Interdisciplinary Postgraduate Program

Business Mathematics

Athens University of Economics – National Kapodistrian University of Athens

‘CUSTOMER DEFAULT PREDICTION-DATA CORRELATIONS’

Master’s Thesis

Ioannis Pararas

Supervisors: Paraskevas Vassalos

Athens, November 2022

List of Tables

List of Figures

Table of Contents

ABSTRACT

For customer-oriented businesses predicting customer default is crucial. In the past few years predicting when a customer might default is one of the critical problems companies face especially in the energy industry. This is significant since this way a company can estimate a customers’ risk level from the analysis of the historical data of its customers and aided by predictive machine learning models it can identify the customers that are at risk of defaulting on their payments.

There are several models that can predict the probability of a customer defaulting. The approach to this problem is handled in a similar fashion of such problems as in the banking industry through credit risk models. After some initial data pre-processing, we move through EDA and data cleaning to the implementation of three different models to achieve this goal, namely Logistic regression, Random Forest classifier and Extreme Gradient Boosting. The metrics used to evaluate model performance was mainly F1 score as well as Precision-Recall Curve (PR-Curve). All models achieved satisfying results with the least performing model achieving an PR-AUC of 90% and Random Forest being the overall best performer with an AUC of 93%.

INTRODUCTION

1.1 Introduction

According to the legislation that has been in effect since 2011 in Greece, household and professional electricity consumers whose contracts have been terminated and are unable to find a new electricity provider are supplied electricity from the so-called universal service. In most countries there are certain time frames in which the consumer has the option of remaining in said service and after said time frame has been exceeded, the electricity supply of the consumers is halted. In Greece those time frames are not as clear. This allows certain consumers to strategically enter this service and stay there for an indefinite period of time. According to data presented from a member of the Energy Regulatory Authority N.Karakatsanis , the number of consumers who have been subject to the universal service in 2020 is up to 135 thousand (Floudopoulos, 2020) .The problem is emphasized more by the fact that the universal service has obtained up to 190 thousand consumers in 2022, similar to a mid-sized provider. (Liaggou, 2022)

Originally the universal service was primarily assigned to PPC (Public Power Corporation) up until 2019 where the RAE (Regulatory Authority for Energy) dictated that these so called ‘bad debtors’ would be assigned to the top 5 power providers in Greece, namely PPC, Protergia, Heron, Elpedison and NRG. (Liaggou H. 2022) .

Moreover, the circumstances from the beginning of 2020 with the arrival of the pandemic started bringing more risk in this industry. Following the appearance of covid-19, household electricity bills have seen significant increases as a result of the lockdown measure taken to halt the spread of the virus. According to IEA reports residential electricity demand in certain European economies has seen an increase of up to 40% during the months of March-April of 2020 as opposed to the same period in 2019 (IEA ,2020). Therefore, it is apparent that the risk originating from the aforementioned bad debtors and other types of customers with questionable financial stability, demands a proactive approach in order to foresee the risk associated with a customer, namely the probability of a customer defaulting on their payment.

In order to have a clearer understanding of customer default, we start off by defining the general context surrounding defaults which is in a certain way an alternative subclass of credit risk.

Credit risk is the potential that a borrower or a group of borrowers will fail to meet their contractual obligations and the future loss that is associated along with it. (Bandyopadhyay, 2016). Typically credit risk modelling has seen applications mostly in the banking industry, where the aim of said models is to calculate the overall ability of a borrower to repay their loan in the originally agreed terms. This is also known as customer default, specifically the failure to make the required payment on a debt, despite the debt’s contractual nature. (Chen, 2022). In the energy industry consumers are considered to have defaulted after they have missed two consecutive payments on their bills.

It is evident that if any number of customers ranging from few to many, were to default on their payments, this could lead to large source of losses for any given company. Therefore, it is clear that the need to evaluate customers and the potential risks associated with them is important.

1.2 Goal of this thesis

The goal of this thesis is to predict the probabilities of customers defaulting in their next months payments using their historical transaction data. Therefore, the problem is defined as supervised imbalanced classification, where the majority class (as it is shown in section 3) are the normal non defaulting customers and the minority class are the defaulters.

1.3 Thesis Structure

Starting off this thesis, we describe the general context surrounding the collaborating company's industry. We define the customer-oriented risks that are associated in the energy industry and we link the subject of this thesis with problems of very similar nature posed in other industries (e.g. banking loan defaults).

In chapter 2 we provide a short review of related projects, the methodology that they adopted and the results that were accomplished.

In chapter 3 a detailed step by step guide of the methodology that was followed is provided. First, we define the problem at hand and provide a detailed description of the provided dataset. After that we review the theory and present the results and insights from the initial steps before modelling, namely in data cleaning, feature engineering and exploratory analysis. At the end of chapter 3 we review the theory behind algorithms (Random forests, Extreme gradient boosting and logistic regression) and the metrics used (Precision-Recall curve, F1-score) to create and evaluate our models.

In chapter 4, having reviewed the methodology and the technical background, we move to the core of the project, where the results of the training and testing of the models are presented and discussed thoroughly.

Lastly, in chapter 5, we discuss the concluding assumptions regarding the overall results of the project , some of the limitations we faced and at the end we provide ideas for possible extensions and improvements of this model.

The above structure is illustrated in figure 1 below:

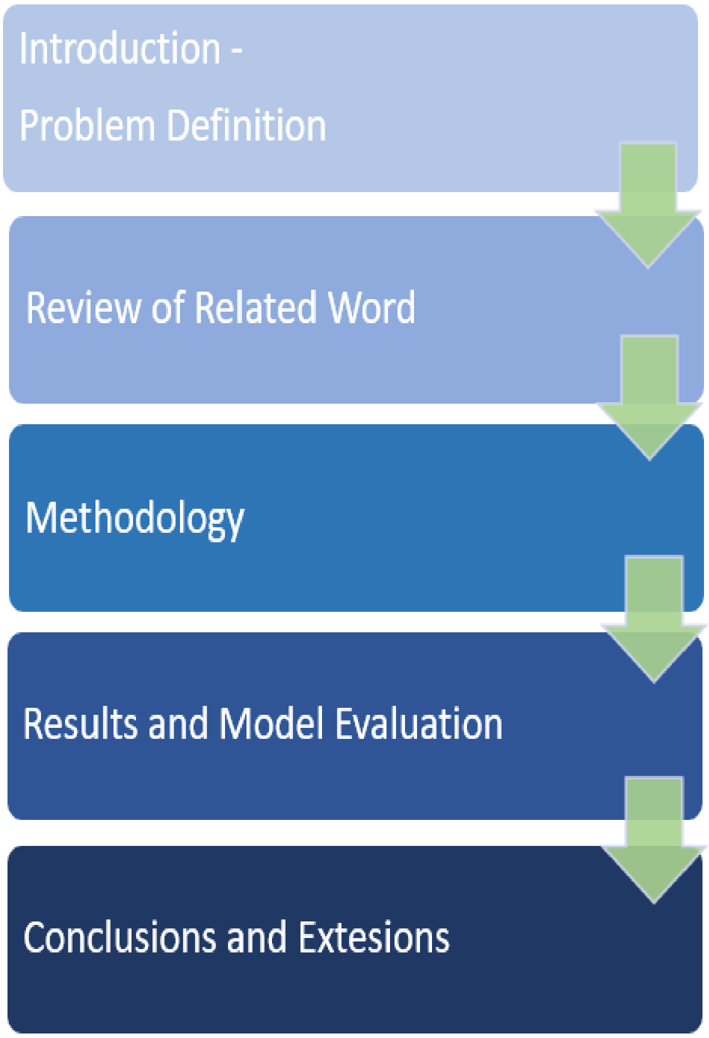


Figure 1: Thesis Structure Flowchart

2. REVIEW OF RELATED WORK

In recent years, customer default prediction has seen many applications and approaches due to its importance for banks and other financial institutions. According to Ahmad Al Qerem et al., predictive modelling for loan defaults is one of the most important tasks for financial institutions. In their study they emphasize the key role of correctly engineering features to improve model performance. The dataset they used was provided from a lending club and contained real world data with a total of 145 attributes and over 43 thousand entries, containing information for loans such as the loan amount and purpose, customer details such as demographic information and features regarding borrower behaviour. Ahmad Al Qerem et al. after data pre-processing applied three feature selection algorithms, namely Information gain, generic algorithm and particle swarm optimization and used F1 score as metric to compare three predictive models, namely Naïve Bayes, Decision Tree and Random Forest. When comparing the unprocessed results with the ones that had undergone data pre-processing, performance improved from 3% to 40% showing the impact that data engineering has in model performance. (Ahmad Al Qerem et al., 2020)

Wang et al. created an ensemble model in an attempt to efficiently take on the class imbalance that is commonly appearing in credit default prediction problems. In order to achieve this goal, they created a standard XGBoost and a proposed a novel CT-XGBoost which was created by using two algorithms to address the class imbalance. Using data that was provided by a Chinese bank that contained information of debtors to said bank, they implemented Logistic Regression, Support Vector Machines, Neural Networks and Random Forest in order to compare the results with the XGB models using the Type I and Type II accuracy metrics. The results demonstrated that using the new proposed XBG model yielded satisfying performance, which was significantly better that all the other models that were implemented. (Wang et al., n.d.)

In another study made by Wijewardhana et al., Logistic Regression, Artificial Neural Networks and affinity analysis models were used to create a model that would predict the probability of a borrower repaying their debt. In their work, they highlight the complications that arise from the limited and incomplete data that is generally available in such tasks. Using a dataset that consisted of 37 variables and over 200.000 entries they concluded that all 3 models offered satisfying results, with the best model being the Neural Network one, that included a clustering step before making the final prediction (Wijewardhana et al., 2018).

Rising Odegua focused in identifying the features that risky customers are associated with in bank loan defaults. The model applied for predicting default was XGBoost in a real-world loan dataset that was provided from a bank and consisted of over 31 attributes with information for loan applications and demographics of customers with a total of 26.897 entries each. Using this dataset Odegua demonstrated that the most important characteristics of a defaulter location and age. The implemented XGB model achieved an F1-score of 87%. (Odegua , n.d.)

In another study Lin Zhu et al. showed that Random Forest is very formidable classifier when dealing with binary classification tasks. Lin Zhu et al. compared 4 different algorithms for loan default prediction, namely Random Forest, Logistic Regression, Decision Trees and Support Vector Machines. The models were applied on a real-world user loan dataset that was composed of 102 attributes and over 115.000 instances. These models were compared using the F1-Score as a metric with the Random Forest model achieving the best performance of all the rest with and F1-Score of 98% and Logistic Regression having the least F1 of 73%. (Zhu et al., 2019)

3. Methodology

In this chapter we lay the theoretical ground for the main analysis of the thesis. In the first segment we state the assumptions we make across the thesis. In the second segment we give a detailed description of the provided dataset and the included features.
Lastly, we provide a short review of the theoretical background of the algorithms used when creating the model as well as the metrics used to evaluate it.
The overall process of the project is illustrated in the flowchart in figure 2.

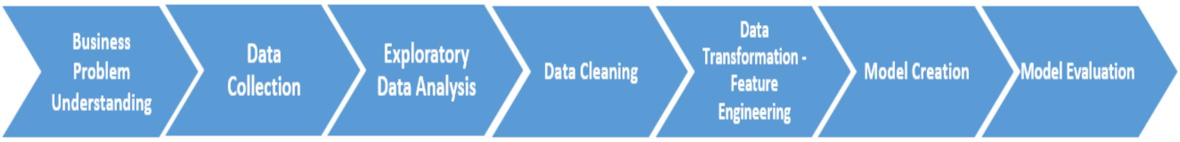
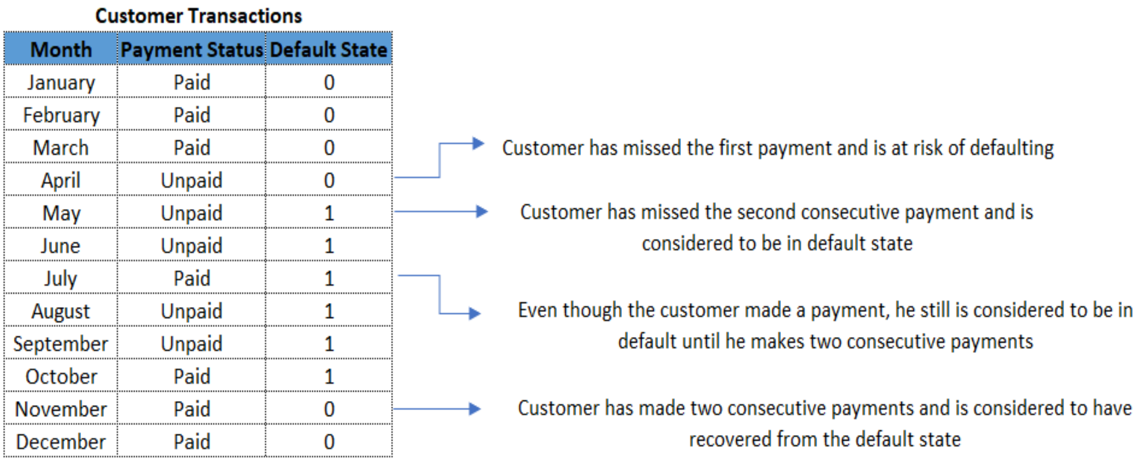

Business Problem Understanding
Data Collection
Exploratory Data Analysis
Data Cleaning
Data Transformation - Feature Engineering
Model Creation
Model Evaluation

Figure 2: Project Flowchart

3.1 Thesis Assumptions

In this segment we state the various assumptions we make in this thesis. The most significant assumption regards to the response variable which is the default state of the customer. The original dataset that was provided did not include this response variable therefore it was created according to the business rules of the collaborating company. According to those rules, if a customer has missed two consecutive payments, then they are considered to be in default. Similarly, if a customer who is in default has paid two consecutive bills, then that customer is considered to have transitioned from the default state to the regular state.



Customer Transactions

Month	Payment Status	Default State
January	Paid	0
February	Paid	0
March	Paid	0
April	Unpaid	0
May	Unpaid	1
June	Unpaid	1
July	Paid	1
August	Unpaid	1
September	Unpaid	1
October	Paid	1
November	Paid	0
December	Paid	0

Customer has missed the first payment and is at risk of defaulting

Customer has missed the second consecutive payment and is considered to be in default state

Even though the customer made a payment, he still is considered to be in default until he makes two consecutive payments

Customer has made two consecutive payments and is considered to have recovered from the default state

Table 1: An indicative example of a customer transitioning to and from a default state.

In our provided dataset, many defaulters still remained in the company's customer-base long after they have stopped paying. This means that even after they were considered to have entered the 'default-state' according to the above assumption the company continued to provide them with electricity, and they continued to not pay their bills. In this thesis we have made the assumption that the cooperation 'ends' when someone defaults for the first time. Consequently, since our goal was to predict the first default of a customer, all transaction after the event of interest were deleted. Since the algorithm is meant to be updated each month with the current month's data, we have removed the last transaction of all customers to train the model to 'look one step ahead'. Therefore, we use n_{i-1} number of observations to predict the outcome at time n_i, where n is the number of observations.

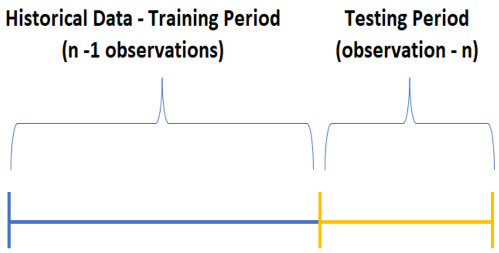


Figure 3: Timeframe of observations.

3.2 Dataset Description

The dataset that was provided included the transactions for 4995 unique customers. The original data consisted of 8 datetime features, 27 numeric features, 12 categorical features and 2 IDs. The data includes customer billing information, such as bill amount and energy amount. Table 1 display all the features that were initially provided along with their respective descriptions.

Feature number	Attribute	Description
1	ΕΠΩΝΥΜΙΑ ΠΑΚΕΤΟΥ	Name of the package the customer uses
2	ΤΥΠΟΣ ΤΙΜΟΛΟΓΙΟΥ	Invoice type
3	ΗΜΕΡΟΜΗΝΙΑ ΤΙΜΟΛΟΓΗΣΗΣ	Invoice date
4	ΗΜΕΡΟΜΗΝΙΑ ΛΗΞΗΣ	Invoice expiration date
5	ΚΑΤΑΣΤΑΣΗ ΠΛΗΡΩΜΗΣ	State of payment : Whether the bill was paid or unpaid
6	ΚΑΤΑΣΤΑΣΗ ΤΙΜΟΛΟΓΙΟΥ	State of invoice
7	ΑΝΟΙΧΤΟ ΥΠΟΛΟΙΠΟ	The unpaid amount of the current bill that was not paid from the customer
8	ΤΙΜΟΛΟΓΗΘΕΝ ΠΟΣΟ	The bill amount for a given month
9	ΥΠΟΛΟΙΠΟ ΕΝΕΡΓΕΙΑΣ	Unpaid energy amount
10	ΑΠΟ ΗΜΕΡΟΜΗΝΙΑ	Start date that the invoice refers to
11	ΕΩΣ ΗΜΕΡΟΜΗΝΙΑ	End date that the invoice refers to
12	ΚΑΤΑΝΑΛΩΣΗ ΗΜΕΡΑΣ	Consumption during the day
13	ΚΑΤΑΝΑΛΩΣΗ ΝΥΧΤΑΣ	Consumption during the night
14	ΠΑΓΙΟ ΗΜΕΡΑΣ	Fixed day charge
15	ΤΙΜΗ ΜΟΝΑΔΟΣ ΗΜΕΡΑΣ	Unit price of day hours

16	ΧΡΕΩΣΗ ΗΜΕΡΑΣ	Day charge
17	ΠΑΓΙΟ ΝΥΧΤΑΣ	Fixed night charge
18	ΤΙΜΗ ΜΟΝΑΔΟΣ ΝΥΧΤΑΣ	Unit price of night hours
19	ΑΠΟ ΕΚΚΑΘΑΡΙΣΗΣ	Clearence from this date
20	ΕΩΣ ΕΚΚΑΘΑΡΙΣΗΣ	Clearence until this date
21	ΗΜΕΡΕΣ ΕΚΚΑΘΑΡΙΣΗΣ	Clearence number of days
22	ΤΕΛΙΚΟΣ	Binary : 1 means the customer received the final invoice and left the company.
23	ΣΥΝΟΛΙΚΟ ΠΟΣΟ	Total amount to be paid
24	ΠΟΣΟ ΠΡΟ ΕΚΠΤΩΣΗΣ	Amount before discount
25	ΠΟΣΟ ΕΚΠΤΩΣΗΣ	Discount amount
26	ΠΟΣΟ ΕΓΓΥΗΣΗΣ	Warranty amount
27	ΑΞΙΑ ΕΝΑΝΤΙ	The amount the customer needs to pay from an approximate energy measurement
28	ΡΗΤΡΑ ΟΤΣ	System threshold clause
29	ΛΟΙΠΕΣ ΧΡΕΩΣΕΙΣ	General charges
30	CLAWBACK	The amount the customer needs to pay if he leaves the company before the contract expires
31	ΚΑΘΥΣΤΕΡΗΣΗ ΠΛΗΡΩΜΗΣ	Number of days that the customer has not paid the bill
32	ΚΑΤΑΣΤΑΣΗ ΑΠΟΠΛΗΡΩΜΗΣ	If a payment is ontime or is overdue
33	ΤΕΛΕΥΤΑΙΑ ΗΜΕΡΟΜΗΝΙΑ ΠΛΗΡΩΜΗΣ	The last time the customer paid
34	ΕΠΩΝΥΜΙΑ ΔΗΜΟΥ	The municipality of the customer
35	ΣΥΝΕΡΓΑΤΗΣ	Collaborator is an external partner that brings customers to the company
36	PhysAddrStreetNumber	Physical Address Street Number
37	PhysAddrPostalCode	Physical Address Postal Code
38	PhysAddrPrefecture	Physical Address Addr Prefecture
39	ΗΜΕΡΟΜΗΝΙΑ ΛΗΞΗΣ ΣΥΜΒΟΛΑΙΟΥ	Expiration date of contract
40	ΣΥΝΟΛΟ ΕΝΕΡΓΕΙΑΣ	Total energy amount to be paid
41	ΣΥΝΟΛΟ ΡΥΘΜΙΖΟΜΕΝΩΝ	These are some standard charges that are applied to all customers
42	ΣΥΝΟΛΟ ΛΟΓΑΡΙΑΣΜΟΥ	The total amount of the bill to be paid
43	ΕΙΝΑΙ ΝΥΧΤΕΡΙΝΟ	If the customer has a clock for measuring the night time consumption
44	DTM2	Square meters of the house
45	ΠΟΣΟ ΔΙΑΚΑΝΟΝΙΣΜΟΥ	Settlement amount
46	ΠΛΗΡΩΤΕΟ	The total amount to be paid in the current bill (includes previous unpaid amounts)
47	ΠΡΟΗΓΟΥΜΕΝΟ ΑΝΕΞΟΦΛΗΤΟ	Total previous unpaid amount
48	ΕΠΩΝΥΜΙΑ_2	Customer Id
49	ΠΑΡΟΧΗ_2	Provision Id

Table 2: Description of Dataset Attributes

3.3 Feature Creation

While the original dataset that was provided contained a large number of features, many of them provided very similar information. Therefore, in order to extract the most information possible more features were created that better described the nuances of the given dataset. Table 2 showcases the represents the features that were created along with their respective explanations.

Feature number	Feature	Description
1	missed_payment	1 if 'ΚΑΤΑΣΤΑΣΗ ΠΛΗΡΩΜΗΣ' is marked as UNPAID , 0 otherwise
2	ΜΗΝΑΣ/ΕΤΟΣ ΛΗΞΗΣ ΤΙΜΟΛΟΓΗΣΗΣ	month and year that the bill expires
3	period	month and year that the customer was billed
4	default_year	year that the customer defaulted
5	avg_open_balance	average open balance of the customer
6	avg ΤΙΜΟΛΟΓΗΘΕΝ ΠΟΣΟ	average amount that the customer was billed
7	avg_daily_consumption	average daily consumption of each customer
8	avg_night_consumption	average consumption during night hours
9	avg_total_amount	average total amount that the customer was billed
10	avg_delay	average payment delay of each customer
11	avg_previous_unpaid	average unpaid amount of each customer
12	avg_ΠΛΗΡΩΤΕΟ	average amount to be paid (including debt)
13	avg ΤΙΜΗ ΜΟΝΑΔΟΣ ΗΜΕΡΑΣ	average price the customer was charged during daytime
14	avg ΥΠΟΛΟΙΠΟ ΕΝΕΡΓΕΙΑΣ	average unpaid energy amount
15	ΤΕΛΙΚΟΣ	1 if the customer left the company , 0 otherwise
16	unpaid_ratio	ration of total unpaid/paid bills of each customer
17	DTM2	square meters of each customers house
18	ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ	length of cooperation between customer and company
19	ΜΗΝΑΣ ΤΙΜΟΛΟΓΗΣΗΣ	month that the customer was last billed

20	number of contracts	number of contracts that the customer has had throughout his cooperation
21	Defaulter	1 if customer defaulted, 0 otherwise
22	number of overdue payments	number of times the customer was late on their payment
23	quarter	the quarter of the year that the customer was last billed

Table 3: List of created features based on the original dataset.

3.4 Exploratory Analysis – Insights

One of the most important steps in a data science project is exploratory analysis. In this step the goal is to explore the data and uncover possible patterns and insights. In this segment the main results of the exploratory analysis and the insights gained are presented as well as behavioural patterns, which acted as a steppingstone in helping identify important features that needed to be included to our model.

After dividing the dataset in two classes, it was deemed important to investigate whether the characteristics of defaulters followed certain patterns.

When investigating payment behaviour, a logical starting point is to look at the amounts the different classes of customers were billed over time. One probable cause of influence of a customer defaulting on their payment is bill amount. Figure 4 displays the average bill amount that defaulters received in contrast to the other customers.

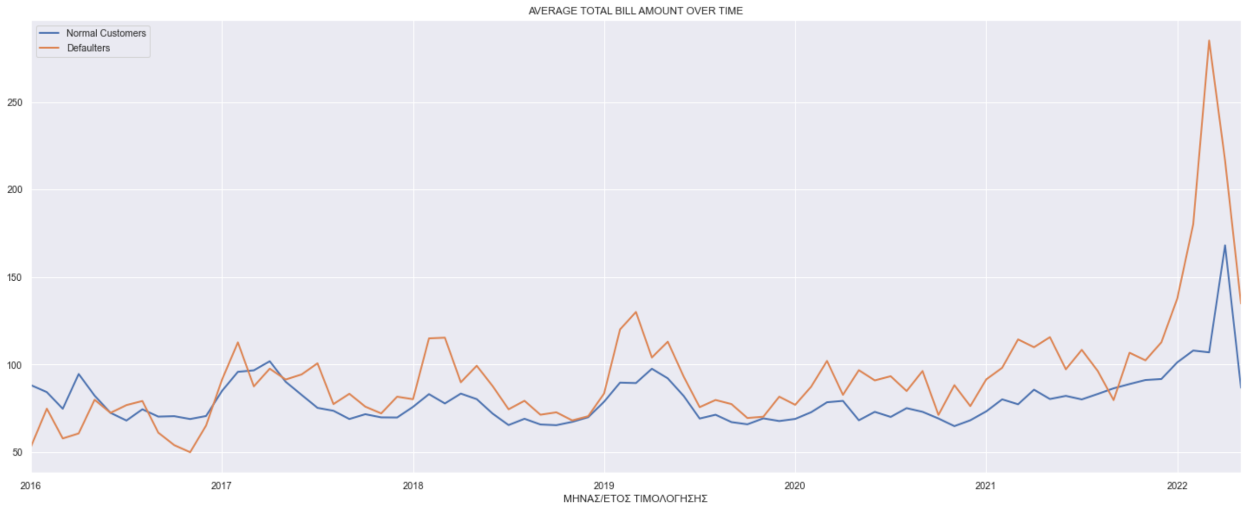


Figure 4: Bill amounts over time

Unsurprisingly defaulters tend to receive higher bill amounts in contrast to non-defaulters, with a large spike being observed for both classes form the beginning of 2022.

High billed amounts indicated a link with the power consumption, as the two are highly correlated. As it was expected power consumption followed similar patterns with bill amounts for both classes, with the defaulters once again maintaining higher values than normal customers.

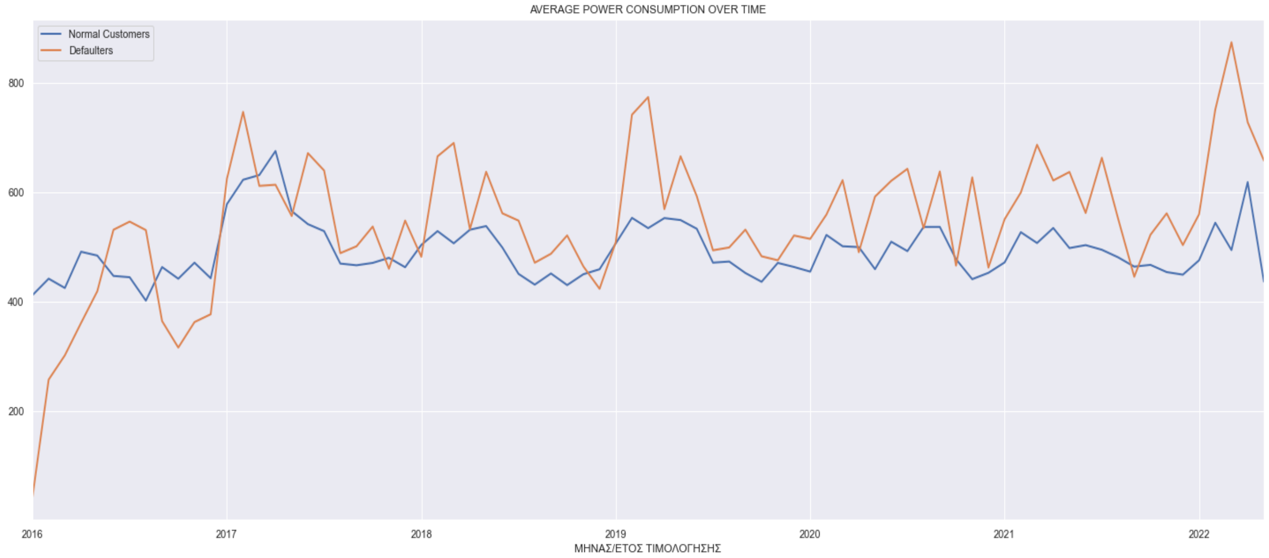


Figure 5: Average power consumption over time

Open balance is also a key factor when it comes to defaults. Open balance is defined in section (3.2) as the amount the customer leaves unpaid for a given bill. When observing figure 6 another pattern is spotted, where defaulters tend to leave bills unpaid with a large increase starting after 2021.

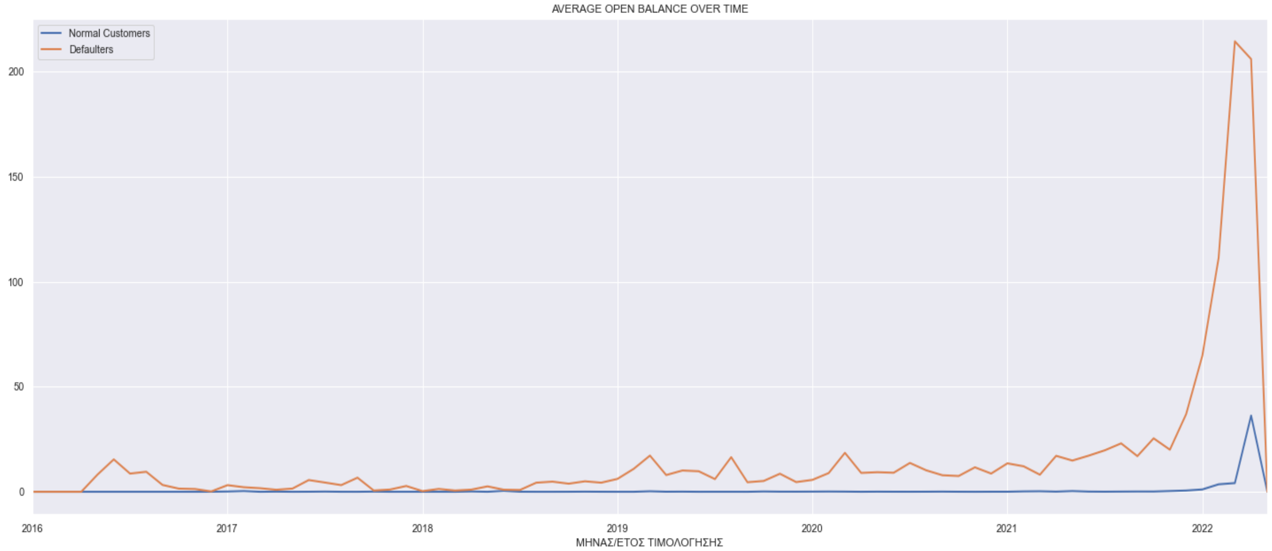


Figure 6: Average open balance over time

Once again, this pattern does not come as a surprise as increased bills and consumption would inevitably lead to increased debt and by extension defaults. This fact is backed by figure 7, where the annualized default rate shows a massive upward trend in 2022.

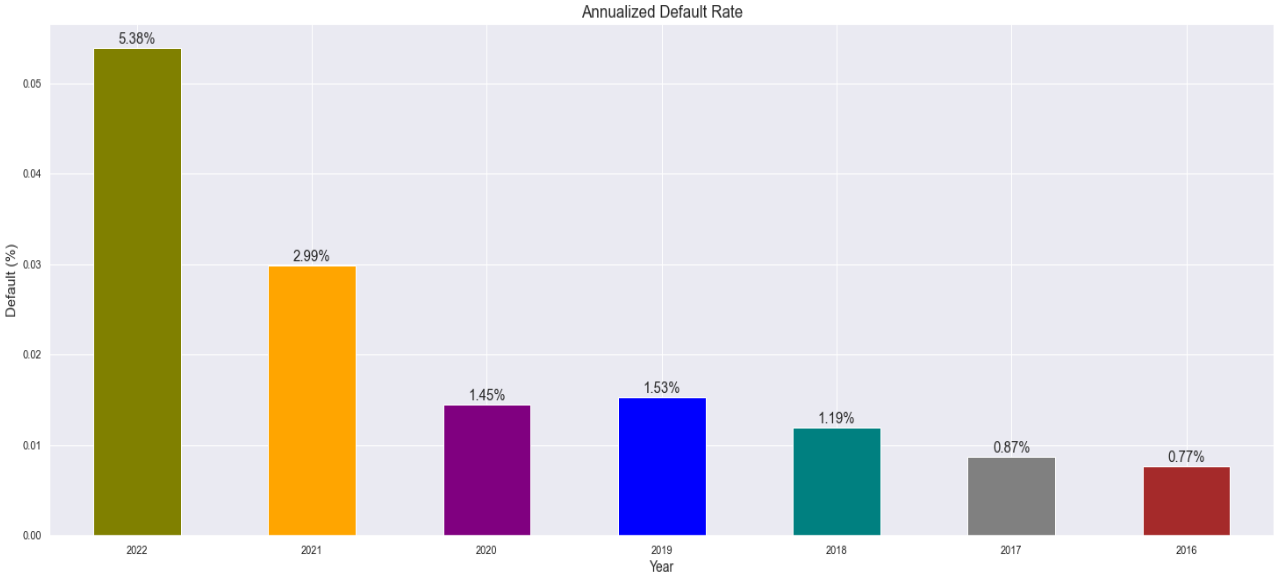


Figure 7: Annualized default rate

Another significant point of interest is whether default is related in some way to seasonality. Graph 8 shows that there is a significantly higher number of defaults during the first and second quarter of the year. This comes as no surprise since the first 4 months of the year are typically the coldest ones in Greece and therefore energy consumption is increased due to heating needs and by extension bills amounts also increase.

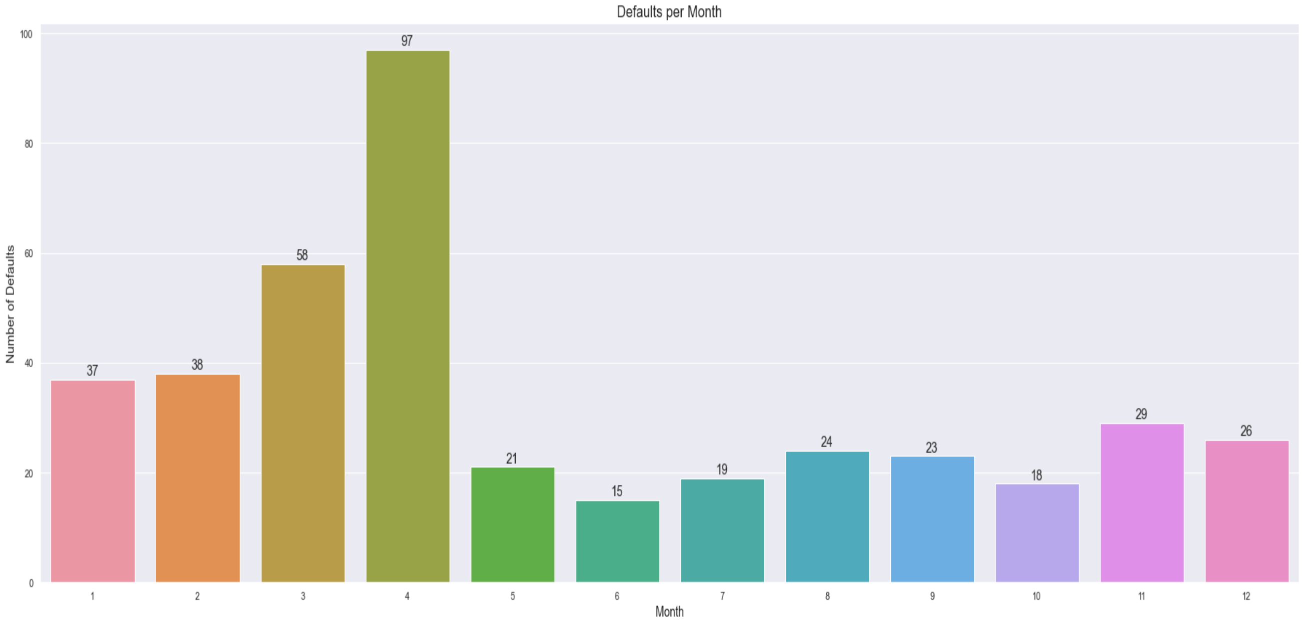


Figure 8: Average open balance over time

Lastly, one important question that needed to be investigated was to identify the length of time before defaulters stopped paying their account. The answer was that the vast majority of defaulters tend to default within a year after the start of their cooperation with the company.

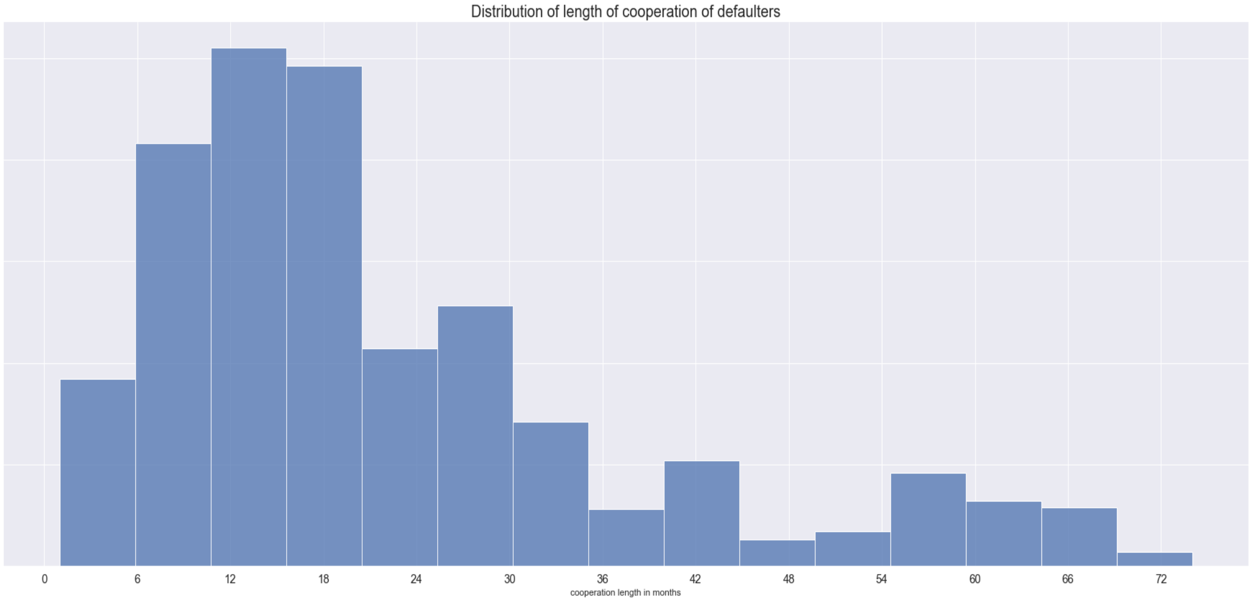


Figure 9: Time distribution before customer default

The last statement holds true, since the vast majority of defaulters do not renew their contracts as as shown in figure 10:

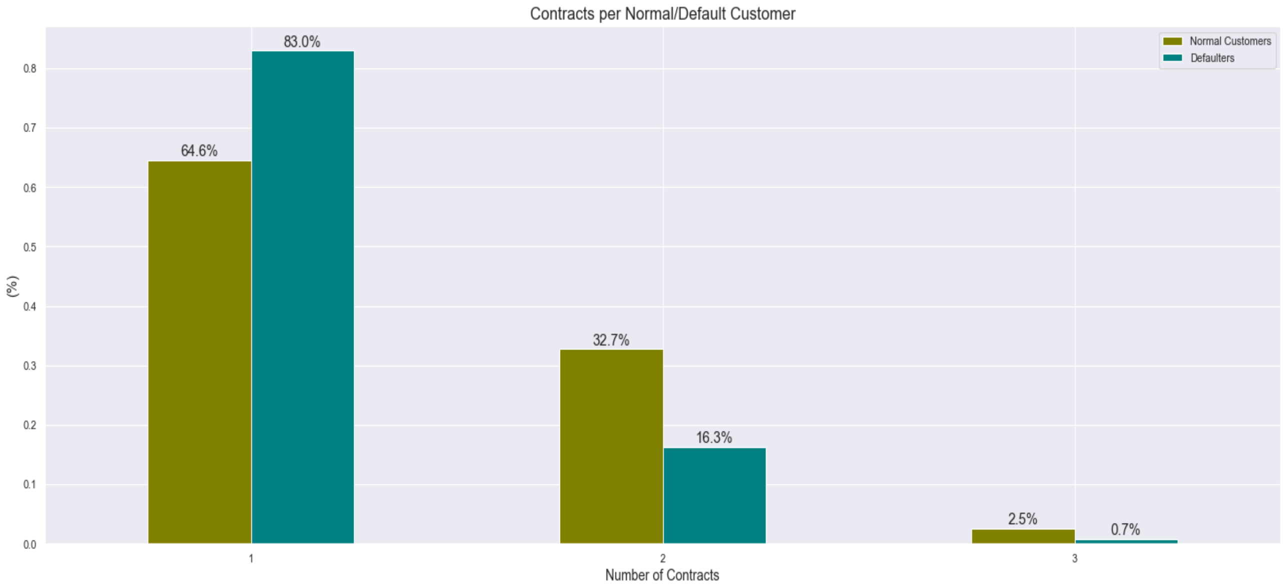


Figure 10: Number of contracts per customer class

3.5 Data Cleaning

Arguably one of the most important steps in a data analysis project is the correction and cleaning of the dataset. In this section we present the cleaning techniques applied for cleaning in this dataset. During the data cleaning segment, the below considerations were made:

Nulls, duplicates and features with low information

First step was to check for duplicate entries, but none we identified. Next step was to remove features that contained mostly nulls, since they did not provide adequate information to be considered in the main dataset. According to figure 11 , features: ‘ΑΠΟ ΕΚΚΑΘΑΡΙΣΗΣ’,’ΕΩΣ ΕΚΚΑΘΑΡΙΣΗΣ’,’ΗΜΕΡΕΣ ΕΚΚΑΘΑΡΙΣΗΣ’,’ΠΟΣΟ ΠΡΟ ΕΚΠΤΩΣΗΣ’ consisted of over 70% nulls so the features were dropped.

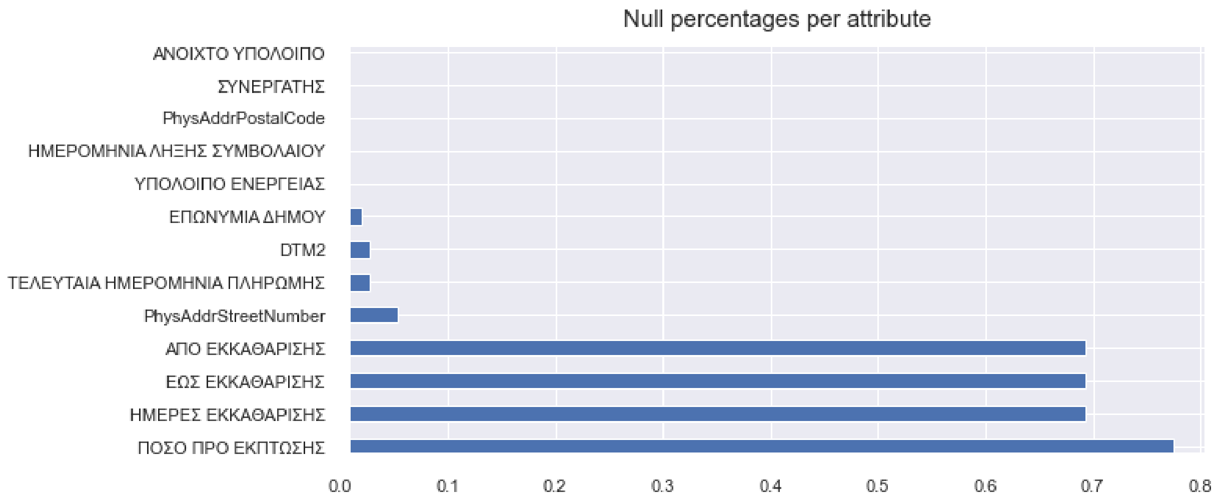


Figure 11: Null percentages per attribute

As for variables ‘ΕΠΩΝΥΜΙΑ ΔΗΜΟΥ’ and ‘DTM2’ if the customer had other non-null entries in these attributes, then nulls were filled using the observation with the highest frequency of each customer. Otherwise, the nulls were filled using the observation with the highest frequency for each feature.

Attributes with only 1 unique value or zero values were removed since they did not offer any additional information. These features were ‘CLAWBACK’,’ ΠΟΣΟ ΔΙΑΚΑΝΟΝΙΣΜΟΥ’, and ‘ΚΑΤΑΣΤΑΣΗ ΤΙΜΟΛΟΓΙΟΥ"’ and ‘ΠΟΣΟ ΕΚΠΤΩΣΗΣ’.

Outliers

An outlier is an observation that has significantly different values than the rest of the observations in a given feature. If outliers are not treated, then the model performance is greatly affected. In this study we implemented the use of boxplots to identify outliers and the IQR method to treat them.

According to IQR method, the data is divided into 3 percentiles namely Q1, Q2, Q3. The IQR range is defined as the difference between the third and first percentile:

$$IQR = Q3 - Q1$$

An observation is considered as an outlier if it lies outside the range:

$$[Q1 - 1.5*IQR, Q3 + 1.5*IQR] \quad (1)$$

Using the IQR method the outliers were replaced with the upper and lower bounds defined by function (1)**.** Figure 12 presents a visual illustration of how boxplots are used to detect outliers. Maximum value is defined by the upper bound of equation (1) and minimum value by the lower bound of equation (1). All points outside this range are considered to be outliers.

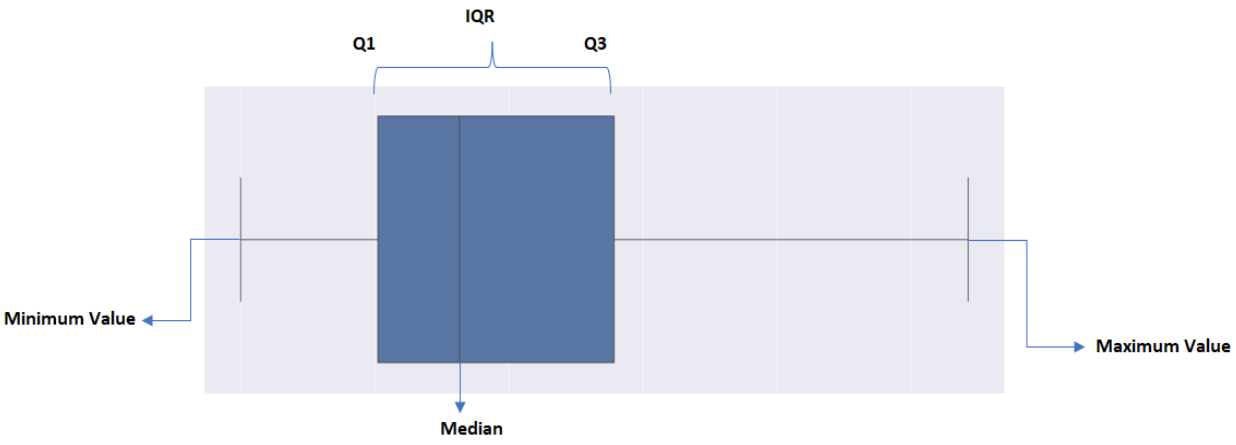


Figure 12: Visual Representation of boxplot

To provide an example, based on figure 13 we can spot outliers on the right side of the boxplot for daily power consumption with abnormal values of over 10.000 kilowatt hours.

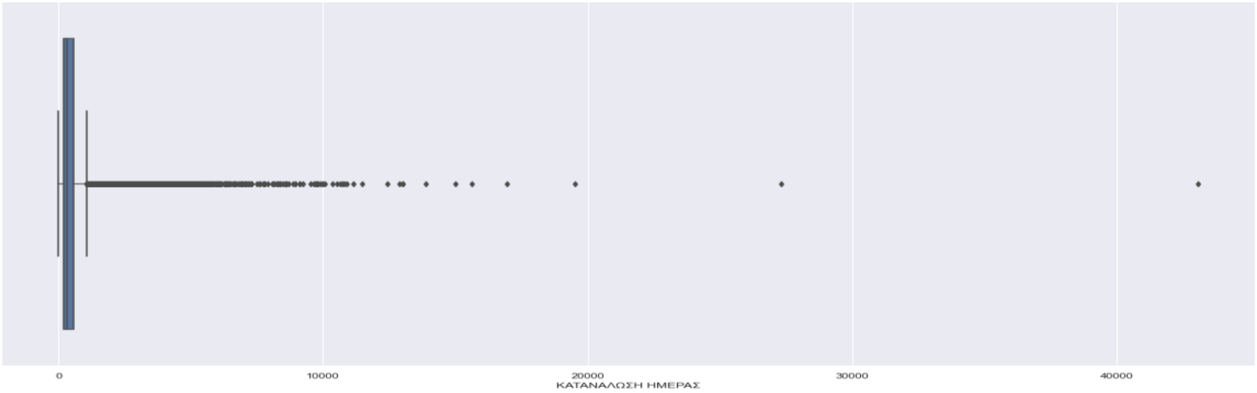


Figure 13: Boxplot for feature ‘ΚΑΤΑΝΑΛΩΣΗ ΗΜΕΡΑΣ’ before IQR

Using the IQR method we replace the spotted outliers with the lower and upper bounds defined from (1). The resulting boxplot is shown in figure 14.

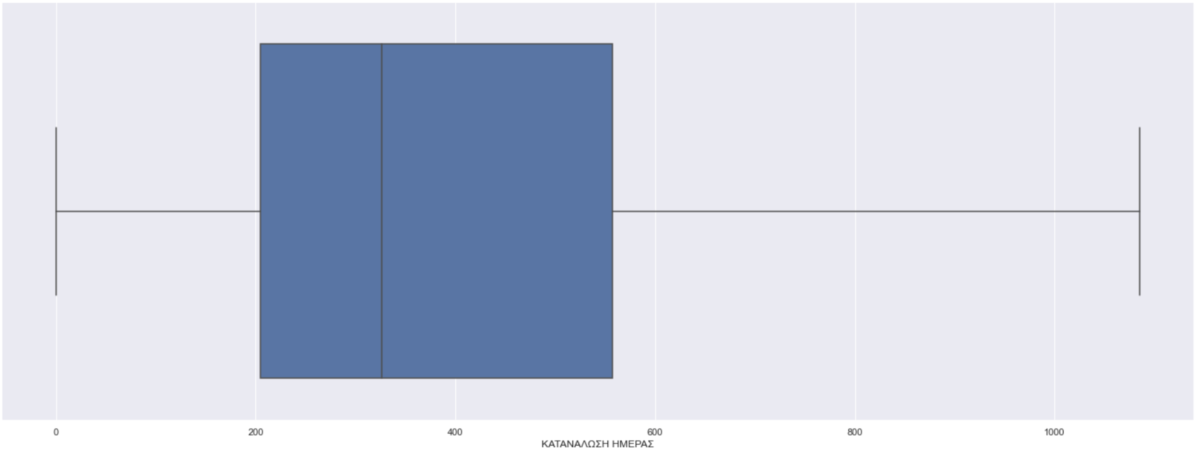


Figure 14: Boxplot for feature: ‘ΚΑΤΑΝΑΛΩΣΗ ΗΜΕΡΑΣ’ after IQR

At this point it is important to note that not all features were treated with the IQR method. Some attributes contained important information that the IQR method registered as outliers. An example is given in figure 15 for the feature ‘ΚΑΘΥΣΤΕΡΗΣΗ ΠΛΗΡΩΜΗΣ’.

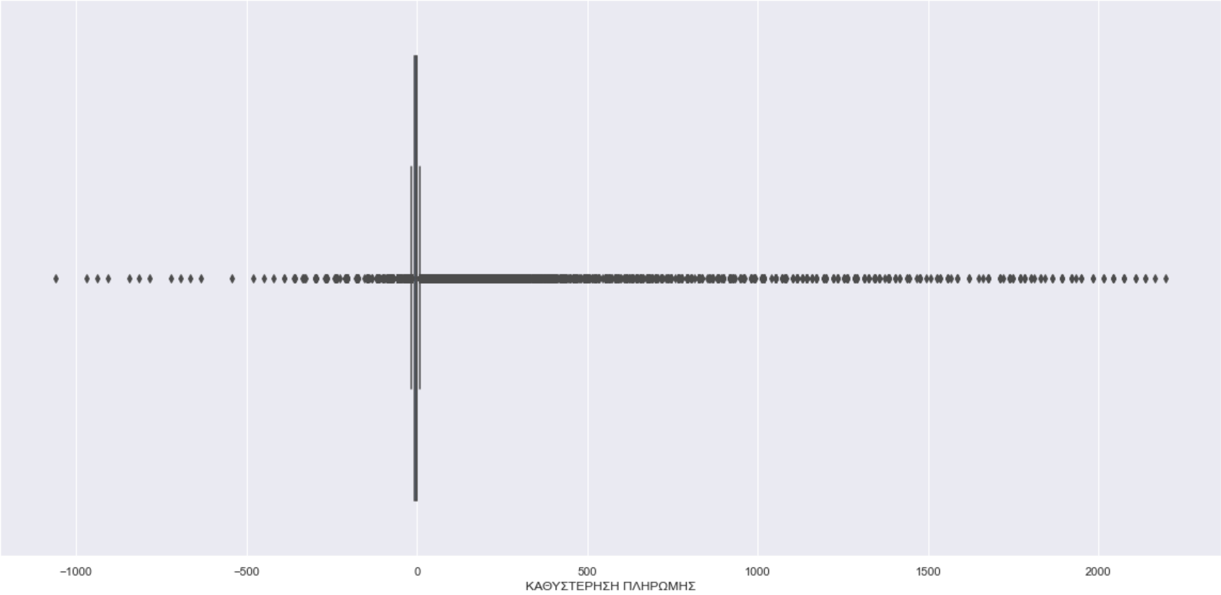
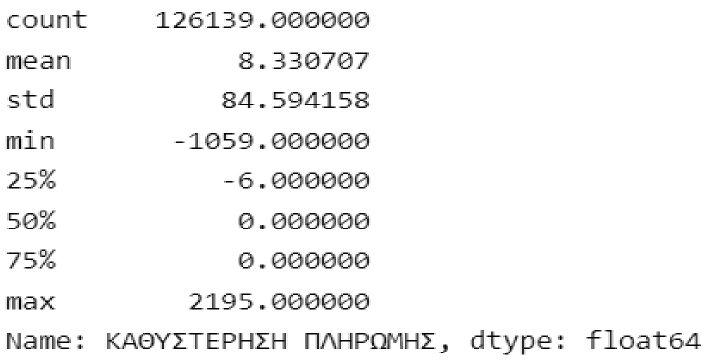


Figure 15: Boxplot of feature: ‘ΚΑΘΥΣΤΕΡΗΣΗ ΠΛΗΡΩΜΗΣ’

From the given boxplot it is evident that this attribute has many extreme values both positive and negative. The problem is however, that since most customer are not late in their payments, this attribute has all three percentiles close to zero, therefore by applying IQR all outliers turn to zero. The summary statistics of this attribute are displayed in table 4 below:



```
count    126139.000000
mean          8.330707
std          84.594158
min       -1059.000000
25%          -6.000000
50%           0.000000
75%           0.000000
max        2195.000000
Name: ΚΑΘΥΣΤΕΡΗΣΗ ΠΛΗΡΩΜΗΣ, dtype: float64
```

Table 4: Summary statistics of feature ‘ΚΑΘΥΣΤΕΡΗΣΗ ΠΛΗΡΩΜΗΣ’

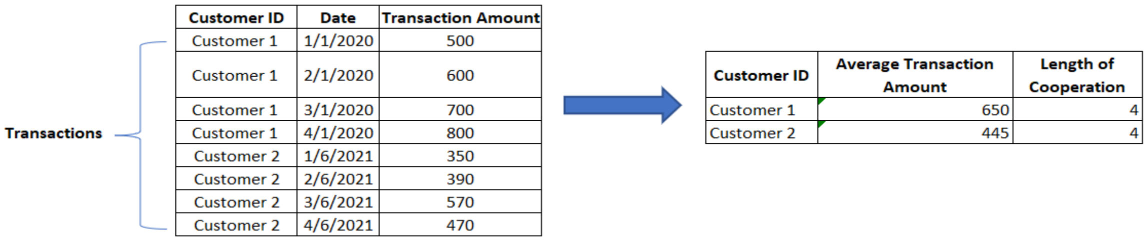
It is for this reason that these features were not treated with IQR and instead were manually capped using business rules, so as to avoid removing useful information. Using this approach, the following features were capped as follows:

- ‘ΚΑΘΥΣΤΕΡΗΣΗ ΠΛΗΡΩΜΗΣ’ had all negative values turned to zero and the positive values were capped at 180 day (6 months).
- ‘ΠΡΟΗΓΟΥΜΕΝΟ ΑΝΕΞΟΦΛΗΤΟ’ had all negatives values turned to zero and the positive values were capped at 2000 euros total debt.

- ‘ΑΝΟΙΧΤΟ ΥΠΟΛΟΙΠΟ’ was capped at 500 euros

3.6 Data Transformation - Feature Engineering

As it was mentioned in (3.1) the original dataset contained the transactions of over 4995 unique customers where each customer owned number of entries equal to the number his transactions. After performing cleaning and analysis, the data needed to be transformed in order to be organically utilized by the machine learning algorithms. To this end, the original dataset was transformed into a new matrix with 4995 entries (which we will refer to as design matrix) in which each entry contained all the information of a specific customer using new created features from the corresponding aggregated statistics. A similar approach is adopted by Zhou Xu in his implementation in ‘Loan Default Prediction with the Berka Dataset’(Zhou, 2020) . The end result was a new design matrix containing rows equal to the number of unique customers.


Transactions

Customer ID	Date	Transaction Amount
Customer 1	1/1/2020	500
Customer 1	2/1/2020	600
Customer 1	3/1/2020	700
Customer 1	4/1/2020	800
Customer 2	1/6/2021	350
Customer 2	2/6/2021	390
Customer 2	3/6/2021	570
Customer 2	4/6/2021	470

Customer ID	Average Transaction Amount	Length of Cooperation
Customer 1	650	4
Customer 2	445	4

Figure 16: Indicative illustration of data transformation process

After the transformation process the new design matrix of our dataset contained 18 features, out of which 4 where categorical variables and 14 were continuous, that were also used in the final models and are given in detail in the table below:

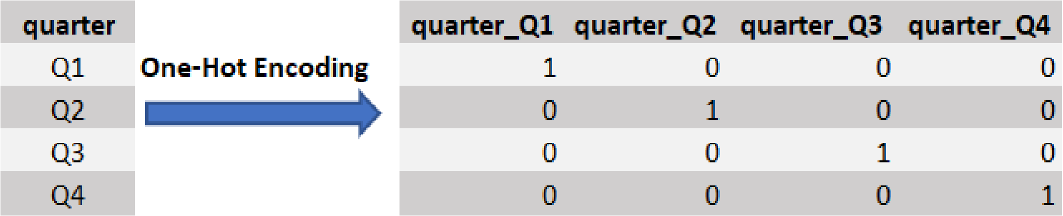
Feature number	Feature	Description
1	avg_open_balance	average open balance of the customer
2	avg_daily_consumption	average daily consumption of each customer
3	avg_night_consumption	average consumption during night hours
4	avg_delay	average payment delay of each customer
5	avg_previous_unpaid	average unpaid amount of each customer
6	avg ΤΙΜΗ ΜΟΝΑΔΟΣ ΗΜΕΡΑΣ	average price the customer was charged during daytime

7	ΤΕΛΙΚΟΣ	1 if the customer left the company , 0 otherwise
8	unpaid_ratio	ration of total unpaid/paid bills of each customer
9	DTM2	square meters of each customers house
10	ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ	length of cooperation between customer and company
11	number of contracts	number of contracts that the customer has had throughout his cooperation
12	Defaulter	1 if customer defaulted, 0 otherwise
13	number of overdue payments	number of times the customer was late on their payment
14	quarter	the quarter of the year that the customer was last billed

Table 5: List of features utilized by the model.

3.6.1 One-Hot Encoding

When using categorical data, in order for it to be correctly used by the machine learning algorithm, it needs to be broken down into binary subclasses for each category. This technique is known as one-hot encoding. Figure 17 present a visual illustration of the way one hot encoding works.


One-Hot Encoding

quarter
Q1
Q2
Q3
Q4

quarter_Q1	quarter_Q2	quarter_Q3	quarter_Q4
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Figure 17: Representation of one hot encoding

3.6.2. Stratified Train-Test Split

When creating a machine learning model, it is important to test its performance with data that the model has never seen, so as to make assumptions of the model's ability to generalize with real world unseen data. To this end we split the dataset into to stratified subsets called the training set and the test set where in each set the ratio of the response variable is preserved. The training set as its name suggests, is used to train the model and make assumptions of its predictive capabilities. The test set represents real world data that the model has never seen and is only used to test the performance of the model. It is of utmost importance that only the train set is used to make decisions about the model and the test set only to test to strictly test its performance, because otherwise there is a high risk of data leakage, meaning that important information regarding the prediction might leak to the test dataset and the resulting model performance will be mistakenly overly optimistic. This is the reason why techniques such as undersampling are applied only in the train set, because the test set is a representation of real-world unseen/unprocessed data.

In our study the dataset used to train the model was contained 70% of the original data and the remaining 30% was used to test the model.

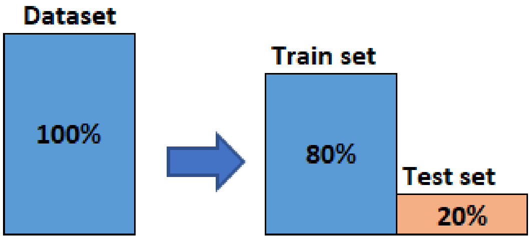
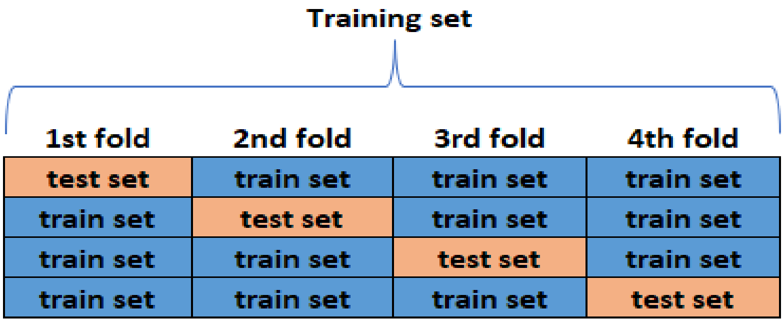


Figure 18: Illustration of Train-Test split

3.6.3. Stratified Cross-Validation

In order to further increase our confidence in the generalization ability of the model we implement cross validation algorithm. Stratified Cross validation splits the training set in a number of equal sized subsets, in our case 5 subsets, and uses one of these subsets as a test set and the rest as a training set while preserving the ratio of the minority with the majority class. This procedure is done iteratively, in each repetition using a different subset as the test set until all possible combinations of the subsets have been used. If the model produces similar results in each repetition, then we have increased confidence in the model's generalization ability. After this procedure is completed, we calculate the mean of a specific metric to gain a stronger understanding of the model's performance.



1st fold	2nd fold	3rd fold	4th fold
test set	train set	train set	train set
train set	test set	train set	train set
train set	train set	test set	train set
train set	train set	train set	test set

Figure 19: Illustration of Cross Validation Procedure

3.6.4 Feature Scaling

Before providing the data to the machine learning algorithms, it is important to scale numeric features in values between [0,1], so as to avoid making a biased model that places more weight on positive values and less on negative values.

In this study we implemented Min-Max scaler, where each numeric feature is scaled by:

$$X_{\mathrm{NEW}} = \frac{X_i - min(X)}{\max(X) - \min(X)}$$

3.6.5 Feature Selection

Model performance, speed and precision is greatly affected by the included number of features and the importance that is associated with them. While there is a great number of feature selection algorithms, each with its own advantages and shortcomings, there is no single best approach since it heavily relies on the underlying problem.

In this study, we performed feature selection right after splitting the data into train and test sets, to avoid the possibility of leaking data to our test set. The process included the following steps:

- Numerical feature selection by identifying the features with predictive power using Pearson Correlation
- Categorical feature selection using chi squared contingency table

Numerical Feature Selection

When choosing which numerical features to include to the model, Pearson's Correlation Coefficient was used. The correlation coefficient is given by equation (3). The values of the coefficient range from -1 to 1, showing complete negative and positive correlation respectively.

$$r = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2 \sum(y_i - \overline{y})^2}} \qquad \textbf{(3)}$$

Where x_i and $y_i$are the values of x and y respectively in a given sample and $\overline{x}$ and $\overline{y}$ are the means of x and y samples respectively.

Calculating the dataset correlation yielded the below heatmap as seen in figure 20.

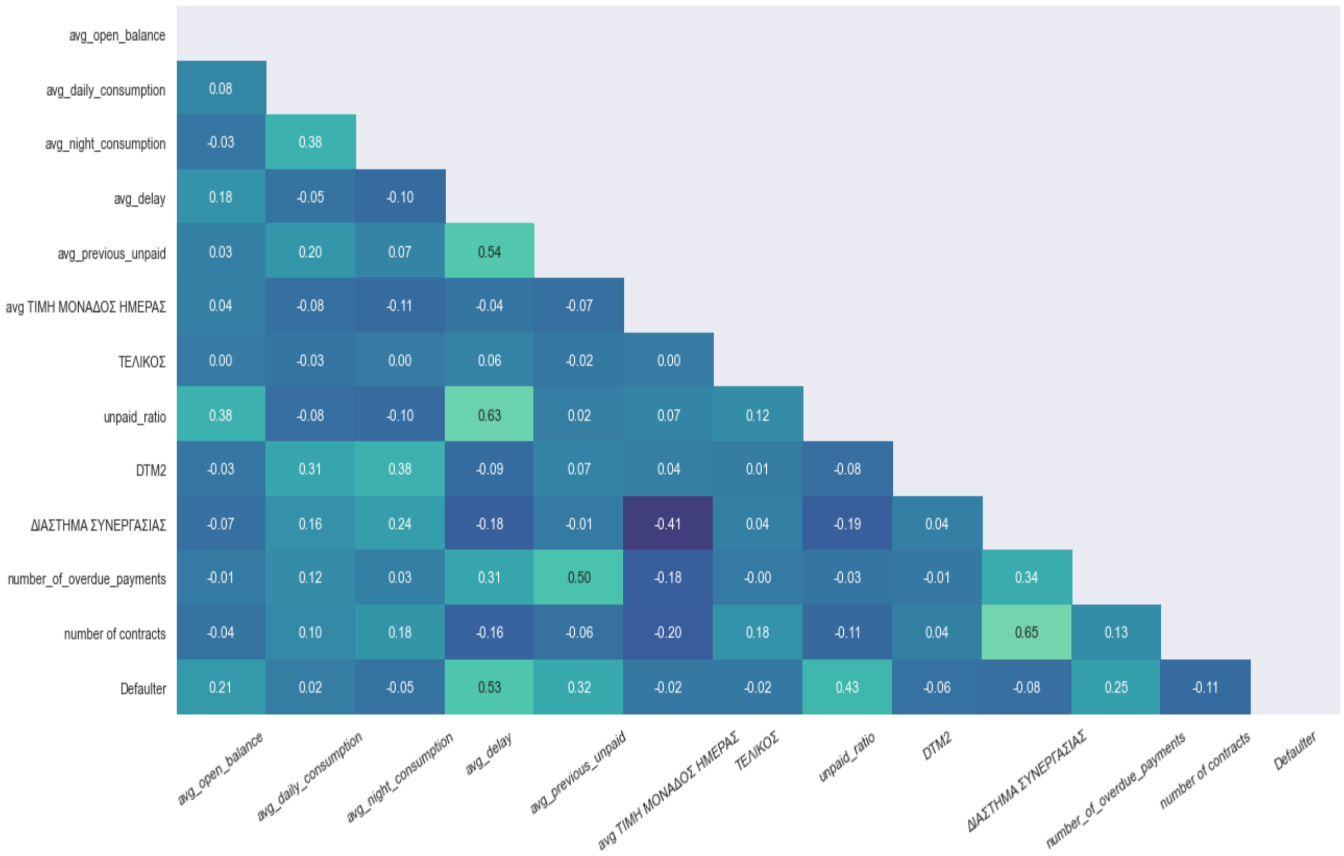
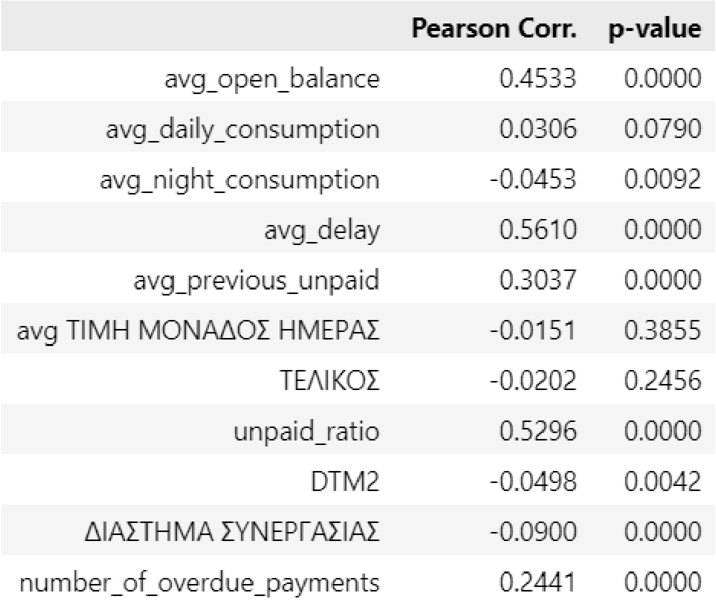


Figure 20: Pearson Correlation Heatmap of Train Set

From the heatmap we can identify which variables have predictive power through their correlation with the target variable ‘Defaulter’.

Variables ‘avg_open_balance’,‘avg_delay’,’avg_previous_unpaid’,’unpaid_ratio’ and ‘number_of_overdue_payments’ all have positive correlation with the response variable of a magnitude at least 0.2. Therefore, we can already make the assumption that these attributes are going to be carry significant information for our model. This came as no surprise since most of these variables were proven to be important in the exploratory analysis of section (3.3).

Calculating the p-values in relation to the response variable for our dataset yielded the results in table number:



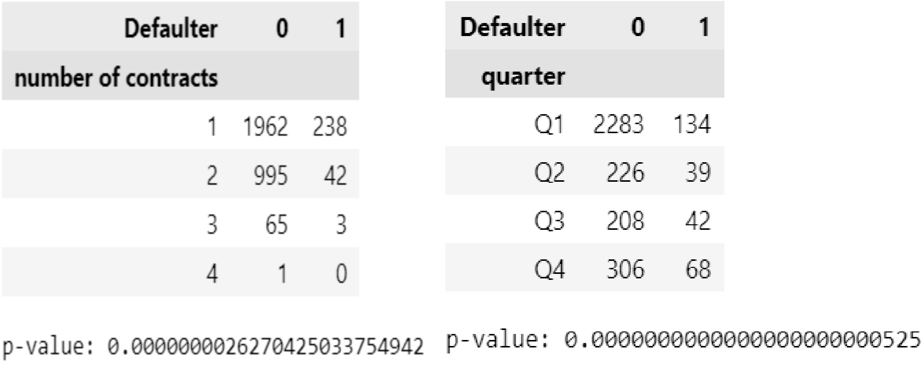
	Pearson Corr.	p-value
avg_open_balance	0.4533	0.0000
avg_daily_consumption	0.0306	0.0790
avg_night_consumption	-0.0453	0.0092
avg_delay	0.5610	0.0000
avg_previous_unpaid	0.3037	0.0000
avg ΤΙΜΗ ΜΟΝΑΔΟΣ ΗΜΕΡΑΣ	-0.0151	0.3855
ΤΕΛΙΚΟΣ	-0.0202	0.2456
unpaid_ratio	0.5296	0.0000
DTM2	-0.0498	0.0042
ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ	-0.0900	0.0000
number_of_overdue_payments	0.2441	0.0000

Table 6: p-values of numerical features in relation to the response variable

Based on the results of table number, only attributes with p-value < 0.05 were kept in the model, thus variables 'avg_daily_consumption' , 'avg_ΤΙΜΗ_ΜΟΝΑΔΟΣ_ΗΜΕΡΑΣ', and 'ΤΕΛΙΚΟΣ' were removed.

<u>Categorical Feature Selection</u>

In order to understand the relationship between categorical variables and the response variable (also categorical) the Chi-square test was utilized. Chi-square tests shows under the null hypothesis H0, whether two categorical are not significantly related, and in the H1 alternative hypothesis that there is significant relations between those variables. Conducting the chi squared test for independence of the categorical variables in our design matrix, yielded us with the two results shown in table 7.



Defaulter	0	1
number of contracts		
1	1962	238
2	995	42
3	65	3
4	1	0

```
p-value: 0.0000000026270425033754942
```

Defaulter	0	1
quarter		
Q1	2283	134
Q2	226	39
Q3	208	42
Q4	306	68

```
p-value: 0.0000000000000000000000525
```

Table 7: Contingency Matrices for Categorical Features

Once again only variables with p-values < 0.05 were considered for the model.

The results of the test were expected since both variables were shown to have significant relationship with the response variable in the exploratory analysis in paragraph (3.3). It should be mentioned that for the attribute ‘number of contracts’ the value 4 was replaced with the next closest value, that is value 3, since it contained only 1 observation and created some computational complications later on.

After completing the feature selection phase, the below 13 variables were used in the model:

Feature number	**Feature**	**Description**
1	avg_open_balance	average open balance of the customer
2	avg_night_consumption	average consumption during night hours
3	avg_delay	average payment delay of each customer
4	avg_previous_unpaid	average unpaid amount of each customer
5	avg ΤΙΜΗ ΜΟΝΑΔΟΣ ΗΜΕΡΑΣ	average price the customer was charged during daytime
6	ΤΕΛΙΚΟΣ	1 if the customer left the company , 0 otherwise
7	unpaid_ratio	ration of total unpaid/paid bills of each customer
8	DTM2	square meters of each customers house
9	ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ	length of cooperation between customer and company
10	number of contracts	number of contracts that the customer has had throughout his cooperation
11	Defaulter	1 if customer defaulted, 0 otherwise
12	number of overdue payments	number of times the customer was late on their payment
13	quarter	the quarter of the year that the customer was last billed

Table 8: Features included in the final model

3.7 Imbalanced Dataset Classification

Classification problems are typically categorized by the number of classes and the size of each class that the response variable contains.

Binary classification, which is our problem case, refers to the task of predicting a binary response variable, meaning that the outcome variable contains only two variables (classes) either 0 or 1. Customer default prediction also falls under the category of binary classification since the goal is to predict whether a customer will default (1) or not default (0) on their loan/payment.

When working on classification tasks, often examples are not equally distributed between the classes.

Imbalanced classification occurs when the number of classes of the response variable are unequally distributed. Imbalanced classification is most commonly encountered in binary classification problems such as fraud detection, or loan default prediction.

In our study the minority class constituted for only 8.5% of the total entries. Figure 21 shows the ratio between defaulters and not defaulters.

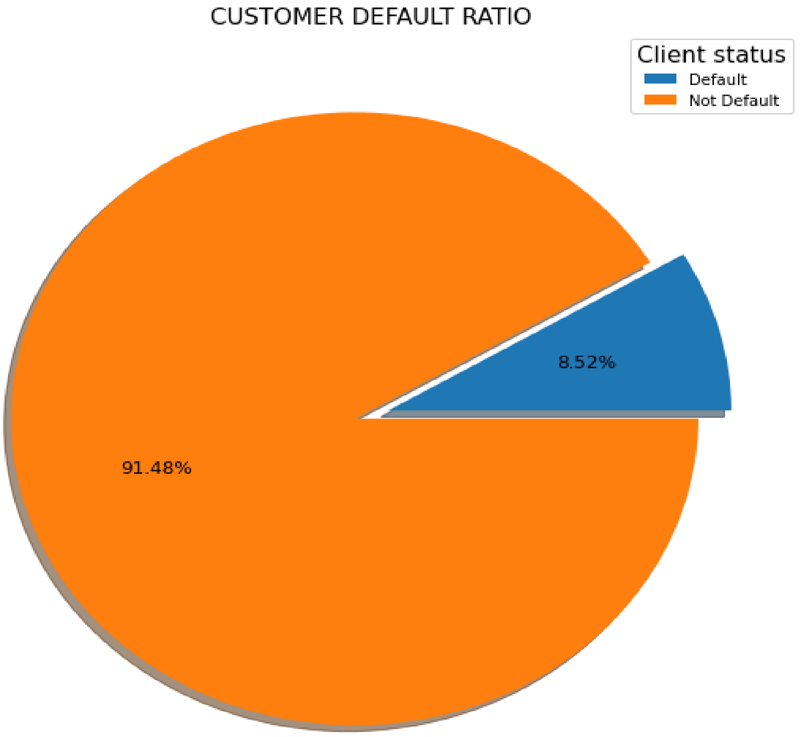


Figure 21: Ratio of Defaulters -Not Defaulters

In such problems if the imbalance between classes is not issued, then the results generated from the model are most likely to be biased, meaning that the model usually always predicts the majority class and still has a high accuracy due to the class imbalance. One important detail is that the techniques that are used to address this class imbalance are used on the training set of the model only and not on the test set. The reason is to prevent information leakage and to prevent the generation of biased results, since the test set is a representation of real-world data that the model is meant to work with.

There are several methods of dealing with class imbalances such as Oversampling and Undersampling. While both methods have several variations, the most frequently used in undersampling are covered in the next section.

In this study, the undersampling of the majority class was achieved through the use of the random undersampling algorithm.

3.7.1. Undersampling

Undersampling refers to the process of making the class distributions equal when dealing with an imbalanced binary classification problem. Undersampling works by taking a smaller sample of the majority class equal to size as the minority size. There are various methods of performing undersampling, however one common disadvantage most share is that there is always the risk of losing crucial samples when selecting a smaller subset from the majority class (mastersindatascience.org, n.d.)

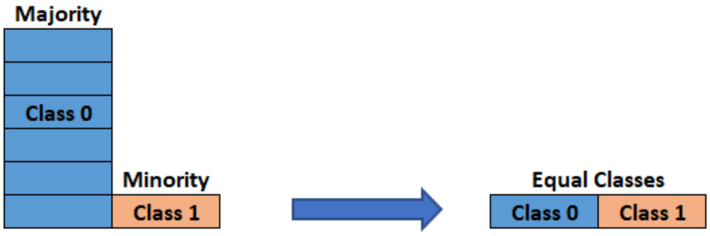


Figure 22: Illustration of Undersampling

Most common techniques for undersampling include:

- Random Undersampling: which is achieved by removing entries from the majority class in a random manner. As mentioned earlier, this technique holds the risk of losing important information when deleting entries at random from the majority class. On the other hand, this method is efficient when there are enough entries in the minority class to fit a useful model. (Brownlee, 2020)
- Condensed Nearest Neighbors (CNN) Undersampling: creates a minimum consistent set , which is a subset of a collection of samples that do not reduce the performance of the model. The way this is accomplished is by enumerating the entries in the given dataset and storing only those that are incorrectly classified by the current contents of the created 'store'. (Brownlee, 2020)

3.8. Model Creation

3.8.1 Logistic regression

When working on binary classification problems, one of the most common algorithms utilised is logistic regression. The main differentiation between standard linear and logistic regression is that in linear regression the result (dependent variable) is a continuous real number, while in logistic regression the dependent variable takes values in [0,1].

The logistic function is defined as:

$$f(x) = \frac{1}{1+e^{-x}}$$

where x takes values in R.

From the definition of the logistic function, we can spot two important properties:

1. $0 < f(x) < 1$ for any real x
2. $\lim_{x\to\infty} f(x)=1$ and $\lim_{x\to-\infty} f(x)=0$

The two aforementioned properties are what make the logistic function able to obtain the probability of a dependent variable belonging to either class 1 or 0. (Wijewardhana et al., 2018)
To generate the logistic regression model, we define logit g(x) as a linear combination of the independent variables where:

$$\text{Logit } g(X)= a+ b_1X_1+b_2X_2+\ldots+b_nX_n$$

The parameters of the logistic regression model are estimated using maximum likelihood.

Therefore, the probability of the response variable Y belonging in class 1 is given by:

$$P\,(Y=1 \mid X=X_1, X_2, \ldots., X_n) = \frac{1}{1+e^{-(a+\Sigma bixi)}} = \frac{1}{1+e^{-g(x)}}$$

Therefore, if the probability of Y belonging to 1 is higher than a predefined threshold (usually set at 50% by default) then the entry will be classified as such.

One important advantage of the logistic regression is the interpretability of its coefficients.
The importance of the coefficients in logistic regression is given by their size. A coefficient b_i interprets as the magnitude of change that the log odds will take if Xi increases or decreases by 1 unit.

In section (3.5) we mentioned that the final model contained some features that deliberately contained outliers. For this reason, we implement an L1-regularized logistic regression which shows more robustness to noise as well as irrelevant data. (Shi et al., 2010) This is a desirable property considering that some features include outliers.
L1 regularization is a linear function of the weights of the covariates, which uses the sum of the absolute values of these weights, that is the norm 1 of weights $||w||_1$ (Jurafsky & Martin, 2021)
The L1 regularized objective function is given by:

$$\hat{w} = argmax_w[\sum_{i=1}^{m} logP(\,y_i|x_i)] - \lambda \sum_{j=1}^{n} |w_j|$$

Where w are the weights, λ is the regularization parameter and m is the number of examples. The first logarithmic term is the loss function of logistic regression which essentially is a metric that shows how much a prediction differs from the actual value.

3.8.2 Random Forest

Since a random forest consists of multiple decision trees, we first review the basic properties of a decision tree in order to understand random forests better.
A decision tree splits the dataset recursively using the decision nodes and stops when we are left with pure leaf nodes, and it finds the best split by maximizing the entropy gain. If a data sample satisfies the condition at a decision node, then it moves towards that leaf (also known as nodes) else it moves to the other leaf. At the end of the tree, when the final leaf is reached then a class label is assigned to it.
One important downside however is that decision trees are likely to overfit the data, especially in cases where the tree expands in many leaves. (Liberman, 2017)

A random forest as its name suggests, uses multiple decision trees to predict the outcome of a classification. The process works by creating multiple decision trees on bootstrapped datasets from the original data and randomly selecting a subset of features to train each tree. With bootstrapping we are ensuring that the same data is not used in every tree, and by extension this aids the model to be less sensitive to the original training dataset. On the other hand, the random feature selection minimizes the correlation that the trees share, since if the same feature were to be used, then all the trees would have the same decision nodes and produce almost identical results.
The new data presented to the model is then classified with each of the trained trees and the end result is the majority vote of those trees. (Yiu , 2019)

The most important upside of random forests is that they are much less prone to overfitting and are robust to outliers and noise and they also do not require a dataset to be normalized because of the branching property of the individual trees.

3.8.3 Extreme Gradient Boosting

XGBoost or Extreme Gradient Boosting, is one of the most utilized algorithms to date known for its scalability in most scenarios (Chen et al., 2016). While this algorithm has the capability to be implemented in regression problems as well, in this section we will focus on the implementation in classification cases. This algorithm utilizes two core techniques, Decision trees and Gradient Boosting.

The philosophy behind boosting is that a more capable predictive model can be created from many models with weak predictive capabilities. Through gradual construction, this method generalizes the model by enabling the optimization of a differentiable loss function. Gradient boosting essentially iteratively combines weak learners into a single powerful learner. A new model is then fitted for each additional weak learner to give a more precise estimate of the target variable, by improving on the residuals of the previous predictor. Therefore, in the Extreme gradient boosting algorithm, the key characteristic is that decision trees take the place of those weak predictors.
According to Dietterich, one important vulnerability of the boosting algorithms is their sensitivity towards outliers (Dietterich, 2000).

According to Chen et al. some of the key features that have promoted the use of this algorithm are the following:

- **Sparse data handling**: It proposes an innovative way to manage sparse data using a tree-based algorithm.

- **Weighted quantile sketch**: To efficiently handle weighted data, XGBoost offers a distributed weighted quantile sketch technique which the majority of tree-based algorithms cannot do.

- **Column block structure for parallel learning**: In-memory units that are called blocks, are categorized and used to store data in memory. This approach, unlike others, allows for reuse of the data layout over successive rounds rather than having to compute it from scratch.

- **Cache aware access:** For XGBoost to obtain the gradient statistics by row index, non-continuous memory access is necessary. Consequently, XGBoost has been created to utilize hardware to its full potential. In order to accomplish this, each thread allocates internal buffers where the gradient statistics can be saved.

- **Out-of-core computation**: When working with large datasets that don't fit in memory, this functionality maximizes the use of the available disk space. The two main techniques used to achieve this is block compression and block sharding. (Chen et al., 2016)

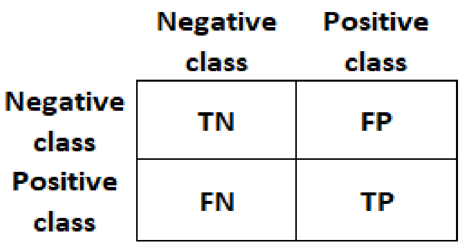
3.9 Classification Metrics

In this section we define the classification metrics we used to evaluate the classification models in this thesis by first defining some basic notations:

Throughout this thesis we refer to class 1 as positive class and to class 0 as negative class.

TP = True positive: the number of samples that were correctly classified in the positive class
FP=False positive: the number of samples that were incorrectly classified in the positive class
TN=True negative: the number of samples that were correctly classified in negative class
FN=False negative: the number of samples that were incorrectly classified in the negative class

- **Confusion matrix**: is a matrix that is used to describe the performance of a model, where it plots the true/false positive/negative metrics as seen in figure:



	Negative class	Positive class
Negative class	TN	FP
Positive class	FN	TP

Figure 23: Indicative Confusion Matrix

- **False positive rate (FPR) =** $\boldsymbol{\frac{FP}{TN+FP}}$

- **Precision =** $\frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$

Precision refers to the number of positive correctly classified in the positive.

- **Recall =** $\frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$

Recall refers to the number of correctly classified predictions in relation to the total population of the positive class.

- **Accuracy** = $\frac{\boldsymbol{TP+TN}}{\boldsymbol{TP+FN+FP+TN}}$

Accuracy refers to the total number of correctly classified samples in relation to the total population.

- **F1 score=** $\frac{2*\boldsymbol{Precision*Recall}}{\boldsymbol{Precision+Recall}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP}+\left(\frac{1}{2}\right)*(\boldsymbol{FP+FN})}$

F1 score is the harmonic mean of precision and recall

- **ROC (Receiver Operating Characteristic) Curve**

ROC plots the FPR (False Positive Rate) in the X-axis and the TPR (True Positive Rate) in the Y-axis of a classifier for all possible thresholds. The lower the threshold of a classifier, the easier it is to classify an example to the positive class. In order to measure the performance of a classifier the AUC (area under curve) is used. A perfect classifier would be a straight line from Y=1 parallel to the X-axis, while a random baseline classifier is a straight line at a 45-degree angle from the start of the axes.

An important shortcoming of the ROC curve is that it is not an optimal choice when working with a highly imbalanced dataset. The reason is that since it plots the False Positive Rate in the X axis it can produce overly optimistic results because the denominator of FPR contains the True Negative classified samples (which is the majority class). This results in the False positive rate remaining very small and starting high in the Y axis which is misleading. (Movahedi et al., 2020)

- **Precision-Recall Curve (PR Curve):**

The PR curve is created by plotting the Precision of a classifier on the Y-axis and Recall of a classifier on the X-axis for all possible thresholds. The performance of a classifier similarly to ROC is obtained from the AUC (area under curve) of the PR curve, as a perfect classifier would have AUC equal to 1.

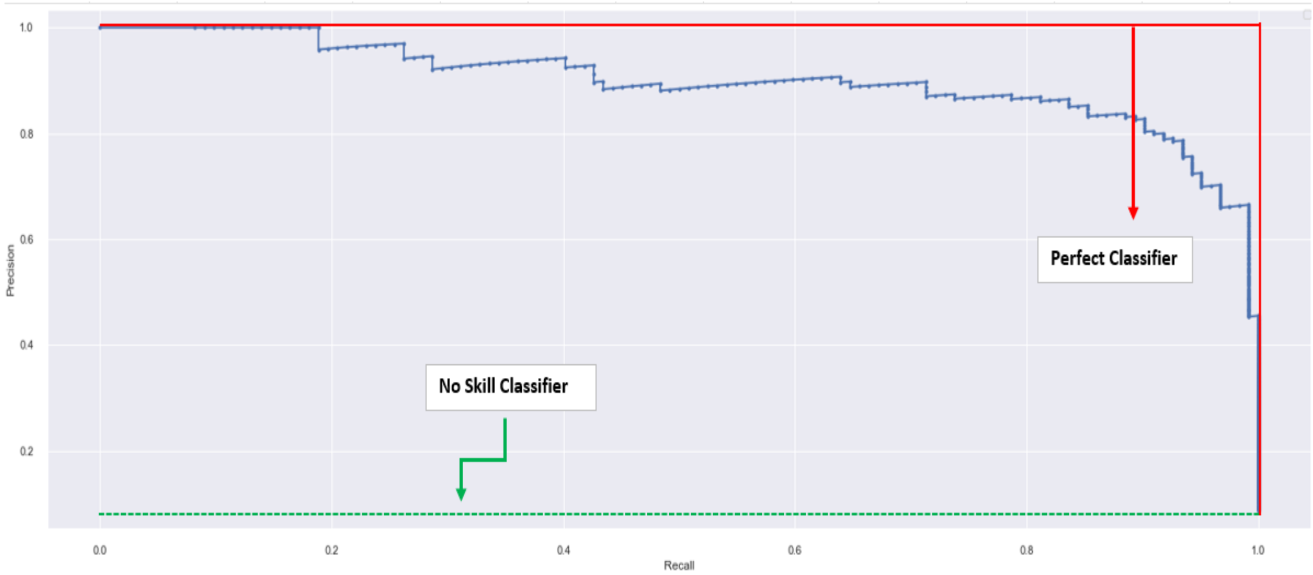


Figure 24: Illustration of PR-Curve

4. Results

In this chapter the results of the three trained algorithms that were covered in section (3.7) are presented.

4.1 Logistic Regression Results

In the training phase when performing 5-fold stratified cross validation, logistic regression showed stable performance, with similar scores in each fold, indicating stable generalization ability.

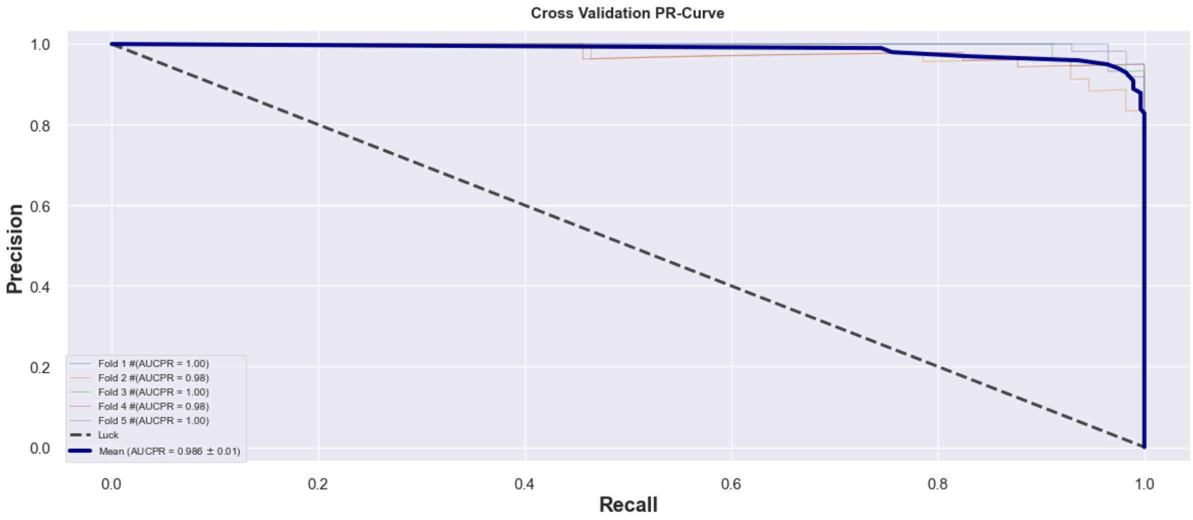
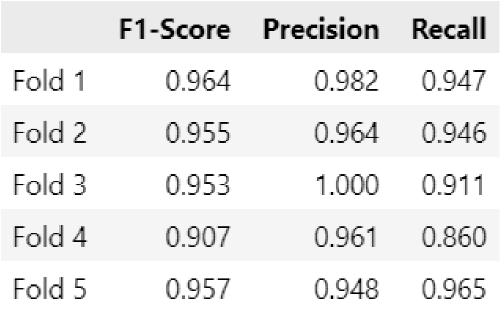


Figure 25: Precision-Recall Curve for Cross Validation of Logistic Regression



	F1-Score	Precision	Recall
Fold 1	0.964	0.982	0.947
Fold 2	0.955	0.964	0.946
Fold 3	0.953	1.000	0.911
Fold 4	0.907	0.961	0.860
Fold 5	0.957	0.948	0.965

Table 9: Cross Validation Scores of Logistic Regression

Testing Results of Logistic Regression

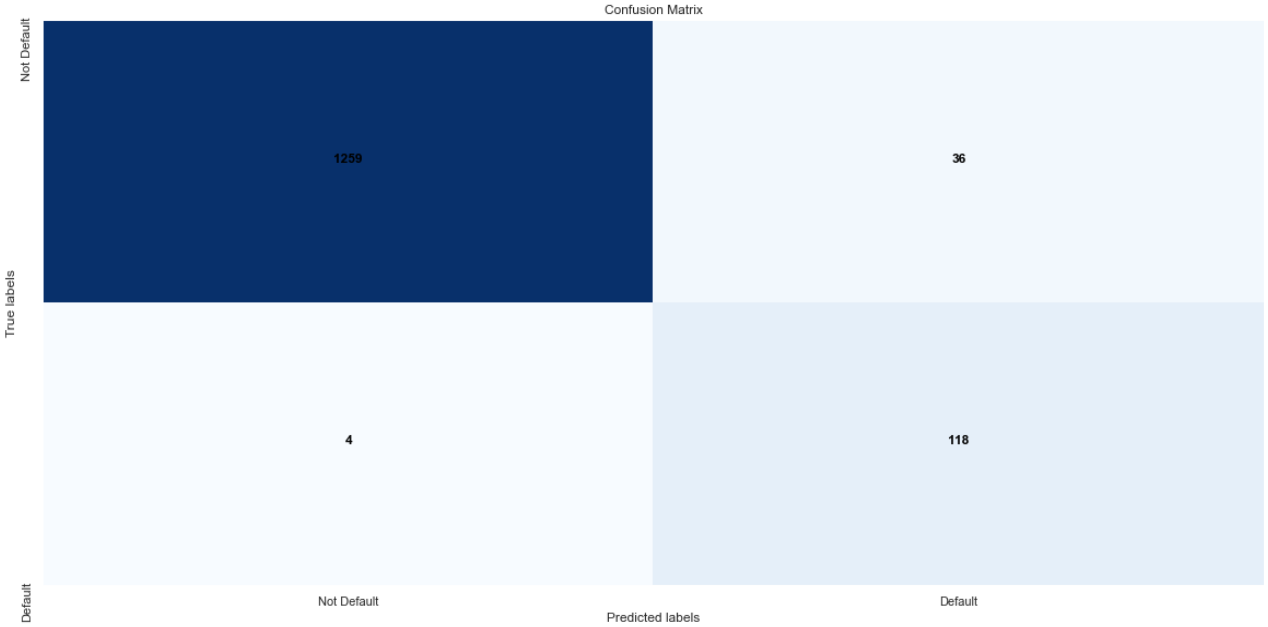


Figure 26: Confusion Matrix of Logistic Regression

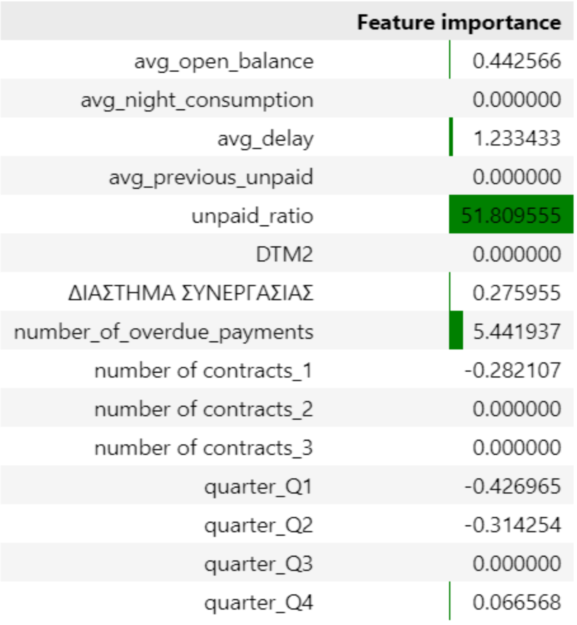
Classification Report: Logistic Regression

	Precision	Recall	F1-Score	Support
0	1.00	0.97	0.98	1295
1	0.77	0.97	0.86	122
accuracy			0.97	1417
macro avg	0.88	0.97	0.92	1417
weigted avg	0.98	0.97	0.97	1417

Table 10: Classification Report of Logistic Regression

We can see from the classification report and the confusion matrix that logistic regression performed decently with an F1-Score of 86%. The model showed high recall of 97% and a precision of 77%, meaning that it has a tendency of classifying Non-Defaulters as Defaulters (high False Positive Rate). The relationship between the resulting Precision and Recall is discussed further on at (4.5).

Lastly the feature importance was obtained to identify which were the key features that logistic regression utilized for its predictions. The coefficient importance is defined as the absolute value of the coefficients of the independent variables. Since we used the l1- regularization logistic regression, the coefficients of the least important features were shrunk to zero.



	Feature importance
avg_open_balance	0.442566
avg_night_consumption	0.000000
avg_delay	1.233433
avg_previous_unpaid	0.000000
unpaid_ratio	51.809555
DTM2	0.000000
ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ	0.275955
number_of_overdue_payments	5.441937
number of contracts_1	-0.282107
number of contracts_2	0.000000
number of contracts_3	0.000000
quarter_Q1	-0.426965
quarter_Q2	-0.314254
quarter_Q3	0.000000
quarter_Q4	0.066568

Table 11: Feature importance of Logistic Regression

The feature that held the most predictive power was unpaid_ratio , but other attributes , namely number_of_overdue_payments, ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ and avg_delay were also important to the model. These results are not unexpected since it was highly anticipated that the most significant characteristics of defaulters would be their performance as customers (unpaid_ratio) , their diligence (number of overdue payments) and their length of cooperation (ΔΙΑΣΤΗΜΑ ΣΥΝΕΡΓΑΣΙΑΣ).

4.2 Random Forest Results

In the training phase when performing 5-fold stratified cross validation, random forest showed excellent performance, with similar scores in each fold, indicating stable generalization ability.

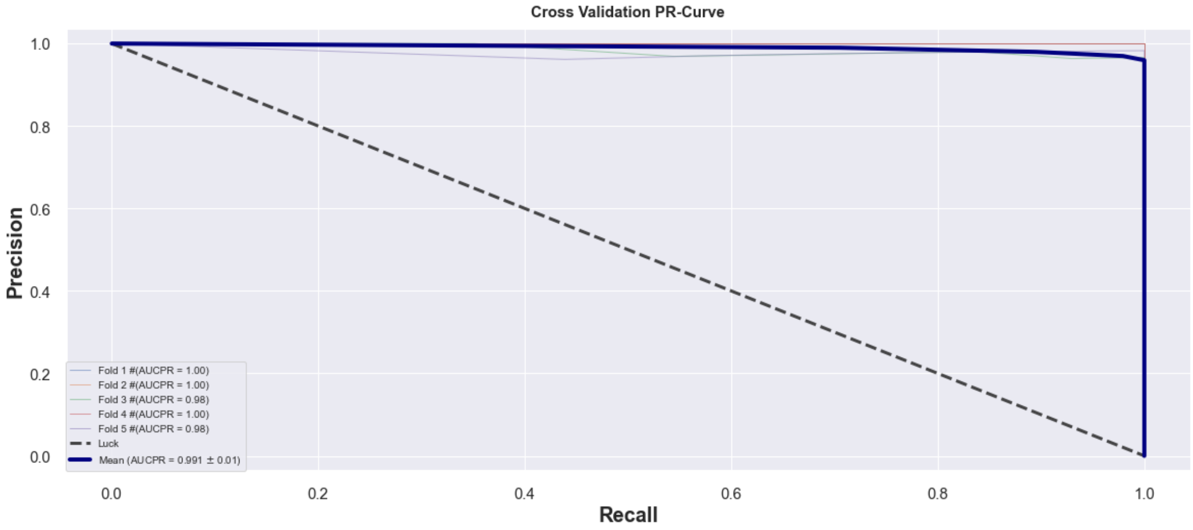
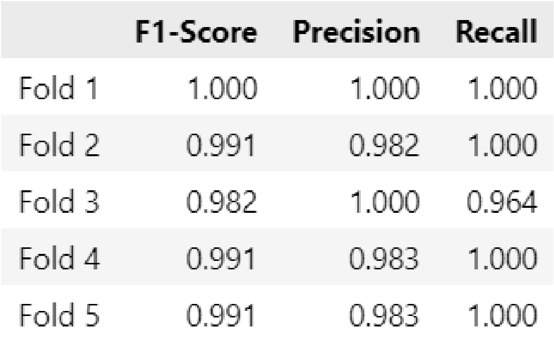


Figure 27: Precision-Recall Curve for Cross Validation of Random Forest



	F1-Score	Precision	Recall
Fold 1	1.000	1.000	1.000
Fold 2	0.991	0.982	1.000
Fold 3	0.982	1.000	0.964
Fold 4	0.991	0.983	1.000
Fold 5	0.991	0.983	1.000

Table 12: Cross Validation Scores of Random Forest

Testing Results of Random Forest

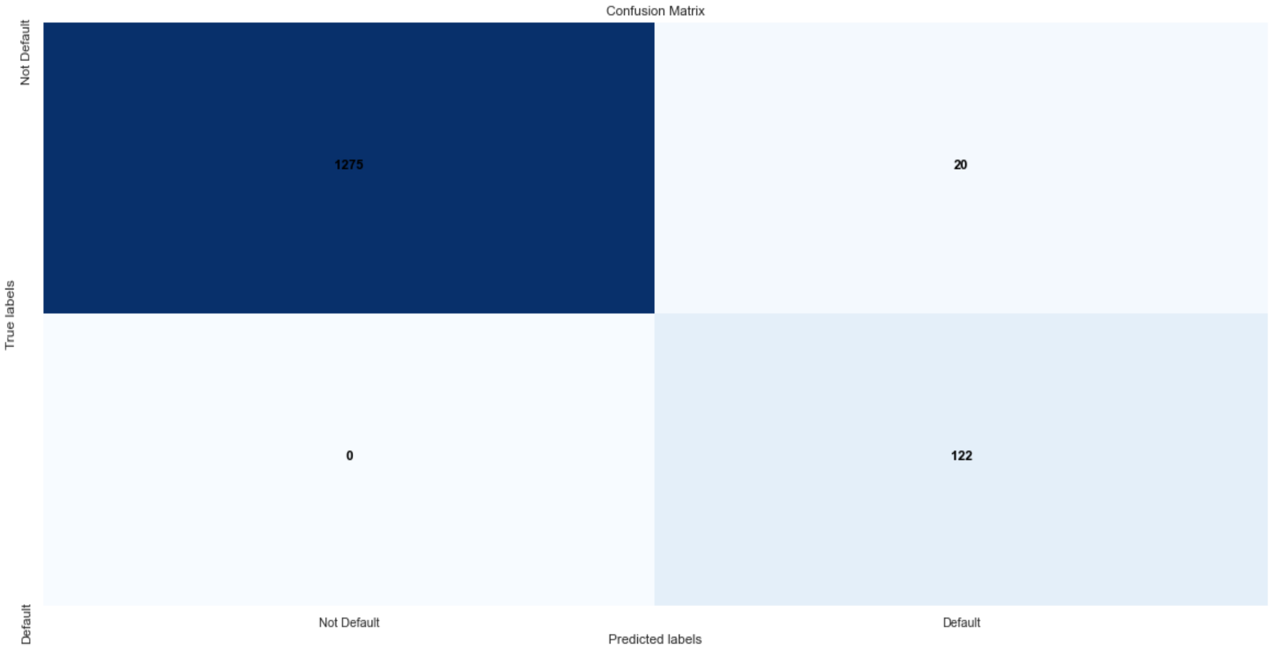


Figure 28: Confusion Matrix of Random Forest

Classification Report: Random Forest

	Precision	**Recall**	**F1-Score**	**Support**
0	1.00	0.98	0.99	1295
1	0.86	1.00	0.92	122
accuracy			0.99	1417
macro avg	0.93	0.99	0.96	1417
weigted avg	0.99	0.99	0.99	1417

Table 13: Classification Report of Random Forest

Random Forest showed excellent performance even on the test set, with a slight tendency to label normal customers as defaulters. In the testing phase the model correctly classified all defaulters in class 1, but mistakenly labelled 20 normal customers as defaults, achieving an F1-Score for class 1 of 92%.

Lastly feature importance was extracted from the model to identify the key attributes that held the most predictive power for Random Forest.

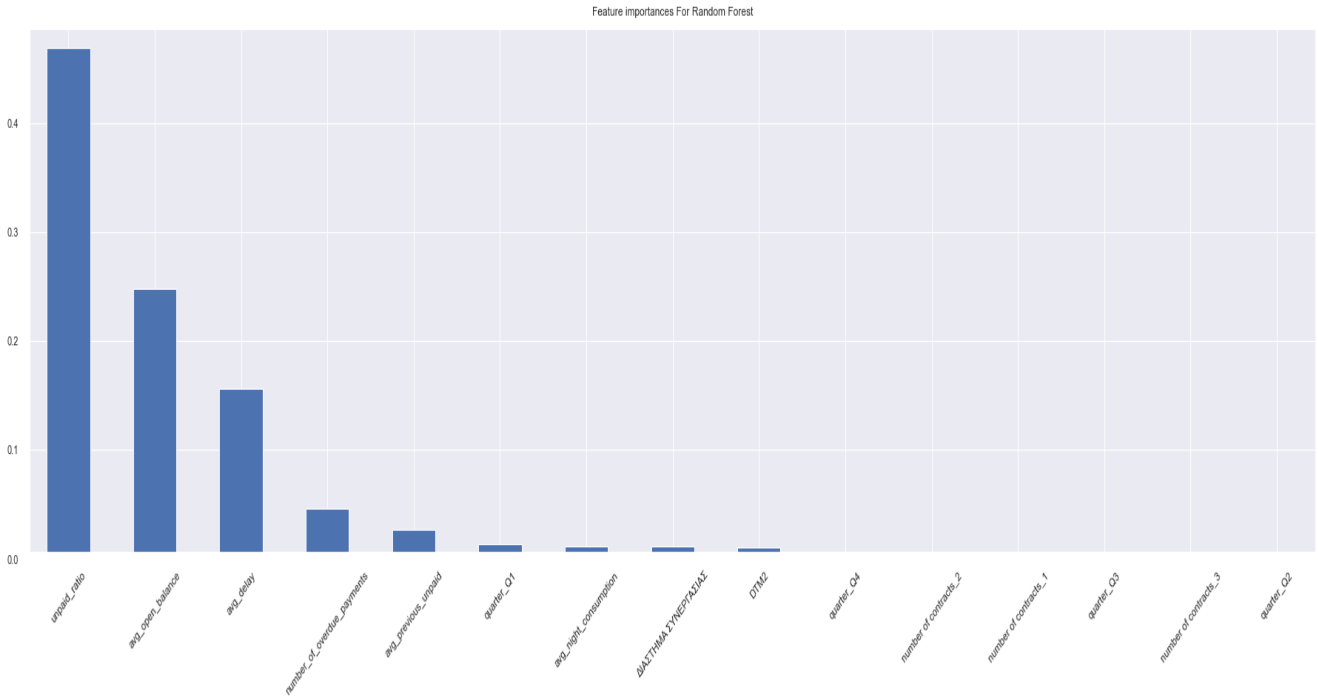


Figure 29: Feature importance of Random Forest

Much like Logistic Regression, Random Forest placed most importance in the unpaid_ratio attribute. Unlike Logistic Regression however, random forest also placed a significant portion of importance in the attribute avg_open_balance, meaning that the algorithm deemed very important on whether a customer tends to leave bills unpaid or not. The rest of the attribute importance was similar to both models.

4.3 XGBoost Results

Training results for 5-fold stratified cross validation which are shown in tables number, xgboost showed excellent performance, with similar scores in each fold, indicating stable generalization ability.

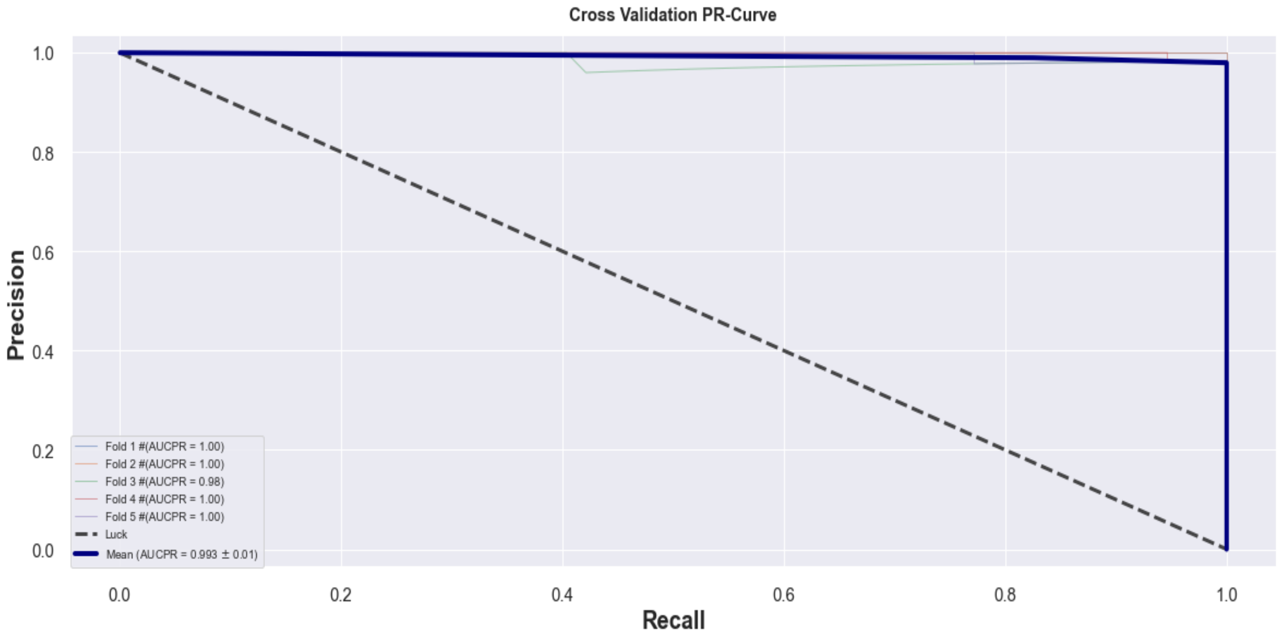
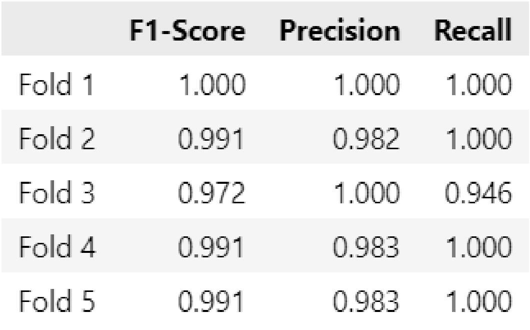


Figure 30: Precision-Recall Curve for Cross Validation of XGBoost



	F1-Score	Precision	Recall
Fold 1	1.000	1.000	1.000
Fold 2	0.991	0.982	1.000
Fold 3	0.972	1.000	0.946
Fold 4	0.991	0.983	1.000
Fold 5	0.991	0.983	1.000

Table 14: Cross Validation Scores of XGBoost

Testing results of XGBoost

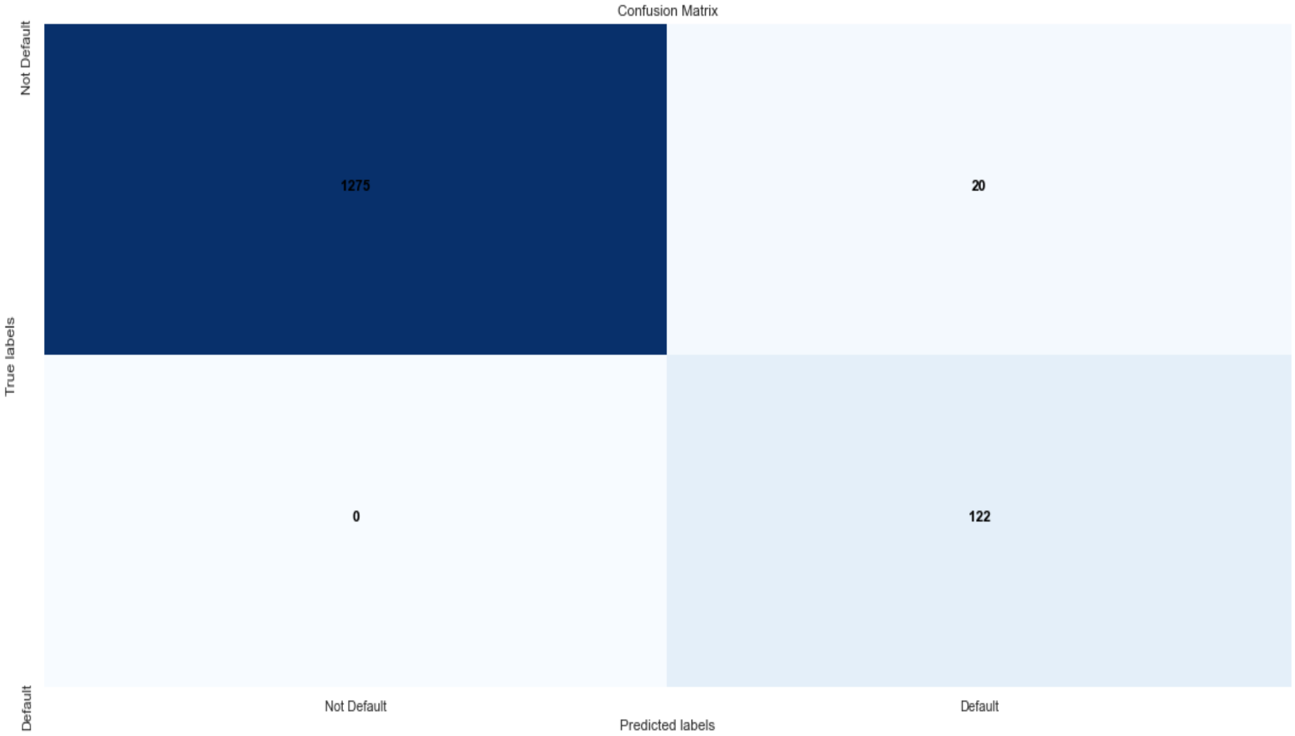


Figure 31: Confusion Matrix of XGBoost

Classification Report: XGBoost

	Precision	**Recall**	**F1-Score**	**Support**
0	1.00	0.98	0.99	1295
1	0.86	1.00	0.92	122
accuracy			0.99	1417
macro avg	0.93	0.99	0.96	1417
weigted avg	0.99	0.99	0.99	1417

Table 15: Classification Report of XGBoost

XGBoost showed excellent performance, identical to Random Forest, which is not surprising given that the two models share many similarities. On the test set the model had a slight tendency to label normal customers as defaulters. In the testing phase the model correctly classified all defaulters in class 1, but mislabelled 20 normal customers as defaults, scoring an F1 for class 1 of 92%.

Feature importance was extracted from the model to identify the key attributes that held the most predictive power for the model.

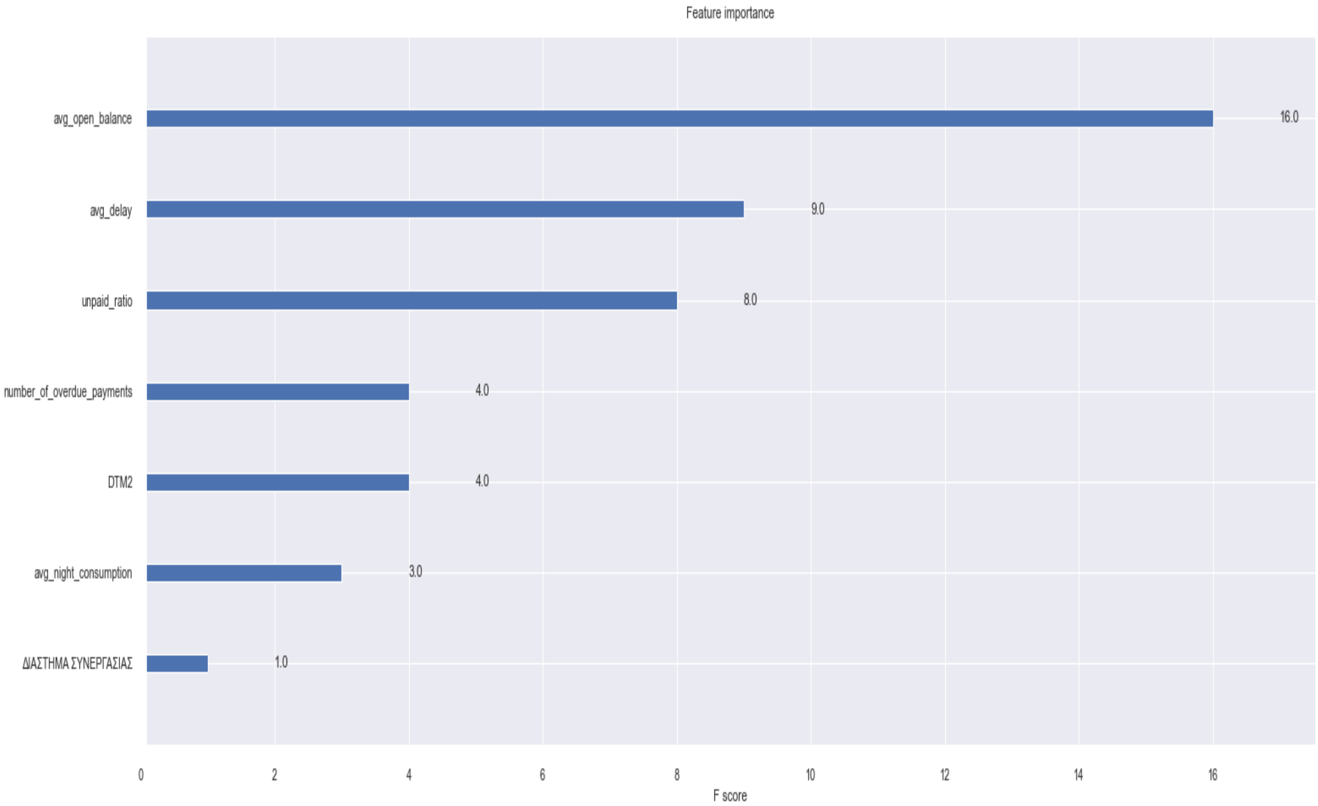


Figure 32: Feature importance of Random Forest

Feature importance for XGBoost produced interesting results. While all the important predictive features were almost the same as the other two models, XGBoost placed most significance in the avg_open_balance feature, meaning that it deemed most important if a customer frequently missed on bill payments, which is a logical assumption.

4.4 Classification Probabilities

After training and testing the three models the probabilities of default were extracted from each model. Figures 33,34 and 35 display the probability distributions of the extracted probabilities for each class for each of the three models. When inspecting the distributions of the positive class and the corresponding mean probabilities it becomes apparent that the random forest and XGBoost are more ‘confident’ in their predictions in comparison to logistic regression, which shows a slightly more spread-out distribution than the aforementioned models.
Tables 16,17,18 contain the mean probabilities that a defaulter and a non- defaulter might belong to either class 1 or 0.

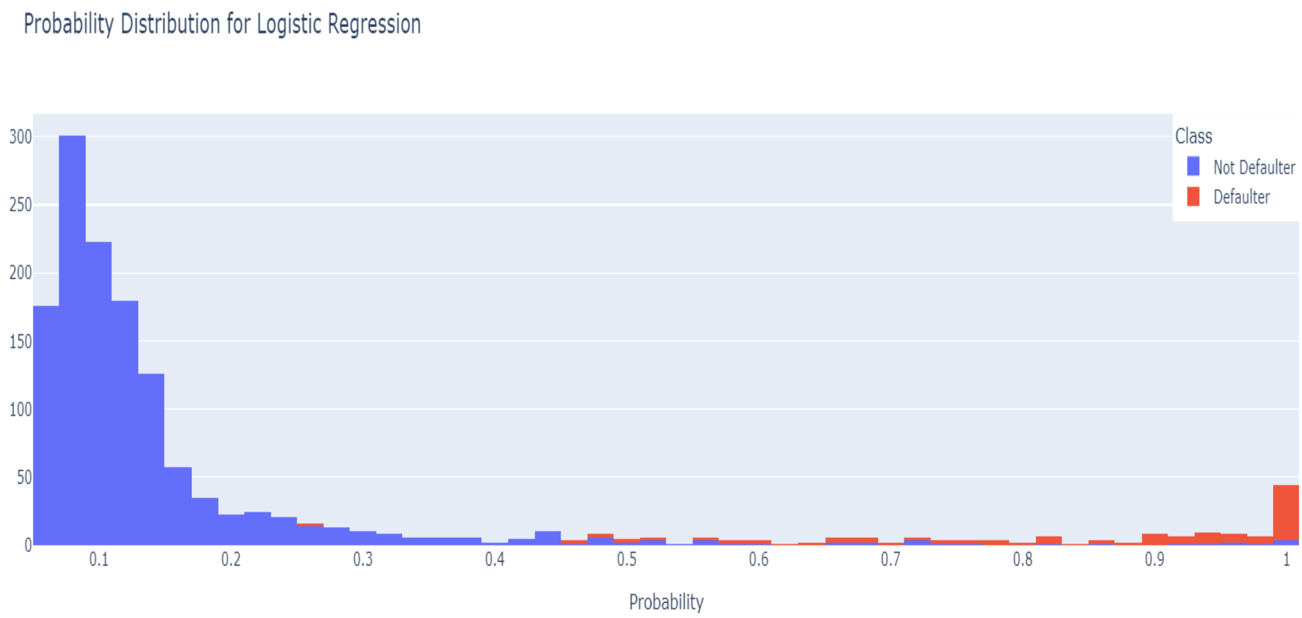


Figure 33: Probability distribution for Logistic Regression

Mean Probabilities: Logistic Regression		
	Class 0 - Not Default	**Class 1 - Default**
Defaulter	0.132	0.867
Not Defaulter	0.856	0.143

Table 16: Transition matrix of Logistic Regression

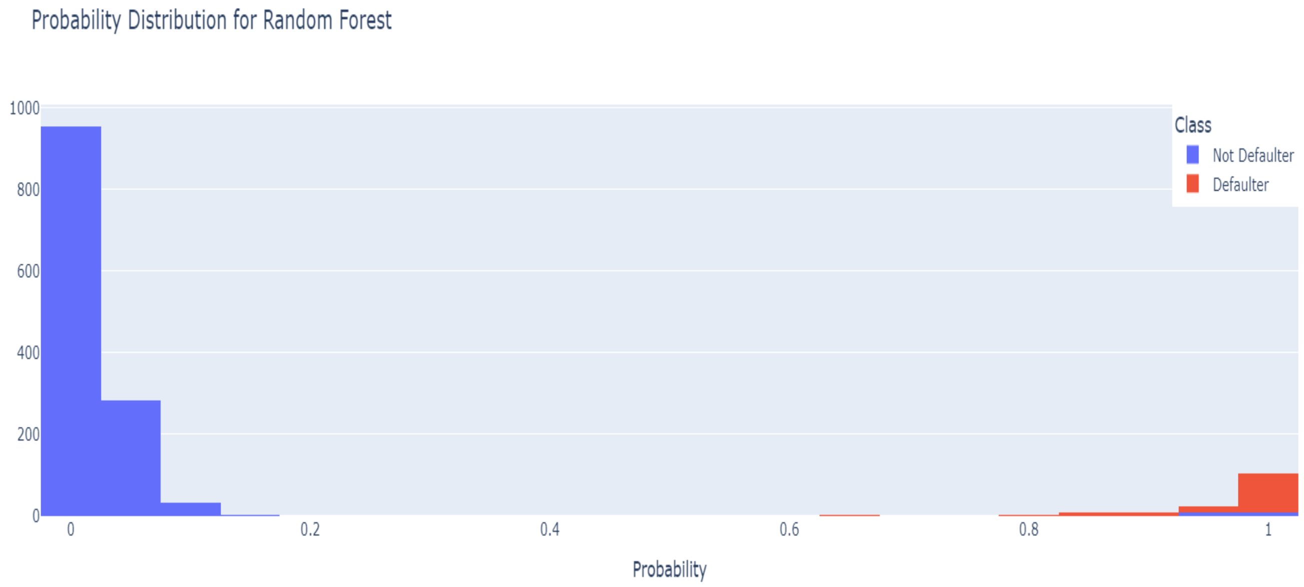


Figure 34: Probability distribution for Random Forest

Mean Probabilities: Random Forest

	Class 0 - Not Default	Class 1 - Default
Defaulter	0.023	0.976
Not Defaulter	0.969	0.030

Table 17: Transition matrix of Random Forest

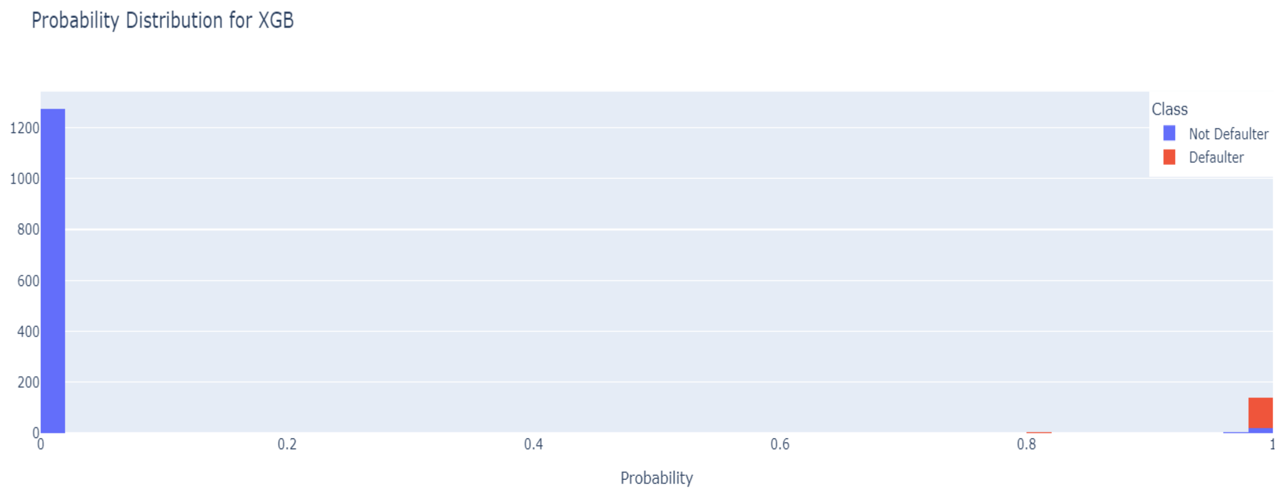


Figure 35: Probability distribution of Positive class for XGBoost

Mean Probabilities: XGBoost

	Class 0 - Not Default	**Class 1 - Default**
Defaulter	0.004	0.995
Not Defaulter	0.981	0.018

Table 18: Transition matrix of XGBoost

4.5 Model Comparison – Overall Results

Model	Class	Metrics		
		Precision	Recall	F1
Logistic Regression	Class 0 - Not Default	100%	97%	98%
	Class 1 -Default	77%	97%	86%
	Macro Average	88%	97%	92%
Random Forest	Class 0 - Not Default	100%	98%	99%
	Class 1 -Default	86%	100%	92%
	Macro Average	93%	99%	96%
XGBoost	Class 0 - Not Default	100%	98%	99%
	Class 1 -Default	86%	100%	92%
	Macro Average	93%	99%	96%

Table 19: Overall Results of all models

As it was mentioned in (3.7.1) all three of the implemented models are probabilistic classifiers, meaning that they classify data to a class, based on an adjustable threshold, that by default is set at a predetermined value of 50%. An examination of the Precision Recall diagrams in figures 36,37 and 38, gives us a better understanding of the optimal threshold for the models.

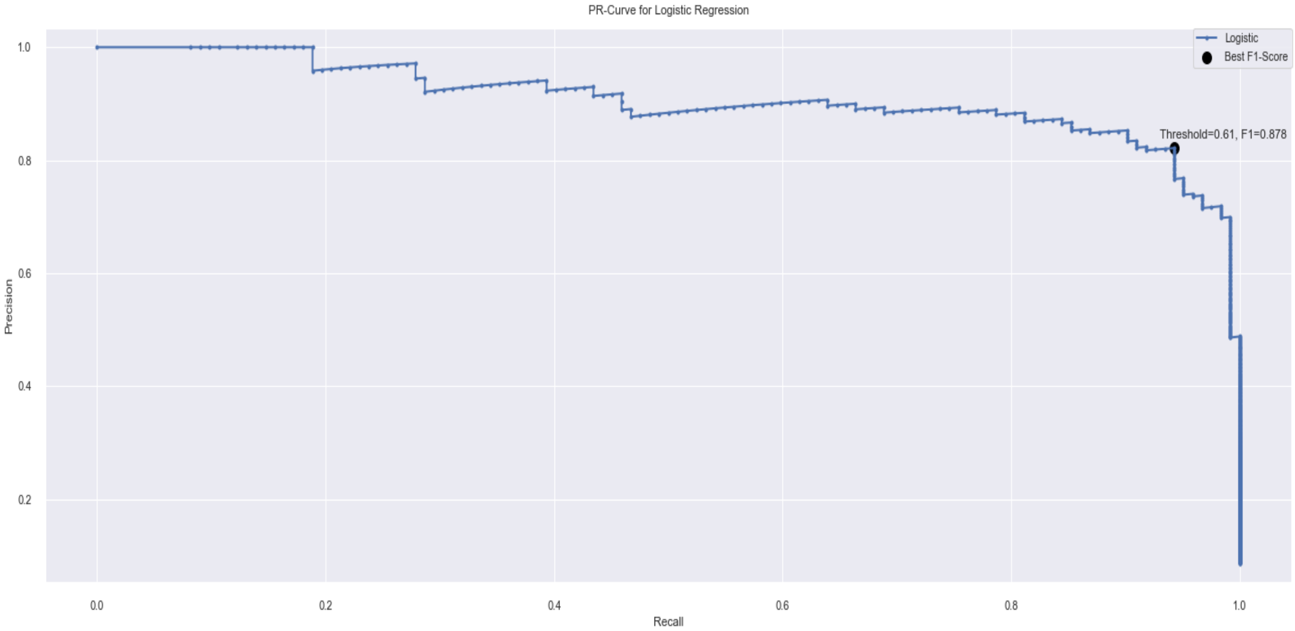


Figure 36: Precision-Recall Curve for Logistic Regression

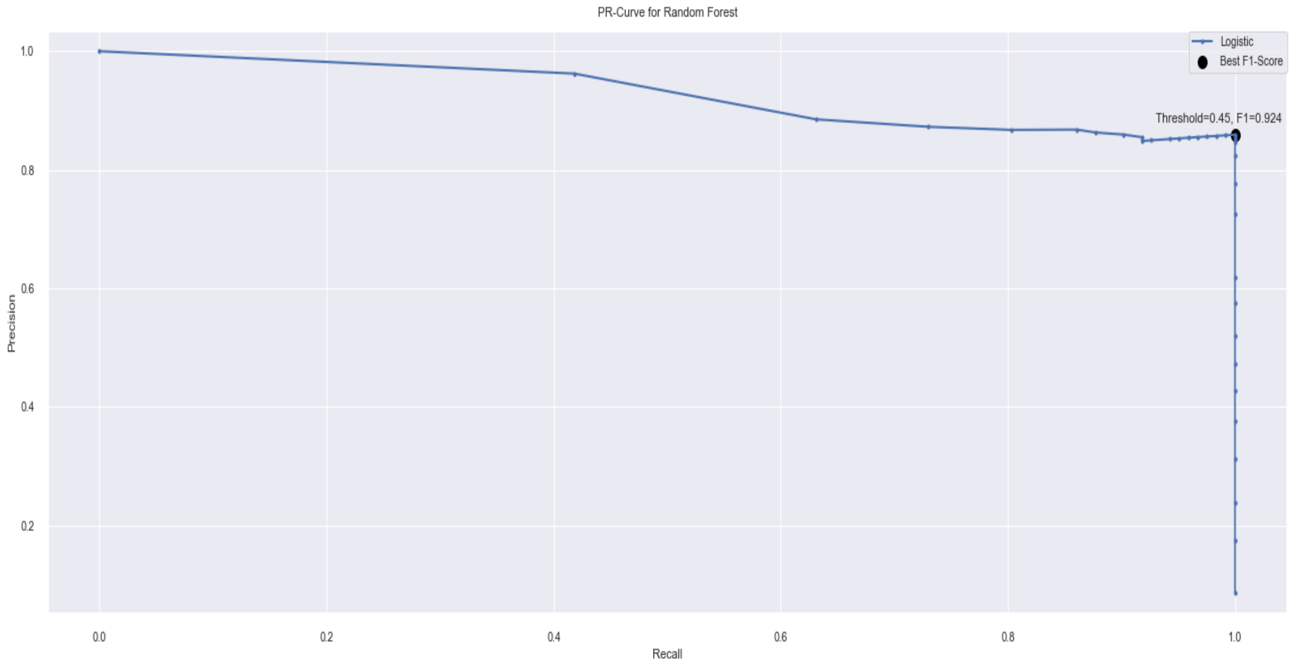


Figure 37: Precision-Recall Curve for Random Forest

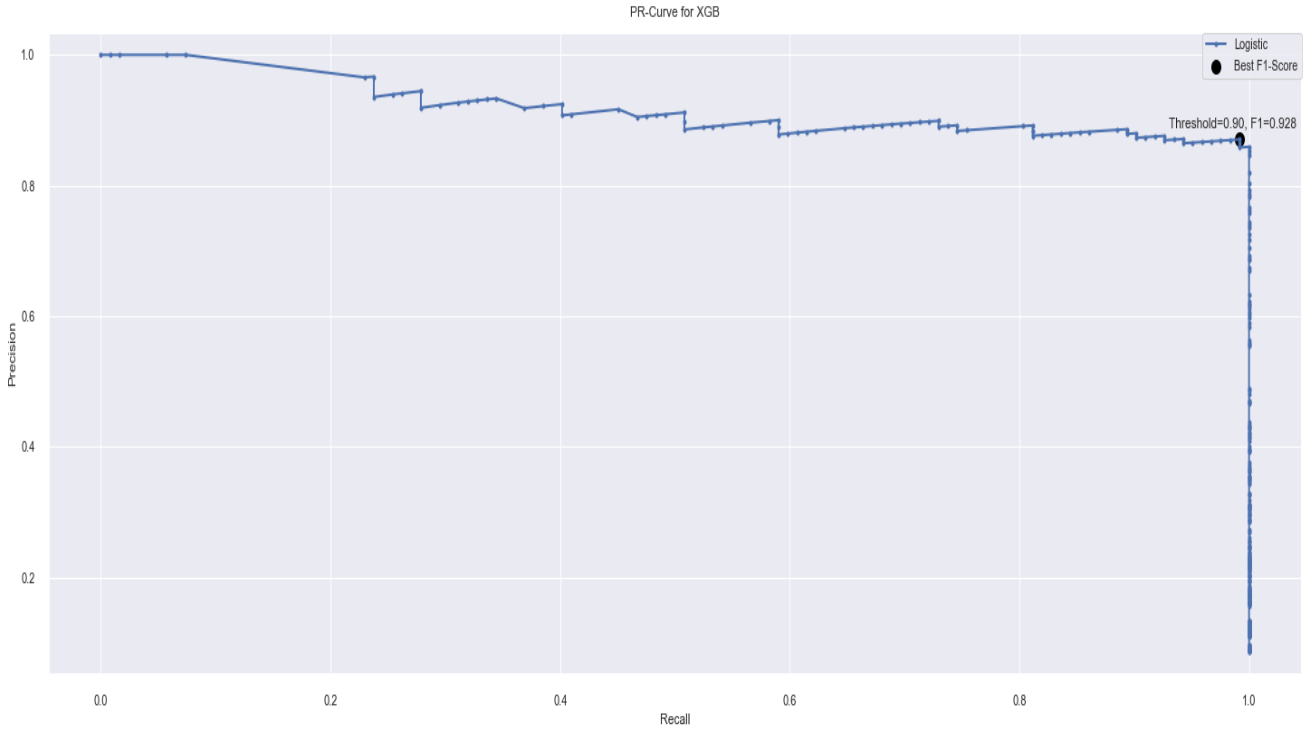


Figure 38: Precision-Recall Curve for XGBoost

As it was stated in (3.8) a perfect classifier would produce both precision and recall equal to 1. Diagrams 36,37 and 38 display the trade-off between these two metrics, and to be more specific, the trade-off between classifying correctly detecting all defaulters (but at the same time misclassifying

some normal customer as defaulters) and correctly classifying all normal customers but letting some defaults go undetected.
Taking a closer look at the Logistic Regression PR-Curve, based on diagram 36, if we were to maximize recall, then precision would be quite low, roughly at 45-50%, which would translate in a rather aggressive approach since many non-defaulters would be flagged (because the number of false positive classifications would be maximized). In figures 36,37,38 the optimal threshold, which is represented by the black dot, is defined as the threshold that maximizes the F1 score, meaning that it gives the best balance between precision and recall. As it is evident from the diagrams the optimal threshold for Logistic Regression is 61% with an F1 score of 87.8%, for Random Forest is 45% with and F1 of 92.4% and lastly for XGB the optimal threshold is 90% with an F1 of 92.8%.

However, this threshold is best determined based on the business needs, and whether or not the business wants to pursue a ‘detect defaults at all costs’ approach, even if that means penalizing regular customers at the same time.
On the other hand, Random Forest and XGBoost displayed more potent predictive capabilities. Taking into consideration the aforementioned business dilemma of setting or not a stricter classification threshold, these two models provide greater flexibility than Logistic Regression, since they have better Precision-Recall ratios.

One interesting observation is produced when inspecting the classification reports of Random Forest and XGBoost as well as their PR-Curves. At first glance both models scored exactly the same results but nevertheless have slightly different PR-Curves. This is a result of the threshold that was previously discussed, because the classification reports concern only a specific threshold. When comparing the two curves while both models achieve the same precision for recall 1, XGBoost indicates greater volatility from Random Forest, which leads us to assume that Random Forest is overall a slightly more stable performer. This is not an unexpected occurrence, however. In section (3.4) we mentioned that there we purposefully left some outliers in some features, because they contained important information that should be included in the model. In section 3.7.3. we mentioned that while XBG has great predictive power, one downside is that it shows a greater sensitivity to outliers than random forest which is robust when faced with extreme feature values. Therefore, this trait most likely accounts for XGB’s more volatile results.

In order to make the results of all three models more comparable in a wider range, the area under the curve was used as a metric to evaluate the overall performance of said models. Results showed that Logistic regression and XGB achieved similar performance in terms of AUC but Random Forest achieved the best out of the 3 models.

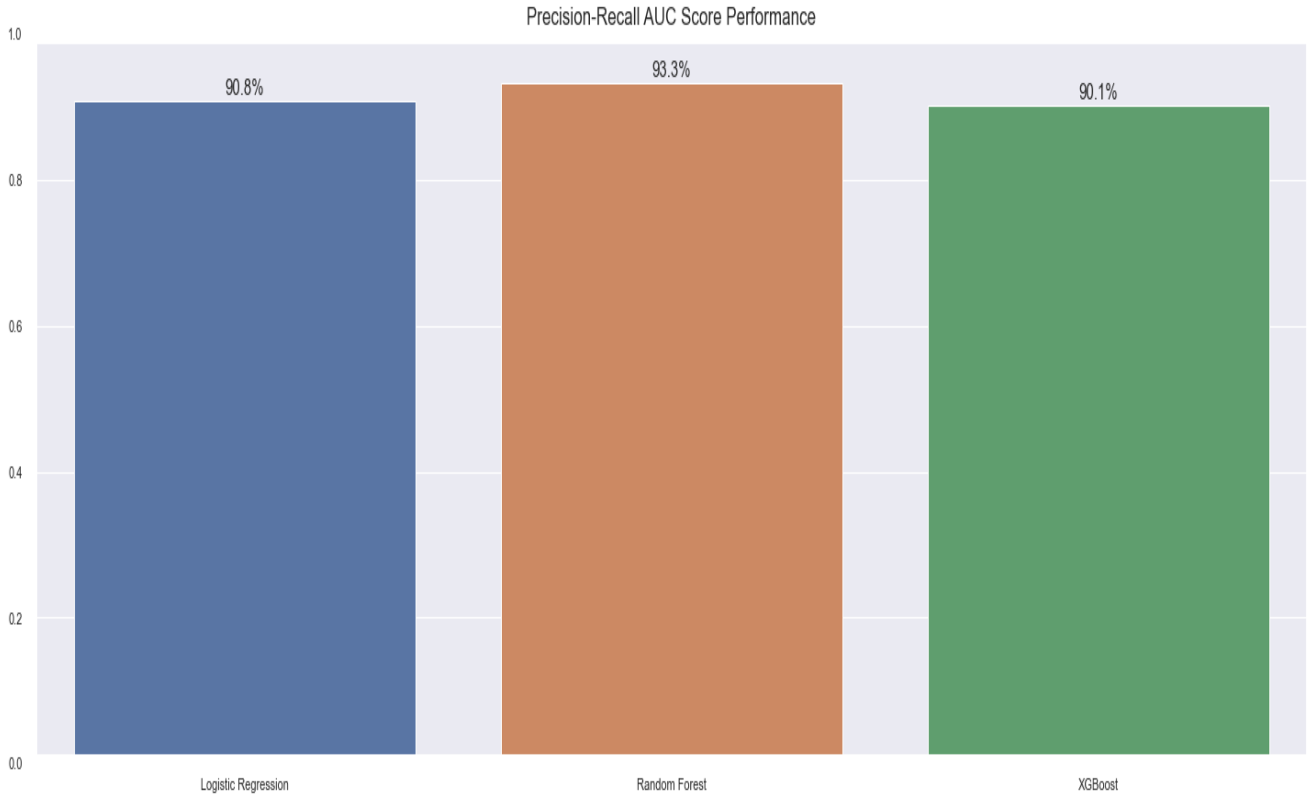


Figure 39: PR-AUC performance for all three models

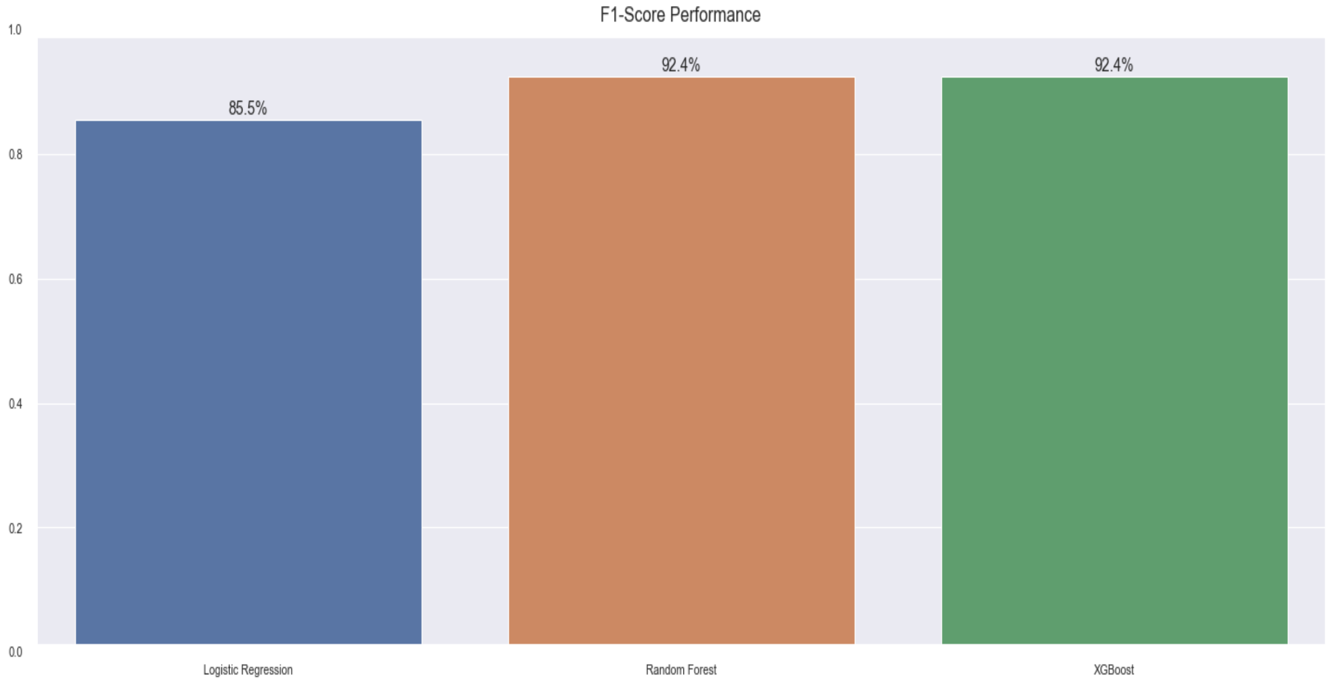


Figure 40: F1-Score for all three models

5. Conclusions

This study covered the prediction of customer default in the energy industry. More specific the goal was to create a model that would predict the probability of a customer defaulting on their next month's payments. Using a dataset that contained over 47 attributes and 120k entries, that was provided from an electricity provider operating in Greece, we performed exploratory data analysis and successfully uncovered useful insights regarding customer behaviour, cleaned the data, engineered the features for optimal model performance and trained three different models, namely Logistic Regression, Random Forest and Extreme Gradient Boosting. Given that it was an imbalanced classification problem, the models were evaluated using the F1-Score and compared using the Area Under the Precision-Recall Curve. Results showed that all three models achieved very satisfying performances with Random Forest gaining a slightly better score than the other two models with an AUC of 93.3%. Overall, the methodologies and implemented models followed along standard practices that have been widely utilized in similar industries such as the banking industry.

Despite the fact that in this study the implemented models showed significant performance capabilities, there are certainly areas for expansion and improvement in the predictive modelling as well as the statistical analysis. For future work, it is highly recommended to include a larger dataset with more unique customers. The inclusion of demographic characteristics as well as salary range information and employment status, has proven to provide significant predictive information and should help in the generalization of this model. Another way to expand the model would be the implementation of customer scoring, which is commonly used in similar projects in financial environments, but typically require a greater number and variety of features.

Lastly looking at this study from a business perspective, the model could be further tailored to the specific rules of the company. In this work, we made the simplification that if a customer missed two consecutive payments, then he is considered as default, but in reality, there are more regulations underlying this issue that could be further looked into. Another area where the implementation of more specific business rules would make an impact on the model, is the definition of the probability threshold of the classification. As it was discussed in (4.4.), it ultimately falls to the business to define which is more important, identifying more high-risk clients and simultaneously penalizing some normal customers or lowering the threshold and take on more risky clients in order to not penalize regular customers.

In conclusion, while this project's main objectives have been accomplished, there is further room for expanding this model in all areas, should it be needed to be implemented in production.

References

al-Qerem, A. (2020). *Default Prediction Model: The significant Role of Data Engineering in the Quality of Outcomes.* Retrieved from DOI : https://iajit.org/PDF/Special%20Issue%202020,%20No.%204A/19489.pdf

Aymane, H. (2022). *XGBoost: Everything You Need to Know*. Retrieved from https://neptune.ai/blog/xgboost-everything-you-need-to-know

Bandyopadhyay, A. (2016, 05). *cambridge.org*. Retrieved from https://www.cambridge.org/core/books/abs/managing-portfolio-credit-risk-in-banks/introduction-to-credit-risk/EF340492026F7127FDBF083BC8196AFA

Brownlee, J. (2020). *Random Oversampling and Undersampling for Imbalanced Classification*. Retrieved from https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/

Brownlee, J. (2020). *ROC Curves and Precision-Recall Curves for Imbalanced Classification*. Retrieved from https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/

Brownlee, J. (2020). *Undersampling Algorithms for Imbalanced Classification*. Retrieved from https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/

Chen, J. (2022, 09). *investopedia.com*. Retrieved from Default: What It Means, What Happens When You Default, Examples: https://www.investopedia.com/terms/d/default2.asp

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Retrieved from https://dl.acm.org/doi/pdf/10.1145/2939672.2939785

Dietterich, T. (2000). Retrieved from An Experimental Comparison of Three Methods: https://web.engr.oregonstate.edu/~tgd/publications/mlj-randomized-c4.pdf

Floudopoulos, H. (2020, 01). *Καθολική υπηρεσία: το άγνωστο 'bad debt' της αγοράς ρεύματος*. Retrieved from capital.gr.

IEA. (2020). *iea.org*. Retrieved from https://www.iea.org/reports/global-energy-review-2020/electricity

Jurafsky, D., & Martin, J. H. (2021). Retrieved from Chapter 5 : Logistic Regression: https://web.stanford.edu/~jurafsky/slp3/5.pdf

Kui, W. (2022). *A Hybrid Algorithm-Level Ensemble Model for Imbalanced Credit Default Prediction in the Energy Industry*. Retrieved from DOI: https://doi.org/10.3390/en15145206

Liaggou, X. (2022, 05). *kathimerini.gr*. Retrieved from https://www.kathimerini.gr/economy/561882088/vomva-gia-tin-agora-reymatos-i-katholiki-ypiresia/

Liberman, N. (2017). *Decision Trees and Random Forests*. Retrieved from https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991

mastersindatascience.org. (n.d.). *Undersampling vs. Oversampling for Imbalanced Datasets*. Retrieved from https://www.mastersindatascience.org/learning/statistics-data-science/undersampling/#:~:text=Undersampling%20is%20a%20technique%20to,information%20from%20originally%20imbalanced%20datasets.

Movahedi, F., Rema, P., & Antaki, J. (2020). Retrieved from Limitations of ROC on Imbalanced Data::

https://arxiv.org/pdf/2010.16253.pdf#:~:text=Conclusion%3A%20The%20ROC%20can%20portray,focusing%20on%20the%20minority%20class.

O'Reilly. (n.d.). Retrieved from The min-max scaling method: https://www.oreilly.com/library/view/feature-engineering-made/9781787287600/aa5580ee-6fb7-4ac2-a1fe-369d95b70168.xhtml

Rising, O. (n.d.). *Predicting Bank Loan Default with Extreme Gradient Boosting.* Retrieved from https://arxiv.org/ftp/arxiv/papers/2002/2002.02011.pdf

Shi, J., Yin, W., Stanley, O., & Sajda, P. (2010). Retrieved from A Fast Hybrid Algorithm for Large-Scale ℓ1-Regularized: https://jmlr.csail.mit.edu/papers/volume11/shi10a/shi10a.pdf

Wijewardhana, U., Bandara, C., & Thesath , N. (2018). *A Mathematical Model for Predicting Debt Repayment: A technical Note*. Retrieved from https://ro.uow.edu.au/cgi/viewcontent.cgi?article=1904&context=aabfj

Yiu, T. (2019). *Understanding Random Forest*. Retrieved from https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Zhou, X. (2020). *towardsdatascience.com*. Retrieved from Loan Default Prediction with Berka Dataset: https://towardsdatascience.com/loan-default-prediction-an-end-to-end-ml-project-with-real-bank-data-part-1-1405f7aecb9e

Zhu, L., Dafeng, Q., Daji, E., Cai, Y., & Kuiyi, L. (2019). *A study on predicting loan default based on the random forest algorithm.* Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050919320277