German Bank Loan Default Prediction

The aim of the project "Loan Default Prediction" is to create a project that is both replicable and suitable for publication. This project will illustrate the step-by-step procedure of creating and assessing machine learning models specifically designed for predicting loan defaults within the Banking Financial sector.

Introduction

The German bank dataset offers insight into the field of predicting loan defaults, a significant challenge for financial institutions. In this scenario, the bank is dealing with the crucial task of identifying customers who may default on their loans. The dataset contains historical information about customers who have taken loans from the bank. The main goal is to create a predictive machine learning model that can effectively predict whether a customer will default or not. This prediction is based on insights derived from various historical features associated with each customer.

The data set has 17 columns and 1000 rows. Columns are described below and each row is a customer.

- checking_balance - Amount of money available in account of customers
- months_loan_duration - Duration since loan taken
- credit_history - credit history of each customers
- purpose - Purpose why loan has been taken
- amount - Amount of loan taken
- savings_balance - Balance in account
- employment_duration - Duration of employment
- percent_of_income - Percentage of monthly income
- years_at_residence - Duration of current residence
- age - Age of customer
- other_credit - Any other credits taken
- housing- Type of housing, rent or own
- existing_loans_count - Existing count of loans
- job - Job type
- dependents - Any dependents on customer
- phone - Having phone or not
- default - Default status (Target column)

The bank has historical information on relevant features for each customer such as employment duration, existing loans count, saving balance, percentage of income, age, default status.

The banking industry faces substantial consequences in terms of financial stability and decision-making when predicting loan defaults. The bank's decision to employ machine learning models for such predictions reflects a forward-thinking approach to risk management. By utilizing a diverse set of customer details like credit history, employment duration, savings balance, age, and loan amount, the bank aims to use data effectively.

This approach can improve the loan approval process, reduce potential losses from defaults, and establish reliable risk management and financial strategies.

Throughout this project, several questions were addressed:

- Which machine learning model proves most effective in predicting loan defaults using the German bank dataset? The project compares the performance of various models to determine the optimal approach for the bank's predictive requirements.
- How can we enhance model performance to minimize false negatives and identify potential defaulters more accurately, aiming for the highest recall performance?
- What impact does a customer's credit history have on the likelihood of loan default? This inquiry offers insights into the importance of creditworthiness in predicting loan defaults.
- While exploratory data analysis (EDA) and data visualization have provided initial answers to many questions, final conclusions require a more in-depth study.

Methods and Materials

Exploratory Data Analysis (EDA)

Data Understanding, Cleaning and Preparation: In the initial phase of the project, I delved into the dataset, thoroughly understanding its structure and column descriptions to grasp the meanings of different features. Subsequently, I conducted a comprehensive exploratory data analysis (EDA) aimed at uncovering any potential issues such as missing values, typos, or duplicates. Although no missing values were found, I addressed some typos to enhance data accuracy. Further categorizing features into numerical, nominal, and ordinal types, I performed summary statistics analyses to extract insights from both numerical and categorical features. To maintain data integrity, duplicate values were identified and removed.

```
In [5]: # StatisticsSummary of the numerical columns
        df.describe().T
```

Out[5]:

	count	mean	std	min	25%	50%	75%	max
months_loan_duration	1000.0	20.903	12.058814	4.0	12.0	18.0	24.00	72.0
amount	1000.0	3271.258	2822.736876	250.0	1365.5	2319.5	3972.25	18424.0
percent_of_income	1000.0	2.973	1.118715	1.0	2.0	3.0	4.00	4.0
years_at_residence	1000.0	2.845	1.103718	1.0	2.0	3.0	4.00	4.0
age	1000.0	35.546	11.375469	19.0	27.0	33.0	42.00	75.0
existing_loans_count	1000.0	1.407	0.577654	1.0	1.0	1.0	2.00	4.0
dependents	1000.0	1.155	0.362086	1.0	1.0	1.0	1.00	2.0

Statistics Summary of the numerical columns

```
In [6]: # Statistics for the categorical columns
        df.describe(include=['O']).T
```

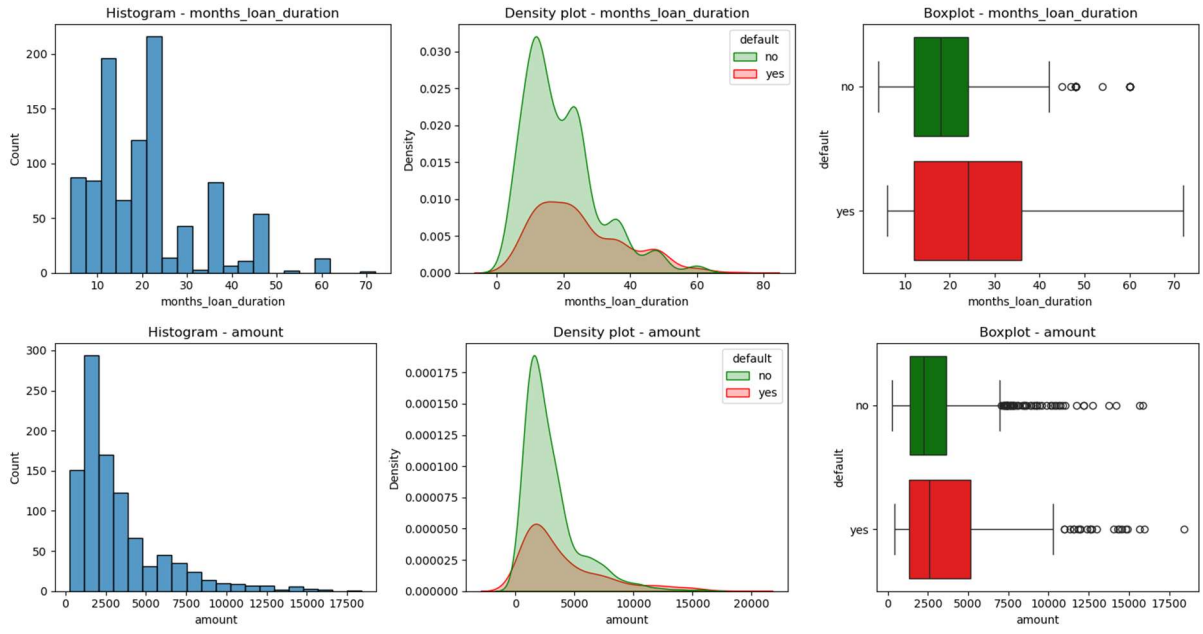
Out[6]:

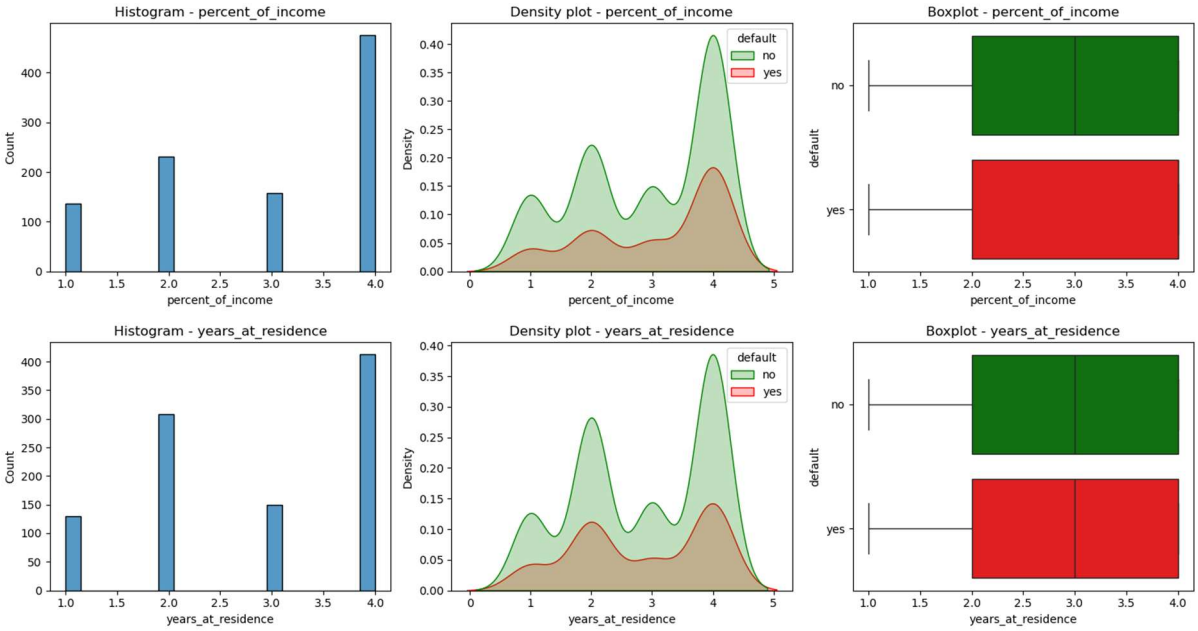
	count	unique	top	freq
checking_balance	1000	4	unknown	394
credit_history	1000	5	good	530
purpose	1000	6	furniture/appliances	473
savings_balance	1000	5	< 100 DM	603
employment_duration	1000	5	1 - 4 years	339
other_credit	1000	3	none	814
housing	1000	3	own	713
job	1000	4	skilled	630
phone	1000	2	no	596
default	1000	2	no	700

Statistics for the categorical columns

NOTE: Some of the columns having 'unknown', 'other', 'none' or 'unemployed' as a category and hence cannot be considered as an ordinal variable even though other categories within the column have a hierarchical order.

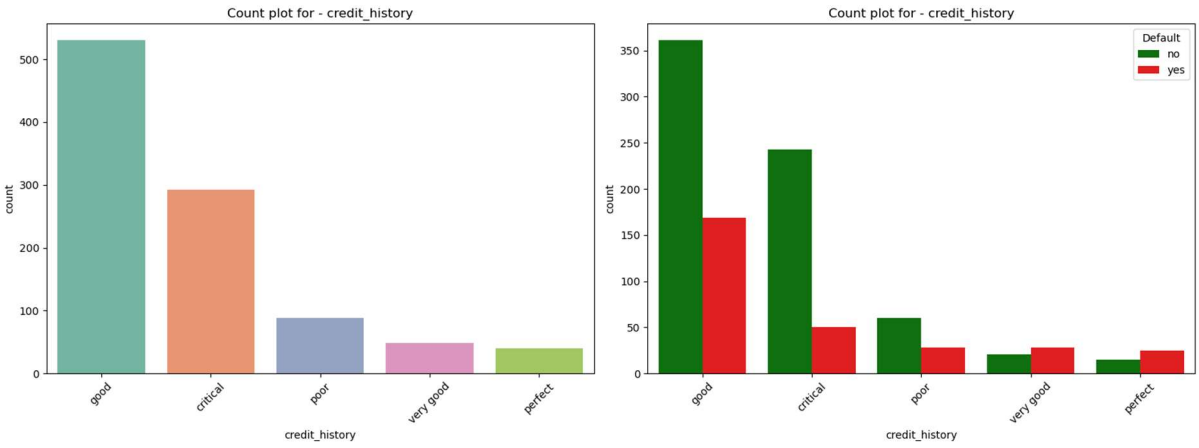
Data Visualization: During the Exploratory Data Analysis (EDA) phase, I utilized a variety of visualizations and analyses to gain deeper insights into the relationships between different features and their influence on loan default behaviour. Histograms, density plots, and boxplots were employed to visually represent the distribution of numerical features, helping identify potential outliers. Additionally, I explored the correlation between numerical features through a pair plot and heatmap. The EDA process also uncovered intriguing research questions, providing initial insights through plotted graphs that may require further in-depth study to establish definitive conclusions.

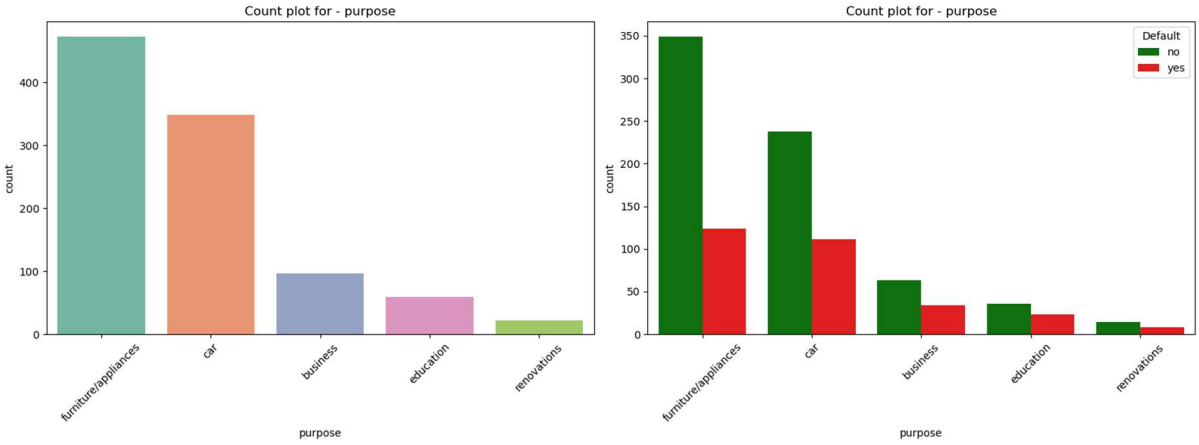




Histogram, Density plot (hue='default') & Boxplots (hue='default') on some numerical column

To analyse categorical variables, I employed count plots to examine how customers are distributed across various categories. I compared these distributions with and without considering the 'default' status as the hue. This approach enabled me to explore differences in behaviour between customers who defaulted on loans and those who did not.

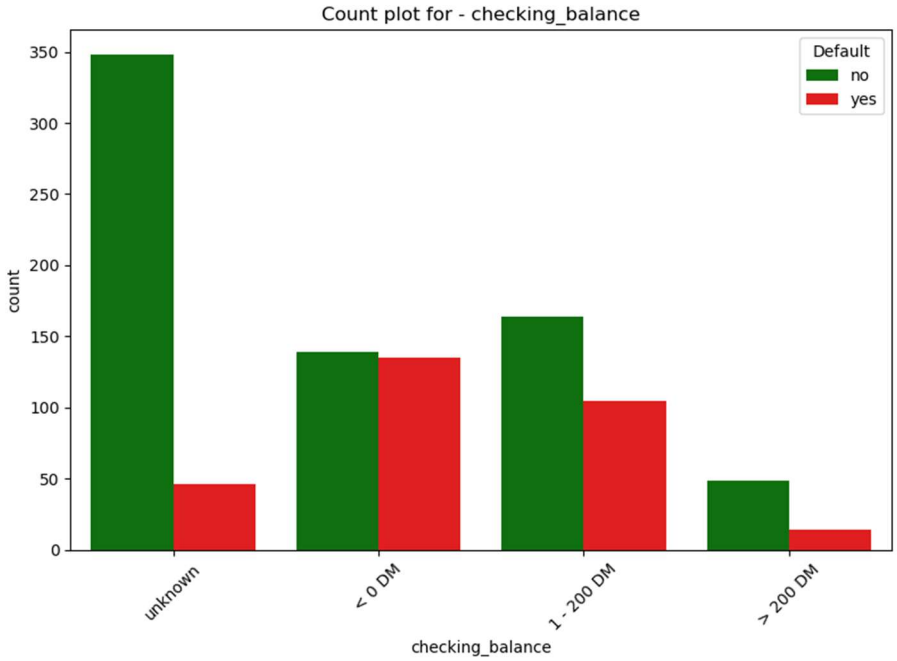




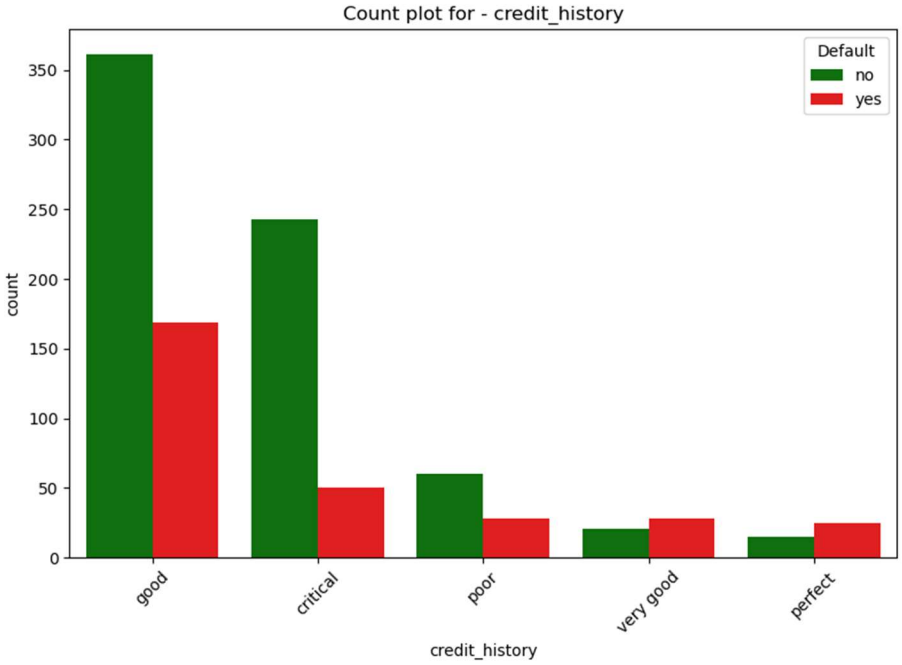
EDA for each categorical column with and without 'default' as the hue

Some Insights from EDA:

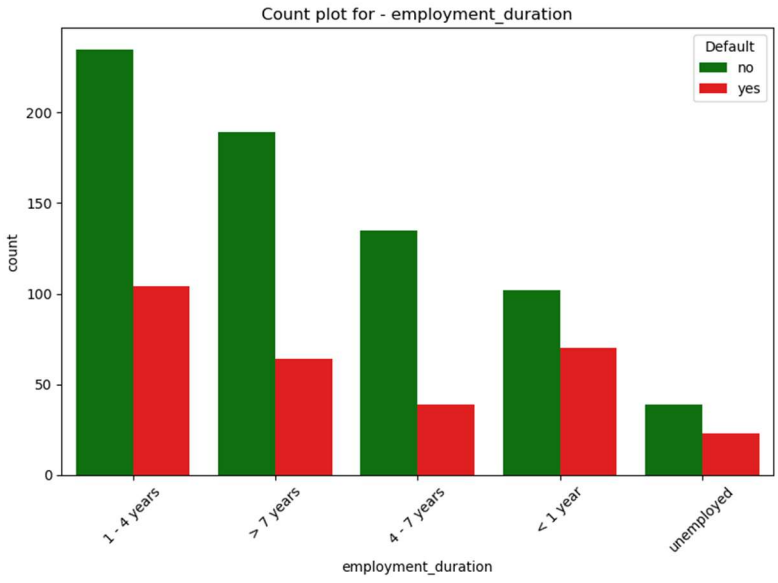
The term 'checking balance' represents the available amount of money in a customer's checking account, which is used for everyday financial transactions. It is a highly liquid portion of the customer's finances. Through data visualization, we can observe that the proportion of customers who default on loans tends to increase when the balance in this liquid account diminishes**.**



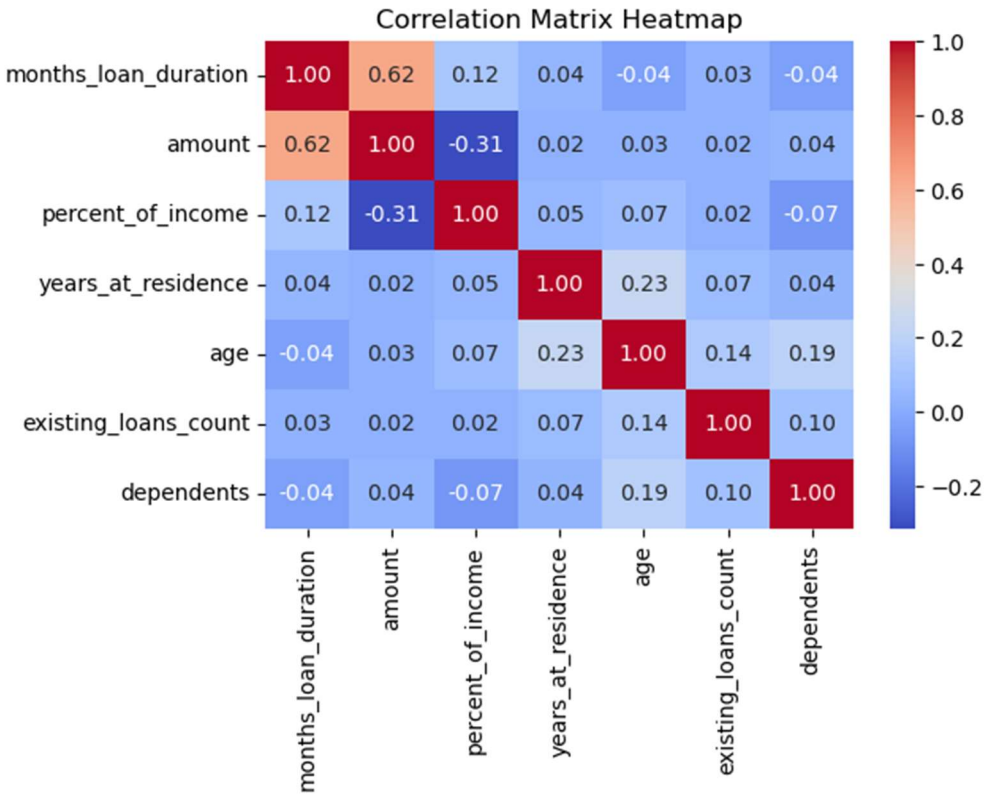
A notable proportion of customers classified with 'very good' and 'perfect' credit history end up defaulting on loans. This observation suggests a potential need for a reassessment of the credit rating system that categorizes customers. While the absolute numbers of such cases are relatively low, the percentage of customers defaulting is higher, warranting further investigation. It's essential to note that the dataset has a limited size of 1000 observations, and a larger dataset may provide a clearer understanding of the situation.



Even without a detailed analysis, a visual inspection of the graph suggests that individuals with a longer employment history tend to have a lower proportion of defaulters. This observation indicates that customers who are employed for an extended period are more financially stable and exhibit fiscal prudence, as reflected in their higher compliance with loan repayments. However, a thorough analysis is required to validate and draw definitive conclusions from this observation.



In general, with a few exceptions, the correlation matrix, heatmap, and pair plot suggest a weak relationship among the numerical predictors. The pair plot, which includes scatterplots (not displayed here), provides a visual representation of these relationships, reinforcing the observation of generally weak correlations among the numerical features.



Heatmap of the correlation matrix among the numerical columns

In summary, the EDA phase provided valuable insights into the dataset's characteristics and behaviour. It helped uncover potential relationships between various features and the likelihood of default. Additionally, it identified key variables that could play a crucial role in predicting loan defaults. These insights served as a strong foundation for the subsequent stages of machine learning model development and evaluation.

Methods used to build ML Models:

Data Preprocessing: At the beginning of the data preprocessing stage, tasks were carried out to ready the dataset for modelling. Nominal categorical features were encoded using one-hot encoding (dummy variables), while ordinal categorical features were encoded based on their ordered ranking with the help of the scikit-learn Python library package.

To ensure uniform scaling across features, standardization was applied using the Standard Scaler method. Following that, the dataset was divided into training and testing sets, maintaining a 75-25 ratio.

Stratification was employed to preserve the class distribution, considering the dataset's nature as an imbalanced classification problem.

Target variable ['default'] (indicates whether the customer defaulted on the loan or not)

Predictors variables

- **Numerical Variables:** ['months_loan_duration', 'amount', 'percent_of_income', 'years_at_residence', 'age', 'existing_loans_count', 'dependents']
- **Categorical Variables:**
  - **Nominal columns:** ['checking_balance', 'purpose', 'savings_balance', 'other_credit', 'housing', 'job', 'phone']
  - **Ordinal columns:** ['credit_history', 'employment_duration']

Model Training and Hyperparameter Tuning:

I explored various supervised machine learning models, including logistic regression, k-nearest neighbours, support vector machines, quadratic discriminant analysis, random forest, gradient boosting, AdaBoost, and XGBoost, to address the prediction of the 'default' class.

For each model, hyperparameter tuning was carried out using the GridSearchCV function and cross-validation on the training set to determine the most effective hyperparameters.

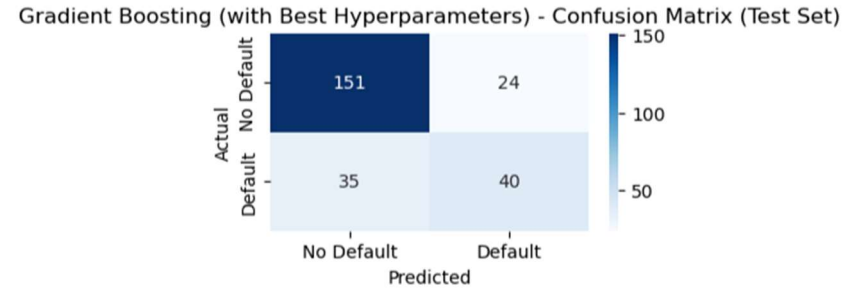
The chosen metric for optimization was 'recall' because the main objective was to minimize false negatives, capturing potential loan defaulters.

Model Fitting and Evaluation:

Following hyperparameter tuning, I applied all the models to the entire training set using the optimal hyperparameters and assessed the model performance on both the training and testing sets. The classification report presented metrics like precision, recall, and F1-score for each class. Additionally, the confusion matrix was visualized to interpret the model's predictions in terms of true positives, true negatives, false positives, and false negatives.

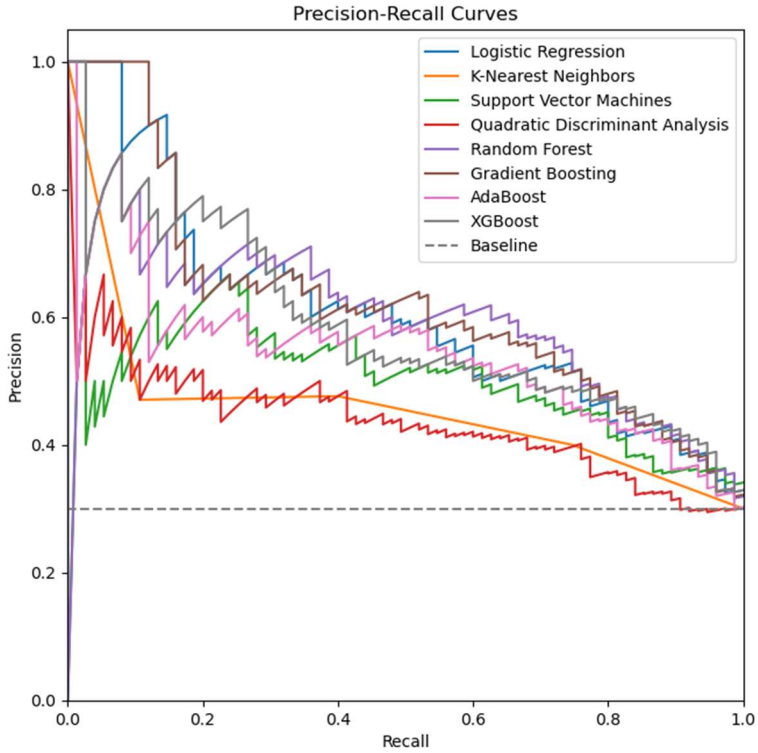
```
Gradient Boosting (with Best Hyperparameters) - Training Set Performance:
              precision    recall  f1-score   support

           0       0.95      0.99      0.97       525
           1       0.98      0.88      0.93       225

    accuracy                           0.96       750
   macro avg       0.97      0.94      0.95       750
weighted avg       0.96      0.96      0.96       750

Gradient Boosting (with Best Hyperparameters) - Confusion Matrix (Training Set):
[[521   4]
 [ 26 199]]
```

```
Gradient Boosting (with Best Hyperparameters) - Test Set Performance:
              precision    recall  f1-score   support

           0       0.81      0.86      0.84       175
           1       0.62      0.53      0.58        75

    accuracy                           0.76       250
   macro avg       0.72      0.70      0.71       250
weighted avg       0.76      0.76      0.76       250
```

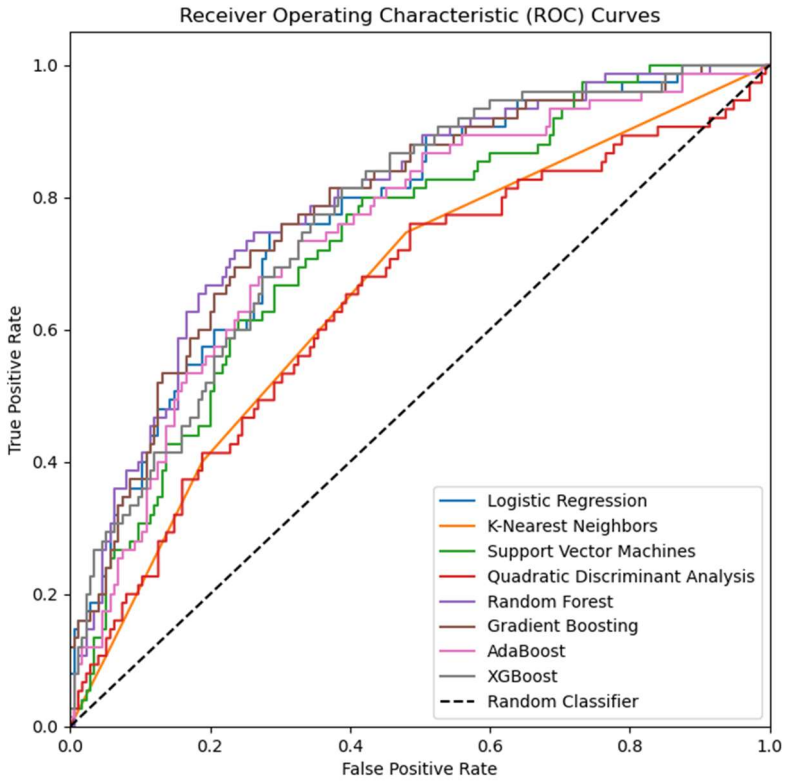


Model Comparison and Selection:

To determine the final model, I utilized precision-recall (PR) curves, along with their associated AUC values, for model comparison. Given the imbalanced nature of the dataset (unequal class distribution), the primary metric for evaluating model performance was the area under the precision-recall curve (AUC-PR). The model with the highest AUC-PR was considered the most effective. Furthermore, I applied a customized threshold to the selected model to optimize 'recall' performance. These analyses provided insights into loan default prediction, emphasizing the trade-offs between precision and recall.



Precision-recall curve for all models (i.e., precision and recall values at different threshold points)



ROC curve (i.e., false positive rate and true positive rate values). Calculate the AUC for the PR-curves to select the best performing model instead of an ROC curve due to imbalanced-class.

Results and Discussion:

The following are the ML Models in Order of Performance (Based on PR-AUC) from lowest to highest, obtained for training on the German bank loan dataset:

1. Quadratic Discriminant Analysis
2. K-Nearest Neighbours
3. Support Vector Machines
4. AdaBoost
5. XGBoost
6. Random Forest
7. Logistic Regression
8. Gradient Boosting

```
AUC-PR Values (Area Under the Curve in a Precision-Recall plot):
                            Model   AUC-PR
                Gradient Boosting 0.619926
              Logistic Regression 0.607566
                    Random Forest 0.587175
                          XGBoost 0.583114
                         AdaBoost 0.539437
          Support Vector Machines 0.494610
              K-Nearest Neighbors 0.457832
Quadratic Discriminant Analysis 0.434089
```

'default' = 1000 datapoints, has 700 rows as 'no' ('0') and 300 rows as ('yes' or '1'). This can be considered as a moderate case of imbalanced classification problem. Hence AUC-PR values are used for model comparison instead of AUC-ROC

Now, let's examine the project results based on the PR-AUC (Precision-Recall Area Under the Curve) scores. These scores assess the balance between precision and recall for each model. PR-AUC holds particular significance in imbalanced classification problems, such as loan default prediction (or similar scenarios in the medical sector), where the positive class (defaults) is a minority. Higher PR-AUC values signify superior performance in accurately identifying instances of loan defaulters.

Low-Moderate Performing Models:

Quadratic Discriminant Analysis (QDA) exhibits relatively low performance with a PR-AUC score of 0.396, the lowest among all models. It struggles to effectively balance precision and recall, leading to less accurate identification of default cases. K-Nearest Neighbors (KNN) performs better than QDA with a PR-AUC of 0.458 but still falls within the lower range of PR-

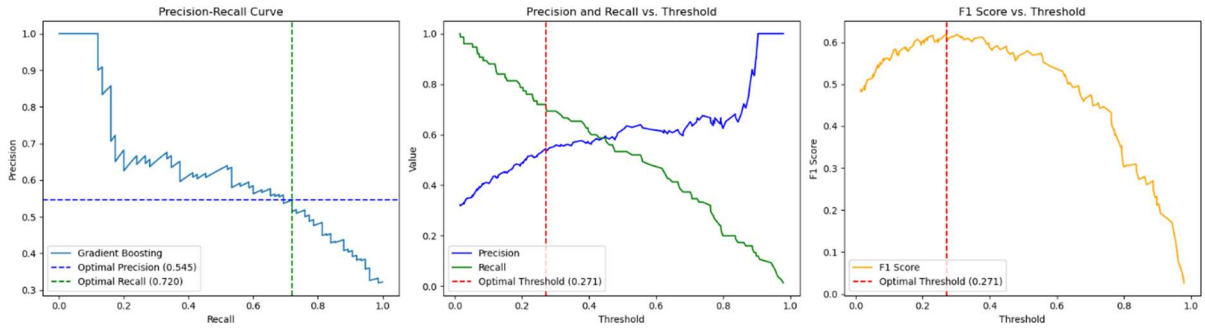
AUC scores. Support Vector Machines (SVM) demonstrates improved balance between precision and recall compared to KNN, achieving a PR-AUC of 0.495.

Best Performing Models:

Among the models considered, AdaBoost with a PR-AUC of 0.539 demonstrates improved performance compared to earlier models, falling within the moderate range. It effectively balances precision and recall, making it more proficient in identifying cases of default. XGBoost, with a PR-AUC of 0.584, further enhances performance beyond AdaBoost. This substantial increase in the PR-AUC score signifies its capability to capture more positive instances while maintaining reasonable precision. Random Forest, with a PR-AUC of 0.587, performs similarly to XGBoost but only marginally better. Logistic Regression, achieving a higher PR-AUC score of 0.608, stands out as one of the top-performing models. It successfully strikes a balance between precision and recall.

Gradient Boosting, both with and without a custom threshold, showcased the most exceptional performance, achieving a PR-AUC score of 0.620. This model excelled by attaining the highest "recall" while maintaining a satisfactory level of precision. As a result, Gradient Boosting emerges as a robust choice for identifying instances of 'default.' The emphasis on recall is particularly significant in this context, aiming to minimize false negatives and mitigate the bank's risk by accurately identifying potential default cases.

```
Optimal Threshold that maximizes F1-score (i.e., optimizing both Precis
ion and Recall): 0.27118255931448304
Optimal Precision: 0.5454545454545454
Optimal Recall: 0.72
```

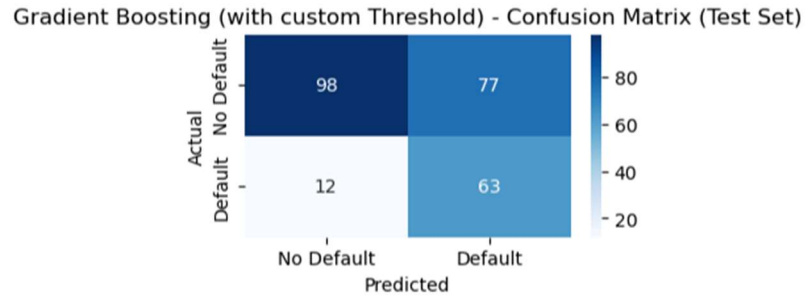


Threshold value that maximizes F1-score performance (i.e., optimizing both Precision and Recall)

Applying a custom threshold of '0.14,' which represents a further 50% reduction from the optimal threshold maximizing F1-score (balancing Precision and Recall), to Gradient Boosting significantly improved its ability to detect 'default' cases. This adjustment in precision-recall trade-off resulted in enhanced overall performance, effectively capturing more true positive instances. This aligns with the bank's objective of identifying potential loan defaulters.

Gradient Boosting, equipped with its best hyperparameters and the custom threshold, achieved impressive results in terms of PR-AUC score. The model adeptly identifies a substantial portion of actual default instances while maintaining a reasonable level of precision. This balance is pivotal for the bank's goal of minimizing false negatives, ensuring that it doesn't miss actual default cases. While the precision for defaults may not be exceptionally high, the elevated recall ensures that the model identifies a significant proportion of genuine defaulters. Considering the goal of maximizing the identification of default cases while managing precision, Gradient Boosting with a custom threshold emerges as the most suitable choice.

```
Custom probability decision threshold: 0.14

Gradient Boosting (with custom Threshold) - Training Set Performance:
              precision    recall  f1-score   support

           0       1.00      0.70      0.82       525
           1       0.59      1.00      0.74       225

    accuracy                           0.79       750
   macro avg       0.79      0.85      0.78       750
weighted avg       0.87      0.79      0.80       750

Gradient Boosting (with custom Threshold) - Confusion Matrix (Training Set):
[[369 156]
 [  1 224]]


Gradient Boosting (with custom Threshold) - Test Set Performance:
              precision    recall  f1-score   support

           0       0.89      0.56      0.69       175
           1       0.45      0.84      0.59        75

    accuracy                           0.64       250
   macro avg       0.67      0.70      0.64       250
weighted avg       0.76      0.64      0.66       250
```



Despite the promising results achieved in predicting loan defaults using Gradient Boosting with a custom threshold, there are several opportunities for improvement and expansion in this project. These possibilities include exploring feature engineering techniques, employing ensemble methods tailored for large datasets, and adopting advanced techniques for

handling imbalanced datasets. As we delve into the limitations of this project, we will also explore potential directions that similar projects could consider for future enhancements.

Limitations and Future Directions:

The dataset's relatively small size, with only 1000 samples, may impact the models' ability to generalize to larger populations. Future efforts could focus on collecting a more extensive dataset that encompasses a diverse range of customer profiles, loan types, and economic conditions to enhance the robustness of the models.

Additionally, the dataset exhibited class imbalance, with a higher number of non-default cases compared to default cases. This imbalance can introduce bias in model results, where the model might favour the majority class. While techniques like oversampling and under-sampling were not fully explored in this project, future work should consider experimenting with these methods to achieve a more balanced representation of classes and enhance overall model performance. It's essential to strike a balance between addressing class imbalance and optimizing the model for better 'recall' performance, especially by using a custom threshold.

In the project, the dataset's features were utilized in their original form without extensive feature engineering or dimensionality reduction. However, exploring additional features derived from existing ones or integrating external data sources could unveil hidden patterns and enhance model predictions. For example, calculating metrics like debt-to-income ratios, credit utilization ratios, or incorporating economic indicators might offer valuable insights into customers' financial health and default risk.

Regarding hyperparameter tuning and model selection, while the project involved tuning hyperparameters for each model, further refinement and experimentation with different combinations could lead to improved model performance. Exploring advanced techniques like neural networks may be considered for future directions to enhance predictive capabilities.

The interpretability of complex models, such as Gradient Boosting, can be challenging. Although Gradient Boosting demonstrated impressive performance, understanding the factors contributing to its predictions remains elusive. In contrast, the Logistic Regression model, with its simplicity and higher interpretability, performed well in this project. It allows for a quantitative understanding of the model and exploration of the individual features' contributions.

Considering external data sources, integrating additional information such as economic indicators, industry trends, or customer behaviour data could further enrich the predictive power of the models.

For instance, incorporating macroeconomic indicators like unemployment rates or inflation could provide contextual information about customer default patterns and contribute to more accurate predictions.

Conclusion:

In this project, my objective was to create a predictive model for predicting loan defaults using historical data from a German bank. I followed a systematic approach that involved exploring and visualizing the data and applying various machine learning algorithms. The goal was to build a reliable model that could help the bank identify potential loan defaulters and manage financial risks effectively.

Throughout the project, I conducted a thorough analysis, uncovering insightful patterns through Exploratory Data Analysis (EDA). This phase shed light on the relationships between different features and their impact on the likelihood of loan defaults. I explored a variety of machine learning models, fine-tuning each of them and evaluating their performance using key metrics and the area under the Precision-Recall curve (AUC-PR). This approach allowed me to prioritize models that effectively balanced precision and recall.

Among the models, Gradient Boosting, a powerful ensemble technique, emerged as the most promising for loan default prediction, especially with certain custom thresholds. It consistently demonstrated robust performance in terms of AUC-PR and recall, making it proficient at identifying potential default cases while minimizing false negatives. By leveraging the strengths of Gradient Boosting, I achieved a balance between precision and recall, enhancing the bank's ability to accurately predict loan defaults.

However, it's important to acknowledge the limitations of this study, including the relatively small dataset, class imbalance, and the potential for more advanced feature engineering. Despite these limitations, the findings emphasize the importance of employing sophisticated machine learning techniques in the field of financial risk assessment.