Loan Approval Automation Report By Ruth

July 30, 2024

1 Loan Approval Automation

Table of Contents

1.1 Executive Summary

The objective of this project is to develop a machine learning model that will automate the rejection and approval of loan applications, with the aim of reducing labor costs and minimizing losses from incorrect decisions.

1.1.1 Key Findings

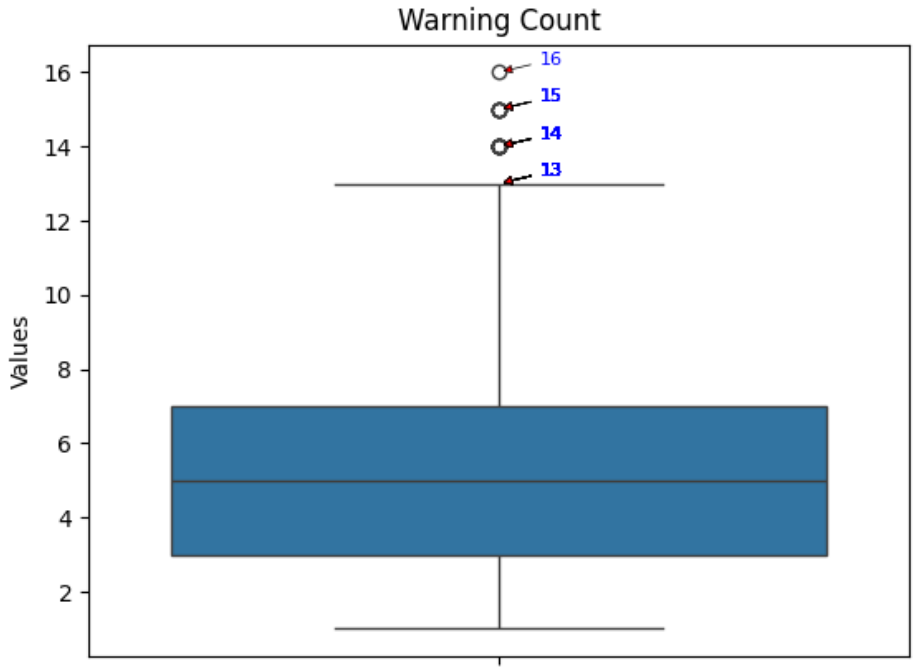
- **Model Performance:** The Random Forest model achieved an accuracy of 84%.

- **Probability Thresholds:** The recommended thresholds for auto-rejecting and auto-approving applications is set at 0.5
- **Cost Savings:** Implementing the model can save approximately 24,655 EUR, which is 50% of the total labor cost from manual processing

1.2 Data Understanding and Preparation

The dataset comprises 9898 records of loan applications that underwent manual approval. It includes the target variable (AR), which indicates whether an application should be accepted or rejected, alongside various characteristics of each application.The following steps were taken to ensure that the data is clean:

- **Standardization of Data Types:** Ensured all data types were consistent and appropriate for analysis.
- **Encoding Categorical Variables:** Applied one-hot encoding to transform categorical variables into a format suitable for machine learning algorithms.
- **Missing Values Analysis:** No missing values were found, ensuring a complete dataset.
- **Duplicate Rows Analysis:** The dataset contained no duplicate rows, affirming its uniqueness.
- **Outlier Detection Analysis:** Outliers were detected in the warning count column.Some of the techniques used to detect outliers were:
  - **Interquartile Range (IQR):** Identified 24 outliers.
  - **Z-Score:** Identified 52 outliers.
  - **Combined Analysis:** A total of 52 unique outliers were observed by combining the IQR and Z-score methods. The outlier values ware assumed to be reasonable and possible in real-life scenarios. Therefore, they were treated as edge cases rather than anomalies and retained in the dataset.

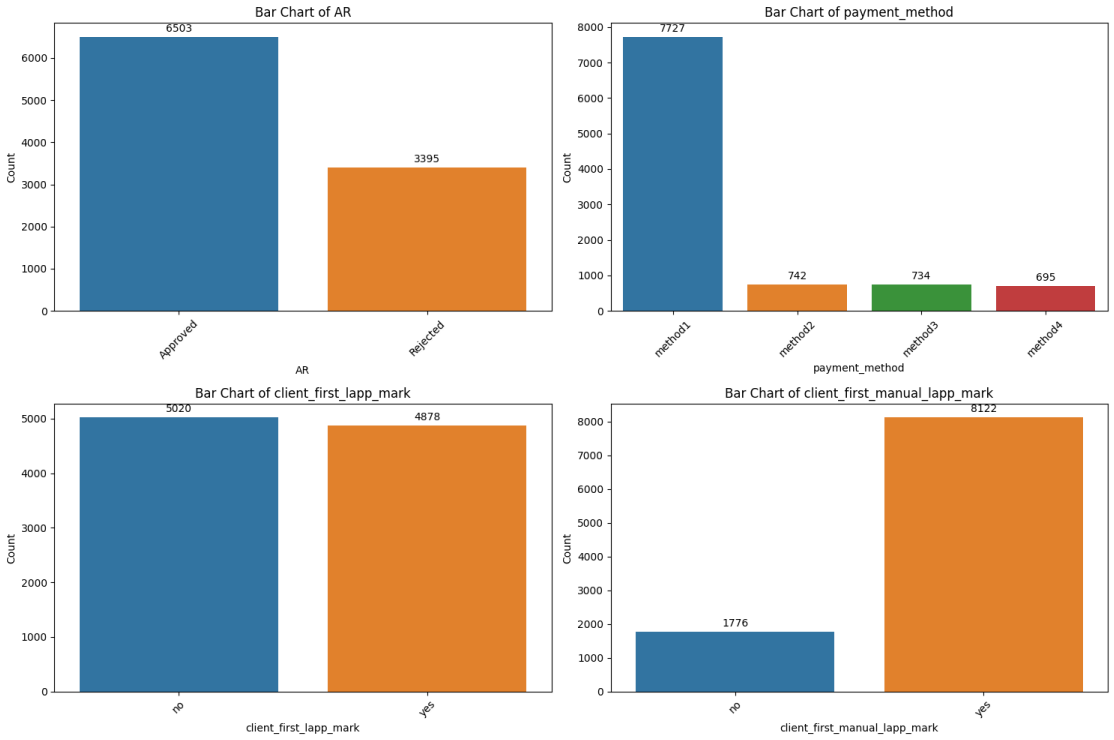


1.3 Exploratory Data Analysis

I conducted an exploratory data analysis (EDA) to understand the distribution and relationships within the cleaned dataset.

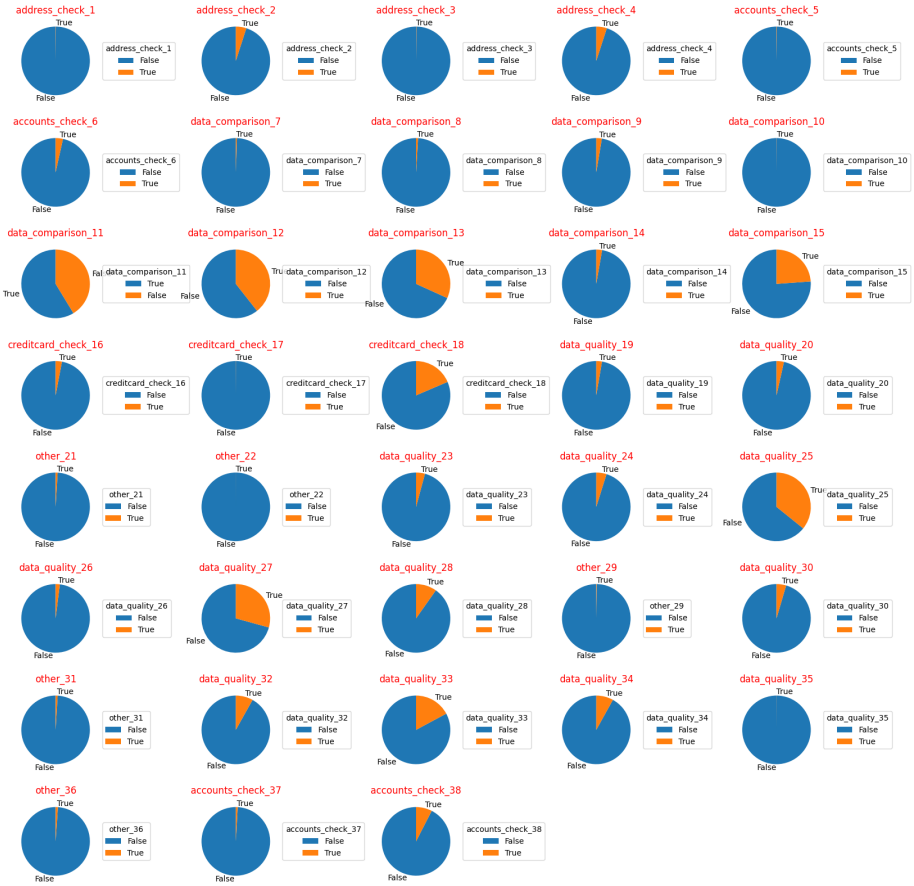
1.3.1 Univariate Analysis

Categorical Fields Identified imbalance in the AR field, with 6,503 for the majority class and 3,395 for the minority class. Other features also showed imbalances.



Data Check Fields Most fields related to data checks had an overwhelming amount of imbalance.

- **Recommendations for the data check fields:** Most fields have a minimal presence of `True` values, suggesting that they might not contribute significantly to the analysis. So focus should be on categories with a higher proportion of `True` values as they are more likely to provide relevant and actionable insights. Fields with an overwhelming majority of `False` values and do not have a strong relationship with the AR column will be dropped, as they may not add substantial value to the analysis.



1.3.2 Correlation Analysis

An analysis of of how the feature interact with each other and most importantly how they interact with the target was done. Features that had multicollinearity were either:

- Combined features to enhance positive relationships with the target.
- Removed less informative features.

Findings and Recommendations:

- **Columns with weak relationship with the target :**
  - **Columns with Coefficients less than or equal to 0:** Since these variables show either a very weak or negative relationship with AR, they may add noise or complexity to the model without providing meaningful predictive power. Removing these columns could simplify the model and potentially improve its performance.
- **Columns with strong realtionship with the target:**

– **Columns with Positive Coefficients:** These variables show a more meaningful positive correlation with the AR and are likely to provide useful information for predictions.

1.4 Models

1.4.1 Model Selection

The following models were tested.

- **Logistic regression**
- **Random Forests**
- **XGBoost**
- **An Ensemble of the three**

Variables Used

- client_first_lapp_mark
- client_first_manual_lapp_mark
- warning_count
- data_comparison_8
- data_comparison_13
- creditcard_check_16
- data_quality_19
- data_quality_24
- data_quality_27
- data_quality_28
- data_quality_32
- data_quality_33
- accounts_check_38
- accounts_check_sum - `An engineered field for all account check fields`
- creditcard_check_sum - `An engineered field for all credit card check fields`
- data_comparison_check_sum - `An engineered field for all data comparison check fields`
- data_quality_check_sum - `An engineered field for all data quality check fields`
- other_check_sum - `An engineered field for all other check fields`
- payment_method
- warning_count&data_comparison_15 - `An engineered field from multicollinear field increased relationship with target`
- warning_count&data_comparison_14 - `An engineered field from multicollinear field increased relationship with target`
- warning_count&data_comparison_12 - `An engineered field from multicollinear field increased relationship with target`
- warning_count&data_comparison_11 - `An engineered field from multicollinear field increased relationship with target`

1.4.2 Model Training and Evaluation

The data was split into training (80%) and testing (20%) sets. SMOTE resampling techniques were applied to address class imbalance. GridSearchCV and RandomizedSearchCV for hyperparameter tuning of the models.

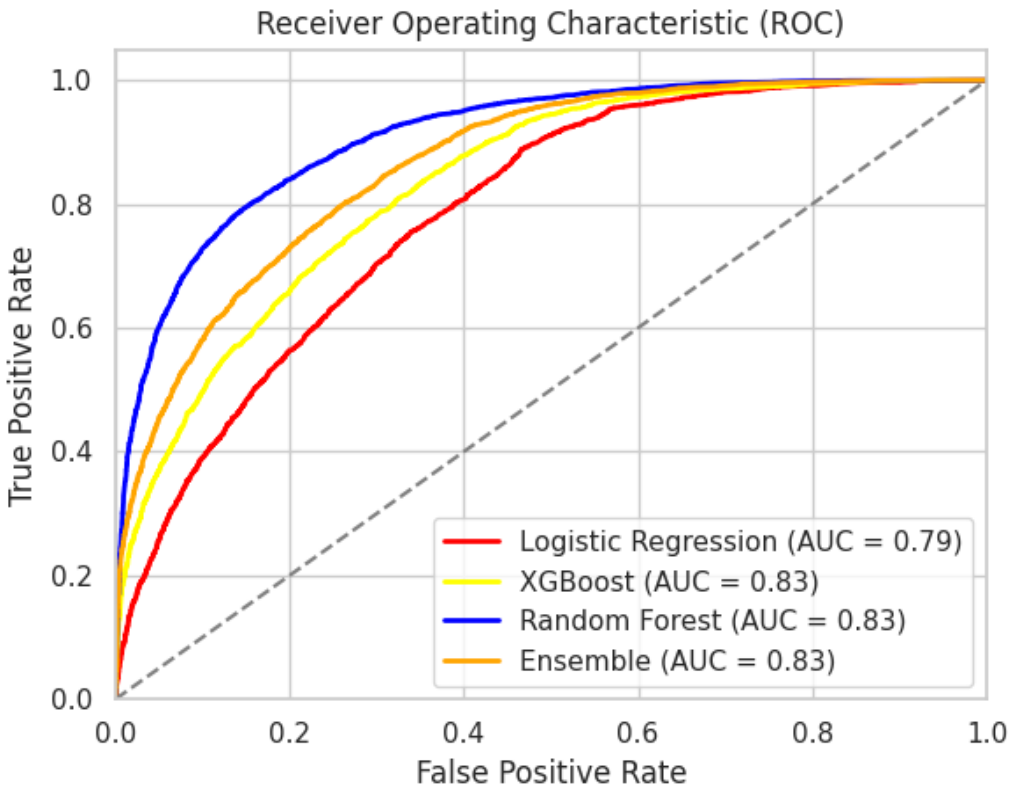
Model Performance To assess and compare the precision of the models, I used the following metrics.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	84.10%	79.34%	72.52%	75.78%
XGBoost	72.02%	60.91%	51.40%	55.75%
Logistic Regression	72.47%	61.80%	51.69%	56.30%
Ensemble	72.54%	63.25%	51.49%	58.54%

The table above shows the performance metrics for four different models: Random Forest, XGBoost, Logistic Regression, and an Ensemble model. The metrics include Accuracy, Precision, Recall, and F1-score, which are essential for evaluating the effectiveness of a classifier.

Among these models, the Random Forest classifier exhibited the highest performance across most metrics, achieving an accuracy of 84.10%, a precision of 79.34%, a recall of 72.52%, and an F1-score of 75.78%. These metrics indicate that the Random Forest model is the most effective in balancing precision and recall, leading to a higher overall performance.

Therefore, we chose the Random Forest classifier due to its superior performance in terms of accuracy, precision, recall, and F1-score compared to the other models. This makes it the most reliable choice for our classification task.



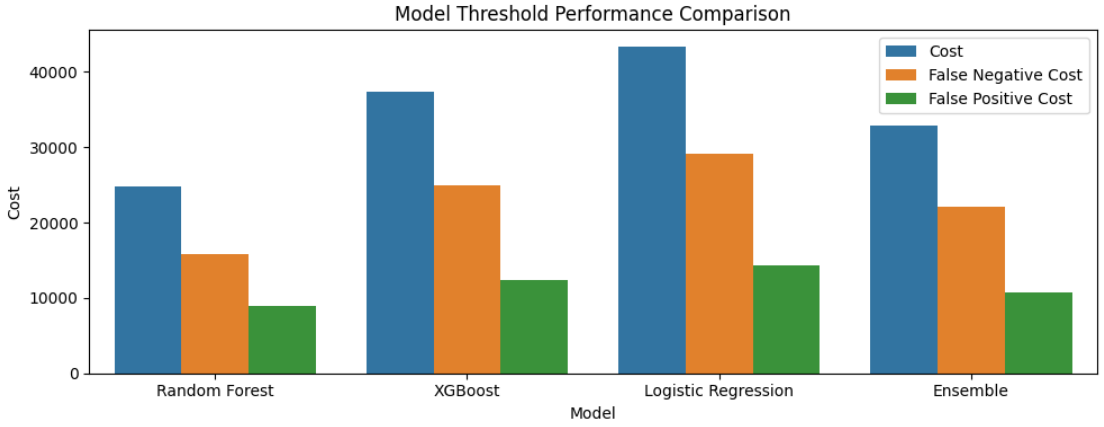
1.5 Probability Limits for Automated Decision Making

Evaluated multiple thresholds from 0.1 - 0.9 by classifying data based on predicted probabilities of various models, and calculating the costs of false negatives and false positives. The threshold that results in the lowest total cost of loss and provides the optimal balance between false negatives and false positives was selected.

1.5.1 Optimal Model Thresholds

Below is a list of optimal thresholds for the models used:

Model	Threshold	Cost	False Negative Count	False Negative Cost	False Positive Count	False Positive Cost
Random Forest	0.50	24,835	933	15,861	641	8,974
XGBoost	0.51	37,337	1,465	24,905	888	12,432
Logistic Regression	0.51	43,367	1,711	29,087	1,020	14,280
Ensemble	0.50	32,892	1,304	22,168	766	10,724



1.5.2 Recommended Threshold

Based on the above , I recommend using the Random Forest model :

- Auto-reject applications with probability >= 0.5
- Auto-approve applications with probability < 0.5

1.6 Business Impact Analysis

1.6.1 Cost and Loss Comparison

The Cost and Loss comparison is based on 9898 loan applications.

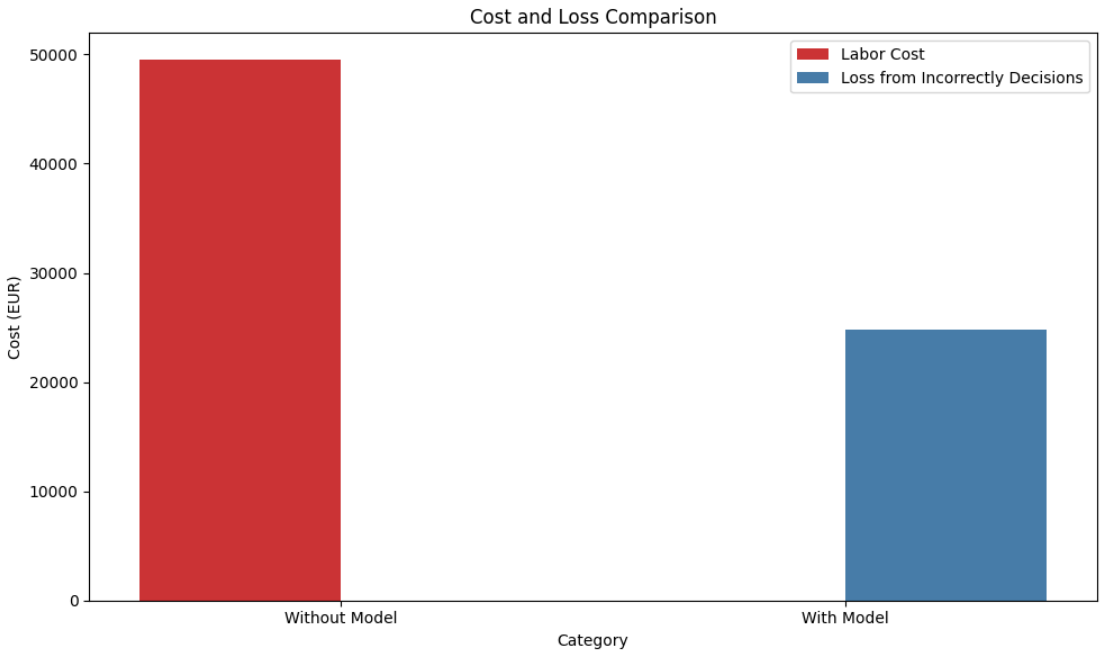
Without Model

- Total labor cost: 49,490
- Total loss from incorrect decisions: 0

With Model Out of the 9,898 loans 1,574 loans were incorrectly labled, 641 loans that should have been approved were rejected and 933 loans that should have been rejected were approved. With the model: - Total labor cost: 0 - Total loss from incorrect decisions: 24,835

Potential Savings Implementing the model saves 24,655 EUR, which is 50% of the labor cost, with potential for increase in savings with further model tuning.

	Without Model	**With Model**
Total Labor Cost	49,490 EUR	0 EUR
Total Loss from Incorrect Decisions	0 EUR	24,835 EUR



1.7 Conclusions

1.7.1 Key Findings

- The machine learning model can effectively automate the loan approval process, achieving an accuracy of 84.10% with the Random Forest classifier. This reduces manual processing costs and minimizes losses from incorrect decisions.
- The optimal threshold for the Random Forest model is 0.5, leading to the best balance between false positives and false negatives and overall reduction of loss .

1.7.2 Improvements

- **Feature Engineering:** Explore additional feature engineering techniques to enhance the model's predictive power. This could involve creating new features or improving existing ones.
- **Explore Different Models:** While the Random Forest model has shown the best performance among the tested models, exploring other machine learning models and techniques can be done to ensure the highest possible accuracy and reliability from the model.

By implementing these recommendations, the company can significantly enhance the efficiency and accuracy of the loan approval process, leading to substantial cost savings and improved decision-making.