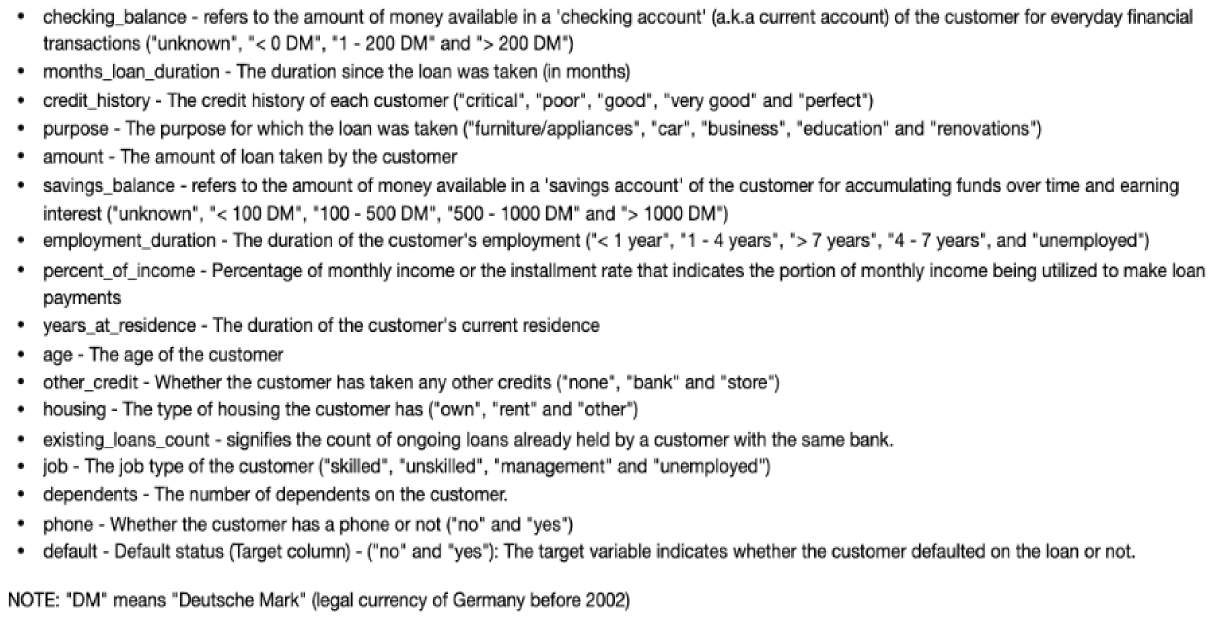
Project Report on
German Bank Loan Default Prediction Project

Introduction

The problem of loan default prediction is one of the most critical and challenging tasks faced by financial institutions, as it has significant implications for both the lenders and the borrowers. A loan default occurs when a borrower fails to repay a loan or credit obligation according to the agreed terms and conditions, resulting in a potential financial loss for the lender, and a negative impact on the creditworthiness and reputation of the borrower. Therefore, it is essential for the lenders to have a reliable and accurate predictive model that can identify customers who are likely to default on their loans, and help the lender make informed and optimal decisions for profit generation and risk management.

The German bank dataset is a publicly available dataset that provides a valuable glimpse into the domain of loan default prediction, as it comprises historical records of customers who have availed loans from a German bank. The primary objective is to develop a predictive machine learning model that can accurately forecast whether a customer will default on the loan or not, by drawing insights from various historical features associated with each customer.

The dataset contains 1000 cases and 17 attributes, including both numerical and categorical variables, such as credit_history, savings_balance, age, amount, and employment_duration. The target variable (default) is a binary variable that indicates whether the customer is a default (yes) or not default (no). The attributes detail of the German Loan dataset is given below:



- 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 since the loan was taken (in months)
- credit_history - The credit history of each customer ("critical", "poor", "good", "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 ("unknown", "< 100 DM", "100 - 500 DM", "500 - 1000 DM" and "> 1000 DM")
- employment_duration - The duration of the customer's employment ("< 1 year", "1 - 4 years", "> 7 years", "4 - 7 years", and "unemployed")
- percent_of_income - Percentage of monthly income or the installment rate that indicates the portion of monthly 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 count of ongoing loans already held by a customer with the same bank.
- job - The job type of the customer ("skilled", "unskilled", "management" and "unemployed")
- dependents - The number of dependents on the customer.
- phone - Whether the customer has a phone or not ("no" and "yes")
- default - Default status (Target column) - ("no" and "yes"): The target variable indicates whether the customer defaulted on the loan or not.

NOTE: "DM" means "Deutsche Mark" (legal currency of Germany before 2002)

Attributes details of the German Loan dataset

The following are some of the intriguing questions that I explore and answer in this project:

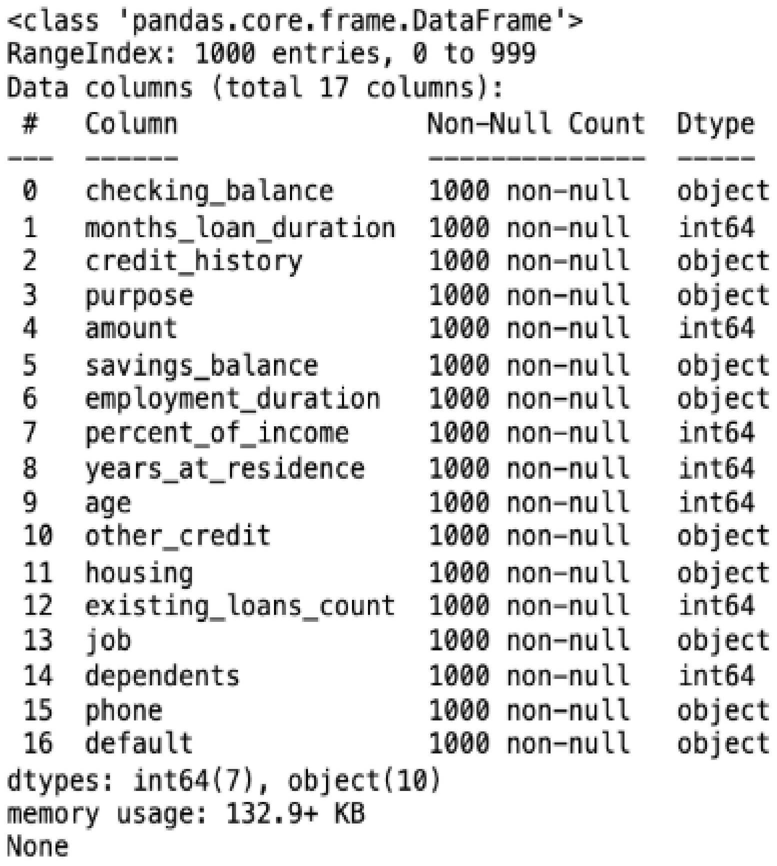
- Which machine learning model performs best in predicting loan defaults based on the features available in the dataset? I will evaluate and compare the performance of various models and choose the most suitable model that meets the specific requirement of the bank.
- What are the best ways to improve model performance and reduce false negatives, so that we can effectively detect potential defaulters by maximizing recall performance?
- How does the past credit behaviour of a customer influence the probability of loan default? This question can reveal the importance of creditworthiness in predicting loan defaults.
- What are the most important features that influence the loan default prediction? I will use feature importance analysis to identify and rank the features that have the most impact on the model performance and the loan default outcome.
- How does the loan amount and duration affect the loan default risk? I will use descriptive statistics and visualizations to examine the relationship between the loan amount, duration, and the loan default status, and to test whether there are significant differences between the defaults and not defaults customers.
- How can we segment the customers into different groups based on their characteristics and behaviour? I will use clustering techniques to group the customers into different clusters, and to analyze the cluster profiles and the loan default rates.

Methods and Materials

Exploratory Data Analysis (EDA)

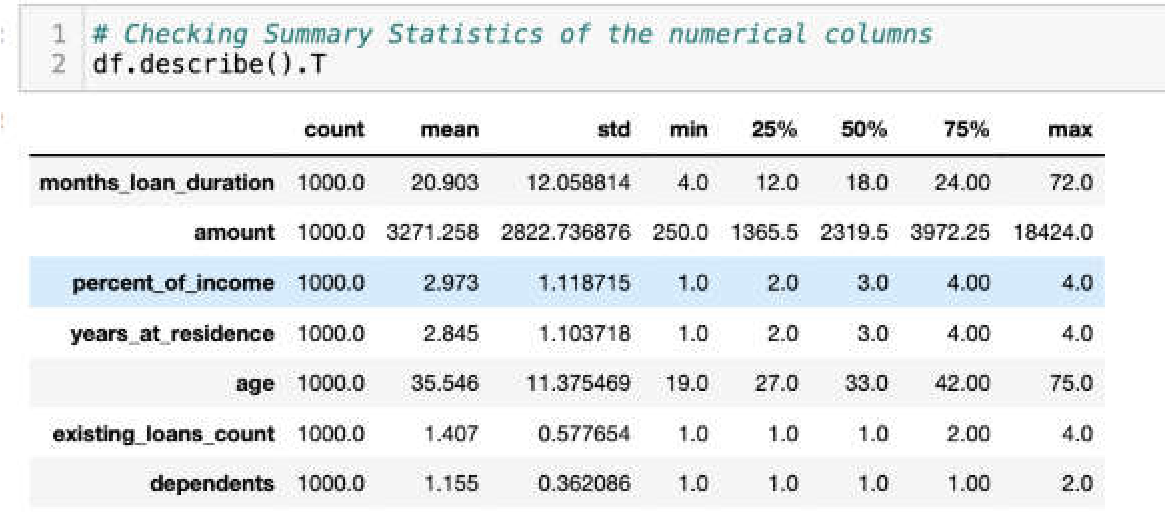
Data Cleaning:

I have conducted a comprehensive exploratory data analysis (EDA) to understand the dataset’s contents and column descriptions, and to identify any potential issues with missing values, typos, and duplicates. I started by exploring the contents and description of dataset to gain insights into meaning and values of the given features. There were no null/missing values or duplicate values in the dataset but I found some typo errors in features and corrected those typos. For better analysis, I have categorised the columns into numerical and categorical features and further divided the categorical columns into nominal and ordinal columns based on the values present in columns.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype 
---  ------                --------------  ----- 
 0   checking_balance      1000 non-null   object
 1   months_loan_duration  1000 non-null   int64 
 2   credit_history        1000 non-null   object
 3   purpose               1000 non-null   object
 4   amount                1000 non-null   int64 
 5   savings_balance       1000 non-null   object
 6   employment_duration   1000 non-null   object
 7   percent_of_income     1000 non-null   int64 
 8   years_at_residence    1000 non-null   int64 
 9   age                   1000 non-null   int64 
 10  other_credit          1000 non-null   object
 11  housing               1000 non-null   object
 12  existing_loans_count  1000 non-null   int64 
 13  job                   1000 non-null   object
 14  dependents            1000 non-null   int64 
 15  phone                 1000 non-null   object
 16  default               1000 non-null   object
dtypes: int64(7), object(10)
memory usage: 132.9+ KB
None
```

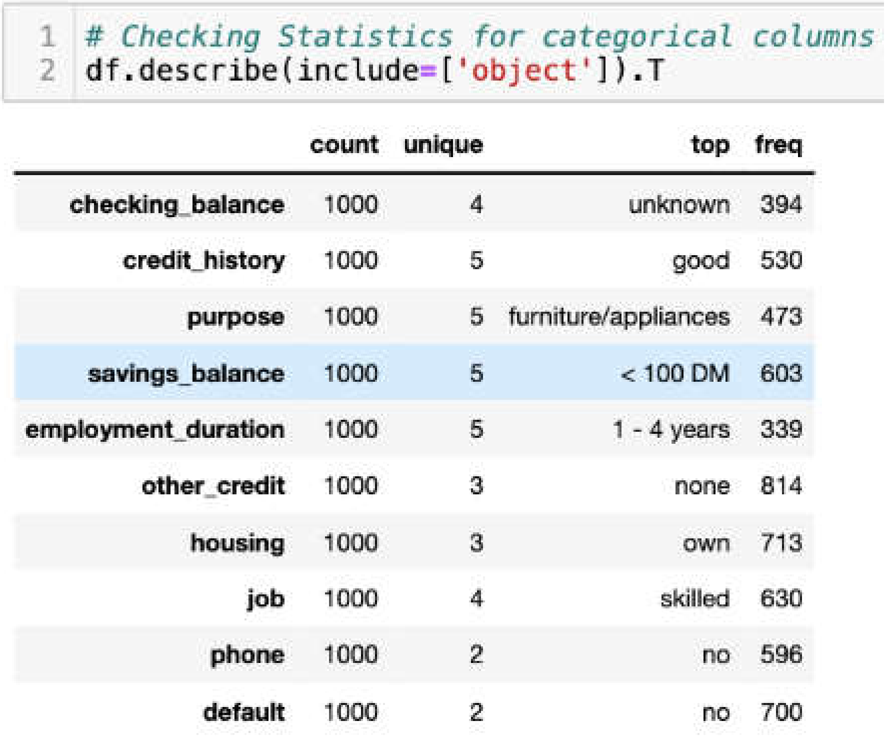
Overall information of dataset



```
# Checking 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 of numerical columns



```
# Checking Statistics for categorical columns
df.describe(include=['object']).T
```

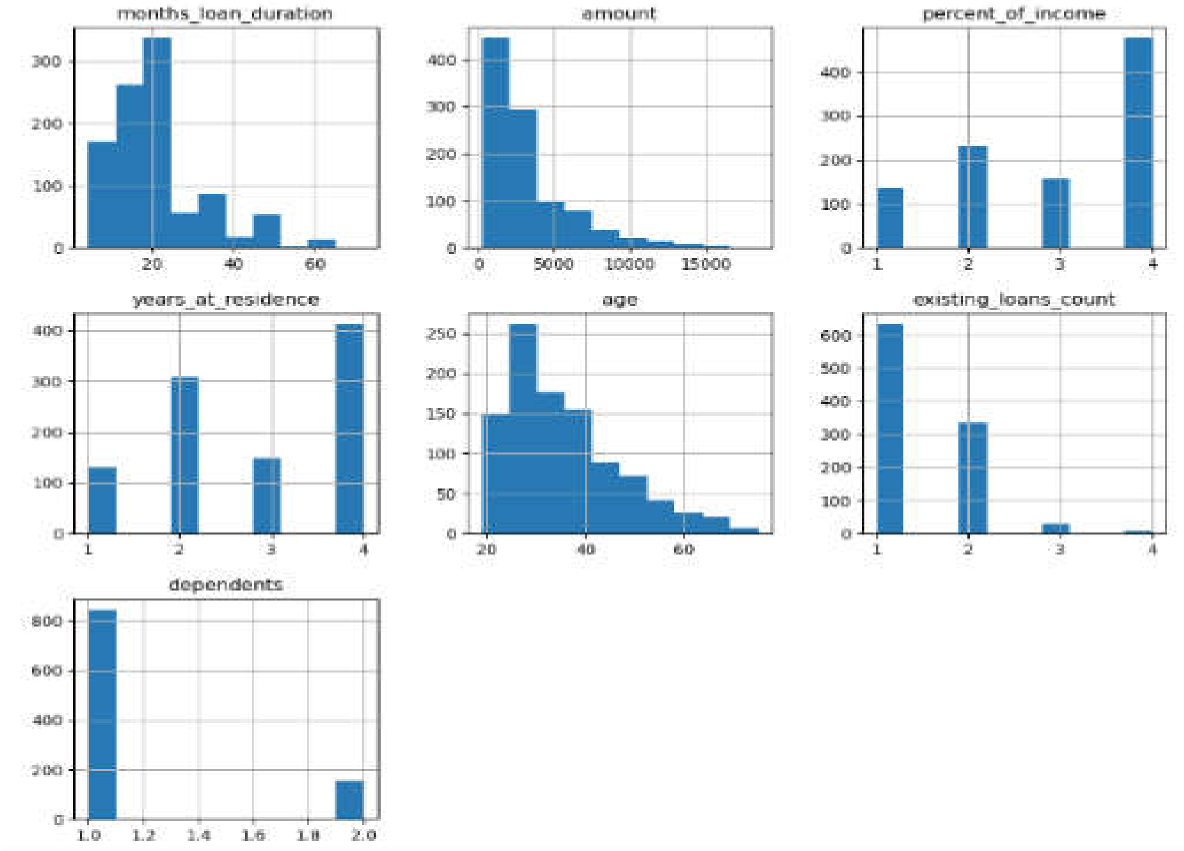
	count	unique	top	freq
checking_balance	1000	4	unknown	394
credit_history	1000	5	good	530
purpose	1000	5	furniture/appliances	473
savings_balance	1000	5	< 100 DM	603
employment_duration	1000	5	1 - 4 years	339
other_credit	1000	3	none	814
housing	1000	3	own	713
job	1000	4	skilled	630
phone	1000	2	no	596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

Summary statistics for categorical columns

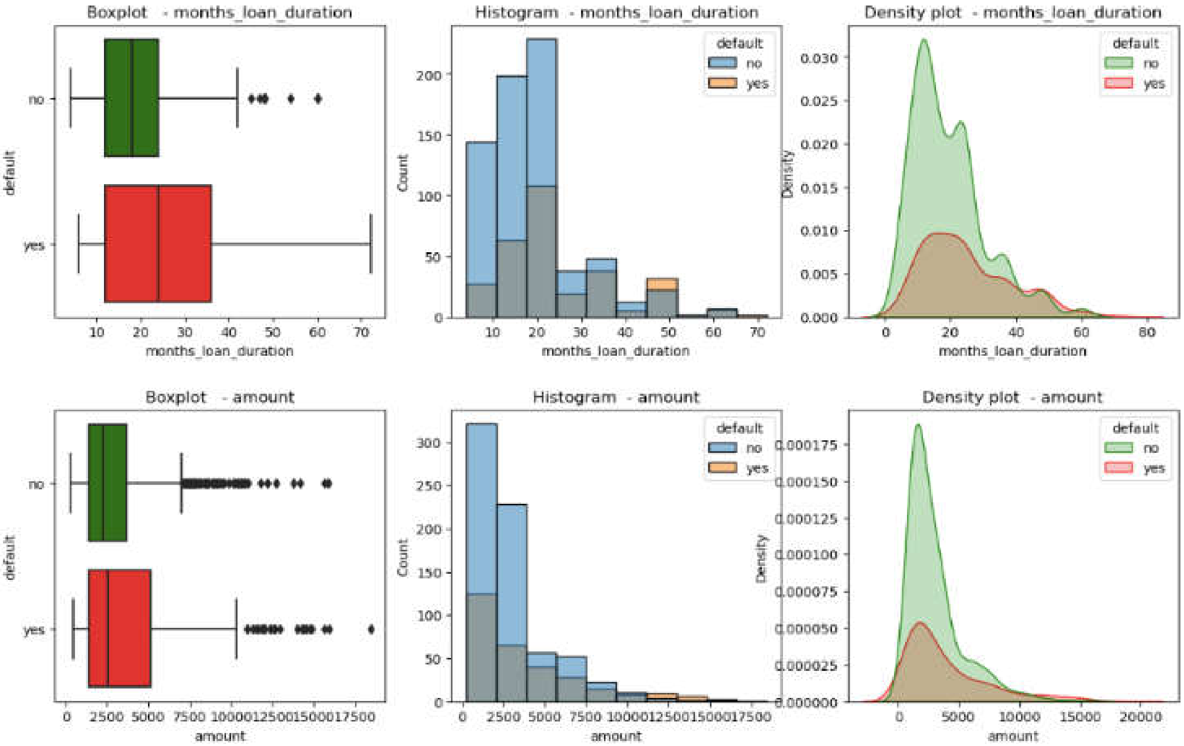
Note: originally the ‘purpose’ column had 6 unique categories of values wherein one category ‘car0’ was a typo error. I merged car0 with car and hence purpose column now has 5 unique categories of values.

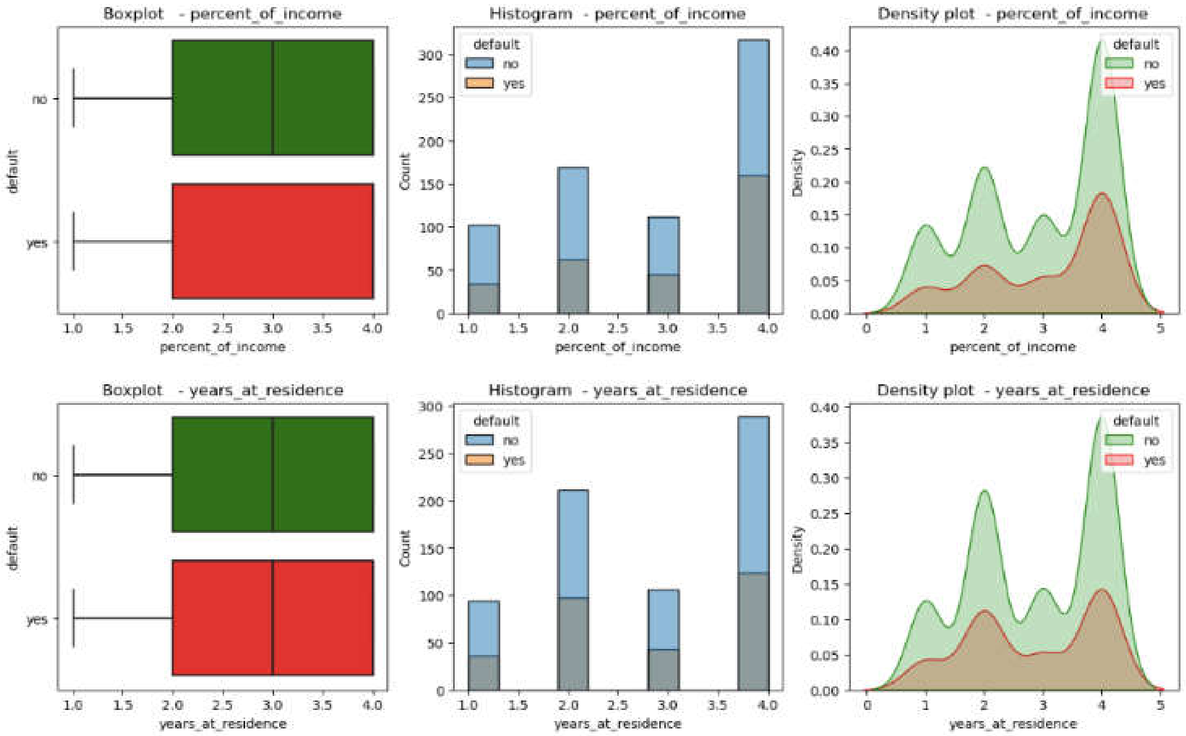
Data Visualization:

Data visualization is an important step in the machine learning process, as it can help us understand the data, explore the patterns and trends, and communicate the results effectively. We used various visualizations, such as histograms, box-plots, and correlation matrices, to examine the distribution and relationship of the variables.

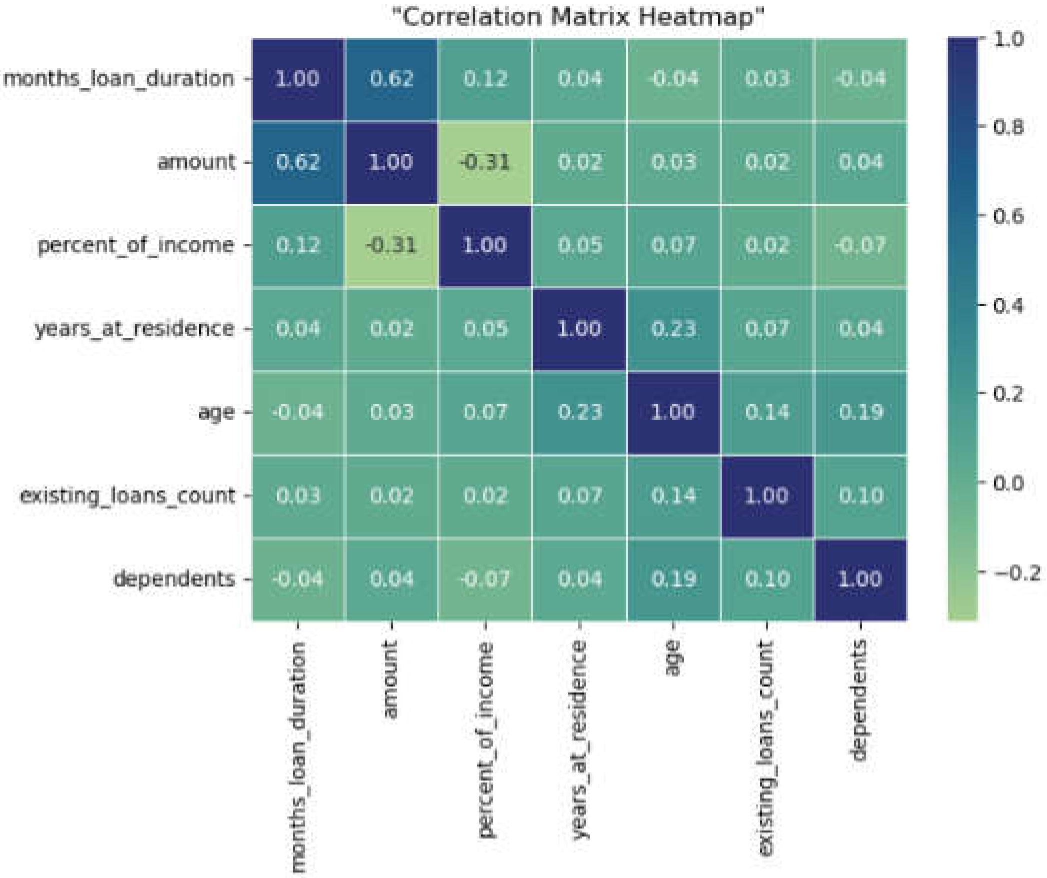


Histograms for numerical columns.





Boxplots, Histograms, and Density plots with hue='default' on selected numerical columns

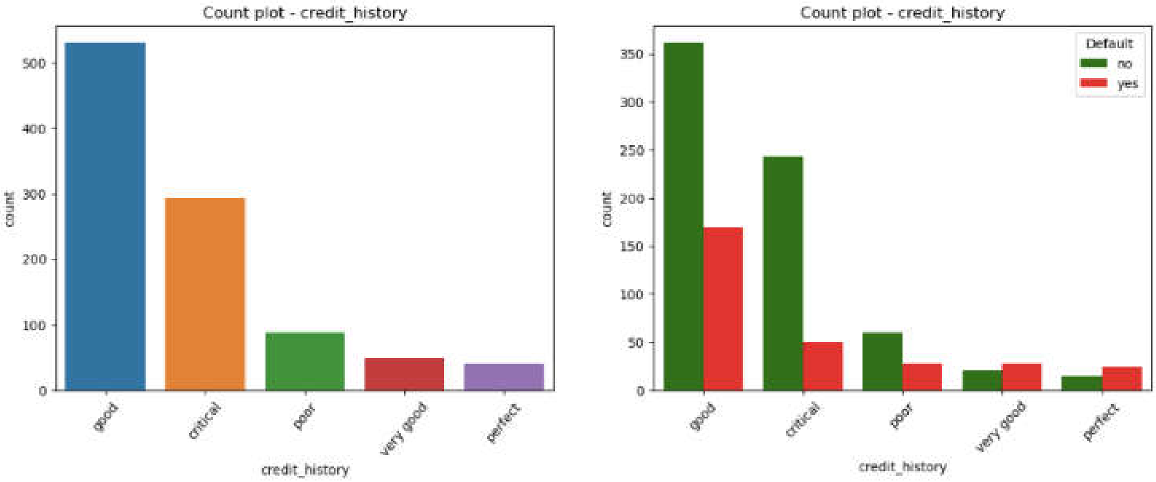


"Correlation Matrix Heatmap"

	months_loan_duration	amount	percent_of_income	years_at_residence	age	existing_loans_count	dependents
months_loan_duration	1.00	0.62	0.12	0.04	-0.04	0.03	-0.04
amount	0.62	1.00	-0.31	0.02	0.03	0.02	0.04
percent_of_income	0.12	-0.31	1.00	0.05	0.07	0.02	-0.07
years_at_residence	0.04	0.02	0.05	1.00	0.23	0.07	0.04
age	-0.04	0.03	0.07	0.23	1.00	0.14	0.19
existing_loans_count	0.03	0.02	0.02	0.07	0.14	1.00	0.10
dependents	-0.04	0.04	-0.07	0.04	0.19	0.10	1.00

Correlation Matrix Heatmap for the numerical columns

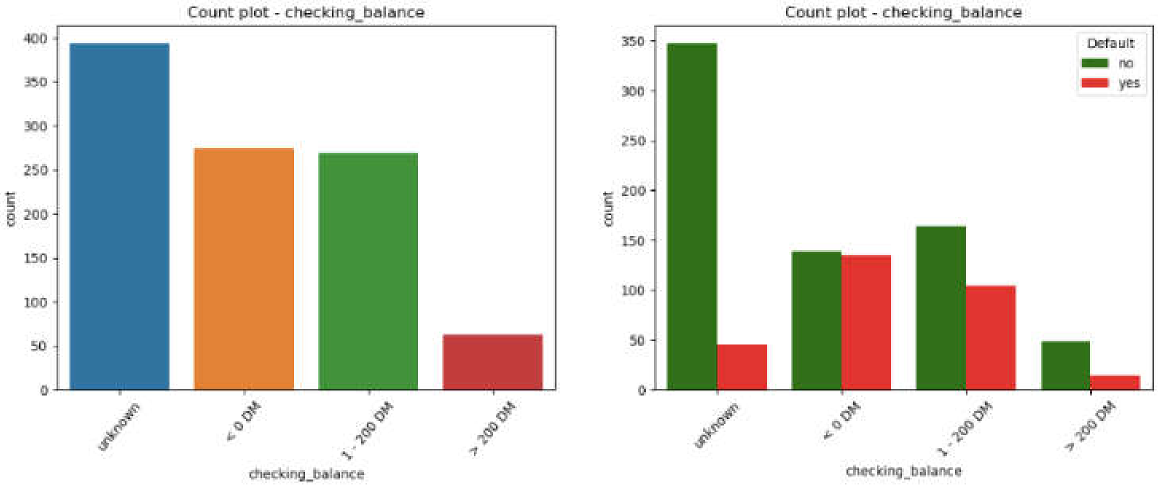
I have used Count plots for analysing the distribution pattern of bank loan customers across different categories. For better comparison of default and non- default customers, I have compared the distribution for each category with and without hue as ‘default’.



Count plot for ‘credit_history’ column with and without hue as ‘default’

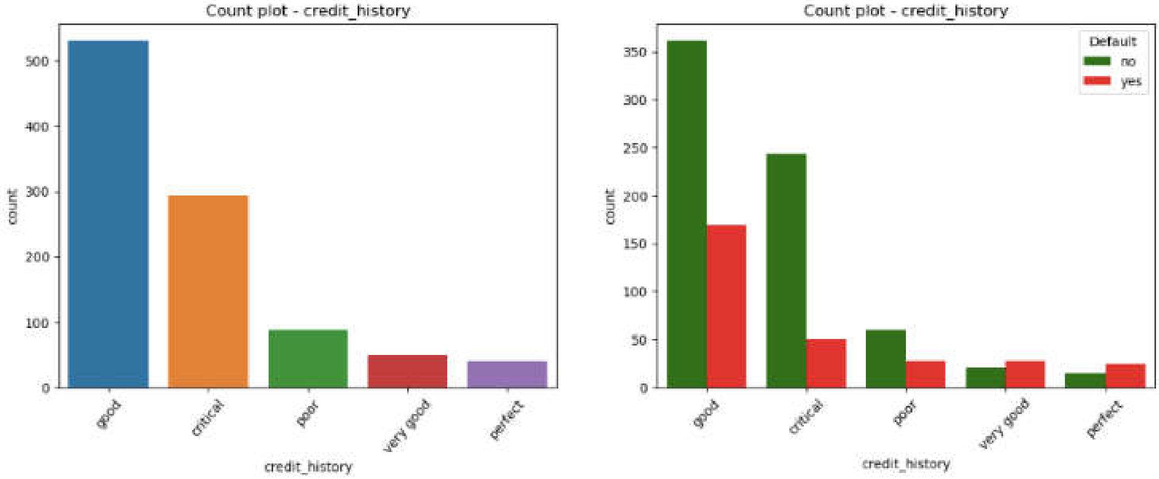
Important Insights from EDA

1. From the description of the loan dataset we know that ‘checking_balance’ refers to the amount of money available in checking account of the customer for everyday financial transactions. From the below count plot of ‘checking_balance’ we can observe that as the availability of this liquid money increases the proportion of defaulters is decreasing.

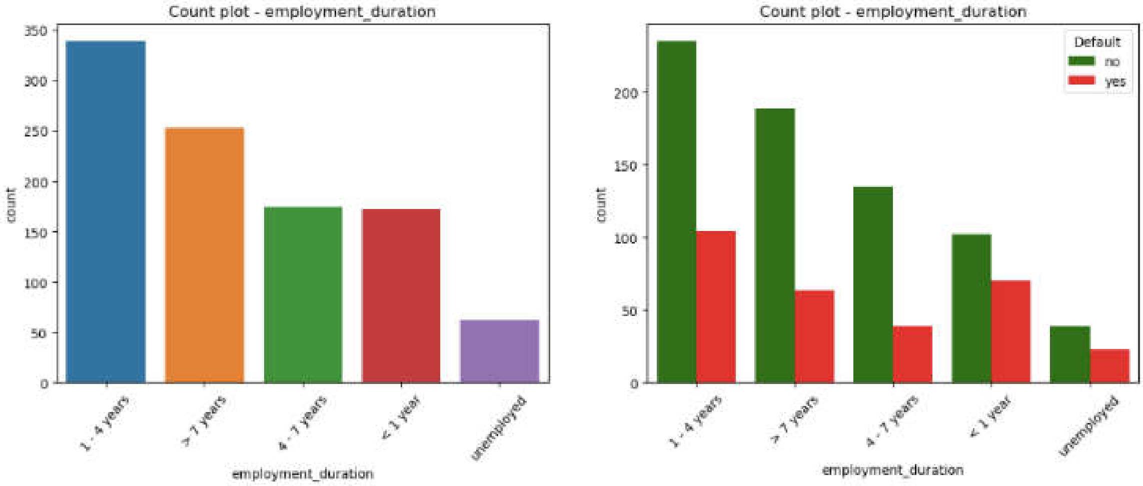


2. From the count plot of ‘credit_history’, I am surprised to observe that proportion of defaulters is higher among the customers who have ‘very good’ and ‘perfect’ credit history as compared to those who have ‘poor’ or ‘critical’ credit history. This finding is surprising since the present credit rating system works on the belief that people with decent credit history very rarely default on their credit/loan obligations.Though the absolute numbers are low, the proportion is high so this needs further study. This

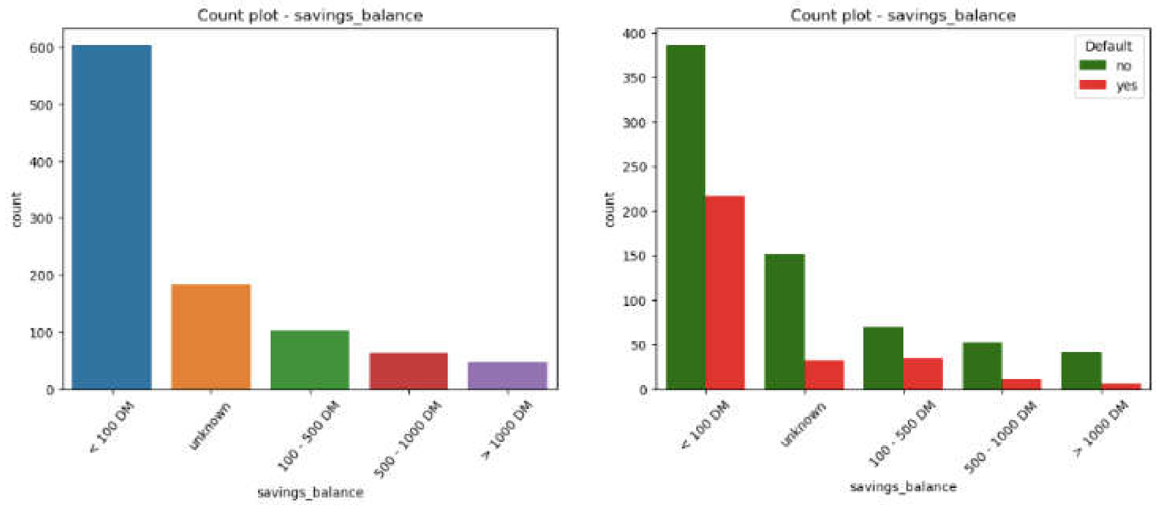
could also be the result of small size of dataset and if we have a bigger sample size, we can validate this finding.



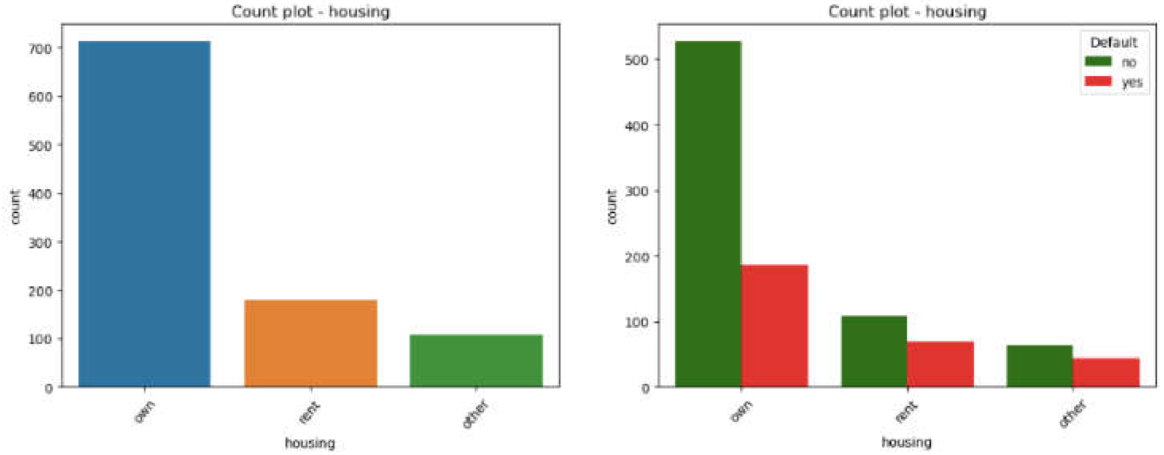
3. From the count plot of ‘employment_duration’ we can observe that customers who are unemployed or freshers (<1 year) have very high default rate and as employment duration increases the default rate decreases. This shows that customers with higher work experience are financially more stable and less prone to default on their loan repayment.



4. Similar to the count plot of ‘checking_balance’, the count plot of ‘savings_balance’ indicates that as the amount of savings balance increases the default rate decreases. This means that customers with higher balances in their savings account are less prone to default on their loans because they can utilize their savings in case of a financial emergency.



5. From the count plot of 'housing', we can observe that customers who are living in rented accommodation have higher proportion of default as compared to those customers who live in their own houses.



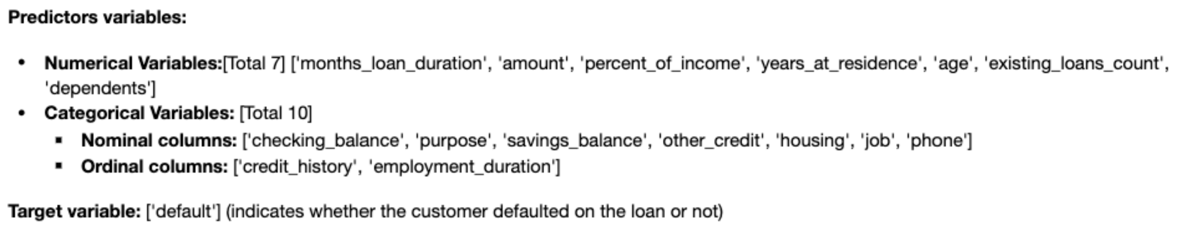
6. In addition, the heatmap of the correlation matrix indicates week relationship among the numerical predictors except few of them. The predictors 'amount' and 'months_loan_duration' have a slightly strong Positive correlation that means loans of higher amount has higher loan duration in months and vice-versa. Similarly the predictors 'amount' and 'percent_of_income' have slightly strong Negative correlation.

In conclusion, through this EDA process I have got a very good understanding about the features of the dataset. I have explored some of the potential relationships between the features and the 'default' behaviour of the customers. These insights are very crucial for the next step i.e. Machine Learning model building for default prediction and for performance optimization of the ML model.

Methods used to build ML Models:

Data Preprocessing:

After data cleaning and data visualization , data preprocessing is done to prepare the dataset for building various Machine Learning models. Data preprocessing involves Data Encoding, Predictor and Response variables selection, Train and Test Data Spliting, and Data Scaling. Data encoding for nominal categorical features is done using one-hot encoding (i.e. using dummy variables) and for ordinal categorical features by using the ordered ranking of values. Data Scaling is done using Standard Scalar method. The Train-Test split of data is done in the ratio of 75-25. Since the dataset is a case of imbalanced classification case, I have used stratification technique while train-test splitting to maintain the class distribution. The predictors and response variables selection is as follows:



Predictors variables:

- **Numerical Variables:**[Total 7] ['months_loan_duration', 'amount', 'percent_of_income', 'years_at_residence', 'age', 'existing_loans_count', 'dependents']
- **Categorical Variables:** [Total 10]
    - **Nominal columns:** ['checking_balance', 'purpose', 'savings_balance', 'other_credit', 'housing', 'job', 'phone']
    - **Ordinal columns:** ['credit_history', 'employment_duration']

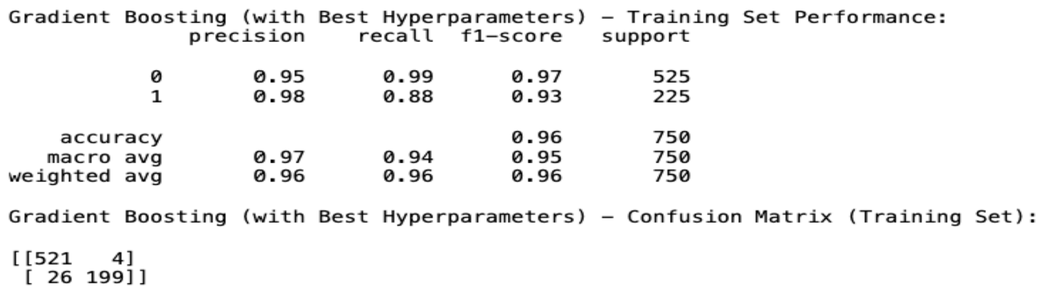
Target variable: ['default'] (indicates whether the customer defaulted on the loan or not)

Hyper-parameter Tuning:

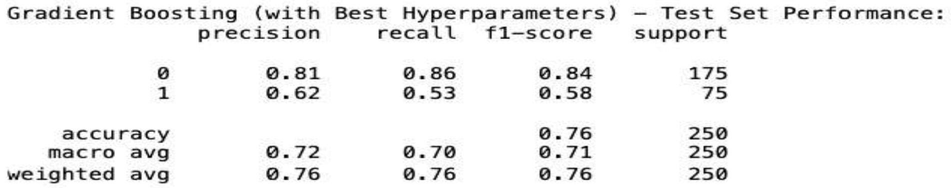
I have build various supervised machine learning models such as 'Logistic Regression', 'Quadratic Discriminant Analysis', 'Support Vector Machines', 'K- Nearest Neighbors', 'Gradient Boosting', 'Random Forest', and 'AdaBoost' for Loan 'default' prediction problem. For hyper-parameter tuning I have used 'GridSearchCV' function and 'Cross-Validation' on the training set to identify the best-performing hyperparameters. Since the primary focus is to minimize false negatives to capture the potential loan defaulters efficiently, I have used 'recall' as the evaluation metric of choice for optimization.

Model Fitting and Evaluation:

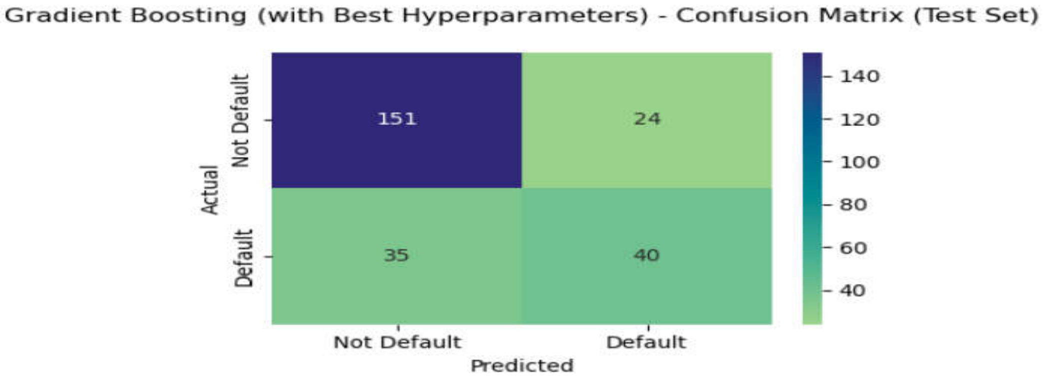
Post model building and hyper-parameter tuning, I fitted all the models on the training set using the optimal hyper-parameters and then 'default' class prediction is done. I have evaluated the performance of each model on both the training and testing datasets. The classification report provided metrics such as precision, recall, and F1-score for each class. I have used the confusion matrix to understand and better visualise the predictions of each model 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]]
```



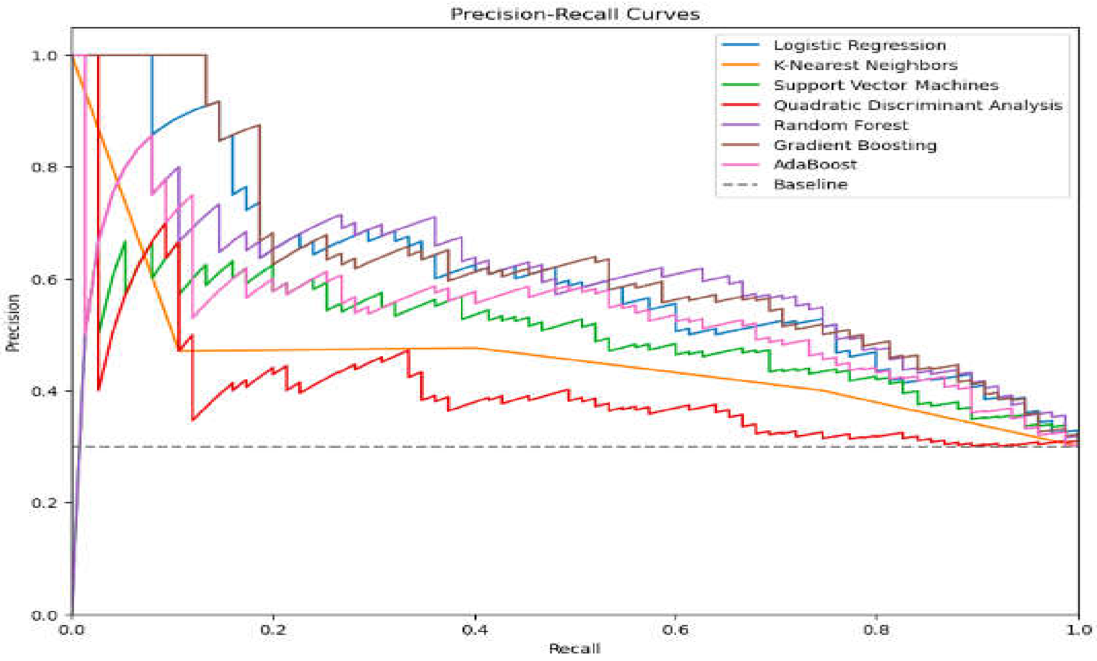
```
Gradient Boosting (with Best Hyperparameters) - Test Set Performance:
              precision    recall  f1-score   support

           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
```

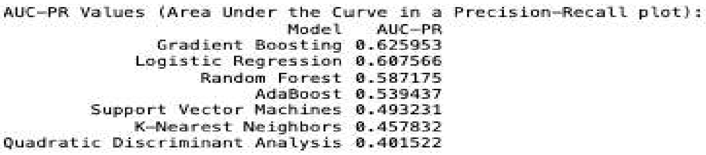


Gradient Boosting (with Best Hyperparameters) - Confusion Matrix (Test Set)

Comparison of Models:

Since the problem in hand is a case of imbalanced-class classification problem, I have used the Area Under the Curve (AUC) for the Precision Recall (PR) or AUC-PR curves instead of ROC curves for comparison of performances of all models and for selection of best performing model. For further optimization of performance of the best selected model, I have applied a custom threshold to the selected model to achieve optimal ‘recall’ performance. The analysis of results highlights the trade-offs between precision and recall.



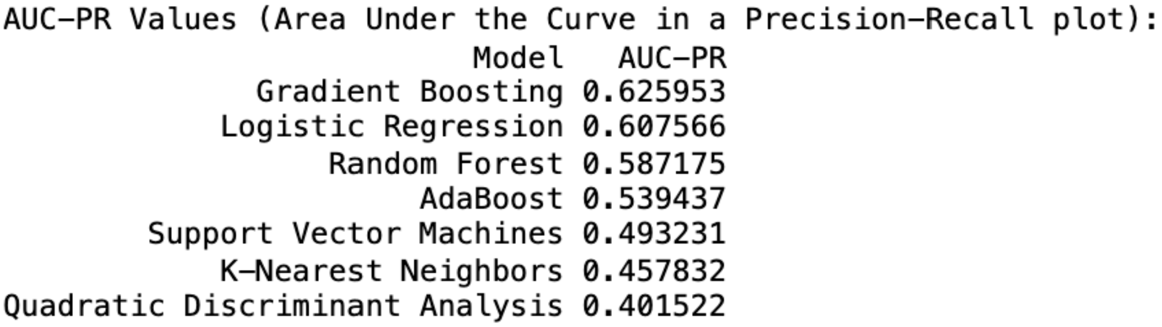


```
AUC-PR Values (Area Under the Curve in a Precision-Recall plot):
                          Model    AUC-PR
              Gradient Boosting  0.625953
            Logistic Regression  0.607566
                  Random Forest  0.587175
                       AdaBoost  0.539437
        Support Vector Machines  0.493231
            K-Nearest Neighbors  0.457832
Quadratic Discriminant Analysis  0.401522
```

Precision-recall curve and AUC-PR values for all models

Results and Discussion:

I applied seven different ML models to predict the credit risk of customers based on the German bank loan dataset. The models were evaluated using the area under the precision-recall curve (AUC-PR) as the performance metric. The results showed that **Gradient Boosting** achieved the highest AUC-PR of **0.626**, followed by Logistic Regression with 0.608**.** The other models had lower AUC-PR values, ranging from 0.58 for Random Forest to 0.40 for Quadratic Discriminant Analysis. The table below summarizes the performance of each model.



```
AUC-PR Values (Area Under the Curve in a Precision-Recall plot):
                           Model    AUC-PR
               Gradient Boosting  0.625953
             Logistic Regression  0.607566
                   Random Forest  0.587175
                        AdaBoost  0.539437
         Support Vector Machines  0.493231
             K-Nearest Neighbors  0.457832
 Quadratic Discriminant Analysis  0.401522
```

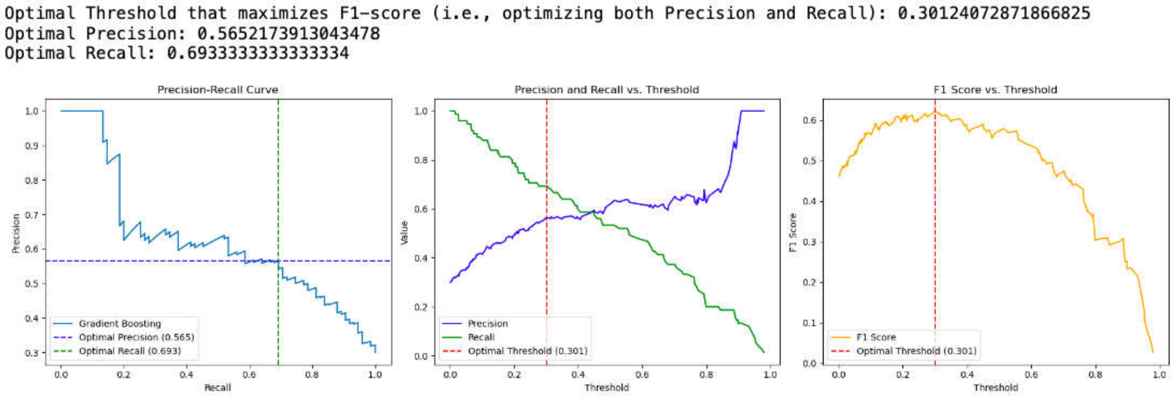
I have used the PR-AUC (Precision-Recall Area Under the Curve) scores to measure the performances of models because PR-AUC is particularly important in imbalanced classification problems like loan default prediction where the positive class (defaults) is a minority. I will now analyse the performance of the models based on AUC-PR scores.

Among the seven ML models, **QDA** had the lowest performance with a PR-AUC score of **0.402**. This model failed to achieve a good trade-off between precision and recall, leading to poor identification of default cases. **KNN** performed slightly better than QDA, but still had a low PR-AUC score of **0.458**. This model could not capture the complex patterns in the data effectively. **SVM** showed a moderate improvement over KNN, with a PR-AUC score of **0.493**. This model was able to balance precision and recall better than KNN.

On the other hand, **AdaBoost** exhibited a higher performance than the previous models, with a PR-AUC score of **0.539**. This model leveraged the ensemble learning technique to enhance the accuracy and robustness of the predictions. **Random Forest** further improved the performance of AdaBoost, with a PR-AUC score of **0.587**. This model utilized the bagging method to reduce the variance and over fitting of the individual trees. **Logistic Regression** achieved the second-highest performance among the models, with a PR-AUC score of **0.608**. This model was able to fit a linear decision boundary to the data and produce reliable probabilities of default. Finally, **Gradient Boosting** attained the highest performance of all the models, with a PR-AUC score of **0.625**. This model applied the boosting method to sequentially correct the errors of the weak learners and generate a strong classifier.

The results suggest that Gradient Boosting is the most effective model for this classification task, while Quadratic Discriminant Analysis and K-Nearest Neighbours are the least effective. The performance difference between the models could be attributed to various factors, such as the complexity, robustness, and interpretability of the models, as well as the characteristics of the dataset, such as the size, imbalance, and noise.

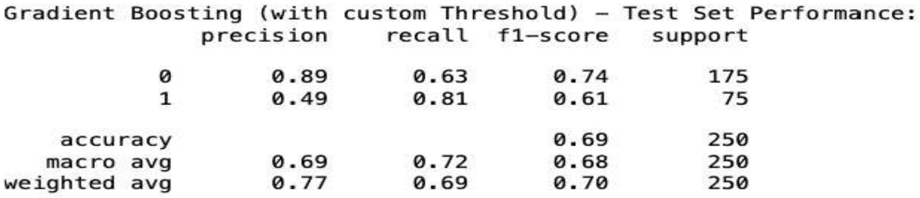
Gradient Boosting shows the best performance with a PR-AUC score of 0.626. This model achieved the highest “recall” while maintaining a reasonable level of precision, making it a strong candidate for identifying ‘default’ cases. The importance of recall is crucial in this context because the focus is on minimizing false negatives to mitigate the bank's risk.



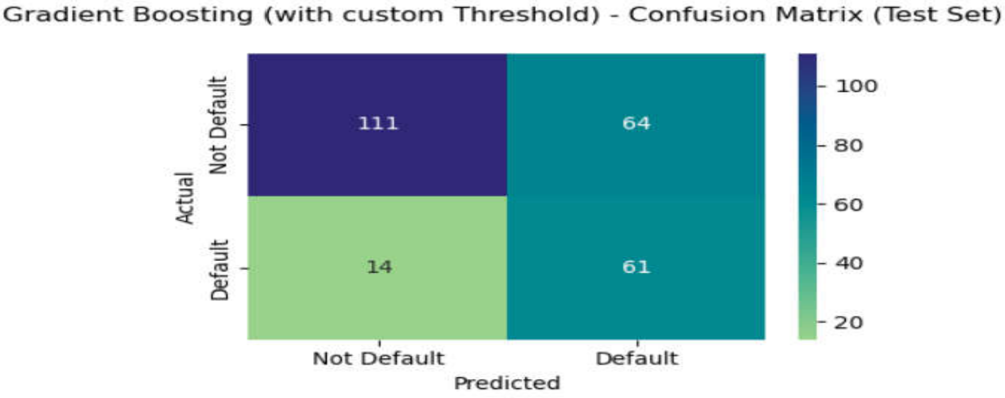
```
Optimal Threshold that maximizes F1-score (i.e., optimizing both Precision and Recall): 0.30124072871866825
Optimal Precision: 0.5652173913043478
Optimal Recall: 0.6933333333333334
```

Threshold value that maximizes F1-score performance

Gradient Boosting with Custom Threshold: Now to further improve the performance of the Gradient Boosting Model in terms of recall, I am applying a custom threshold of ‘0.18’, which is about 40% further reduction from the optimal threshold (0.30). Gradient Boosting model with its best hyper-parameters and custom threshold achieves best results in terms of PR-AUC score and hence the most suitable model. While the precision for defaults may not be as high, the high recall ensures that the model identifies a substantial number of actual defaulters.



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

           0       0.89      0.63      0.74       175
           1       0.49      0.81      0.61        75

    accuracy                           0.69       250
   macro avg       0.69      0.72      0.68       250
weighted avg       0.77      0.69      0.70       250
```



Gradient Boosting model with custom threshold achieves best results but still there are avenues for improvement such as using hyper-parameter tuning more extensively, trying more methods and using feature engineering, etc.

Limitations and Future Directions:

One of the limitations of this project is the lack of feature engineering. I have used the original features of the dataset without creating any new features or transforming the existing ones. Feature engineering is an important step in ML modeling, as it can enhance the predictive power and interpretability of the models by extracting more information from the data. In future works, I could explore some techniques for feature engineering, such as feature selection, feature extraction, feature scaling, feature encoding, and feature interaction. These techniques could help me to reduce the dimensionality, noise, and redundancy of the data, as well as to capture the non-linear and categorical relationships among the features.

Another limitation is the use of the same hyper-parameter tuning method for all the models. I have used the GridSearchCV method, which performs an exhaustive search over a predefined grid of hyper-parameters. This method is simple and easy to implement, but it can be computationally expensive and inefficient, especially when the grid is large and the model is complex. In future works, I could experiment with some more advanced and efficient methods for hyper-parameter tuning, such as RandomSearchCV, Bayesian Optimization, Genetic Algorithms, and Gradient-based Optimization. These methods could help me to find the optimal hyper-parameters in a shorter time and with less computational resources.

A third limitation is the small sample size of the dataset. The German bank loan dataset contains only 1000 samples, which is relatively small for ML modeling. This small sample size could affect the generalization capabilities of the models, as they might not be able to learn the true patterns and distributions of the population. Moreover, the small sample size could also increase the risk of overfitting and underfitting, as well as the variance and bias of the models. In future works, I could incorporate a larger and more representative dataset, with more samples and more diverse features, to improve the robustness and performance of the models.

A fourth limitation is the class imbalance of the dataset. The loan dataset has a skewed distribution of the target variable, as there are more non-default cases than default cases in the 'default' column. This imbalance could lead to biased model results, as the model might favour the majority class over the minority class. This could result in lower precision and recall for the default cases, which are more important and costly for the bank. In future works, I could experiment with some methods to deal with the class imbalance problem, such as oversampling and under-sampling. These methods could help me to achieve a more balanced representation of the classes, and to improve the sensitivity and specificity of the models.

Conclusion:

In this project, I aimed to develop a predictive model for loan default prediction using historical data from a German bank. The project followed a systematic approach that consisted of data exploration, visualization, pre-processing, and modelling. The goal was to build a reliable and accurate model that could assist the bank in identifying potential loan defaulters and mitigating financial risks.

The project experimented with different machine learning models, such as Logistic Regression, Gradient Boosting, Random Forest, and Support Vector Machine. The models were evaluated using various performance metrics, such as accuracy, precision, recall, and F1-score. The project also used the area under the precision-recall curve (AUC-PR) as the main criterion to select the best model, as it reflects the trade-off between precision and recall for imbalanced data. The project also performed feature importance analysis to identify the most influential features for loan default prediction.

The project found that Gradient Boosting was the best model for loan default prediction, as it achieved the highest AUC-PR and F1-score among all the models. Gradient Boosting is a powerful ensemble technique to sequentially correct the errors of the weak learners and generate a strong classifier. The project also found that credit history was the most important feature for loan default prediction, as it reflects the past behaviour and reliability of the customers.

However, the project also acknowledged some of the limitations and challenges of the study, such as the relatively small and imbalanced dataset and the need for more advanced feature engineering. The project suggested some possible future directions for further improvement and exploration, such as using larger and more diverse datasets and applying more sophisticated feature engineering techniques. The project can provide valuable insights and recommendations for the bank to enhance its loan approval process and minimize potential losses arising from loan defaults.

---- End of Report ----