INFO – 521
INTRODUCTION TO MACHINE LEARNING

GERMAN BANK LOAN DEFAULT PREDICTION

FINAL PROJECT REPORT

Submitted by
Sundar Ram Subramanian (SID: 23946913)

Table of Content

Table of Figures

Table of Tables

INTRODUCTION

Context of the Problem

In the financial services industry, particularly within banking, managing credit risk effectively is crucial to maintaining profitability and stability. One of the persistent challenges banks face is the risk of loan defaults, which can lead to significant financial losses. Predictive analytics has emerged as a vital tool in this domain, enabling financial institutions to assess and mitigate risks proactively. By leveraging historical data to predict future outcomes, banks can identify potential loan defaulters before a default occurs, allowing for better credit risk management and more informed lending decisions.

Purpose and Objectives

The primary objective of this project is to develop a machine learning model that predicts the loan defaults (classification) using historical customer data from a German bank. This model aims to assist the bank in enhancing its credit risk assessment processes by identifying key predictors of default. Through this initiative, the bank can not only reduce the incidence of loan defaults but also extend credit more confidently and responsibly. The project utilizes various machine learning techniques to explore how different models perform and which features most significantly impact the prediction of defaults.

Research Questions

This project is structured around a series of analytical and modeling steps to answer key questions that will inform the development and refinement of the predictive model:

1. What relationships exist between various features?
2. Which features are most strongly correlated with loan default?
3. Which model shows the most promise and can be recommended for operational use to predict loan defaults accurately for the following different business scenarios?
    a. Catch as many defaulters as possible
    b. Reduce false alarms there by retaining the customers
    c. A well-balanced approach addressing both catching defaulters and reducing false alarms

METHODS AND MATERIALS

Data Description

The dataset used for this project is the German_bank.csv which consists of historical data from a German bank regarding customer loans. This dataset includes 1,000 rows, each representing an individual customer, and 17 columns that detail various aspects of the customer's financial, personal & professional background, which are relevant to the assessment of their likelihood of defaulting on a loan.

The data encompasses a range of features, from basic demographic information to detailed financial records, as outlined in the dataset dictionary below: (*Table 1*)

Feature	Description
checking_balance	Amount of money available in the customer's account
months_loan_duration	Duration since the loan was taken
credit_history	Credit history of each customer
purpose	Purpose for which the loan was taken
amount	Amount of loan taken
savings_balance	Balance in the savings account
employment_duration	Duration of the customer's current employment
percent_of_income	Percentage of monthly income that goes towards loan repayment
years_at_residence	Duration of current residence
age	Age of the customer
other_credit	Presence of other credits apart from the main loan
housing	Type of housing (e.g., rent or own)
existing_loans_count	Number of existing loans the customer has
job	Type of job held by the customer
dependents	Number of dependents relying on the customer
phone	Whether the customer has a phone registered under their name
default	Default status (target variable)

Table 1 Dataset Dictionary

This data is primarily structured for the task of predictive modeling, with the *default* column serving as the target variable indicating whether a customer has defaulted on a loan. The rest of the features provide a basis for building predictive models to assess risk profiles effectively.

Exploratory Data Analysis (EDA)

OVERVIEW AND STATISTICAL SUMMARIES:

Initial investigations using *df.info()* confirmed that there are *no missing values* across all columns, indicating a clean dataset that is ready for further analysis without the need for imputation strategies. The dataset features a mix of numerical and object type variables. There are 7 numerical & 10 object types in the dataset. **Numerical type** variables include months_loan_duration, amount, percent_of_income, years_at_residence, age, existing_loans_count and dependents while **object type** includes checking_balance, credit_history, purpose, savings_balance, employment_duration, other_credit, housing, job, phone, and default.

Despite datatypes being a mix of numerical & object type, there are certain variables with numerical type but categorical in nature. Further investigating the data using the *value_counts()* for each of the feature, we can deduce that checking_balance, credit_history, purpose, savings_balance, employment_duration, percent_of_income, years_at_residence, existing_loans_count, other_credit, housing, job, phone, dependents & default are **categorical** in nature and months_loan_duration, loan_amount, and age are **continuous** in nature. These classifications help identify which variables represent distinct categories and which ones are measurable quantities.

Statistical summaries were generated using *df.describe(include='all').T*, providing a thorough understanding of both numerical and categorical data distributions. For numerical features, key statistics such as the mean, standard deviation, and range were examined: **Loan Duration (*months_loan_duration*):** Ranges from 4 to 72 months, with an average duration of approximately 20.9 months. The 75th percentile is at 24 months, suggesting a prevalence of shorter-term loans within the dataset. **Loan Amount (*amount*):** Loan amounts vary significantly from 250 to 18,424, with a mean of approximately 3,271. The standard deviation of 2,822 indicates high variability in loan amounts, reflecting diverse borrowing needs among customers. **Age (*age*):** Ranges from 19 to 75, with average age of 35. The 75th percentile is 42 years indicating that most of the customers are in the middle age group. Categorical features were analyzed for frequency distribution, revealing prevalent categories and potential anomalies: **Checking Balance (*checking_balance*):** The most common

category is 'unknown', occurring in 394 instances, indicating a significant number of customers prefer not to disclose their checking account balances. **Credit History (*credit_history*):** Most customers (530 occurrences) have a 'good' credit history, suggesting a customer base with generally reliable financial behavior. **Purpose of Loan (*purpose*):** The predominant use of loans is for 'furniture/appliances', noted in 473 cases, underscoring common consumer borrowing purposes.

NULL VALUE CHECK:

As mentioned in the overview, Initial investigations using *df.info()* confirmed that there are *no missing values* in the data indicating a clean dataset that is ready for further analysis without the need for imputation strategies.

ANALYSIS OF CLASS IMBALANCE:

The balance between the classes in the target variable *default* was examined to detect any signs of class imbalance. The data was found to be moderately imbalanced having a ratio of 70:30 (*Figure 1*). 70% of the customers have not defaulted while only 30% of the customers in the dataset have defaulted. This imbalance may affect model performance, especially in case of Logistic regression while tree-based models mostly take care of moderate imbalances.

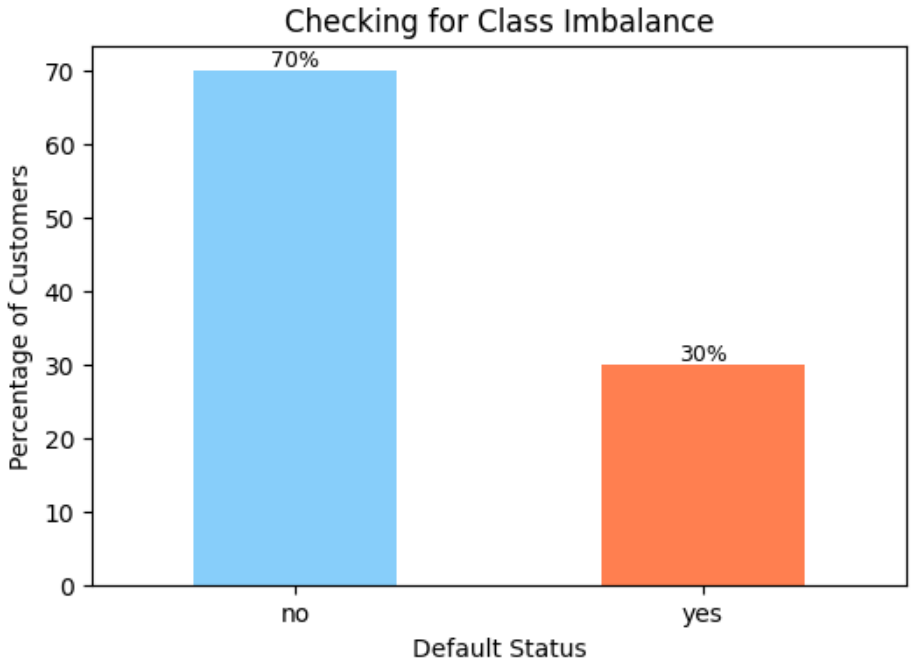

Figure 1 Bar chart representing Class Imbalance

CORRELATION ANALYSIS:

Correlation Analysis was performed to understand the relationships and potential multicollinearity between features. Heatmaps and pair plots from seaborn package were utilized to visually represent these correlations, providing a clear view of how features relate to each other and to the target variable.

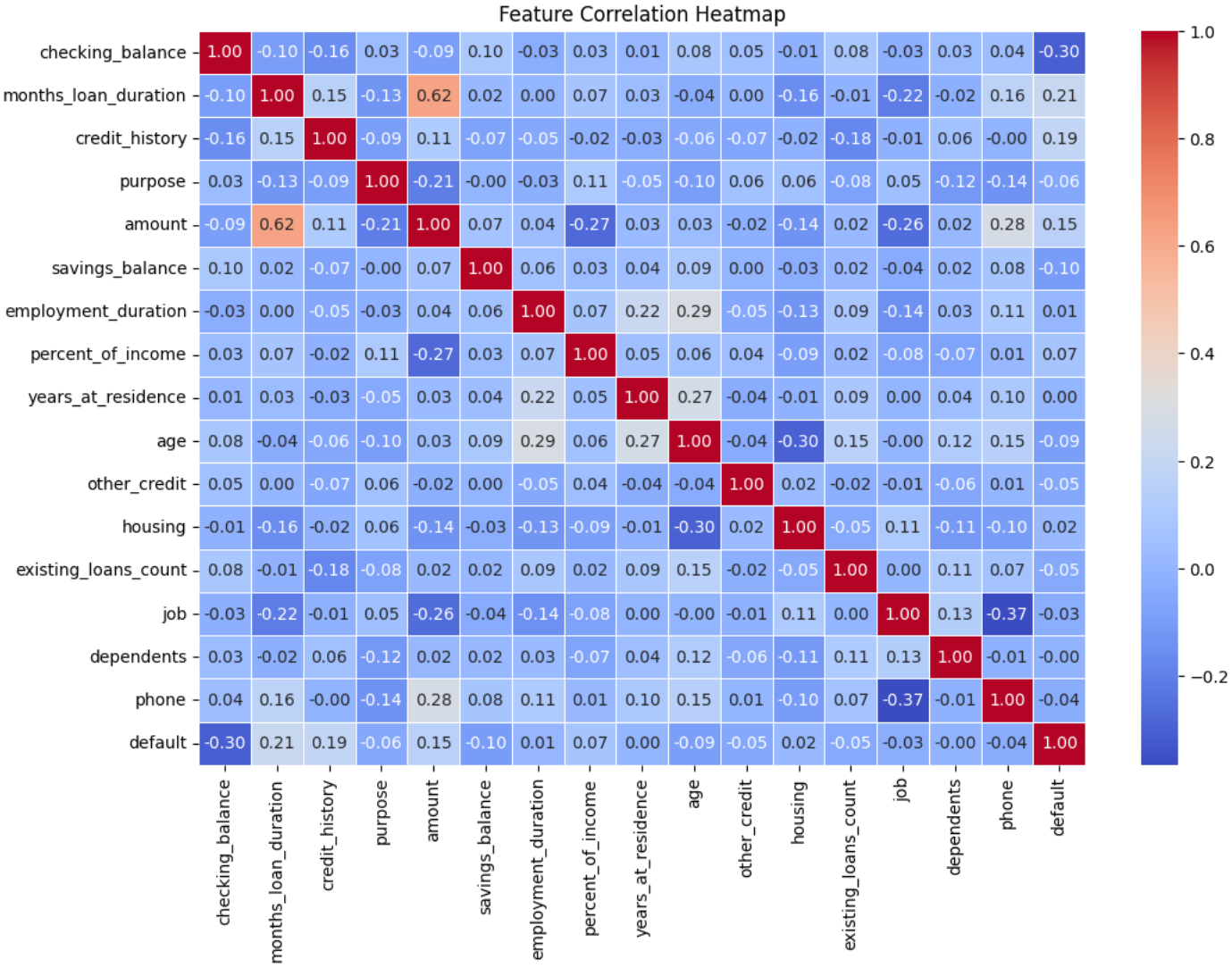

Figure 2 Heat Map - Correlation Analysis

The insights from the heatmap indicate a few notable correlations and interactions between *default* & other variables. Features *checking_balance*, *months_loan_duration*, *credit_history*, and *loan_amount* display a positive correlation with *default*, with *checking_balance* showing the strongest correlation among them. This suggests that higher loan amounts or extended loan durations might be associated with increased default risks. Conversely, *savings_balance* and *age* exhibit a negative correlation with *default*, although these correlations are not particularly strong. Other features such as *purpose*, *other_credit*, *existing_loans_count*, *job*, and *phone* also demonstrate a weak negative correlation with *default*. Meanwhile, *employment_duration*,

percent_of_income, and *housing* show a slight positive correlation. Features like *years_at_residence* and *dependents* appear to have almost no correlation with *default*.

In terms of correlation amongst predictors, there is a very strong positive correlation between *loan_amount* and *months_loan_duration*, which logically aligns with the notion that larger loans typically come with longer repayment periods. Additionally, *age* and *years_at_residence* are positively correlated, suggesting that older individuals tend to have lived at their current residences for longer periods.

Regarding multicollinearity, the analysis reveals no excessively strong correlations among the variables, indicating that multicollinearity does not pose a significant concern in this dataset. This lack of strong multicollinearity is beneficial for modeling, as it suggests that the predictive power and statistical significance of individual predictors can be reliably interpreted without substantial interference from overlapping data influences.

Figure 3 Pair Plot - Correlation Analysis

The insights derived from the pair plot in *Figure 3* are consistent with previously observed correlations using the heat map. It has been observed that individuals with higher loan amounts tend to default more frequently. Additionally, there seems to be a slight trend indicating that people with a higher count of existing loans are more prone to default. Furthermore, the data suggest that younger individuals are more likely to default, reinforcing the link between age and financial reliability.

HISTOGRAM ANALYSIS:

Histograms were generated for all numerical features to visualize their distributions. This analysis helps in identifying the skewness of the distributions, detecting outliers, and understanding the underlying trends that might influence customer default rates.

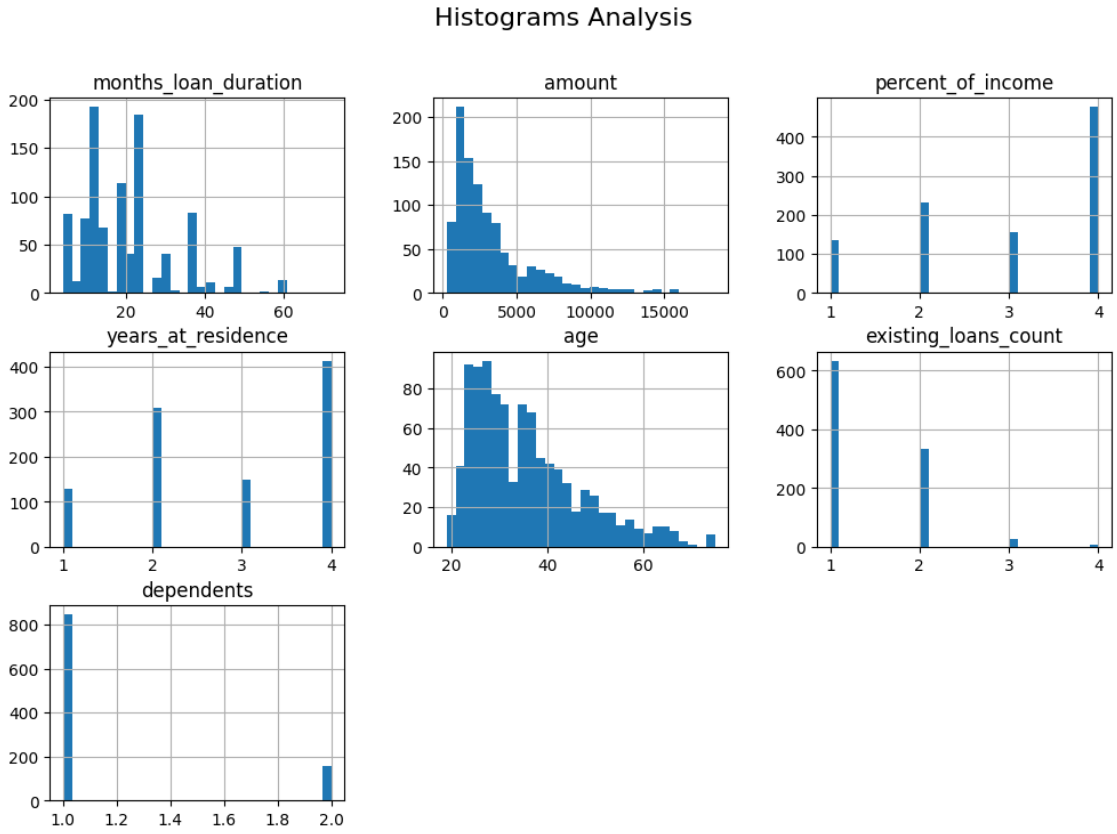

Figure 4 Histogram Analysis

The histograms reveal that the distributions of *amount*, *age*, and *percent_of_income* are right-skewed. This skewness suggests the presence of outliers, which could potentially impact the overall analysis. Hence, the next step is to detect and analyze the outliers. It can also be observed that certain variables are categorical in nature, but with numeric datatype.

OUTLIER ANALYSIS:

A box plot, also known as a whisker diagram, is a standardized way of displaying the distribution of data based on a five-number summary: minimum, first quartile (Q1), median, third quartile (Q3), and maximum. It helps identify outliers, symmetry of data, and the tightness of the data spread. In this project, box plot visualization of the Seaborn library has been leveraged for outlier analysis. Refer Figure 5. It is preferable to consider only the continuous features as other categorical features have only up to 4 classes in each of them.

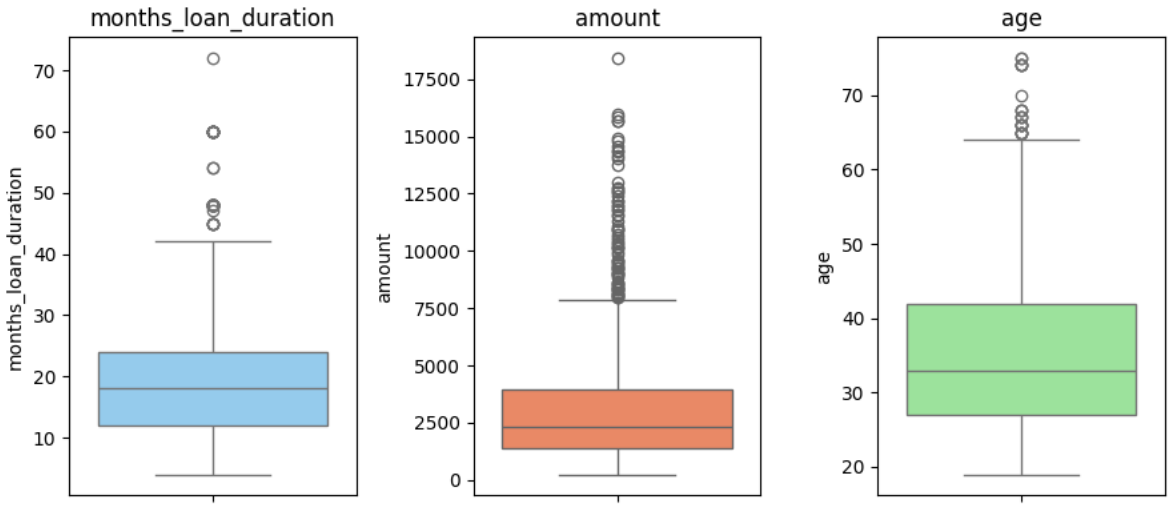

Figure 5 Box Plot - Outlier Analysis

The box plot of loan durations shows that the median loan duration falls between 18 to 20 months. The interquartile range (IQR), which represents the middle 50% of the data, extends from 12 to 24 months. Notably, there are outliers in the data, with some loans extending beyond 40 months, reaching up to 50 months or more. This indicates that a small segment of customers opts for significantly longer loan durations.

The box plot of loan amounts indicates a high degree of skewness, with a concentration of data points below 5000. However, there are numerous outliers, with some loan amounts reaching as high as 15,000 to 18,000. These high-value loans likely represent high-risk or large-scale financial undertakings by certain borrowers.

Regarding the age of borrowers, the median age is clustered around 32 to 35 years. The interquartile range spans from 25 to 45 years, showing a broad distribution of borrower ages. Notably, there are outliers among older individuals, particularly those aged 65 and above, suggesting that loans are being accessed by some customers well into retirement.

Though there are outliers present across these variables, it is important to retain these data points in the analysis. This approach ensures the preservation of the dataset's originality and integrity, recognizing that these outliers represent actual values rather than errors or missing data.

CATEGORICAL FEATURES ANALYSIS:

We have already visualized the relationship between variables using heat map and pair plot. The distribution of numerical variables was also observed in Histogram analysis. Now, the count plot of seaborn library has been leveraged to understand the distribution in categorical variables. Refer Figure 6.

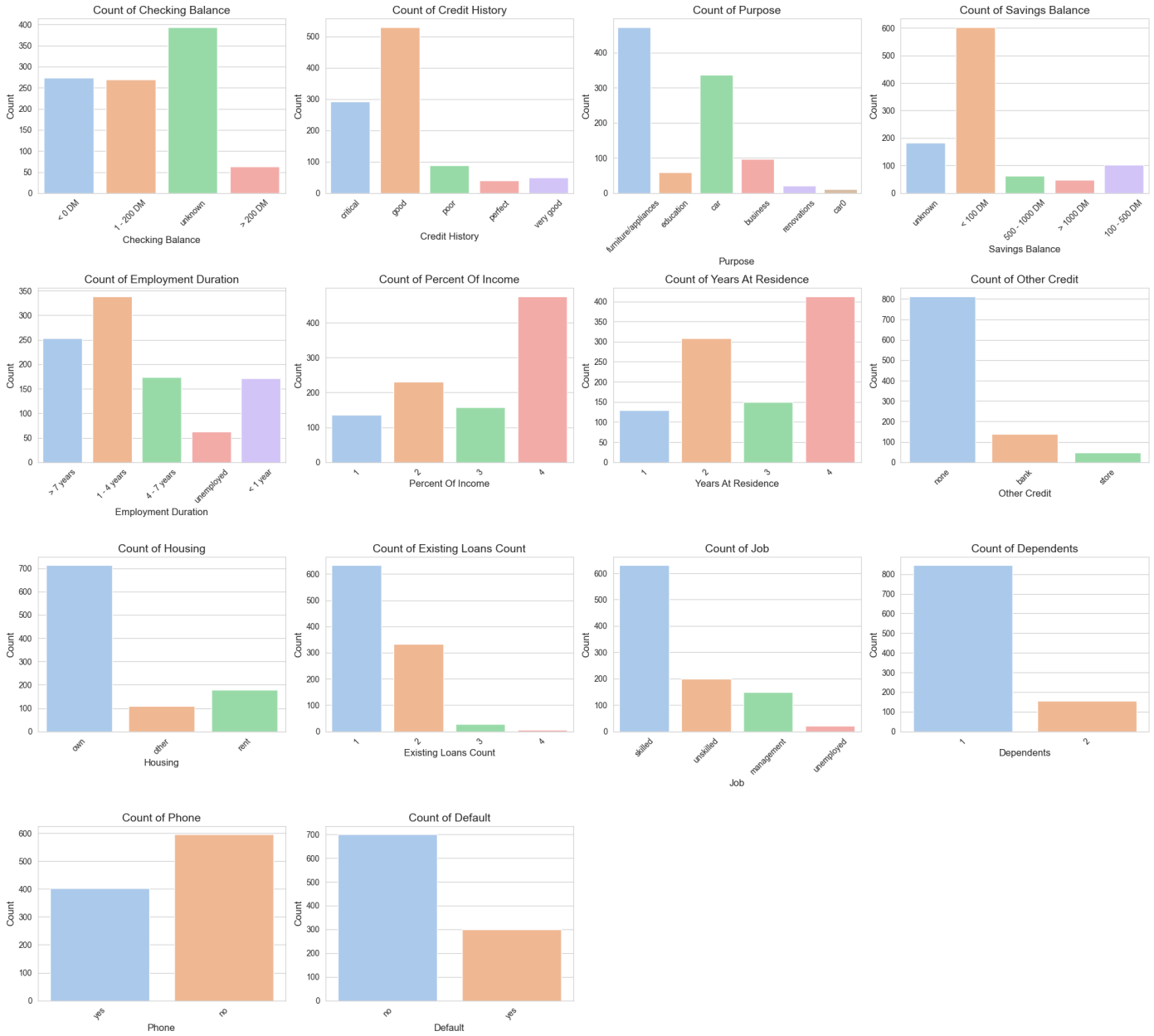

Figure 6 Count Plot - Categorical Features

A noticeable trend can be observed in the checking_balance is that a large portion of the population either does not disclose or does not have available data regarding their checking balances, with many accounts recorded as "unknown." Those that do have data present tend to fall under "< 0 DM" or "1 - 200 DM," with higher balances over "200 DM" being less frequently reported, indicating a clustering of accounts in the lower balance ranges.

In terms of credit history, the data reveals that most individuals maintain a "good" credit history, supporting the view of a reliably performing borrower base. However, only a minority achieve a "perfect" credit rating, underscoring the rarity of an impeccable financial record.

When examining the purpose of loans, vehicles and furniture/appliances emerge as the primary reasons for borrowing, reflecting common consumer financial needs. Conversely, loans for education and renovations appear less frequently, suggesting these are less prioritized or perhaps funded through other means.

Analyzing savings balances, most individuals report having less than "100 DM," indicating a general trend of low savings. Conversely, very few report savings exceeding "1000 DM," pointing to limited saving capacity across the dataset.

Employment duration insights indicate a substantial number of individuals have relatively short employment periods ranging from "1 to 4 years," hinting at either a young workforce or a high job turnover rate. At the extremes, the numbers dwindle for those "unemployed" or those with long-term employment over "7 years," highlighting a concentration in mid-range employment tenure.

Focusing on the percent of income dedicated to loan payments, the most common commitment level is "4%," suggesting a trend towards higher financial obligations relative to income. In contrast, fewer individuals devote as little as "1%" of their income to such payments, possibly reflecting cautious financial behavior or limited engagement with credit facilities.

Residential stability is another area of interest, with many having lived at their current residence for "4+" years, suggesting a degree of residential stability. Those residing at a location for only "1 year" are fewer, which may reflect recent relocations or a less stable living situation.

Regarding other credit obligations, the majority have no other credit commitments, indicative of either a strategic choice to avoid multiple concurrent debts or a focus on managing more substantial singular loans. The presence of "bank" or "store" credits is minimal, perhaps due to their specific or occasional nature.

The housing status largely shows that most people own their homes, which may correlate with financial stability and security. Rental and other housing arrangements are less common, potentially indicating a preference for or accessibility to homeownership over other options. In examining existing loan counts, most individuals manage just "1" loan, while those handling "3" or more are notably rare, highlighting a general reluctance or inability to juggle multiple loans.

The workforce composition reflects a predominance of "skilled" workers, aligning with a potentially well-educated or trained demographic. Those categorized as "unemployed" form a minimal part of the dataset, perhaps pointing to selective lending practices focusing on those currently employed.

Dependency ratios show most individuals support "1" dependent, influencing their financial planning and needs, while those with "2" dependents are less common, suggesting smaller average family units or differing financial responsibilities. The distribution of phone ownership is roughly equal, posing implications for how financial services and communications are conducted.

Lastly, the default rates indicate that the majority have not defaulted on loans, affirming a general pattern of reliability in repayments. However, there is a presence of minority who have defaulted.

Data Pre-Processing

Data preprocessing is a crucial step in the machine learning pipeline. It involves transforming raw data into a clean and usable format, which can significantly improve the performance of machine learning models. In this project we focus on *Encoding* & *Scaling.*

ENCODING:

Machine learning algorithms typically require numerical input, so we need to convert the categorical values into numbers. This process of conversion is called encoding. There are two types of encoding Label Encoding & One hot encoding.

Label Encoding are generally preferred in Tree based models as these models can handle numbers directly, while one hot encoding is suitable for Linear models as they prevent false ordinal relationships by creating separate binary columns for each class. However, the latter increases the dimensionality of dataset, which again affects the model performance. Hence, we prefer to use the Label encoding in all our categorical columns.

SCALING:

Scaling is the process of transforming features to a similar range, which is essential for algorithms that rely on distance calculations. In our dataset, we can consider the example of age and loan_amount columns which are essentially two different ranges. Hence, Scaling is essential for our dataset. There are two common methods for scaling: Standardization & Min-Max Scaling.

Standardization transforms the data such that data is centered around 0 with unit variance (mean = 0, std. deviation = 1). This is best suited for normally distributed (Gaussian) data. Min-Max scaling scales data between 0 and 1 or any defined range. This is best for non-Gaussian distributions and when preserving zero-based relationships.

It is also well known that tree-based models are scale-invariant, as Decision Trees split features based on thresholds, which is anyways not changed by scaling. Therefore, scaling does not have any impact in tree-based model outcome but might improve training efficiency for trees.

However, in this project we prefer to go ahead with scaling as we must examine models irrespective of parametric or tree based. We prefer to scale the data with Standardization method as it preserves outlier effects but reduces their impact on model training. It also ensures features are comparable in magnitude while keeping relationships intact. Min-Max scaling, on the contrary, does not make the mean 0 and is sensitive to outliers. If outliers exist, Min-Max compresses most data points into a very small range. Min-Max is preferred when working with images or pixel data.

Understanding Predictors-Target Relationship

The encoded and scaled data from data pre-processing is further divided in to target variable (*default*) and predictors (all other features except default). We calculate two correlation coefficients on this dataset: Pearson Correlation that measures linear relationships & Spearman Correlation that measures monotonic relationships (which can be non-linear). Table 2 shows observed correlations and its interpretations.

Feature	Pearson Correlation	Spearman Correlation	Interpretation
Checking Balance	-0.3024	-0.2878	Mostly Linear
Months Loan Duration	0.2149	0.2057	Mostly Linear
Credit History	0.1937	0.2115	Slightly Non-Linear
Purpose	-0.0557	-0.059	No Strong Relationship
Loan Amount	0.1547	0.0871	Some Non-Linearity
Savings Balance	-0.1031	-0.0988	Mostly Linear
Employment Duration	0.0089	0.0031	Mostly Linear
Percent of Income	0.0724	0.0737	Mostly Linear
Years at Residence	0.003	0.0026	No Effect
Age	-0.0911	-0.1122	Some Non-Linearity
Other Credit	-0.0539	-0.0579	No Strong Relationship
Housing	0.0193	0.0239	No Effect
Existing Loans Count	-0.0457	-0.0473	No Effect
Job	-0.0328	-0.0368	No Effect
Dependents	-0.003	-0.003	No Effect
Phone	-0.0365	-0.0365	No Effect

Table 2 Pearson & Spearman Correlation

The dataset predominantly exhibits linear relationships, with only minor non-linear interactions among a few features. These small differences suggest that the overall data does not demonstrate significant non-linearity.

Model Building

Model Building process in any Machine Learning project typically include problem definition, data collection, data preprocessing, exploratory data analysis (EDA), feature engineering, model selection, model training, model evaluation, model tuning, model testing, model deployment, and model maintenance. Now, with problem defined, data loaded, EDA performed, and data prepared to some extent by preprocessing, the focus is now on further preparation of data by splitting them to training & testing sets, model selection, model training & model evaluation followed by model tuning for optimization. Model testing, Model deployment and model maintenance are beyond the scope of this project. An iterative training, evaluation, and optimization, approach is followed to leverage the individual strengths of each model and handle complexity effectively.

DATA SPLITTING AND PREPARATION

The dataset is initially divided into predictors (all features except default) and target variable (*default*). Further, train and test sets are formed adhering to a common practice. The choice of train-test split ratio depends on: *Dataset size*, (1000 rows in this case and is relatively small), and *Model generalization*: We need enough training data for the model to learn well. In most of the cases 80:20 ratio, has worked out to be ideal and the same is followed here. This crucial step ensures that a substantial portion of data was available for training the models while retaining a separate subset for an unbiased evaluation of model performance. We have leveraged scikit-learn package of python and train_test_split method with specific hyperparameter *stratify=y,* so that the class imbalance is maintained in both the train & test sets.

MODEL SELECTION

Considering the dataset's complexity, which includes a mix of numerical and categorical data, presence of outliers, and class imbalance, a diverse array of models was necessary. The selection encompassed both **parametric models**, non-linear & linear models which required the addition of a class_weight="balanced" parameter to address the imbalance, and **tree-based models**, which naturally handle such disparities more effectively due to their inherent algorithms that do not make predictions based on probability distributions but split the data based on decision thresholds. But in extreme imbalance cases (e.g., 95% vs. 5%), adding class weights or SMOTE oversampling could improve tree-based models. We have again utilized scikit-learn package of python in building the models.

The models selected, trained & evaluated cover a broad spectrum of machine learning techniques. The Parametric Models include Linear Discriminant Analysis, Quadratic Discriminant Analysis & Logistic Regression (adjusted for class balance using hyperparameter *class_weight="balanced"*). Tree based models include Adaptive Boosting Classifier, Gradient Boosting Classifier, Random Forest Classifier & Bagging Classifier.

MODEL TRAINING & VALIDATION

Each model is trained using 5-fold cross-validation technique during training to ensure comprehensive learning and to mitigate the risk of overfitting, thus enhancing the robustness and reliability of the predictive models. Post training through cross validation, the model is trained again in the entire training set and predictions are made in test set.

PERFORMANCE METRICS

Some of the important metrices for any classification problem are summarized in the table 3 below.

Metric	Description	Use Case
Accuracy	The ratio of correct predictions to the total predictions made.	Best used when the classes are balanced.
Precision	The percentage of positive predictions that are correct.	Important when false positives are costly.
Recall	The proportion of actual positive instances correctly identified by the model.	Crucial when false negatives are costly.
F1-Score	A harmonic mean of precision and recall, providing a balance between the two.	Useful when dealing with imbalanced classes.
AUC-ROC Score	Measures the model's ability to distinguish between classes.	Beneficial when decision thresholds are adjusted.

Table 3 Classification Metrics

EVALUATION CRITERIA

The following metrics are shortlisted considering the business questions that need to be answered:

Metric	Reason for Consideration
Precision	Given the problem nature (Loan Default), where false positives can lead to the loss of good customers.
Recall	Critical in scenarios such as Loan Default, where false negatives result in financial loss to the bank by lending money to defaulters.
F1-Score	Necessary due to the imbalance in the default class and to achieve a balance between precision and recall.
AUC Score	Important for properly distinguishing between positive and negative classes, ensuring accurate classification.

Table 4 Final Classification Metrics

In addition to the metrics in table 4, we have also evaluated cross validation F1-score to test F1 score, to understand how well the model generalizes.

MODEL OPTIMIZATION THROUGH HYPERPARAMETER TUNING AND FEATURE IMPORTANCE

Post-initial evaluations, the top three performing models are selected for further optimization via Grid Search Cross Validation, focusing on fine-tuning the model hyperparameters to extract maximum performance. Post fine-tuning of models, feature importance analysis was conducted for each of the tuned models to identify and retain only the most impactful features, that lead to build a streamlined model that improved the key performance metrics.

ENSEMBLE MODELING

To leverage the complementary strengths of the top models, a Voting Classifier ensemble of best performing finetuned models was implemented. This ensemble approach was evaluated to see if it enhanced the predictive accuracy beyond the individual models.

FINAL MODEL SELECTION

To conclusively determine the best-performing model for recommendation & deployment, a comprehensive comparison is made across all models—from the initial individual models to the refined models and the final ensemble on evaluation metrics.

For the scenario based on balancing out the defaulters & non defaulters, a Geometric Mean of the metrics was calculated to identify the model that best harmonized precision, recall, F1-score, and AUC, ensuring a robust decision-making tool for the bank.

RESULTS

Individual Models

The results of individual models in cross validation F1-score & test F1-score are summarized in the code output snippet as in left side code output snip in Figure. The right-side snip displays the results of other test data metrics such as Precision, Recall, F1-Score & Area under the ROC curve. This also includes the confusion matrix from the values of which precision & recall are calculated.

	Model	Mean CV F1-Score	Test F1 Score
4	LogisticRegression	0.562261	0.567901
6	QDA	0.490990	0.542373
1	GradientBoosting	0.483574	0.558559
0	AdaBoost	0.469141	0.568807
3	BaggingClassifier	0.455401	0.601942
2	RandomForest	0.421137	0.565657
5	LDA	0.367161	0.434783

	Model	Accuracy	Precision	Recall	F1-Score	AUC Score	Confusion Matrix
2	RandomForest	0.785	0.717949	0.466667	0.565657	0.800714	[[129, 11], [32, 28]]
3	BaggingClassifier	0.795	0.720930	0.516667	0.601942	0.788333	[[128, 12], [29, 31]]
1	GradientBoosting	0.755	0.607843	0.516667	0.558559	0.778452	[[120, 20], [29, 31]]
0	AdaBoost	0.765	0.632653	0.516667	0.568807	0.771726	[[122, 18], [29, 31]]
6	QDA	0.730	0.551724	0.533333	0.542373	0.757619	[[114, 26], [28, 32]]
4	LogisticRegression	0.650	0.450980	0.766667	0.567901	0.752738	[[84, 56], [14, 46]]
5	LDA	0.740	0.625000	0.333333	0.434783	0.745119	[[128, 12], [40, 20]]

Figure 7 Results - Individual Models

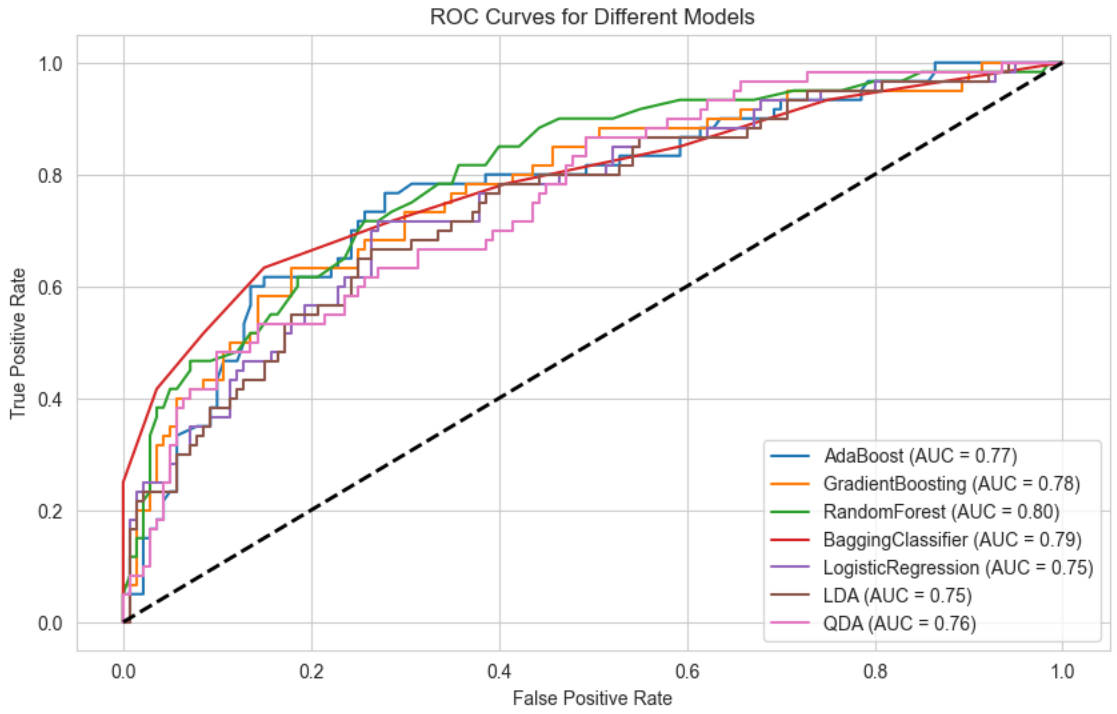

Figure 8 ROC Curves - Individual Models

Figure 8 visualizes the ROC curves that represents the trade-off between a model's True Positive Rate (TPR) and False Positive Rate (FPR), that helps us assess how well a model can distinguish between positive and negative classes. A curve closer to the top-left corner indicates a better performing model.

Hyperparameters Tuned Models

Similarly, Figure 9 displays the code output snip containing the best parameters along with Cross Validation F1-score & test F1-score of the hyper parameters tuned models (Logistic Regression, Bagging Classifier & Random Forest). Figure 10 represents the code output snippet of other test metrices of hyper parameters tuned models & Figure 11 visualizes the ROC curves of the same.

	Model	Best Params	Best CV F1 Score	Test F1 Score
1	Logistic Regression	{'C': 0.1, 'penalty': 'l2', 'solver': 'lbfgs'}	0.567664	0.567901
0	Bagging Classifier	{'bootstrap': True, 'max_features': 1.0, 'max_samples': 0.5, 'n_estimators': 50}	0.501874	0.566038
2	Random Forest	{'bootstrap': False, 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}	0.506441	0.563636

Figure 9 CV F1-score v/s test F1-score (Hyper parameters tuned)

	Model	Precision	Recall	F1 Score	AUC Score	Confusion Matrix
0	Logistic Regression tuned	0.450980	0.766667	0.567901	0.751071	[[84, 56], [14, 46]]
1	Bagging Classifier tuned	0.652174	0.500000	0.566038	0.788571	[[124, 16], [30, 30]]
2	Random Forest tuned	0.620000	0.516667	0.563636	0.788512	[[121, 19], [29, 31]]

Figure 10 Test scores (Hyper parameters tuned)

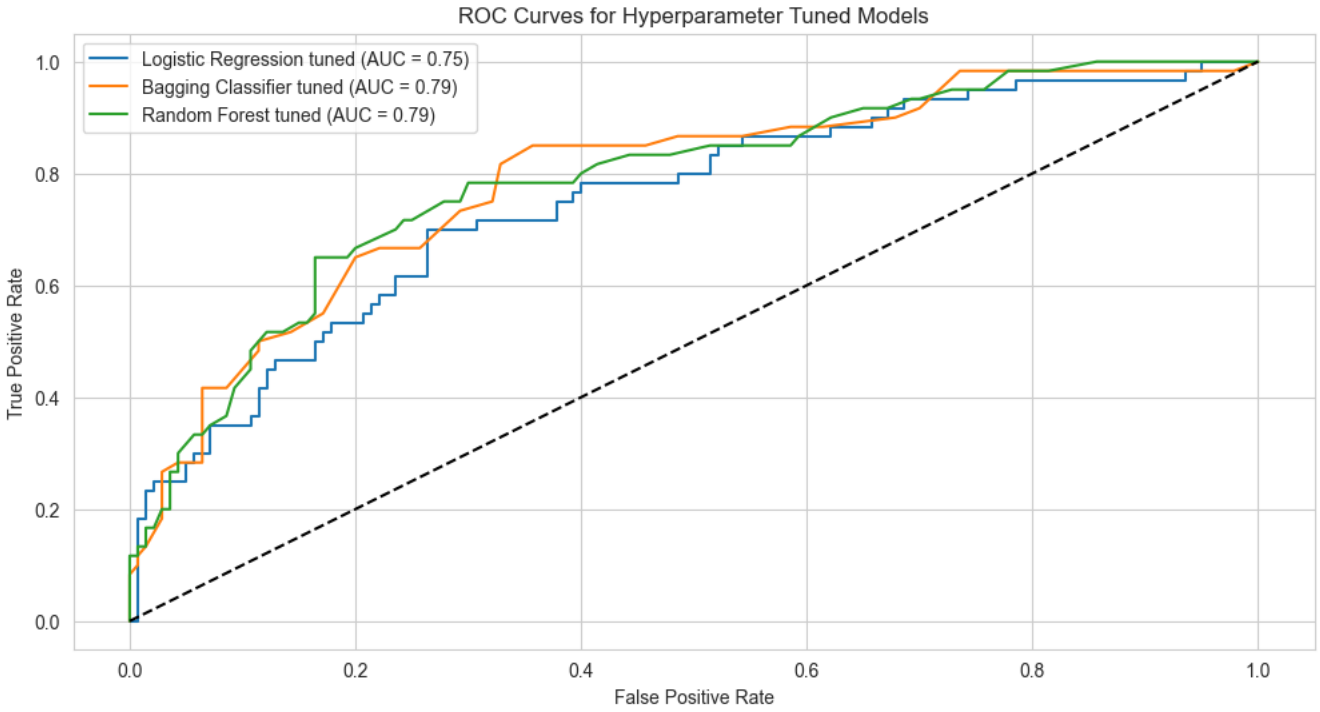

Figure 11 ROC Curves - Hyper parameters tuned

Feature Importance

Results of feature importance are shown in the code output snip of Figure 12.

Feature	Logistic Regression	Bagging Classifier	Random Forest
amount	0.228204	0.246454	0.161543
age	0.103192	0.124112	0.122930
checking_balance	0.608539	0.145724	0.119416
months_loan_duration	0.296273	0.059232	0.108834
credit_history	0.288397	0.078460	0.060494
savings_balance	0.195658	0.027776	0.056908
employment_duration	0.122244	0.055085	0.054157
years_at_residence	0.020065	0.023198	0.049153
purpose	0.014395	0.030074	0.048978
percent_of_income	0.266200	0.051513	0.048258
job	0.004412	0.042084	0.040364
other_credit	0.155032	0.058679	0.033848
housing	0.121092	0.023529	0.032070
existing_loans_count	0.029579	0.000000	0.023396
phone	0.176645	0.009352	0.021794
dependents	0.041861	0.024727	0.017858

Figure 12 Feature importance (Hyper parameters tuned)

Models with most impactful features

Figure 13 displays the code output snippet of various test metrics at the top & ROC curves at the bottom portion, of models only with the most impactful features.

	Model	Precision	Recall	F1 Score	AUC Score	Confusion Matrix
1	Bagging Classifier tuned (with selected top features)	0.666667	0.533333	0.592593	0.778810	[[124, 16], [28, 32]]
0	Logistic Regression tuned (with selected top features)	0.460000	0.766667	0.575000	0.784762	[[86, 54], [14, 46]]
2	Random Forest tuned (with selected top features)	0.583333	0.466667	0.518519	0.733929	[[120, 20], [32, 28]]

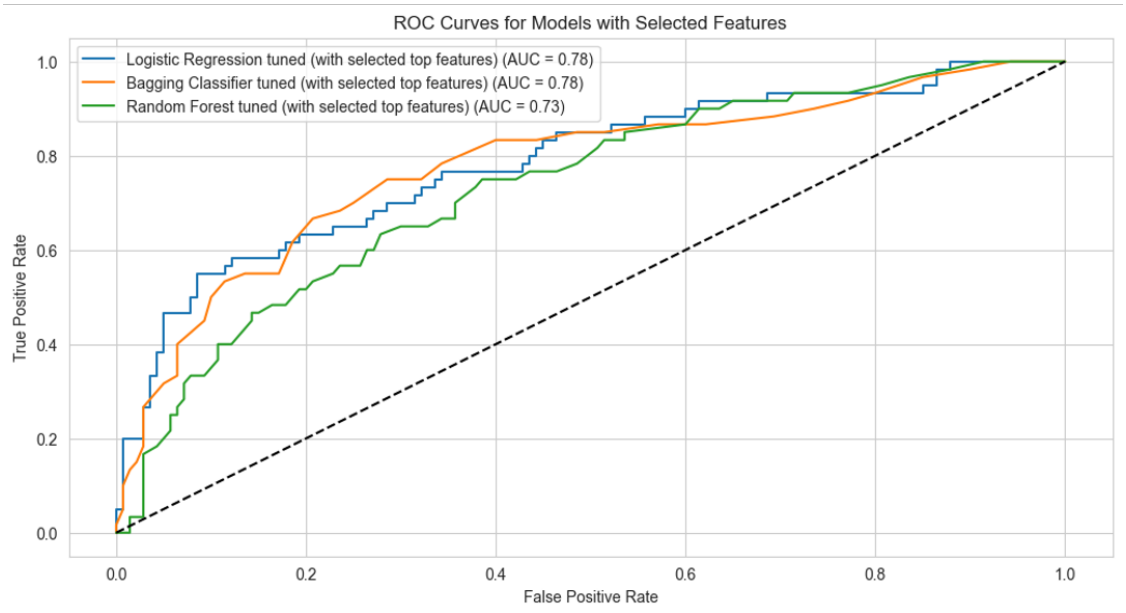

Figure 13 Test scores & ROC Curve - Models with impactful features

Ensemble Model (Voting Classifier)

The top portion of Figure 14 displays the code output snip of the test scores of Ensemble model, while the bottom portion visualizes the ROC curve.

	Model	Precision	Recall	F1 Score	AUC Score	Confusion Matrix
0	Ensemble (Logistic + Bagging)	0.569444	0.683333	0.621212	0.808929	[[109, 31], [19, 41]]

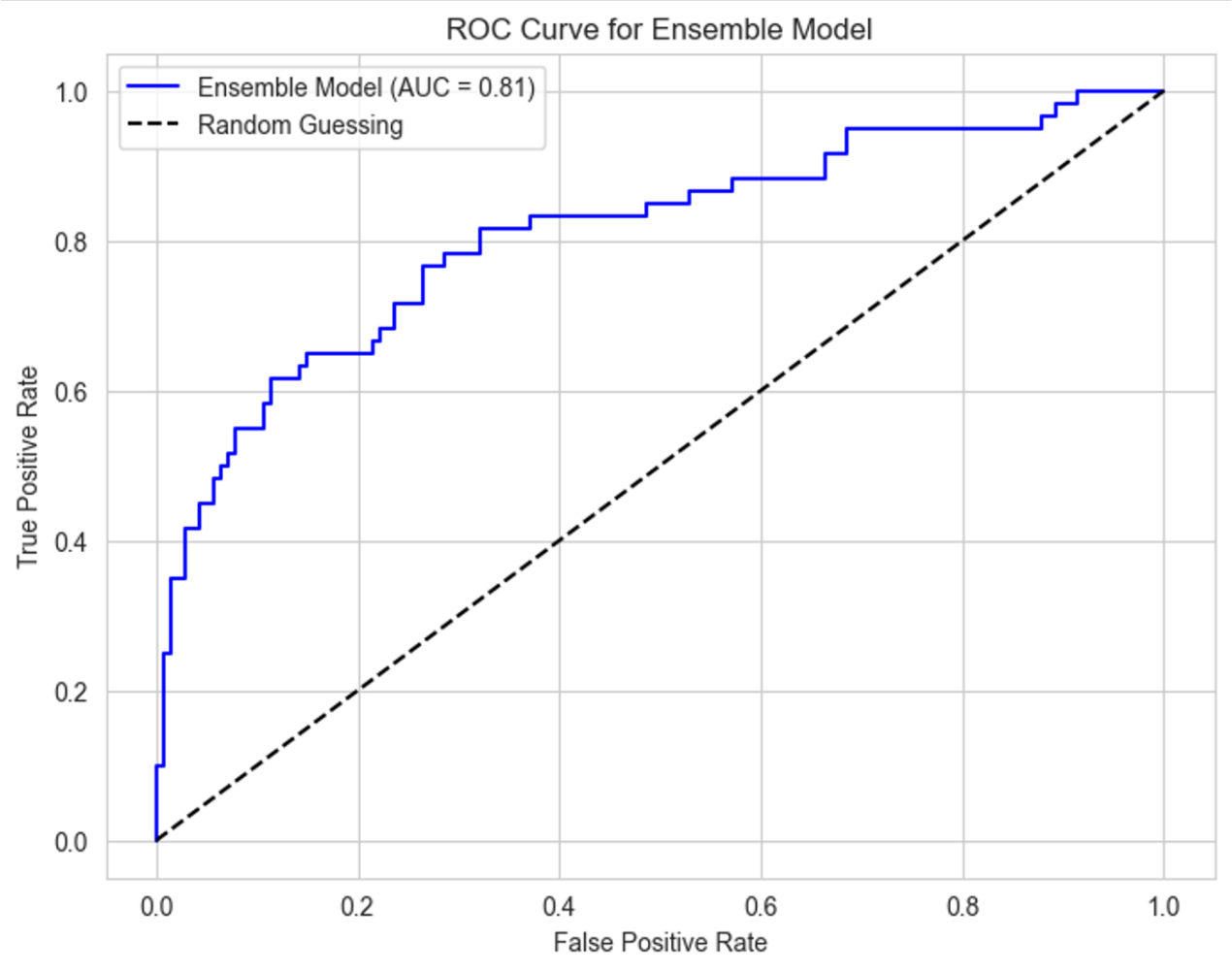

Figure 14 Test scores & ROC Curve - Ensemble Model (Logistic Regression + Bagging classifier)

DISCUSSION

The prime objective of the discussion is to interpret the results so that the research questions in the introduction section could be answered. While the first two questions can be answered from the EDA performed, the last question (question 3) need to be answered based on the prediction model performances shared in the

results section. Table 5 helps us in understanding how the metrics are related to the question and aids us in shortlisting the best performing models in each step of model building and optimization.

Research question	Business Requirement	Metric Interpretation	Metric Requirement
3. a	Catch as many defaulters as possible	Reduce False Negatives	Highest Recall
3. b	Reduce false alarms there by retaining the customers	Reduce False Positives	Highest Precision
3. c	A well-balanced approach addressing both catching defaulters and reducing false alarms	A good F1 score and AUC	Highest F1-score/ AUC

Table 5 Business requirement v/s Metrics interpretation

Individual Models

From left side snip of Figure 7, the evaluation of individual models reveals distinct performances, based on the comparison of Cross validation F1-score & Test F1-score. This essentially indicate how well a model can generalize on unseen data. Bagging Classifier, Logistic Regression, and Gradient Boosting shows the best results. In contrast, LDA and Random Forest demonstrated weaker outcomes. The Bagging Classifier leads with a 0.6019 Test F1 Score, indicating strong capabilities in balancing precision and recall, though it shows signs of underfitting as evidenced by a significant gap between its cross-validation and test F1 scores. Logistic Regression, with closely aligned CV and Test scores, exhibits good generalizability. Gradient Boosting, while consistent, ranks slightly below the first two in terms of performance. LDA and Random Forest lag in effectiveness, with LDA unable to handle complex relationships in the data and Random Forest displaying underfitting in cross-validation scenarios.

The Bagging Classifier, despite its top performance on the Test F1 Score, shows a potential risk of underfitting, which could limit its efficacy in broader applications. Logistic Regression stands out for its stability and generalizability across different datasets, making it a reliable choice. Gradient Boosting, although consistent, might require adjustments to enhance its performance to match the leading models. The poor performance of LDA suggests it struggles with the dataset's complexity, while Random Forest's significant discrepancy between its CV and Test scores indicates a pronounced underfitting issue, reducing its reliability. The primary limitation noted in the analysis is the underfitting observed in both the Bagging Classifier and Random Forest models. This underfitting could affect their performance.

However, in addition to the CV v/s test F1-scores comparison, we have other metrics such as precision, recall, F1-scores & AUC in the right-side snip of Figure 7. With the understanding of Business & Metric interpretations as in Table 4, the best model to address 3a would be Logistic Regression (with the highest recall of 0.7667) while the best model to address 3b would be Bagging Classifier (with the highest precision of 0.7209). Scenario 3c is addressed by two models namely Bagging Classifier (considering highest F1-score of 0.6019) and Random Forest (considering highest AUC of 0. 8007).

However, to further narrow down, we shortlist the models that address the business scenarios and further enhance shortlisted models performance and reliability by fine tuning their hyper parameters.

Hyperparameters Tuned Models

From Figure 9, it is observed that the dataset predominantly showcases linear dynamics, which is evident from the superior performance of Logistic Regression in both cross-validation (CV) and test environments, with the best F1 scores (0.5676 and 0.5679 respectively). With minimal discrepancy between the two, there is negligible overfitting. This model, fine-tuned with hyperparameters *C: 0.1*, *penalty: l2*, and *solver: lbfgs*, emerges as the most reliable for generalization. In contrast, while the Bagging Classifier and Random Forest tree-based models known for handling non-linear relationships—also perform well, but slightly lag behind Logistic Regression. The Bagging Classifier, has the hyper parameters *n_estimators: 50, max_features: 1.0, max_samples: 0.5, bootstrap: True,* while the Random Forest has the hyper parameters *n_estimators: 100, max_depth: None, min_samples_split: 2, min_samples_leaf: 1, bootstrap: False.*

Further, Figure 10, it can be observed that the Bagging Classifier excels in precision with a score of 0.652, indicating it produces fewer false positives and more confident predictions. Logistic Regression achieves the highest recall at 0.767, effectively identifying actual defaulters, albeit with a trade-off in precision. It also offers the most balanced F1 Score of 0.568. For AUC scores, which measure the ability to distinguish between classes, both the Bagging Classifier and Random Forest stand out with a score of 0.789, highlighting strong class separation.

On analysing the confusion matrix in figure 10, Logistic Regression is tailored to maximize recall, capturing a higher number of defaulters but at the expense of accuracy, as evidenced by a significant number of false positives. The Bagging Classifier and Random Forest show a preference for precision, making them conservative in predicting defaulters but more accurate in dismissing non-defaulters (Fewer false positives 16

& 19 respectively). This approach reduces the likelihood of false alarms but misses some true default cases(Higher False Negatives 30 & 29 respectively).

It can be observed that each model's strength in one area is counterbalanced by a weakness in another. The AUC scores for the tree models are impressive, but this doesn't fully translate into superiority in other metrics like recall or precision individually.

Moving forward, a deeper analysis of the feature importance within each model could provide insights into potential improvements or adjustments that could enhance their performance.

Feature Importance

Figure 12 presents feature importance across three predictive models—Logistic Regression, Bagging Classifier, and Random Forest. On Analysing, we can observe that certain features consistently play significant roles in predicting outcomes. Key features like *amount*, *checking balance*, and *credit history* are universally important across all models. Logistic Regression emphasizes *checking balance* and *credit history*, indicating these factors' linear relationship with default probability. In contrast, tree-based models such as Bagging and Random Forest prioritize *amount* and *age*, reflecting their capability to capture complex interactions and dependencies between variables.

The consistent importance of *amount* and *checking balance* across models highlights their significance, with higher loan amounts and lower balances linked to increased default risks. Logistic Regression's focus on *checking balance* and *credit history* suggests a linear interpretation of risk, whereas the tree models' focus on *amount* and *age* indicates their strength in modelling non-linear relationships that may affect loan repayment. The inclusion of *purpose* as a significant feature in tree-based models points to nuanced risk assessments depending on the loan's intent. The table 6 below effectively summarizes the interpretations on the common important features across models.

Further, to move forward we retain only the features that are found to be identified as most important in each of the models *checking_balance*, *months_loan_duration*, *credit_history*, *percent_of_income*, *amount*, *age*, *purpose*. We will re-train and evaluate the models performance based on this recreated improved dataset.

Feature	Interpretation
Checking Balance	Customers with low checking balances are at higher risk of default.
Loan Amount	Higher loan amounts increase default risk.
Credit History	A bad credit history signals a higher likelihood of default.
Months Loan Duration	Longer loan durations might increase risk, possibly due to changes in financial stability.
Age	Older customers may have stable financials (lower risk), while younger ones may have less financial history.
Purpose	Certain loan purposes (e.g., car loans, business loans) might be riskier than others.

Table 6 Interpretation - Feature Importance

Models with most impactful features

Analysing figure 13, we can observe that both the Bagging Classifier and Logistic Regression have shown improvements in various metrics when utilizing a subset of impactful features compared to using all features. In contrast, Random Forest's performance has diminished.

Bagging Classifier excels in precision with a score of 0.667, indicating it is more reliable in predicting defaults accurately with fewer false positives. Logistic Regression achieves the highest recall at 0.767, effectively identifying a larger proportion of actual defaulters, albeit with a higher rate of false positives (54). The Bagging Classifier also leads in balancing precision and recall, as evidenced by its top F1 score of 0.593. For overall classification capability, Logistic Regression's AUC score of 0.785 indicates it has the best performance across models in distinguishing between classes.

However, Logistic Regression, while adept at identifying defaulters (high recall), compromises by misclassifying a significant number of non-defaulters as defaulters. Bagging Classifier, while more precise, fails to catch a considerable number of defaulters (lower recall), which could lead to losses if these defaulters cause significant financial damage.

Given the enhanced performance of the Bagging Classifier and Logistic Regression with selected features, these models are identified as prime candidates for further model optimization and hybrid ensemble approach that combines the strengths of Logistic Regression and Bagging Classifier.

Ensemble Model (Voting Classifier)

Ensemble methods combine the predictions from multiple models to improve the overall performance, typically yielding more accurate and robust results than any single model could achieve. One effective ensemble technique is the *Voting Classifier*, which aggregates the predictions of several base estimators to produce a final prediction based on the majority voting of these estimators. This approach leverages the strengths of various individual models, potentially compensating for their individual weaknesses.

In this specific ensemble of Logistic Regression & Bagging Classifier models with most impactful features, we are utilizing the *Voting Classifier* in a *soft* voting configuration. Soft voting involves averaging the probability estimates from each of the constituent models rather than a simple majority vote found in *hard* voting.

From figure 14, we can observe that Ensemble approach seems to have a good balance across metrics with a test precision of 0.56, test recall of 0.68, test F1-score of 0.62 and AUC score of 0.8. However, the results of all the above approaches are combinedly evaluated to arrive at final model recommendation.

Final Model Recommendation

Figure 15 displays the code output snip of test metrics of all the approaches. Based on the business scenarios, we can observe that **Individual base Logistic Regression model** performs the best to catch as many defaulters as possible (Highest Recall – 0.76), **individual base Bagging Classifier model** performs the best to reduce false alarms (Highest Precision – 0.72)and **Ensemble model (Voting Classifier)** is the best balanced model ensuring both catching defaulters and reducing false alarms (Highest F1-score – 0.62 & AUC – 0.81).

	Model	Precision	Recall	F1-Score	AUC Score
0	RandomForest	0.717949	0.466667	0.565657	0.800714
1	BaggingClassifier	0.720930	0.516667	0.601942	0.788333
2	GradientBoosting	0.607843	0.516667	0.558559	0.778452
3	AdaBoost	0.632653	0.516667	0.568807	0.771726
4	QDA	0.551724	0.533333	0.542373	0.757619
5	LogisticRegression	0.450980	0.766667	0.567901	0.752738
6	LDA	0.625000	0.333333	0.434783	0.745119
7	Logistic Regression tuned	0.450980	0.766667	0.567901	0.751071
8	Bagging Classifier tuned	0.652174	0.500000	0.566038	0.788571
9	Random Forest tuned	0.620000	0.516667	0.563636	0.788512
10	Bagging Classifier tuned (selected features)	0.666667	0.533333	0.592593	0.778810
11	Logistic Regression tuned (selected features)	0.460000	0.766667	0.575000	0.784762
12	Random Forest tuned (selected features)	0.583333	0.466667	0.518519	0.733929
13	Ensemble (Logistic + Bagging)	0.569444	0.683333	0.621212	0.808929

Figure 15 Test Metrics - All approaches

However, for identifying the best model that balance out the defaulters & non defaulters, a Geometric Mean of the metrics was calculated to identify the model that best harmonized precision, recall, F1-score, and AUC, ensuring a robust decision-making tool for the bank. Figure 16 shows the Geometric mean calculated for all the models. It is now evident even from Geometric mean that Ensemble model of Logistic Regression & Bagging Classifier, both being the final tuned with only impactful features is the best balanced model that balances out both Recall & Precision.

	Model	Precision	Recall	F1-Score	AUC Score	Geometric Balanced Score
0	RandomForest	0.717949	0.466667	0.565657	0.800714	0.624141
1	BaggingClassifier	0.720930	0.516667	0.601942	0.788333	0.648399
2	GradientBoosting	0.607843	0.516667	0.558559	0.778452	0.607891
3	AdaBoost	0.632653	0.516667	0.568807	0.771726	0.615462
4	QDA	0.551724	0.533333	0.542373	0.757619	0.589681
5	LogisticRegression	0.450980	0.766667	0.567901	0.752738	0.620041
6	LDA	0.625000	0.333333	0.434783	0.745119	0.509699
7	Logistic Regression tuned	0.450980	0.766667	0.567901	0.751071	0.619697
8	Bagging Classifier tuned	0.652174	0.500000	0.566038	0.788571	0.617668
9	Random Forest tuned	0.620000	0.516667	0.563636	0.788512	0.614260
10	Bagging Classifier tuned (selected features)	0.666667	0.533333	0.592593	0.778810	0.636464
11	Logistic Regression tuned (selected features)	0.460000	0.766667	0.575000	0.784762	0.631601
12	Random Forest tuned (selected features)	0.583333	0.466667	0.518519	0.733929	0.567330
13	Ensemble (Logistic + Bagging)	0.569444	0.683333	0.621212	0.808929	0.664980

Figure 16 Geometric Mean

Limitations & Future Scope

Despite following an approach of incremental improvements in overall scores of the models, we can see that the final scores average between 60% & 75%, which has further scope for improvement. Some spaces that could be explored as a future scope can be:

(i) increasing the size of the dataset by collecting more customer data for both defaulters & Non Defaulters

(ii) Feature engineering of predictors space, &

(iii) Other models like SVM & XGBoost

CONCLUSION

This project has successfully leveraged various techniques of Machine Learning from exploring data to training parametric & tree based machine learning models to analyse and predict customer default status from historical customer data from a German bank. A robust framework comprising Exploratory Data Analysis, data pre-processing, Model building that included training base ML models to Ensemble models, a step by step model optimization including hyper parameter tuning, feature importance analysis, feature selection was built. This reliable approach helped us answer the 3 research questions on feature relationships, important predictors of default & three business scenarios of i) catching as many defaulters as possible, ii) reducing false alarms there by retaining the customers and iii) a well-balanced approach addressing both catching defaulters and reducing false alarms. Further, a solid foundation has been laid for the future enhancements.