Contents

Introduction

This report focuses on predicting churn in a given dataset. Churn refers to customers discontinuing a product or service. This issue has significant implications for businesses, leading to revenue loss and increased costs. The objective is to develop accurate predictive models using machine learning techniques to identify potential churn cases. By doing so, businesses can take proactive measures to retain valuable customers. The report evaluates various algorithms and performance metrics to determine the best model for churn. The insights gained will drive data-driven decision-making and improve customer retention strategies.

Problem Overview

Model Selection

To tackle the prediction problem of customer churn, we consider various machine learning models. These models can be broadly categorized into classification models, such as Logistic Regression, Random Forest Classifier, Support Vector Machines, and Gradient Boosting Classifier, which aim to classify customers as churned or not based on their features.

Model Considerations

In addressing the prediction problem of customer churn, we explore a range of machine learning models that are suitable for classification tasks. Each model offers unique advantages and considerations based on the nature of the data and the problem at hand.

- **Logistic Regression**

Logistic Regression is a popular choice for binary classification problems. It models the relationship between input features and the probability of customer churn. Logistic Regression provides interpretable coefficients that indicate the impact of each feature on the likelihood of churn.

- **Random Forest Classifier**

Random Forest Classifier is an ensemble learning technique that combines multiple decision trees to make predictions. It can handle non-linear relationships, capture complex interactions among features, and provide feature importance rankings. Random Forest Classifier is known for its robustness and ability to handle high-dimensional data.

- **Support Vector Machines (SVM)**

Support Vector Machines is a powerful classification algorithm that finds the optimal hyperplane to separate churned and non-churned customers. It can handle both linear and non-linear decision boundaries, making it suitable for complex datasets. SVM is effective in handling high-dimensional feature spaces and offers good generalization capabilities.

- **Gradient Boosting Classifier**

Gradient Boosting Classifier is an ensemble method that builds a strong predictive model by combining multiple weak learners. It sequentially trains decision trees, each focusing on correcting the mistakes of the previous trees. Gradient Boosting Classifier is known for its high predictive performance and ability to handle complex relationships.

Challenges and Limitations

While developing a predictive model for customer churn, several challenges and limitations may arise:

a. Imbalanced Data: The dataset may exhibit class imbalance, where the number of churned customers is significantly smaller than non-churned customers. This can affect the model's ability to accurately predict churn.

b. Feature Selection: Selecting the most relevant features from the available data is crucial. Feature engineering techniques and domain expertise can help identify the most influential predictors of churn.

c. Model Evaluation: Evaluating the performance of the predictive models requires careful consideration of appropriate evaluation metrics such as accuracy, precision, recall, and F1 score. Handling imbalanced data requires additional techniques such as area under the Receiver Operating Characteristic (ROC) curve or Precision-Recall curve.

Model Evaluation

To assess the performance of the predictive models, various evaluation metrics can be employed, such as accuracy, precision, recall, and F1 score. Additionally, techniques such as cross-validation and stratified sampling can provide robust estimates of the model's performance on unseen data.

In conclusion, developing an accurate and reliable predictive model for customer churn prediction requires careful consideration of model selection, feature engineering, and evaluation metrics. By addressing the challenges and limitations associated with the problem, businesses can gain insights into customer behavior and implement effective retention strategies to mitigate churn.

Data Exploration

The dataset includes the following columns (Figure 1):

- CustomerID: A unique identification number assigned to each customer.
- Geography: The region where the customer is registered.
- Gender: The gender of the customer (Male or Female).
- Age_Band: The age range to which the customer belongs.
- TenureYears: The duration, in years, since the customer's first bank account opening.
- EstimatedIncome: The estimated yearly income of the customer.
- BalanceEuros: The total financial assets (savings/deposits) of the customer.
- NoProducts: The number of total products the customer holds with the bank.
- CreditCardholder: An indicator of whether the customer is a credit cardholder.
- CustomerWithLoan: An indicator of whether the customer has taken a loan (consumer/mortgage).
- Digital_TRX_ratio: The ratio of digital transactions over physical transactions for the customer.
- Inactive (target): An indicator of customer inactivity in the last 3 months (binary).

	CustomerID	Geography	Gender	Age_Band	TenureYears	EstimatedIncome	BalanceEuros	NoProducts	CreditCardholder	CustomerWithLoan	Digital_TRX_ratio	Inactive
0	5188208	Rest_GR	Male	18-25	0	40684	50086.2	1	0	0	0.38	0
1	8683784	Thessaloniki	Female	65+	4	2429.51	0	1	1	0	0.33	1
2	3512360	Athens	Male	45-55	4	41694.5	26852.7	1	1	1	0.72	0
3	7104818	Rest_GR	Male	25-35	5	74523.3	90325.6	1	0	0	0.08	0
4	6712745	Rest_GR	Female	25-35	9	111050	100537	2	0	0	1.38	0

Figure 1: Churn dataset

During the data exploration phase, we examined the dataset for any missing values (NAs). Fortunately, our analysis revealed that there were no missing values present in the dataset. This ensures the completeness of the data and provides a solid foundation for further analysis and modelling.

We performed data exploration using pie charts (Figure 2) to gain insights into the variables. Here are some key findings:

- Churn: Approximately 1 in 5 customers experienced churn within the last 3 months.
- Gender: Males constitute around 10% more of the customer base compared to females.
- Geography: Half of the customers have registered in the region of Athens.
- Age Band: Nearly 2 in 3 customers fall within the age range of 25-45 years.
- NoProducts: Approximately 96% of the customers have 1 or 2 products.
- CreditCardholder: 7 out of 10 customers hold a credit card.
- CustomerWithLoan: Half of the customers have taken a loan.

These insights provide valuable information about the customer base, allowing us to better understand the distribution and characteristics of the dataset.

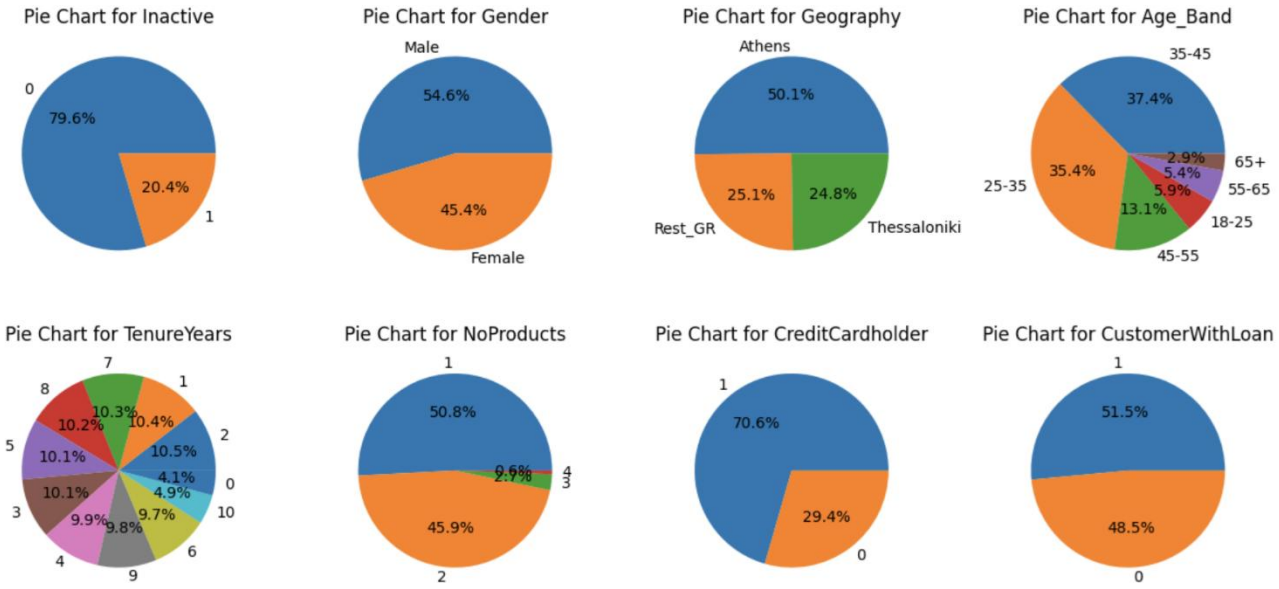

Figure 2: Pie charts regarding Churn dataset

A subplot with multiple bar charts was generated, showcasing the relationship between different categorical features and churn (Figure 3). Here are the observations:

- Gender: When the gender is female, there is a higher likelihood of churn compared to males.
- Age Group: Customers belonging to the age groups of 25-35 and 35-45 are more likely to churn.
- Geography: Customers who registered in the "Rest of Greece" region have a higher probability of churn.
- NoProducts: Customers with only one product are more likely to churn compared to those with multiple products.

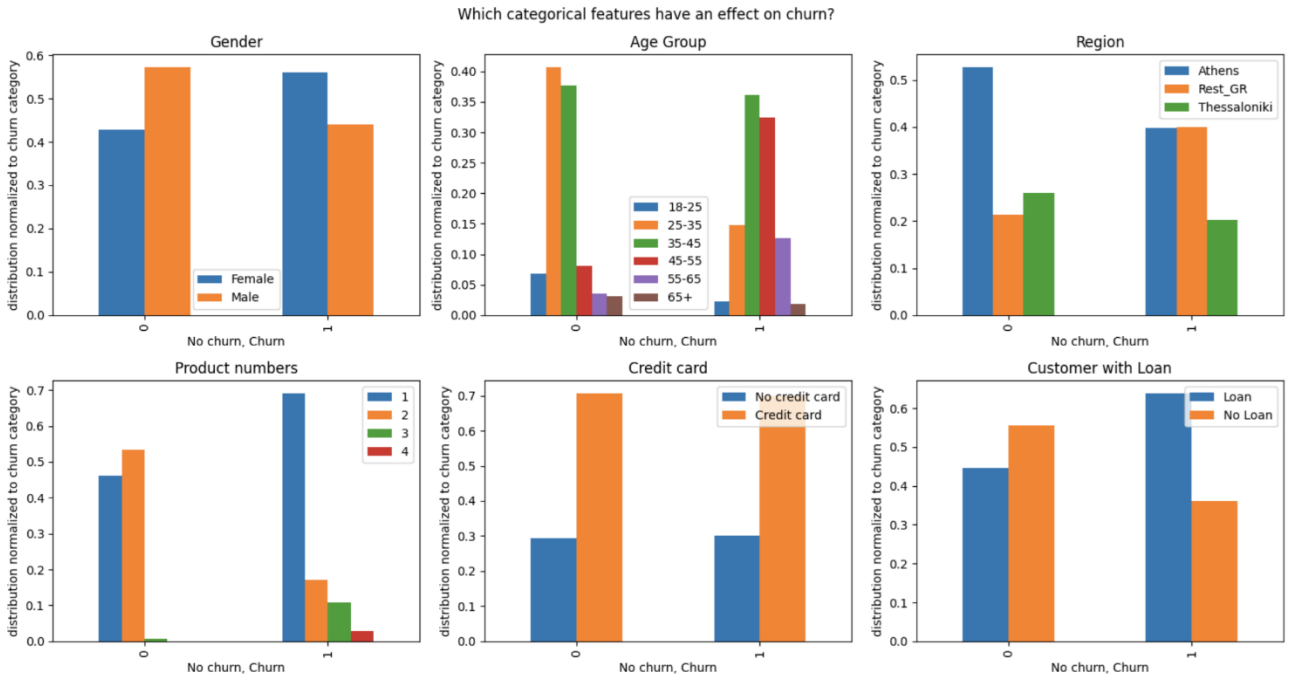

Figure 3: Bar plots showing churn by category.

By examining the balance, we find that as the balance in euros increases, the likelihood of a customer churning also increases (Figure 4). This suggests that customers with higher account balances are more likely to churn.

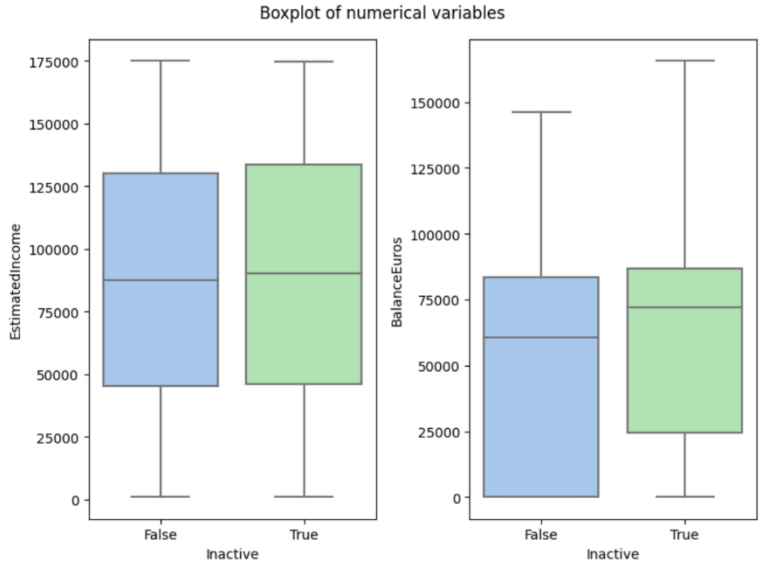

Figure 4: Box plots for estimated income and balance

We created a correlation plot (Figure 5) to examine the correlations between the numerical features, both in relation to the target variable and among themselves. We observed a positive correlation between churn and balance in euros, indicating that as the balance increases, the likelihood of churn also increases. However, we did not observe significant correlations among the other independent variables.

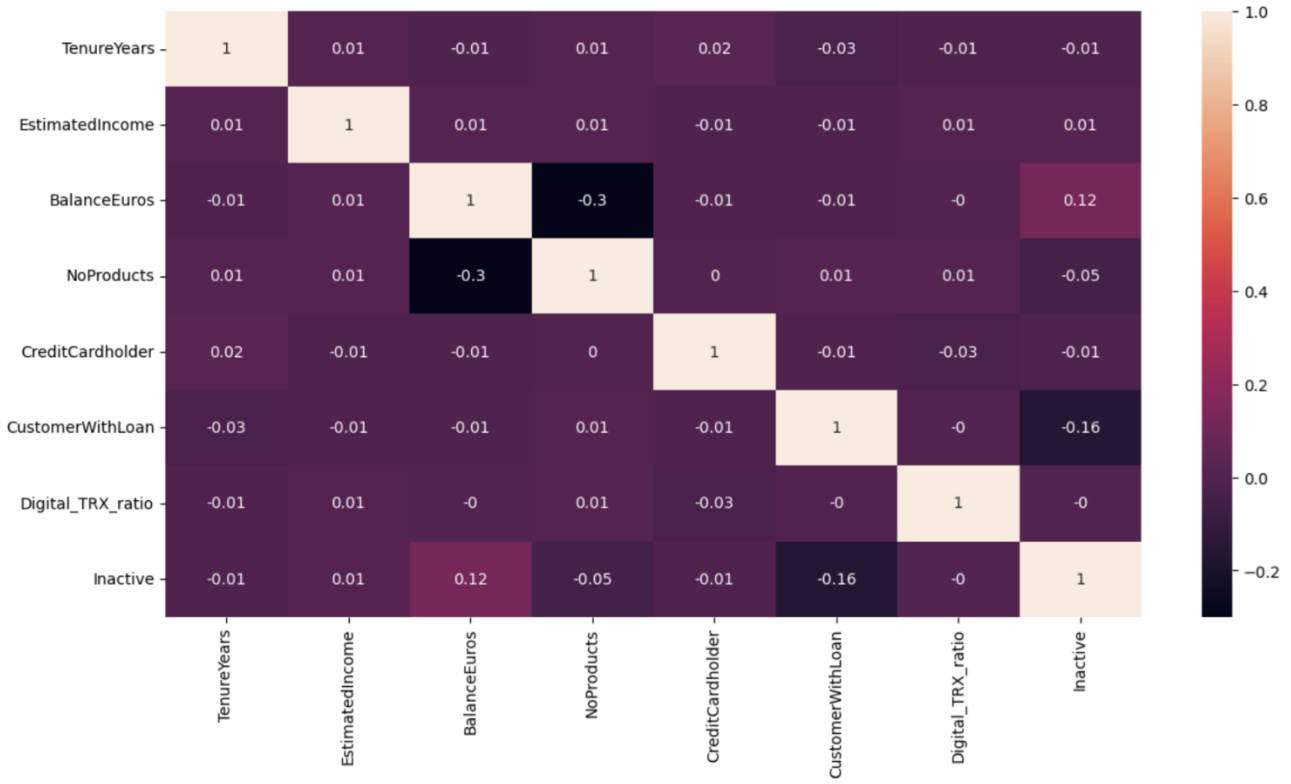

Figure 5: Correlation plot for Churn dataset

Modelling Process

Data Preparation. Data preparation is a crucial step in the modelling process, as it ensures that the data is in a suitable format for the machine learning algorithms. In our case, the data preparation involved several key steps.

Firstly, we converted data types to their appropriate formats, ensuring consistency and accuracy in the dataset. We then proceeded to drop the “CustomerID” column that was not relevant to our modelling task, streamlining the dataset for further analysis.

During the data exploration phase, we conducted an analysis to identify and examine the presence of outliers in the dataset. To detect outliers, we employed the interquartile range (IQR) method. By calculating the IQR and determining the upper and lower bounds, we were able to identify values that fell significantly above or below the normal range. Upon thorough analysis, we found that the EstimatedIncome and BalanceEuros columns did not exhibit any outliers. This implies that the values within these columns were within an expected and reasonable range.

Next, we focused on scaling the numeric features. Scaling is essential to ensure that features with different scales do not disproportionately influence the modelling process. By applying Standardization we transformed the numeric features into a common scale, enabling fair comparisons and accurate modelling.

Another critical aspect of data preparation was encoding categorical variables. Categorical variables, such as gender or geography, cannot be directly used in most machine learning algorithms. Therefore, we employed one-hot encoding to convert categorical variables into numerical representations that could be easily processed by the models.

Lastly, we divided the dataset into training and testing sets. The training set was used to train our models, while the testing set served as an independent dataset to evaluate their performance. Splitting the data allowed us to assess how well the models generalized to unseen data, providing a reliable measure of their predictive capabilities.

Dealing with imbalanced data is a common challenge in classification problems. Imbalanced data occurs when one class is significantly more prevalent than the other, leading to biased models that struggle to accurately predict the minority class. To address this issue, various techniques have been developed. Oversampling involves creating synthetic samples of the minority class, while undersampling reduces the number of instances in the majority class. Another approach is the use of ensemble methods and boosting algorithms that assign higher weights to the minority class. However, it's important to note that not all datasets with class imbalance require these techniques. In cases where the class imbalance is relatively moderate, like the dataset at hand where the churn class represents around 20% of the data, the need for advanced techniques may be less pronounced.

Overall, through comprehensive data preparation, we ensured that our dataset was transformed into a suitable format for modelling, enabling us to proceed with training and evaluating various machine learning algorithms.

Comparative Evaluation of Classification Models

In order to identify the best models for our churn prediction task, we utilized the LazyClassifier library, which provides a convenient way to evaluate and compare multiple machine learning models (Figure 6). This library allows us to quickly assess the performance of various classifiers without the need for extensive manual configuration.

This method fits each classifier on the training data and generates predictions for the testing data. By employing the LazyClassifier library and evaluating multiple models, we are able to compare their performance metrics and identify the most promising ones for further analysis. This approach saves us time and effort in manually configuring and training individual models, allowing us to efficiently explore a wide range of classifiers and select the ones that yield the best results for our churn prediction problem.

After evaluating various models using the LazyClassifier, it was found that the LGBMClassifier performed the best.

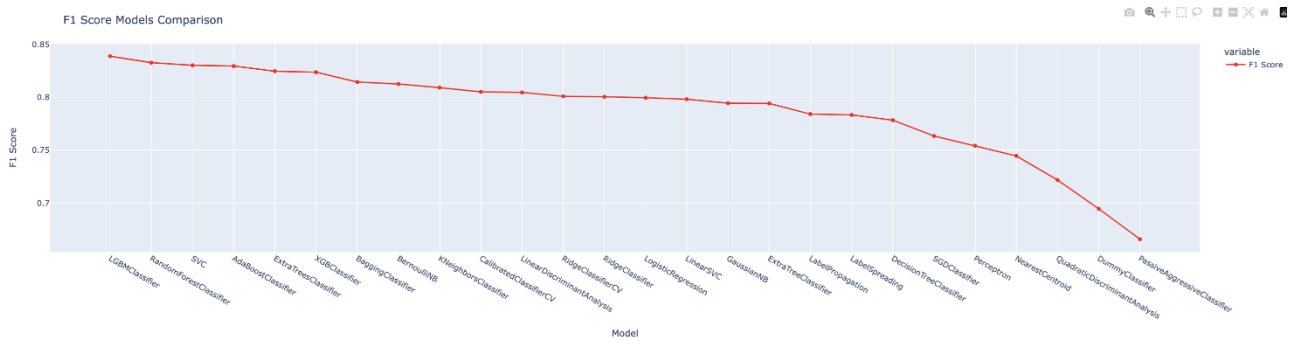

Figure 6: Comparison of classification models for churn

Following the selection of the LGBMClassifier as the best model, we will proceed to utilize it for our classification task. However, in addition to the LGBMClassifier, we will also explore the application of logistic regression. While the LGBMClassifier excels in terms of predictive performance and handling complex datasets, logistic regression offers advantages in terms of interpretability.

LGBMClassifier

Theory

The LGBMClassifier, which stands for Light Gradient Boosting Machine, is a powerful gradient boosting framework known for its efficiency and accuracy in handling large-scale datasets. It utilizes the gradient boosting algorithm, which combines multiple weak learners to create a strong predictive model. The LGBMClassifier is particularly well-suited for classification tasks and has a reputation for delivering

excellent results. Its ability to handle imbalanced datasets, handle missing values, and provide fast and efficient training makes it a popular choice in machine learning applications.

Other very well-known boosting algorithms are the XGBoost and the CatBoost. The main differences between these algorithms lie in their optimization techniques, handling of categorical features, and implementation details. While all three algorithms are powerful and widely used, the choice between them often depends on the specific requirements of the task, dataset characteristics, and computational considerations.

Parameters tuning

The next step was to perform model tuning and evaluation using the LGBMClassifier.

First, a parameter grid is defined, specifying different combinations of hyperparameters such as the number of estimators, learning rate, and maximum depth. The GridSearchCV object is then created, which conducts an exhaustive search over the parameter grid using cross-validation. The scoring metric chosen for optimization is 'recall', which emphasizes the ability of the model to correctly identify the positive class (churned customers) out of all actual positive instances.

In this particular task, the choice of using 'recall' as the scoring metric and focusing on its value in the evaluation is highlighted. Recall is a measure of the model's ability to correctly identify all positive instances out of the total actual positive instances (churned customers). Given the context of churn prediction, it is crucial to minimize the number of false negatives (churned customers falsely classified as non-churned). By prioritizing recall, the model aims to maximize the detection of churned customers, ensuring that they are not overlooked or misclassified. This emphasis on recall helps in capturing as many true positive cases as possible, even at the cost of potentially higher false positive rates.

During the fitting process, the GridSearchCV object trains and evaluates the LGBMClassifier model on different combinations of hyperparameters using the training data. The best parameters and the corresponding best score (recall) are obtained.

Next, the LGBMClassifier is initialized with the best parameters and trained on the entire training dataset. This best model is then evaluated on the test data by making predictions using the predict method.

Finally, evaluation metrics are computed to assess the performance of the best model. The classification_report function generates a comprehensive report, including precision, recall, F1-score, and support for each class. The confusion_matrix function calculates the number of true positive, true negative, false positive, and false negative predictions, which provides further insight into the model's performance.

Regarding the most well-known evaluation metrics in classification problems:

- Precision: Precision measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive. It provides an indication of the model's accuracy in classifying positive cases. Higher precision implies fewer false positives.
- Recall: Recall, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances. It quantifies the model's ability to capture positive cases. Higher recall implies fewer false negatives.
- F1-score: The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance by considering both precision and recall. F1-score reaches its best value at 1 and worst value at 0, considering a perfect balance between precision and recall.
- Support: Support represents the number of occurrences of each class in the test data. It provides insights into the distribution of classes and can help identify any class imbalance issues.

Model Evaluation

The classification results reveal that the model performs well in predicting the 'Active' class (non-churn), with a precision score of 0.87, indicating that 87% of the predicted 'Active' instances are correct. The

recall score of 0.95 signifies that the model captures 95% of the actual 'Active' instances. The F1-score, which balances precision and recall, is 0.91, indicating a good overall performance for the 'Active' class.

However, the model struggles to accurately predict the 'Inactive' class (churn). It has a lower precision score of 0.72, suggesting that only 72% of the predicted 'Inactive' instances are correct. The recall score of 0.46 reveals that the model identifies only 46% of the actual 'Inactive' instances. The F1-score for the 'Inactive' class is 0.56, indicating a relatively weaker performance compared to the 'Active' class (Figure 7 & Figure 8).

In terms of churn prediction (class 'Inactive'), the model's performance could be further improved. Identifying more actual churn instances (higher recall) and improving the precision of the predictions for churn (higher precision) would be beneficial. Fine-tuning the model, exploring additional features, or applying different techniques specifically tailored for imbalanced datasets may help enhance the model's performance on churn prediction.

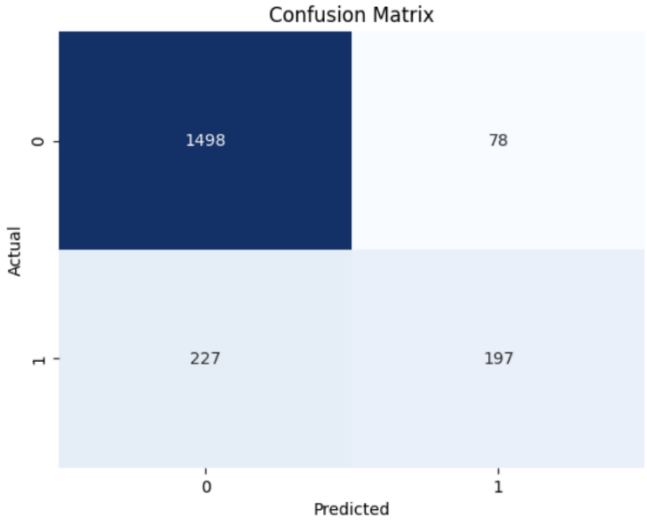

Figure 7: Confusion matrix for LGBM classifier

	precision	recall	f1-score	support
Active	0.87	0.95	0.91	1576
Inactive	0.72	0.46	0.56	424
accuracy			0.85	2000
macro avg	0.79	0.71	0.74	2000
weighted avg	0.84	0.85	0.83	2000

Figure 8: Model performance metrics for LGBM classifier

Among the features examined, several important factors emerge as influential in predicting churn (Figure 9). The first significant feature is the Estimated Income, which represents the customers' estimated income level. Similarly, the Balance variable, reflecting the customers' account balance, holds importance. Another influential feature is the Digital TRX Ratio, which measures the ratio of digital transactions made by customers. A higher ratio suggests a greater reliance on digital channels, indicating a potentially more tech-savvy and engaged customer base. The Tenure Years feature also plays a vital role in predicting churn. Lastly, the number of products used by customers is a significant predictor. Customers with a greater number of products may have a higher level of engagement and dependency on the company's offerings. This multi-product usage may lead to a stronger bond with the services and a lower likelihood of churning.

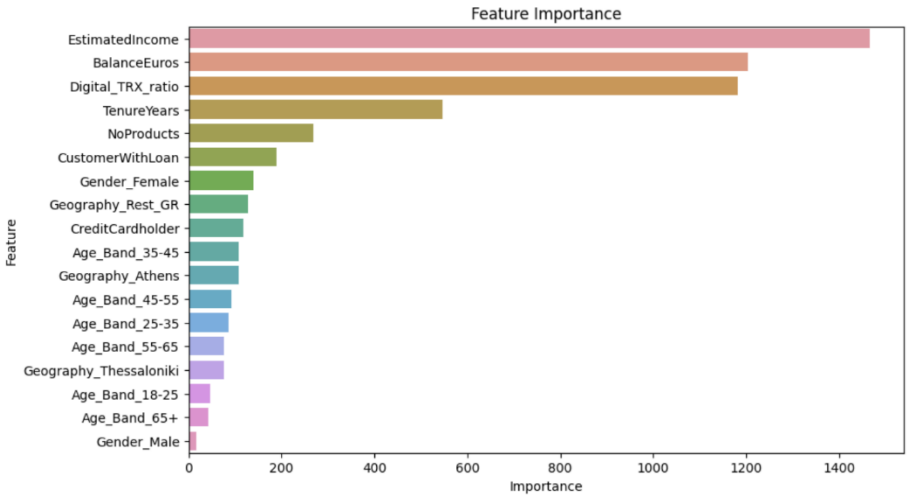

Figure 9: Feature importance for LGBM classifier

Custom Threshold to increase the Recall

Typically, when we make predictions in a classification problem, we assume that if the probability is above 0.5, then it is classified as "true," otherwise it is classified as "false." However, we have the flexibility to change this threshold and lower it in order to predict more instances as "true" and increase the recall score. A simple algorithm was created to find a threshold that achieves a desired recall score (66%) within a specified range. My goal was to have a recall score ~ 66%, i.e., to predict 2 out of 3 churn customers.

To achieve this, the algorithm iterates over a range of threshold values, adjusting the predicted labels based on each threshold. The adjusted labels are then used to calculate the recall score. If the recall score falls within the desired range and is closer to the desired recall score than the previous best recall score, the current threshold and recall score are updated as the optimal threshold and best recall score, respectively.

Finally, the optimal threshold and corresponding best recall score are printed. Additionally, the code specifies a new threshold (optimal_threshold_final) and adjusts the predicted labels accordingly. Evaluation metrics, such as classification report and confusion matrix, are computed using the adjusted labels to assess the performance of the model at the final threshold.

Regarding the issues that this process addresses, it is focused on improving the prediction of churn instances. By adjusting the threshold and considering more instances as "true," we aim to increase the recall score. This is important in churn prediction as we want to identify as many actual churn cases as possible, even if it means having more false positives. This approach allows for a trade-off between precision and recall, emphasizing the importance of capturing all churn instances while accepting a certain level of misclassification.

The results of the process indicate that the optimal threshold for predicting churn is 0.28, and the corresponding best recall score achieved is approximately 0.67. This means that by setting the threshold at 0.28, we can capture around 67% of the actual churn cases correctly. The optimal threshold value of 0.28 implies that if the predicted probability of churn for a customer is higher than this threshold, we will classify them as a churn case. On the other hand, if the predicted probability is below the threshold, we will classify them as a non-churn case.

The best recall score of 0.67 suggests that the model is performing reasonably well in identifying churn cases, capturing a significant portion of them. This is important in a churn prediction task because it allows businesses to take proactive measures to retain customers who are at risk of churning. However, it is important to note that setting a lower threshold to increase recall comes at the cost of potentially increasing false positives, i.e., classifying some non-churn cases as churn. This trade-off between recall

and precision should be considered based on the specific context and priorities of the business (Figure 10).

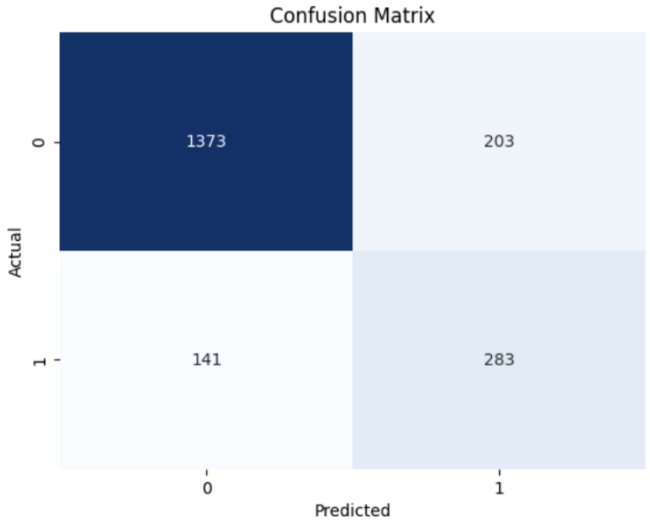

Figure 10: Confusion matrix for LGBM classifier using custom threshold in order to improve Recall

Logistic Regression

Logistic regression is a statistical model used for binary classification. It estimates the probability of an event occurring by fitting a logistic function to the data. One advantage of logistic regression is its interpretability, as it provides coefficients representing the impact of each feature on the likelihood of the target variable.

In our process, we used logistic regression for churn prediction. We began by performing feature selection using Recursive Feature Elimination with Cross-Validation (RFECV) to identify the most important features. Then, we created a logistic regression model and tuned its hyperparameters using GridSearchCV, focusing on the penalty (L1 or L2 regularization) and the inverse of regularization strength (C).

The best model with optimal hyperparameters was fitted to the selected features using the training data. We evaluated the model's performance on the test data using classification metrics such as precision, recall, and F1-score. The performance is worse than the LGBMClassifier model's performance (Figure 11).

	precision	recall	f1-score	support
Active	0.85	0.95	0.90	1576
Inactive	0.67	0.36	0.47	424
accuracy			0.83	2000
macro avg	0.76	0.66	0.68	2000
weighted avg	0.81	0.83	0.81	2000

Figure 11: Model performance metrics for Logistic Regression

A confusion matrix was also generated to assess the model's ability to predict true positive, true negative, false positive, and false negative cases (Figure 12).

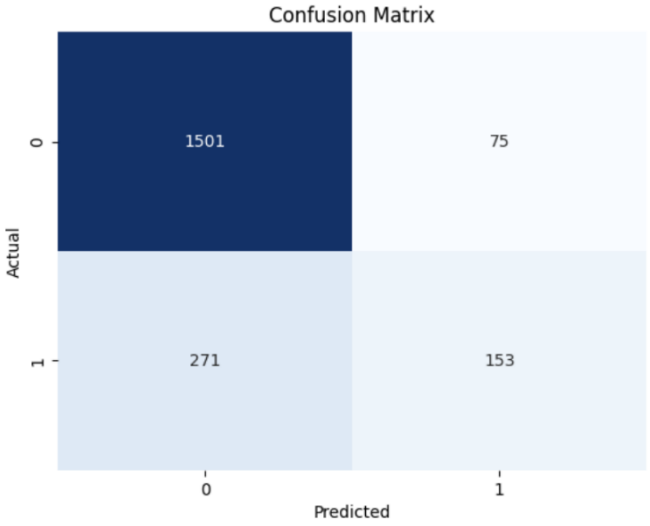

Figure 12: Confusion matrix for Logistic Regression

Additionally, we summarized the model by examining the coefficients and odds ratios of the selected features. The odds ratio provides a clearer interpretation of the coefficients. It represents the multiplicative change in the odds of the target variable for a one-unit increase in the corresponding feature. For instance, the odds ratio of 12.54 for "Age_Band_45-55" implies that customers in this age range are about 12.54 times more likely to churn compared to the reference category.

Other features such as "BalanceEuros" and "Geography_Rest_GR" also have positive coefficients and odds ratios, indicating their positive influence on churn. Conversely, "CustomerWithLoan" has a negative coefficient and odds ratio below 1, suggesting a negative association with churn (Figure 13).

Model Summary:

	Feature	Coefficient	Odds Ratio
0	Age_Band_25-35	0.06	1.06
1	Age_Band_35-45	1.02	2.77
2	Age_Band_45-55	2.53	12.54
3	Age_Band_55-65	2.64	14.08
4	Age_Band_65+	0.90	2.46
5	BalanceEuros	0.19	1.21
6	CustomerWithLoan	-0.94	0.39
7	Geography_Rest_GR	0.73	2.06
8	Geography_Thessaloniki	0.05	1.05
9	Intercept	-2.30	0.10

Figure 13: Coefficients extracted from Logistic Regression

Summary and Future Steps

In this project, we focused on predicting customer churn in a financial institution using machine learning techniques. We started by preprocessing the data, including encoding categorical variables, and scaling the features. We then trained several classifiers, including the LGBMClassifier and Logistic Regression, to identify the best model for churn prediction. Evaluation metrics such as precision, recall, and F1-score were used to assess model performance, with a particular emphasis on recall due to its significance in identifying churn cases accurately.

The results showed that the chosen model achieved promising performance, with an overall accuracy of 85%. The precision and recall scores for active customers were 0.87 and 0.95, respectively, indicating a high level of correctly identified non-churn customers. However, the precision and recall scores for inactive customers were lower at 0.72 and 0.46, respectively, suggesting a room for improvement in accurately identifying churn cases.

A custom algorithm was implemented to identify an optimal threshold for churn prediction. By adjusting the threshold, we aimed to increase the recall and accurately identify more customers who are likely to

churn. Our goal was to find a threshold that maximizes the recall while maintaining a desired range. The algorithm successfully determined an optimal threshold of 0.28, resulting in a recall score of 0.667. This means that 66.7% of the actual churn cases were correctly identified by the model. By fine-tuning the threshold, we were able to increase the model's sensitivity to churn cases and minimize the chances of missing potential churners.

To enhance the churn prediction process, there are several areas that the company could consider focusing on.

- First, it would be beneficial to gather more relevant data, such as customer interaction history, feedback, or customer satisfaction surveys, to gain deeper insights into customer behavior and potential reasons for churn.
- Additionally, exploring advanced feature engineering techniques, such as creating interaction terms or capturing nonlinear relationships, could help capture more nuanced patterns in the data.
- Performing a clustering analysis similar to the RFM analysis would be beneficial for churn prediction, as the resulted categories would be important factors for our models.
- Moreover, the company could invest in implementing proactive retention strategies based on the churn predictions. By identifying customers at high risk of churn, personalized interventions, such as targeted marketing campaigns, loyalty programs, or personalized offers, can be developed to incentivize customers to stay.
- To identify customers at risk of churn, the company can adopt a methodology similar to ours, involving the determination of a custom threshold. This approach would enable a more proactive marketing strategy; however, it is crucial to consider potential customer dissatisfaction due to the intensified marketing efforts. Additionally, the threshold should be defined by weighing the cost associated with these strategies, ensuring a balance between effective retention measures and the financial investment required.
- The company should aim to acquire more young customers between the ages of 18-35, as they are less likely to churn, based on our findings.
- The company should analyze the business impact of our solution (or of any relevant action) by tracking key performance indicators (KPIs) related to customer churn, such as churn rate, customer retention, and customer lifetime value.
- Finally we should implement a system to continuously monitor and update our churn prediction model. As customer behaviors and preferences evolve, it is essential to regularly assess the model's performance and make necessary adjustments to enhance its accuracy and effectiveness.

Overall, this project laid the foundation for accurate churn prediction, but there is room for improvement and additional steps that can be taken to enhance the customer retention efforts. By leveraging the insights gained from the model and adopting proactive retention strategies, the company can work towards reducing customer churn, improving customer satisfaction, and ultimately achieving long-term business growth.