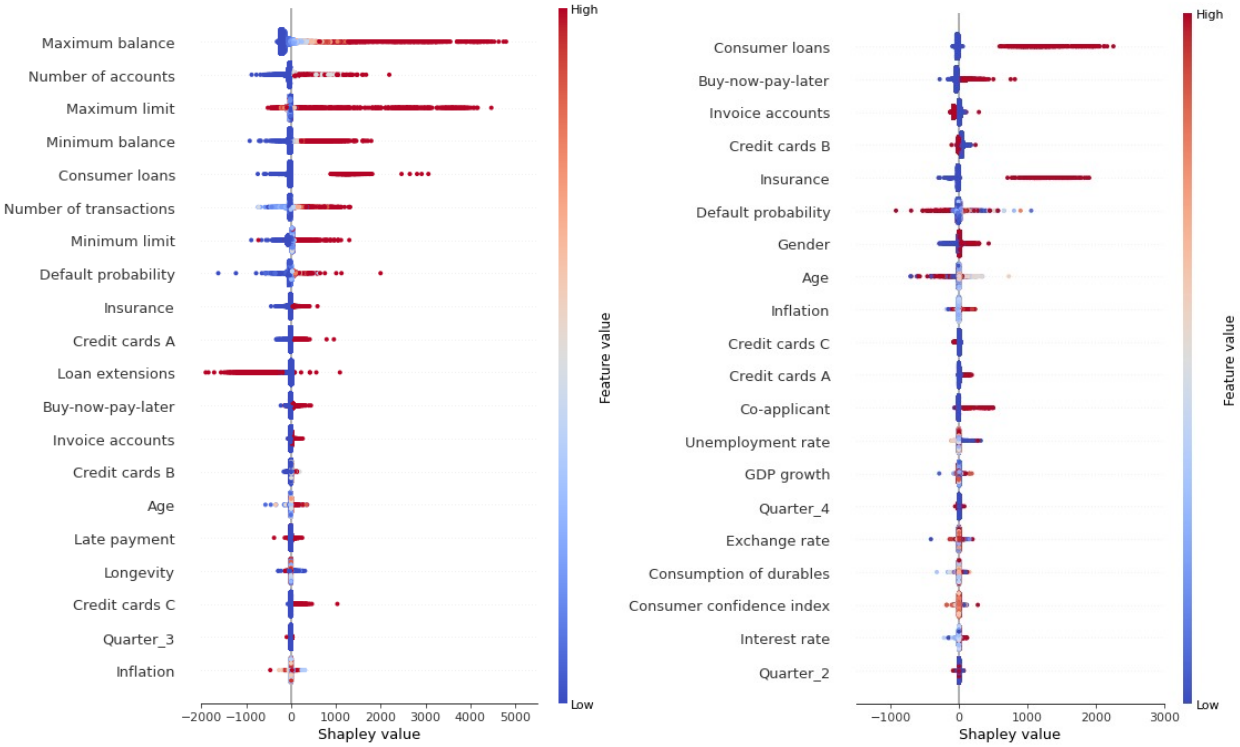
9.9 XGBoost: Shapley values



(a) Recurring customers

(b) New customers

Figure 8: Average absolute Shapley values of the 20 most important variables in XGBoost



(a) Recurring customers

(b) New customers

Figure 9: Shapley values of the 20 most important variables in XGBoost

9.10 Scatter plots, histograms and correlation coefficients for the data

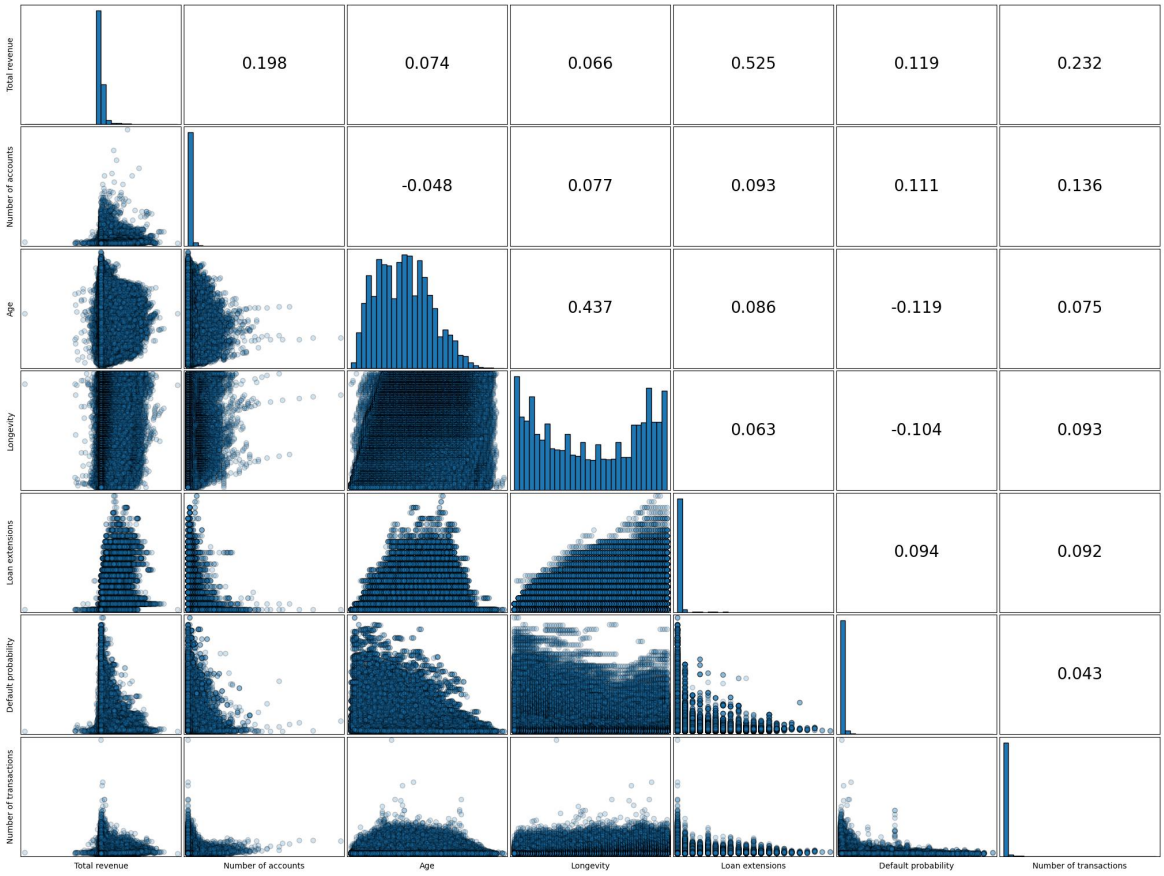


Figure 10: Scatter plots, correlation coefficients and histograms of continuous variables for recurring customers

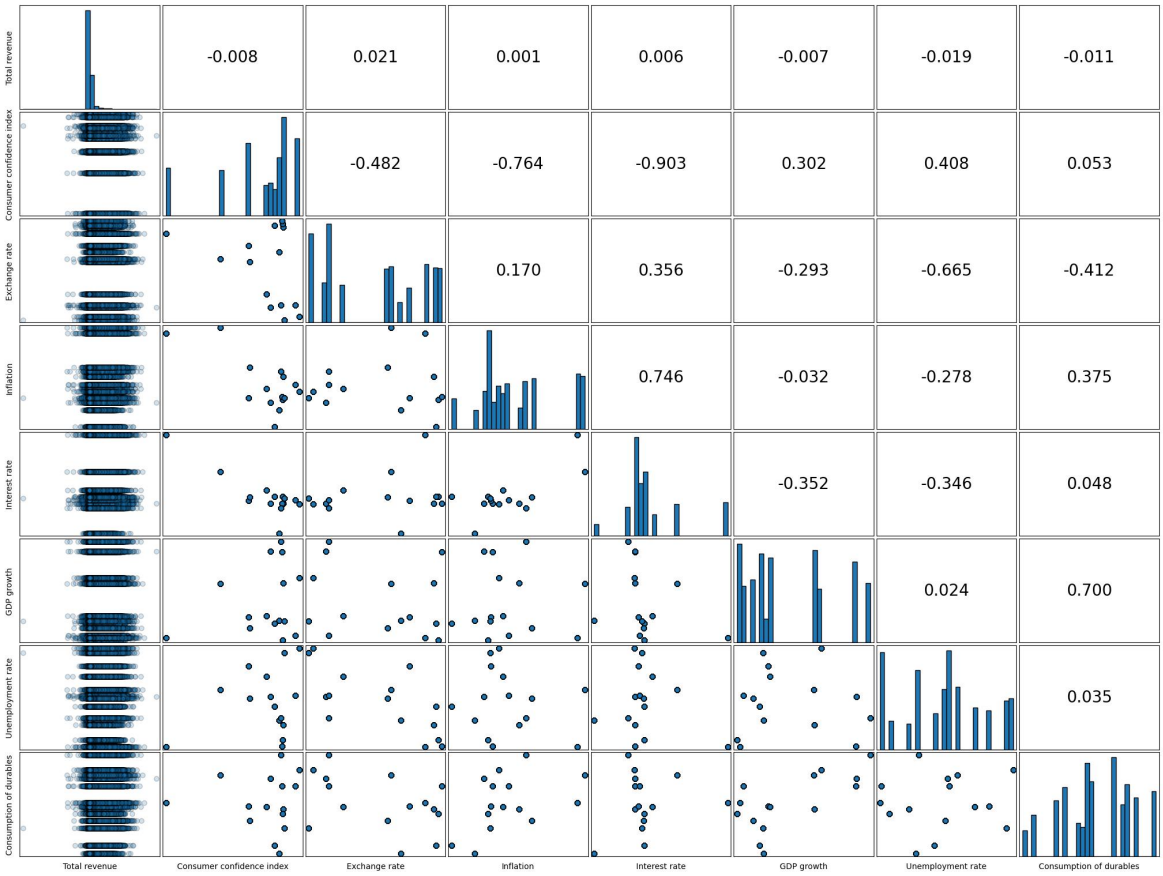


Figure 11: Scatter plots, correlation coefficients and histograms of continuous variables for recurring customers

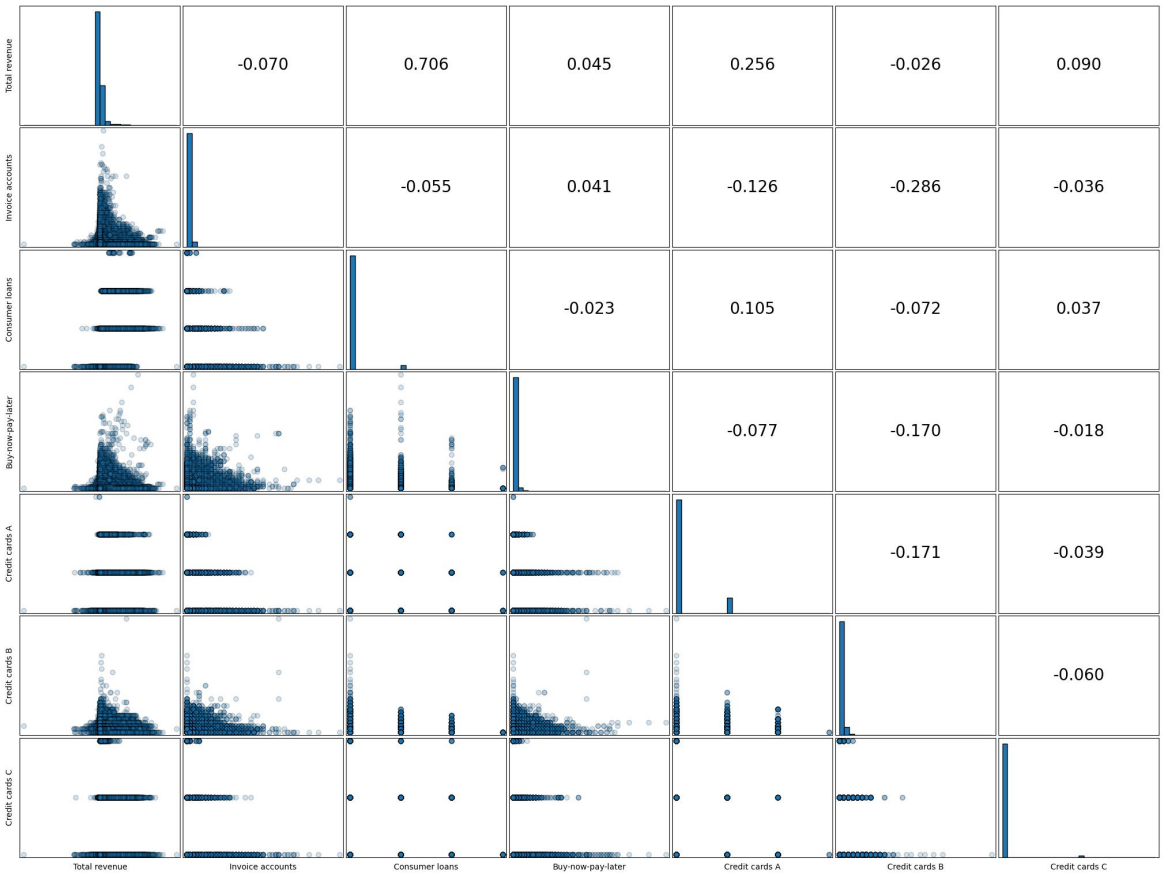


Figure 12: Scatter plots, correlation coefficients and histograms of continuous variables for recurring customers

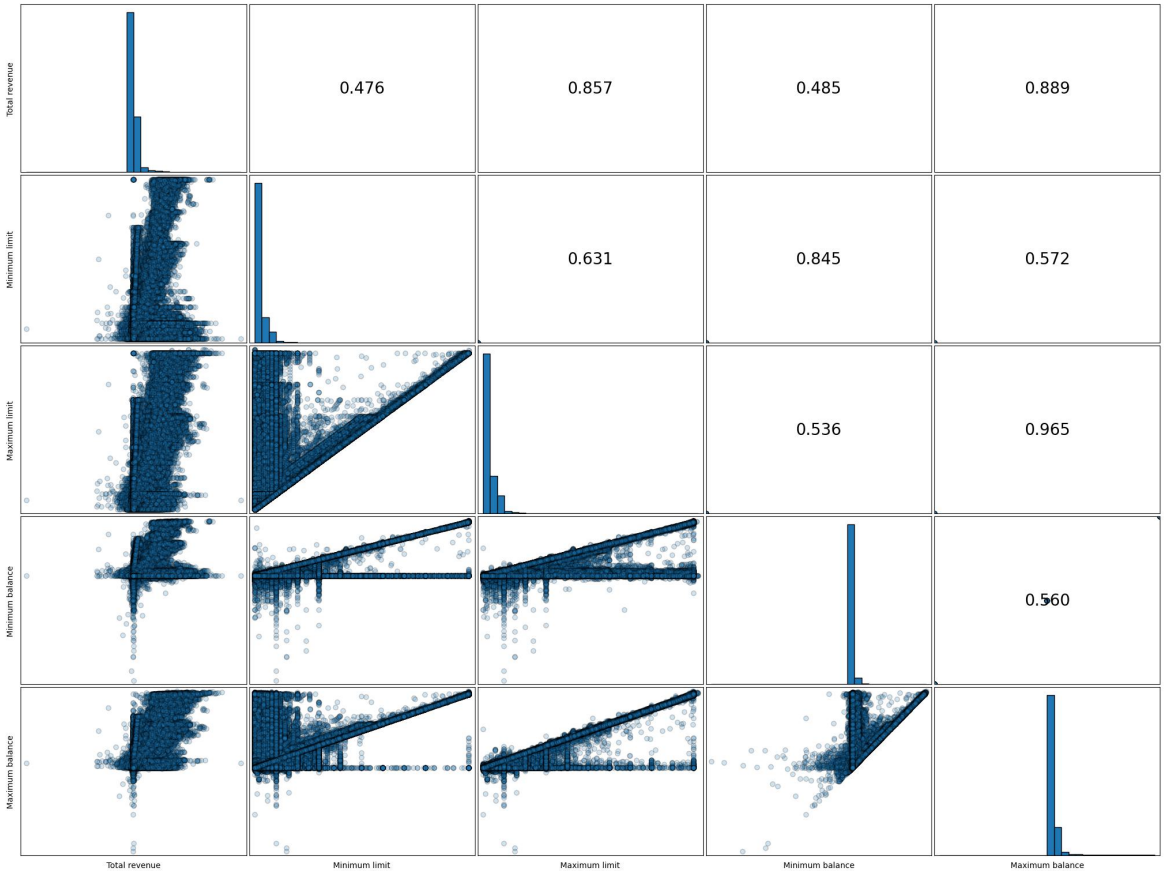


Figure 13: Scatter plots, correlation coefficients and histograms of continuous variables for recurring customers

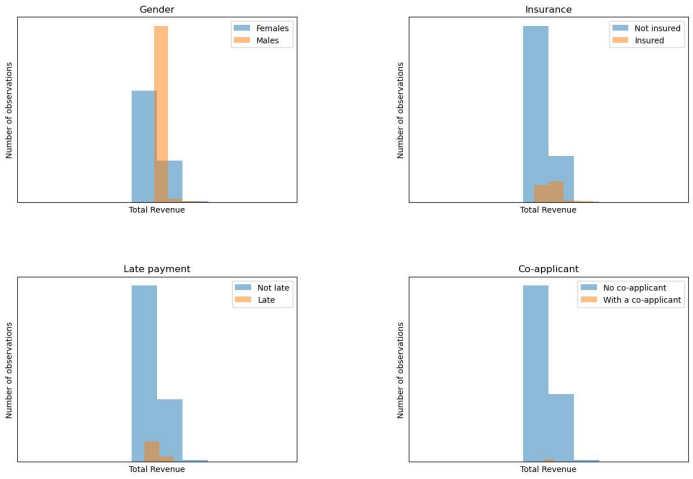


Figure 14: Histograms of indicator variables for recurring customers

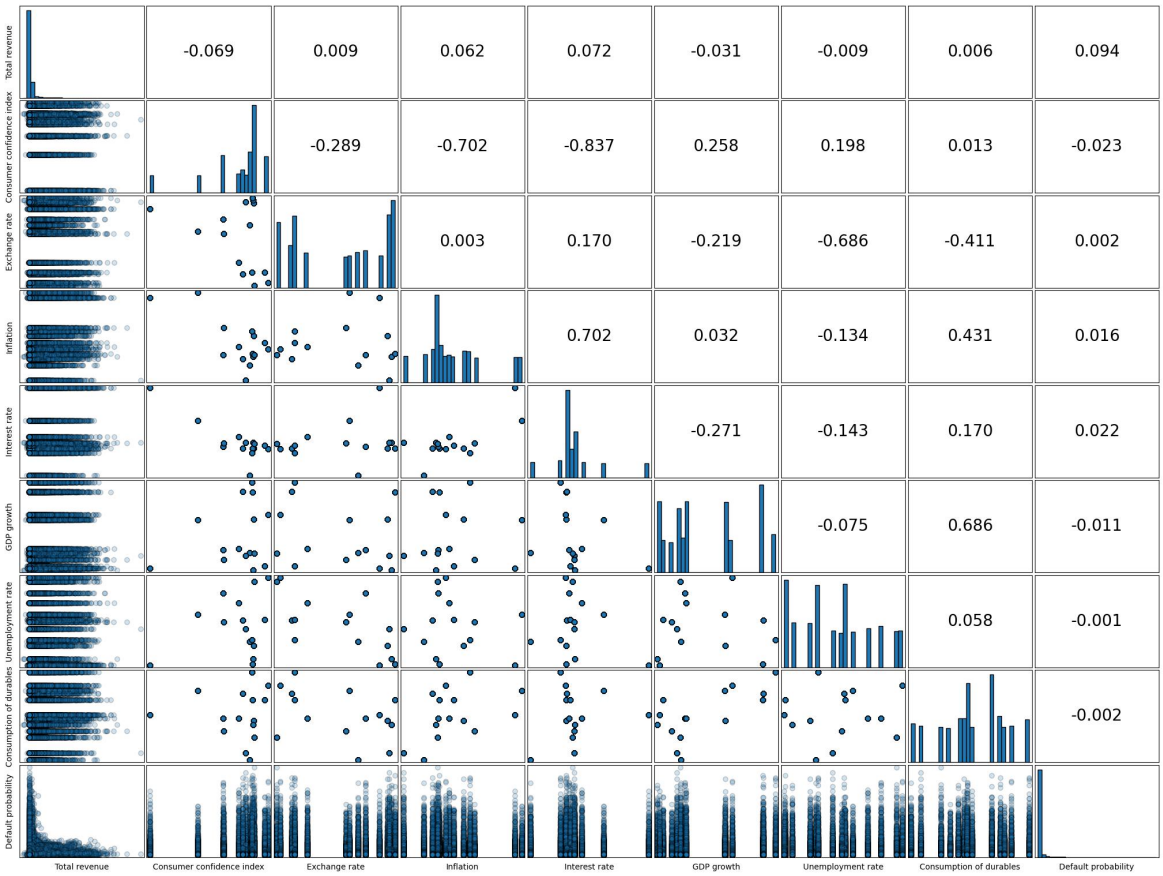


Figure 15: Scatter plots, correlation coefficients and histograms of continuous variables for new customers

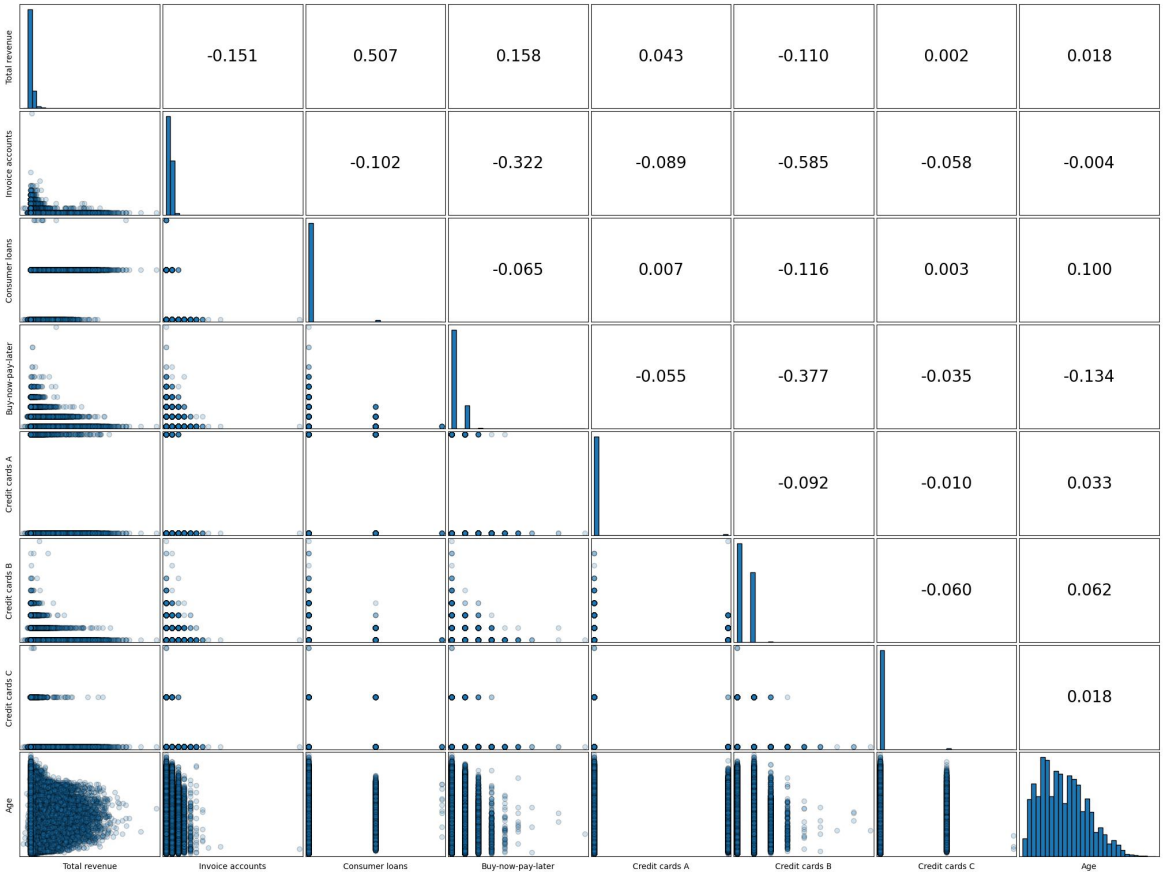


Figure 16: Scatter plots, correlation coefficients and histograms of continuous variables for new customers

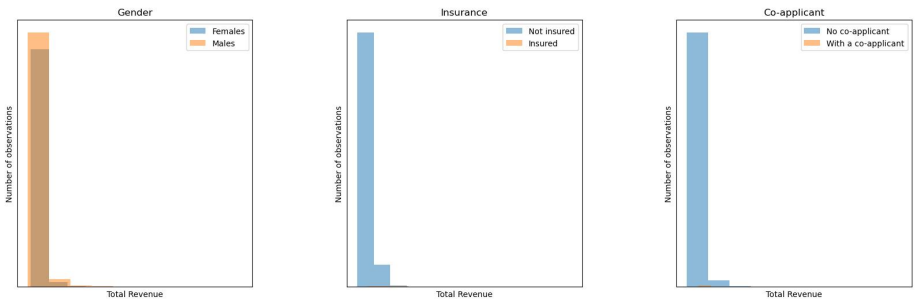


Figure 17: Histograms of indicator variables for new customers