Report: **Loan Default Prediction Project**

The objective of the “Loan Default Prediction Project” is to build a reproducible and publishable project that demonstrates the process of developing and evaluating Machine Learning models for loan *default* prediction in the Banking Financial sector.

Introduction

The German bank dataset provides a valuable glimpse into the domain of loan default prediction, a critical challenge faced by financial institutions. In this context, the bank is confronted with the pressing issue of identifying customers who are likely to default on their loans. The dataset comprises historical records of customers who have availed loans from the bank, and the primary objective is to develop a predictive machine learning model that can accurately forecast whether a customer will default or not, by drawing insights 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 - refers to the amount of money available in a 'checking account' (a.k.a current account) of the customer for everyday financial transactions ("unknown", "< 0 DM", "1 - 200 DM" and "> 200 DM")
- months_loan_duration - The duration, in months, since the loan was taken
- credit_history - The credit history of each customer ("good", "critical", "poor", "very good" and "perfect")
- purpose - The purpose for which the loan was taken ("furniture/appliances", "car", "business", "education" and "renovations")
- amount - The amount of loan taken by the customer
- savings_balance - refers to the amount of money available in a 'savings account' of the customer for accumulating funds over time and earning interest ("< 100 DM", "unknown", "100 - 500 DM", "500 - 1000 DM" and "> 1000 DM")
- employment_duration - The duration of the customer's employment ("1 - 4 years", "> 7 years", "4 - 7 years", "< 1 year" and "unemployed")
- percent_of_income - The installment rate, expressed as a percentage of disposable income, indicates the portion of income being utilized to make loan payments
- years_at_residence - The duration of the customer's current residence
- age - The age of the customer
- other_credit - Whether the customer has taken any other credits ("none", "bank" and "store")
- housing - The type of housing the customer has ("own", "rent" and "other")
- existing_loans_count - signifies the quantity of ongoing loans (currently active credit lines) already held by a customer with this bank
- job - The job type of the customer ("skilled", "unskilled", "management" and "unemployed")
- dependents - Whether the customer has any dependents
- phone - Whether the customer has a phone ("no" and "yes")
- default - Default status (Target column - "no" and "yes"): The target variable indicating whether the customer defaulted on the loan or not.

NOTE: "DM" stands for "Deutsche Mark" (previously legal currency of Germany)

Contextual details to understand the German Bank dataset.

The stakes are high in the banking industry, where ‘default’ prediction can significantly impact a bank's financial health and decision-making. The bank's intent to leverage machine learning models to predict defaulting customers underscores a proactive approach to risk management. With access to an array of customer-specific information, including factors such as credit history, employment duration, savings balance, age, and loan amount, the bank can harness the power of data to enhance its loan approval process and minimize potential losses arising from loan defaults. A reliable predictive model can mitigate risks, optimize lending practices, and ensure prudent risk management financial strategies.

Some Questions that were answered through the course of this project:

- Which machine learning model performs best in predicting loan defaults based on the provided German bank dataset? Comparing the performance of various models to select the most effective approach for the bank's predictive needs.
- How can we optimize model performance to minimize false negatives and identify potential defaulters more effectively (to achieve the highest recall performance)?
- How does a customer's credit history impact the likelihood of loan default? This question can provide insights into the significance of creditworthiness in predicting loan defaults.
- Many such questions are answered as EDA and Data Visualization was performed. But final conclusions on them needs to be done with further deeper study of the same.

Methods and Materials

Exploratory Data Analysis (EDA)

Data Understanding, Cleaning and Preparation:

I began by thoroughly understanding the dataset's contents and column descriptions to gain insights into the features and their meanings. I conducted a comprehensive exploratory data analysis (EDA) to identify any potential issues with missing values, typos, and duplicates. There were no missing values in the dataset as such, but some typos were corrected. I recognized the features as numerical variables and categorical variables (further into nominal type and ordinal type) for further analysis. I conducted summary statistics analyses to gain insights into the numerical and categorical features. Duplicate values were also checked and removed to ensure data integrity.

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

	count	unique	top	freq
checking_balance	1000	4	unknown	394
credit_history	1000	5	good	530
purpose	1000	5	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 Qualititive columns

```
# Summary Statistics of the numerical columns
df.describe().T
```

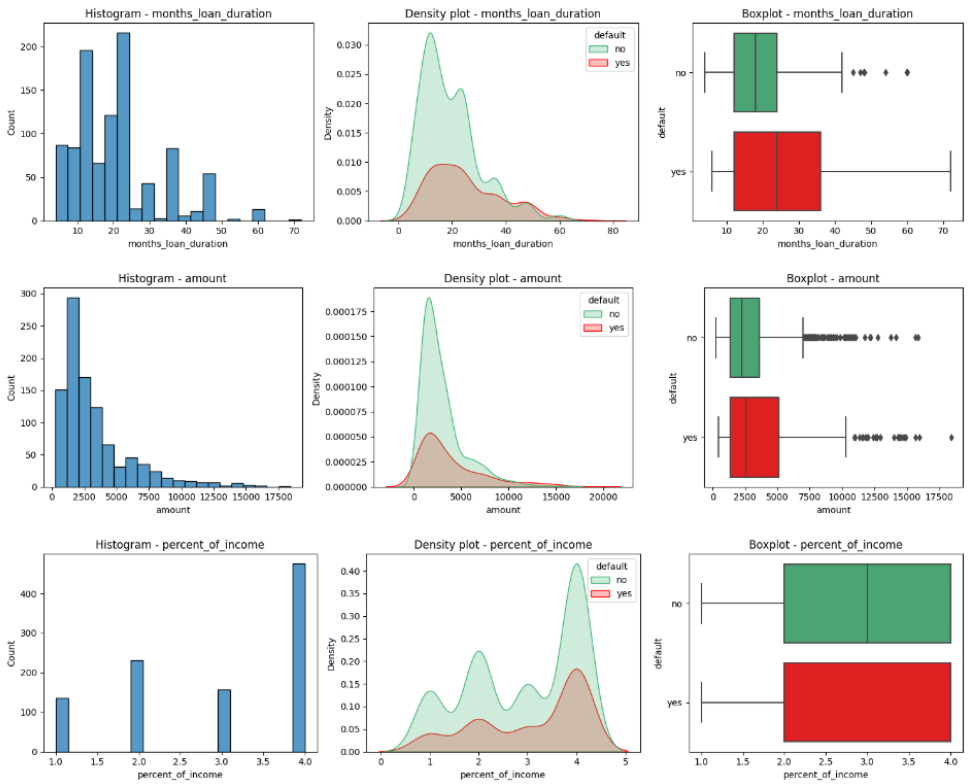
	count	mean	std	min	25%	50%	75%	max
months_loan_duration	1000.0	20.903	12.058814	4.0	12.0	18.0	24.00	72.0
amount	1000.0	3271.258	2822.736876	250.0	1365.5	2319.5	3972.25	18424.0
percent_of_income	1000.0	2.973	1.118715	1.0	2.0	3.0	4.00	4.0
years_at_residence	1000.0	2.845	1.103718	1.0	2.0	3.0	4.00	4.0
age	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

Summary Statistics for the Numerical Qualitative 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.

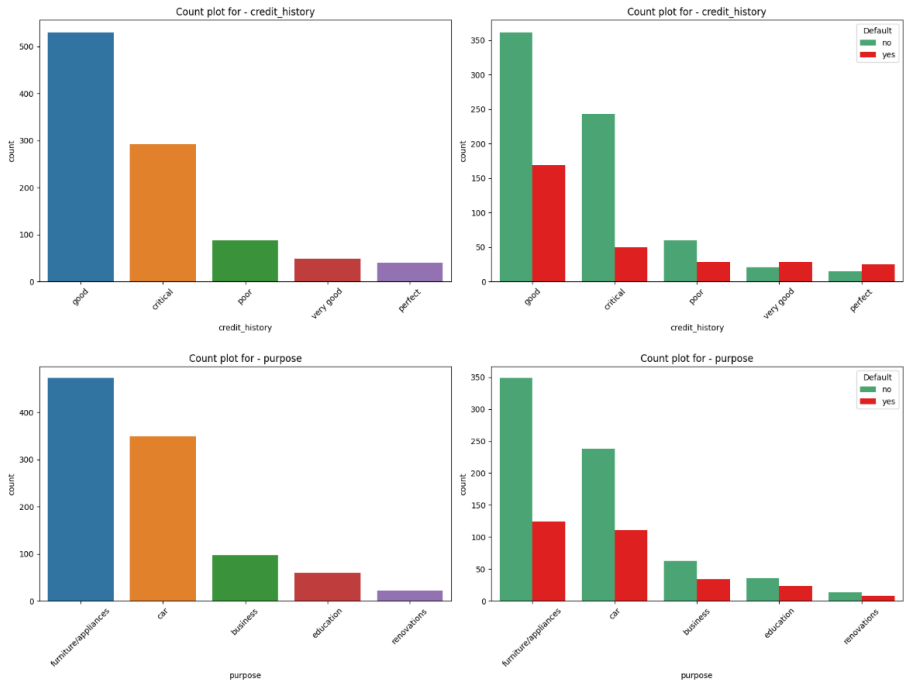
Data Visualization:

The EDA phase involved a series of visualizations and analyses to better understand the relationships between different features and their impact on loan ‘default’ behaviour. I used histograms, density plots, and boxplots to visualize the distribution of numerical features and identify potential outliers. Correlation between numerical features was explored using a pairplot and heatmap. EDA also raises some interesting Research Questions that hint at an answer through the plotted graphs and may need further detailed study to establish definitive conclusions.



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

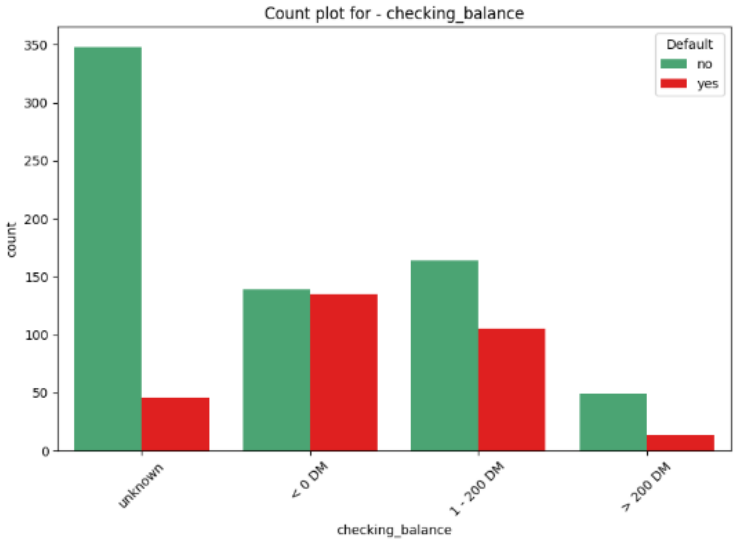
For categorical variables, I used count plots to analyse the distribution pattern of customers across different categories. I compared the distributions with and without considering the 'default' status as the hue, which allowed me to explore differences in behaviour between customers who defaulted and those who didn't.



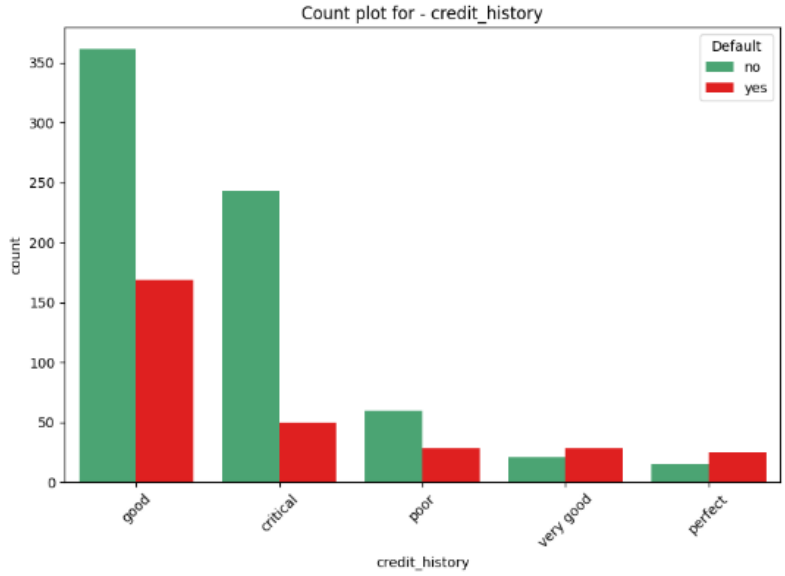
EDA for some categorical columns with and without 'default' as the hue

Some Insights from EDA:

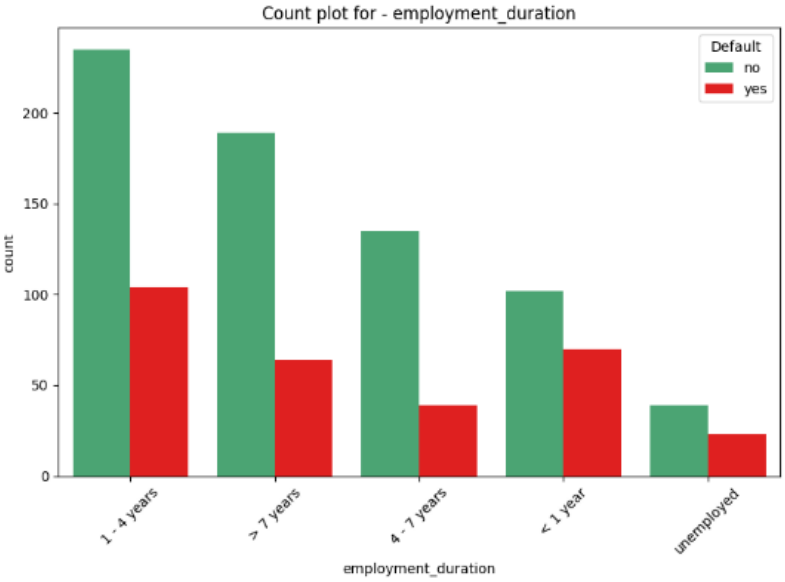
- ‘checking_balance’ refers to the amount of money available in a 'checking account' (a.k.a current account) of the customer for everyday financial transactions. This is a very liquid part of the customers finances. From the Data Visualization we can observe that the proportion of defaulters increases when this liquid balance starts drying up.



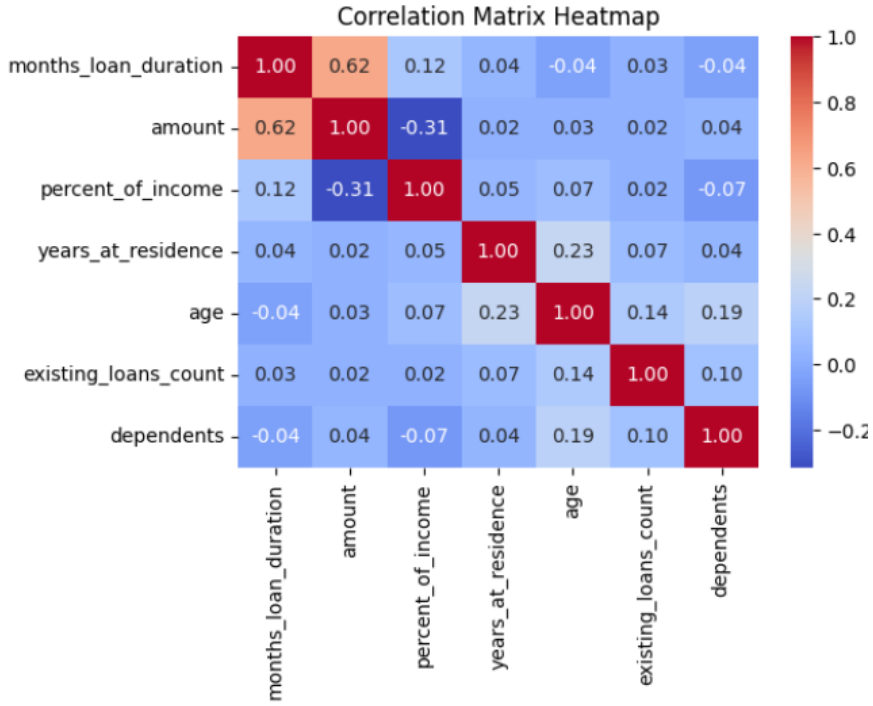
- There is a high proportion of defaulters among the customers who have ‘very good’ and ‘perfect’ credit history. This probably necessitates a review into the credit rating system that categorizes customers. Though the absolute numbers are low, the proportion of customers defaulting is higher, hence this needs further study. This could also be the case due to low volume in this dataset of just 1000 observations and a higher sample of data can reveal the picture better.



- Though this needs further analysis, just by eye-balling the graph we can see that, the higher number of years someone is employed and working, the proportion of defaulters’ in that category is lesser. This shows the fiscal prudence of the customer who is financially stable as he is employed for a longer duration of time and is compliant with their loan repayment.



- By and large, except couple of them, the correlation matrix and the heatmap indicates weak relationship among the numerical predictors. The same can be visualized in the pairplot as well (scatterplot is not displayed here).



Heatmap of the correlation matrix among the numerical columns

Overall, the EDA phase allowed me to gain an understanding and behaviour of the dataset, uncover potential relationships between features and 'default' behaviour, and identify variables that might play a crucial role in predicting loan defaults. These insights provided a solid foundation for the subsequent steps involving machine learning model development and evaluation.

Methods used to build ML Models:

Data Preprocessing:

Initially data preprocessing tasks were performed to prepare the dataset for modelling. This involved encoding nominal categorical features using one-hot encoding (dummy variables) and encoding the ordinal categorical features by their ordered ranking using the scikit-learn python library package. To ensure consistent scaling, I standardized all the features using the Standard Scaler method. The data was then split into training and testing sets with a 75-25 ratio, using stratification to maintain the class distribution (because this dataset is a case of 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']

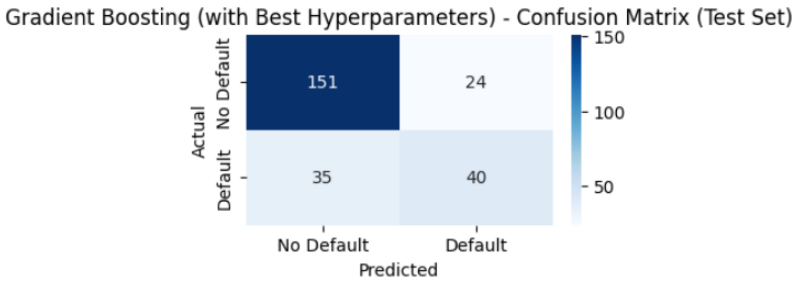
Model Training and Hyperparameter Tuning:

I explored a variety of supervised machine learning models, including ['Logistic Regression', 'K-Nearest Neighbors', 'Support Vector Machines', 'Quadratic Discriminant Analysis', 'Random Forest', 'Gradient Boosting', 'AdaBoost', 'XGBoost'] to solve the 'default' class prediction. For each model, hyperparameter tuning was conducted using GridSearchCV function and cross-validation on the training set to identify the best-performing hyperparameters. The evaluation metric of choice for optimization was 'recall', as the primary focus was on minimizing false negatives to capture potential loan defaulters.

Model Fitting and Evaluation:

After hyperparameter tuning, I fitted all the models on the entire training set using the optimal hyperparameters and then evaluated the model performance on both the training and testing sets. The classification report provided metrics such as precision, recall, and F1-score for each class. The confusion matrix was also visualized to understand the model's predictions in terms of true positives, true negatives, false positives, and false negatives.

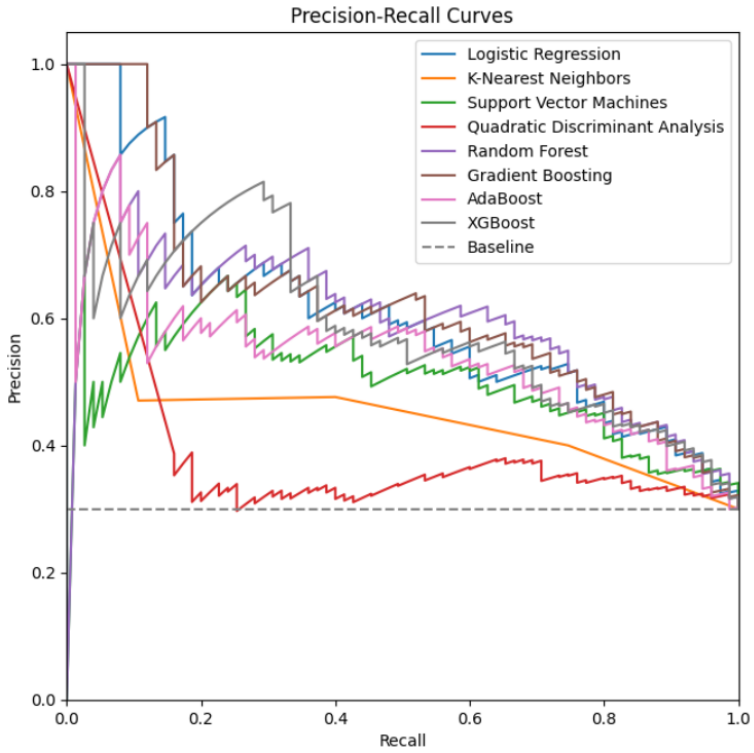
```
Gradient Boosting (with Best Hyperparameters) - Training Set Performance:
              precision    recall  f1-score   support

           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
```



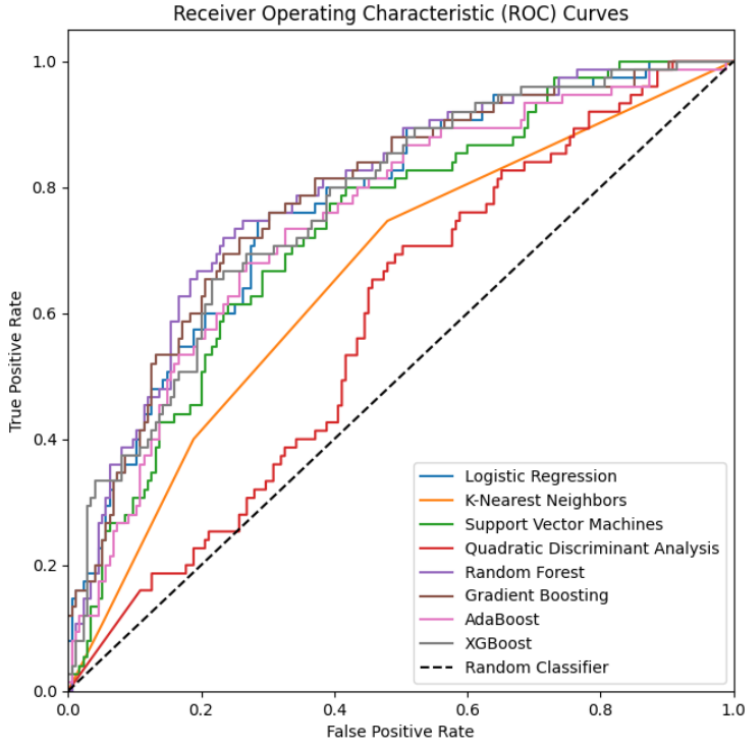
Model Comparison and Selection:

To select the final model, I used the **precision-recall (PR) curves** (and ROC curves for reference), along with their corresponding **AUC values**, for model comparison. The area under the precision-recall curve (AUC-PR) was the primary metric for assessing model performance due to the imbalanced nature of the dataset (unequal number of class classification problem). The model with the highest AUC-PR was considered the best performer. Additionally, I applied a custom threshold to

the chosen model to achieve optimal ‘recall’ performance. The analyses revealed insights into loan default prediction, highlighting 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 Neighbors
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	0.584157
AdaBoost	0.539437
Support Vector Machines	0.494610
K-Nearest Neighbors	0.457832
Quadratic Discriminant Analysis	0.395876

'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.

Let's analyse the results of the project based on the PR-AUC (Precision-Recall Area Under the Curve) scores, which measure the trade-off between precision and recall for each model. PR-AUC is particularly important in imbalanced classification problems like loan default prediction (or many such use-case in the medical sector), where the positive class (defaults) is a minority. Higher PR-AUC values indicate better performance in identifying positive instances of loan defaulters.

Low-Moderate Performing Models:

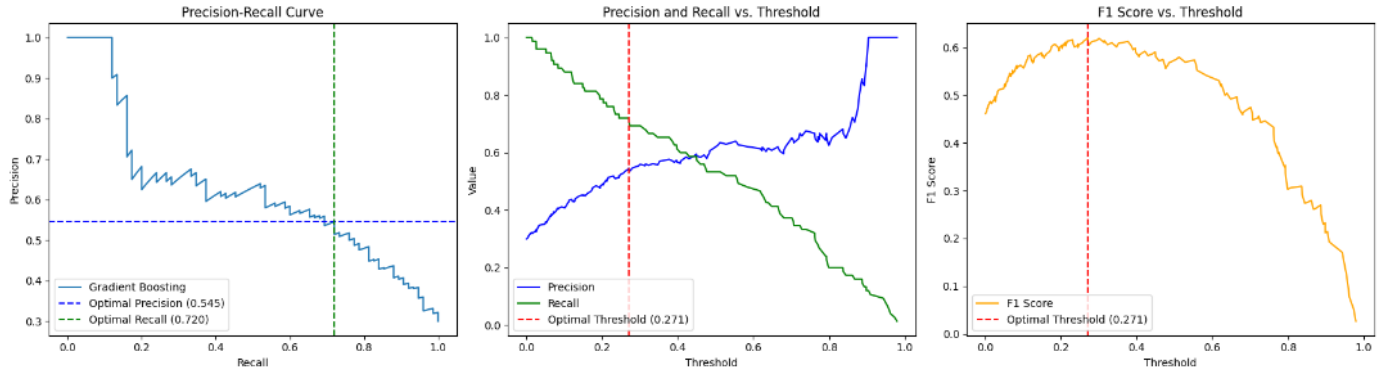
Quadratic Discriminant Analysis (QDA) with PR-AUC 0.396 has a relatively low performance. With its PR-AUC score being the lowest among the models. It struggles to balance precision and recall effectively, resulting in less accurate identification of default cases. K-Nearest Neighbors (KNN) with PR-AUC 0.458 performs better than QDA but still falls in the lower range of PR-AUC scores. Support Vector Machines (SVM) with PR-AUC 0.495 demonstrated a better balance between precision and recall compared to KNN.

Best Performing Models:

AdaBoost with PR-AUC 0.539 showcases better performance than the previous models but still falls in the moderate range. It strikes a balance between precision and recall, making it more effective in identifying default cases. XGBoost with PR-AUC 0.584 further improves upon the performance of AdaBoost. It shows a significant increase in PR-AUC score, indicating its ability to capture more positive instances while maintaining reasonable precision. Random Forest with PR-AUC 0.587 performance is similar to XGBoost but only slightly better. Logistic Regression with PR-AUC 0.608 achieves a higher PR-AUC score compared to previous models. It strikes a balance between precision and recall, making it one of the *top-performing models*.

Gradient Boosting (with and without Custom Threshold): Gradient Boosting demonstrated the *best performance* with a PR-AUC score of 0.620. This model achieved the highest "recall" while maintaining a reasonable level of precision, making it a strong candidate for identifying 'default' cases. The importance of recall is crucial in this context because the focus is on minimizing false negatives (missed default cases) to mitigate the bank's risk.

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

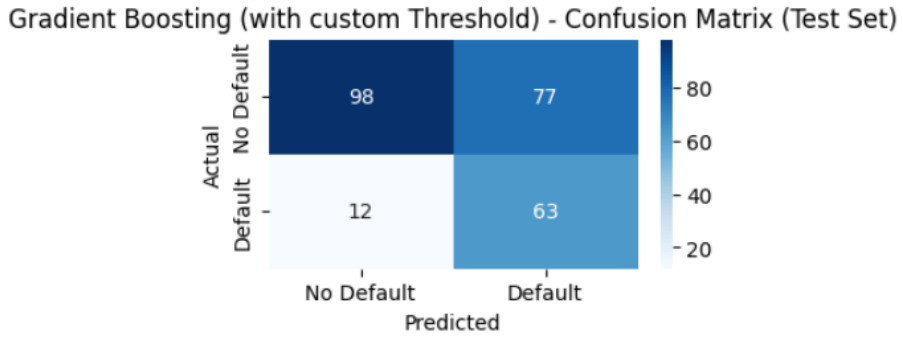


Threshold value that maximizes F1-score performance (i.e., optimizing both Precision and Recall)

Applying a custom threshold of '0.14', which is about 50% further reduction from the *optimal threshold* that maximizes F1-score (i.e., optimizing both Precision and Recall), to Gradient Boosting further enhanced its ability to identify 'default' cases. This trade-off in precision-recall resulted in better overall performance in terms of capturing true positive instances, which aligns with the bank's goal of identifying potential loan defaulters.

Gradient Boosting, with its best hyperparameters and custom threshold, achieves impressive results in terms of PR-AUC score. This model effectively identifies a significant portion of the actual default instances while maintaining a reasonable level of precision. This is crucial for the bank's objective of reducing false negatives (missing actual default cases). While the precision for defaults may not be as high, the high recall ensures that the model identifies a substantial portion of actual defaulters. For the goal of identifying as many default cases as possible while managing precision, Gradient Boosting with a custom threshold is 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
```



While this project presented promising results in predicting loan defaults using Gradient Boosting with a custom threshold, there are several avenues for improvement and expansion, such as feature engineering, ensemble methods (for a large dataset), and advanced techniques for handling imbalanced datasets. With this, let us will look at the limitations of this project and potential directions such similar projects can adopt in the future.

Limitations and Future Directions:

Sample Size and Generalizability: The dataset's size, consisting of 1000 samples (relatively small sample), might affect the models' generalizability to larger populations. Future directions could involve collecting extensive dataset with a diverse range of customer profiles, loan types, and economic conditions to enhance model robustness.

Class Imbalance: The dataset exhibited class imbalance, with a higher number of non-default cases compared to default cases. This imbalance can lead to biased model results, where the model may favour the majority class. While techniques like oversampling and under-sampling were not fully explored in this project, future work should involve experimenting with these methods to achieve a more balanced representation of classes and enhance model performance (though preference was given to better ‘recall’ performance using a custom threshold).

Feature Engineering: The dataset's features were utilized in their original form without extensive feature engineering (dimensionality reduction). Exploring additional features derived from the existing ones or incorporating external data could potentially uncover hidden patterns and improve model predictions. For instance, calculating debt-to-income ratios, credit utilization ratios, or economic indicators could provide valuable insights into customers' financial health and default risk.

Hyperparameter Tuning and Model Selection: While the project involved hyperparameter tuning for each model, further fine-tuning and exploring different combinations of hyperparameters could lead to better-performing models. Future directions could include experimenting with more advanced techniques such as neural networks, which may offer improved predictive capabilities.

Model Interpretability: Interpreting the decisions of complex models like Gradient Boosting can be challenging. While Gradient Boosting demonstrated impressive performance, understanding the factors that contribute to its predictions remains elusive. In this project itself, the Logistic Regression model also performs well with a high AUC-PR scores. It is a simple model with higher interpretability with parametric values that can be used to quantitatively understand the model better and the contribution of the individual features to the model can be explored.

External Data Sources: Incorporating external data sources, such as economic indicators, industry trends, or customer behaviour data, could further enrich the predictive power of the models. For instance, macroeconomic indicators like unemployment rates or inflation could provide context to customer default patterns and contribute to more accurate predictions.

Conclusion:

In this project, I embarked on a journey to develop a predictive model for loan default prediction using historical data from a German bank. I undertook a systematic approach, encompassing data exploration, visualization, and the application of various machine learning algorithms. The goal was to build a reliable and accurate model that could assist the bank in identifying potential loan defaulters and mitigating financial risks.

Through rigorous analysis, I uncovered insightful patterns within the dataset. Exploratory Data Analysis (EDA) shed light on the relationships between various features and their impact on loan default probabilities. I delved into a range of machine learning models and each model was meticulously fine-tuned and evaluated, considering key performance metrics and the area under the Precision-Recall curve (AUC-PR) to prioritize models that effectively managed the trade-off between precision and recall.

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

However, it's crucial to acknowledge the limitations of this study, including the relatively small dataset, class imbalance, and the need for more advanced feature engineering. Despite these limitations, the findings underscore the significance of employing sophisticated machine learning techniques in the realm of financial risk assessment.