Documentation On:

Risk Analysis for Home Credit Default: Exploratory Data Analysis and Predictive Modeling

ABSTRACT

Housing loan prediction is a major challenge for financial institutions because accurate prediction can reduce risk and improve decision-making during the loan process. This study investigates a machine learning-based approach to predict home loans using real world data including borrowers, financial profiles, credit history, and loan terms. Various preprocessing techniques such as handling missing data and normalizing numerical features are used to ensure data quality.

The model development process optimizes prediction accuracy by integrating multiple al gorithms including logistic regression, decision trees, and integration techniques. Key m etrics such as accuracy, precision, recall, and F1 score are used to evaluate model perf ormance. Factor analysis provides lenders with a better idea of the most important factors affecting default. The study also provides guidance on interpretation to ensure that the results are complete and understandable to financial decision makers. Sustainable financial potential. Future work will include incorporating live data and exploring deep learning models to further improve predictions.

INDEX

Topics **Page no:**

INTRODUCTION

Risk analysis for home credit default involves evaluating the likelihood that a borrower will fail to make payments on a home loan (default) and the potential impact on the lender. Financial institutions conduct risk analysis to minimize losses, allocate appropriate resources, and make informed decisions about lending. It involves both qualitative and quantitative assessments, often utilizing statistical models and data analytics. A remarkable number has examined the factors affecting outstanding and problematic debt levels, and some have investigated the financial behavior in terms of responsibility, debt repayment, and credit misuse[1].

A wide range of socioeconomic, demographic, psychological, situational, and behavioral factors was explored, and their role in predicting the investigated outcome domain at various time-points was analyzed. Predictors that are commonly found to have a strong effect on dependent variables are important findings for practical applications. In this context, it can be estimated what kind of personality, attitudinal, behavioral, and situational characteristics will be searched in alternative data sources for digital lending[1].

Objectives:

1. Perform exploratory data analysis (EDA) to understand the distribution and characteristics of the home credit dataset.
2. Identify key factors and features influencing credit default through data visualization and statistical analysis.
3. Develop predictive models using machine learning algorithms to predict the likelihood of default for home credit applicants.
4. Evaluate the performance of the predictive models using appropriate evaluation metrics and techniques.
5. Provide insights and recommendations for mitigating credit default risk based on the analysis and modeling results.

This Project will focus on:

1. Exploratory Data Analysis (EDA): Understanding the dataset, its features, trends, and patterns.
2. Predictive Modeling: Using machine learning models to predict the likelihood that a customer will default on their home credit loan.

LITERATURE SURVEY

The literature survey emphasizes research on P2P loan default prediction and credit risk assessment using machine learning methods. Studies highlight advanced techniques like Random Forests, CNNs, SVMs, and XGBoost for predicting defaults with applications on Lending Club data. Logistic regression and innovative approaches like label propagation and MIL are applied for assessing borrower creditworthiness. For credit risk, ensemble learning, fuzzy inference systems, and GAMs enhance predictive accuracy. While Random Forests show promise, the survey suggests opportunities for improvement in handling imbalanced datasets and combining hybrid models for robust predictions.

redit scoring has become a vital tool in financial institutions' decision-making processes, with significant focus on improving prediction models for loan defaults. Traditional statistical techniques, such as Support Vector Machines (SVM), Logistic Regression, and Decision Trees, have been widely used for loan default prediction. SVMs, often combined with algorithms like Naïve Bayes, show high accuracy and fast execution times. Logistic regression, known for its simplicity and interpretability, is another popular method for predicting loan defaults, offering clear insights into the factors affecting defaults. Decision Trees, using algorithms like C4.5, have been employed for risk assessment by classifying borrowers based on various attributes.

In addition to these traditional methods, advanced techniques like Artificial Neural Networks (ANNs) and Genetic Programming (GP) are increasingly being used. Neural networks, especially multi-layered perceptrons, have shown high prediction accuracy, handling complex data well. GP, an extension of genetic algorithms, has also emerged as a powerful method, combining deep learning with evolutionary principles to identify patterns and generate classification rules, outperforming traditional models.

Recent research in loan prediction has focused on improving credit scoring through various machine learning (ML) and deep learning (DL) algorithms. Papers highlight the success of models like Random Forest, which offers high accuracy in predicting loan defaults, outperforming other algorithms such as logistic regression, decision trees, and support vector machines (SVM). Studies suggest that data preprocessing, including cleaning and feature selection, plays a crucial role in enhancing prediction accuracy. Models like XGBoost, decision trees (C4.5), and neural networks have also shown promise in credit risk assessment, although no single model emerges as universally superior. Researchers emphasize the importance of algorithm choice, feature selection, and data quality for effective loan approval decisions.

This research highlights various machine learning and deep learning models applied to loan prediction and default risk assessment. Studies have compared the performance of algorithms such as Random Forest, Decision Trees, Support Vector Machines (SVM), and Logistic Regression. Results show that Random Forest typically outperforms others in terms of accuracy, with some studies achieving over 80% accuracy using data preprocessing techniques like SMOTE. Additionally, methods such as C4.5 in Decision Trees, CNN for time-series analysis, and XGBoost for feature selection have also been explored, demonstrating diverse approaches to improving loan default prediction models. These advancements help businesses enhance decision-making by more accurately assessing credit risk and selecting qualified applicants.

Researchers have explored various machine learning methods for predicting loan defaults, including Logistic Regression, Decision Trees, Random Forest, and XGBoost. Logistic Regression is favored for its simplicity and performance in predicting default probabilities. Decision Trees are effective when there are fewer attributes and larger samples. Random Forest, by aggregating decision trees, outperforms individual trees, while XGBoost, an enhancement of Gradient Boosting, shows state-of-the-art results. Studies have demonstrated that models like XGBoost and Random Forest outperform traditional methods, achieving higher prediction accuracy in loan default scenarios.

PROPOSED SYSTEM

The data in the field of Finances tend to be very variable and collecting such data can be a very tedious task, but in this case, Home Credit has done most of the heavy lifting to provide us as clean of data as possible.

The dataset provided contains a vast number of details about the borrower. It is separated into several relational tables, which contain applicants' static data such as their gender, age, number of family members, occupation, and other necessary fields, applicant's previous credit history obtained from the credit bureau department, and the applicant's past credit history within the Home Credit Group itself. The dataset is an imbalanced dataset, where the negative class dominates the positive class, as there are only a few number of defaulters among all the applicants.

Data Specifications:-

There are 10 .csv files in total. They are:- HomeCredit_columns_description.csv- 36.51 KB, POS_CASH_balance.csv-374.51 MB, application_test.csv- 25.34 MB, application_train.csv-158.44 MB, bureau.csv- 162.14 MB, bureau_balance.csv- 358.19 MB, credit_card_balance.csv- 404.91 MB, installments_payments.csv- 689.62 MB, previous_application.csv- 386.21 MB, sample_submission.csv- 523.63 KB

application_{train|test}.csv- This is the maintainable, broken into two files for Train (with TARGET) and Test (without TARGET). Static data for all applications. One row represents one loan in our data sample.

bureau.csv- All client's previous credits provided by other financial institutions were reported to the Credit Bureau. For every loan in our sample, there are as many rows as the number of credits the client had in the Credit Bureau before the application date.

bureau_balance.csv- Monthly Balances of previous credits in the Credit Bureau. It has one row for each month of history of every previous credit reported to Credit Burea.

POS_CASH_balance.csv- Monthly Balance snapshots of previous POS (point of sales) and cash loans that the applicant had with HomeCredit.

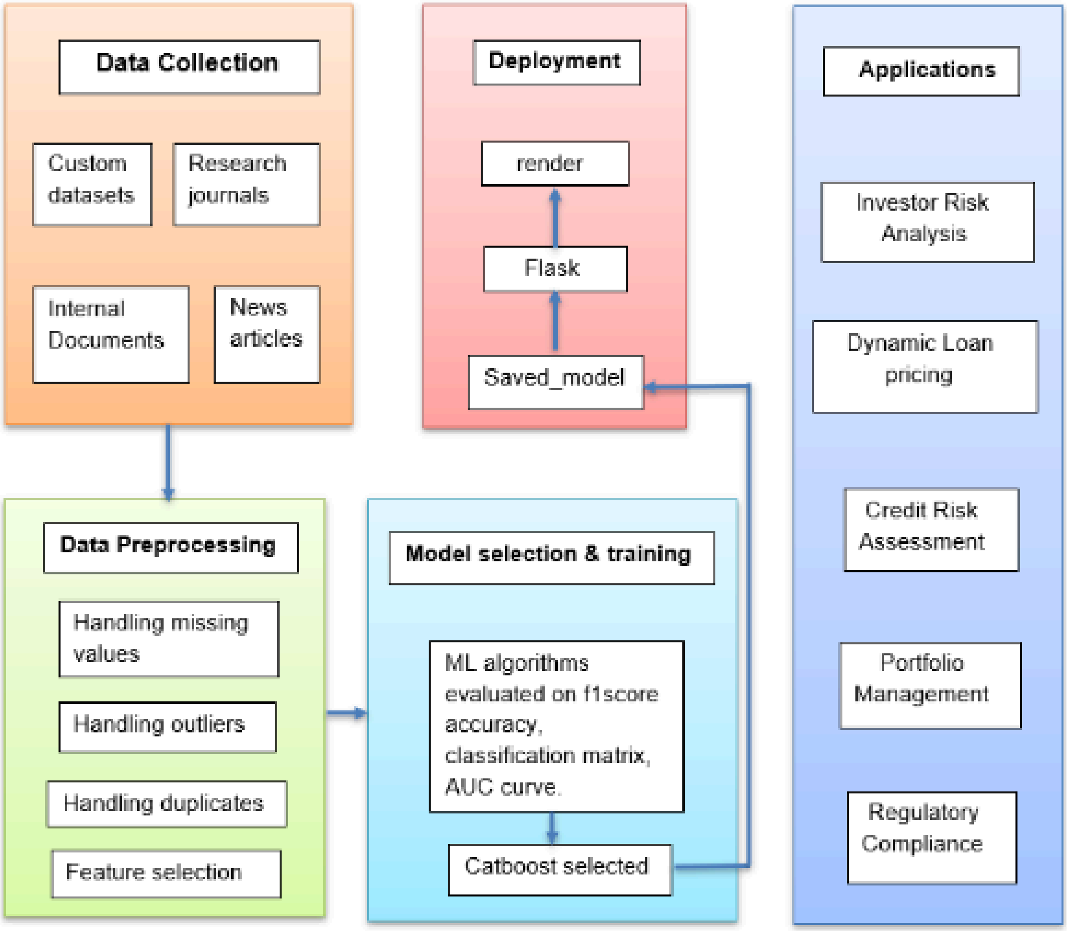
credit_card_balance.csv- Monthly Balance snapshots of previous credit cards that the applicant has with Home Credit.

previous_application.csv- All Previous applications for Home Credit loans of clients who have loans in our sample. There is one row for each previous application related to loans in our data sample.

installments_payments.csv- Repayment History for the previously disbursed credits in Home Credit related to the loans in our sample. There is a) one row for every payment that was made plus b) one row each for missed payment.

HomeCredit_columns_description.csv- Thisfile contains descriptions for the columns in the various data files.

System Workflow:-



1. **EDA:-** For the data analysis, we will follow following steps:

For each table, we will first check basic stats like the number of records in tables, Data in each table , number of NaN values, etc.

Next, we will explore some of the features with respect to the target variable for each table. We Will be employing the following plots For **Categorical Features**, we will mostly be using **Bar Plots**. For **Continuous/numeric features**, we will be using **Kernel Density Plot**

We will be drawing observations from each plot and note important insights generated from the plots.

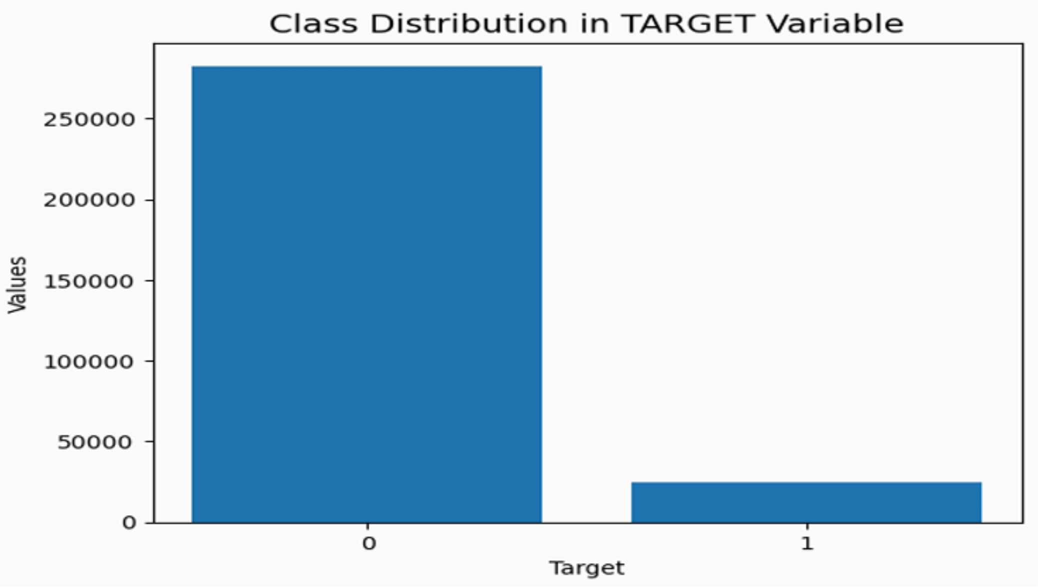


Fig 1- From the distribution of Target variable, one thing that we can quickly notice is the Data Imbalance. There are only 8.07% of the total loans that had actually been Defaulted. This means that Defaulters is the minority class. On the other hand, there are 91.9% loans which were not Defaulted. Thus, Non-Defaulters will be our majority class. The Defaulters have been assigned a Target variable of 1 and Non-Defaulters have been assigned Target Variable 0.

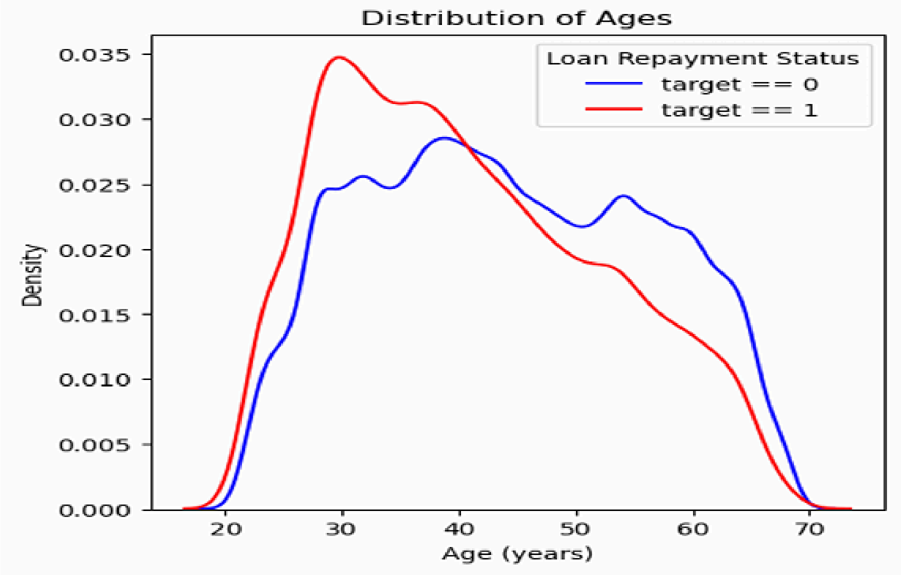


Fig 2- From the above plot, we can observe the peak of Age of people who Default to be close to 30 years. This means that the Defaulters are usually younger than Non-Defaulters.

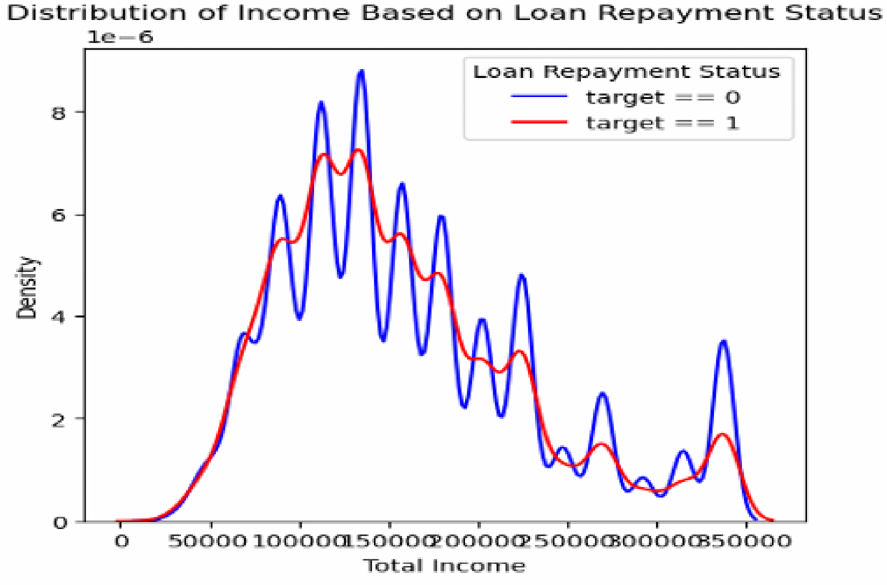


Fig 3- Here we observe that both distributions have similar peaks around lower income ranges (50,000–200,000), but the density for defaults is slightly lower overall. The distribution tails off similarly for higher incomes, indicating that both groups have similar income profiles, though the repayment success rate slightly varies across income levels.

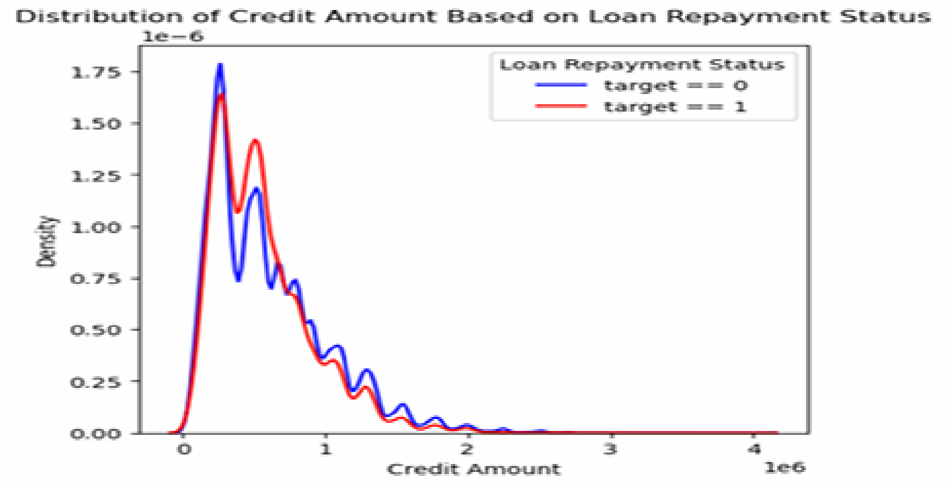


Fig 4- Most loans are concentrated at lower credit amounts (near 0–500,000), with successful repayments (blue line) slightly exceeding defaults (red line) in density. As the credit amount increases, both groups' densities decrease, but the successful repayment group consistently has a higher density across most credit ranges.

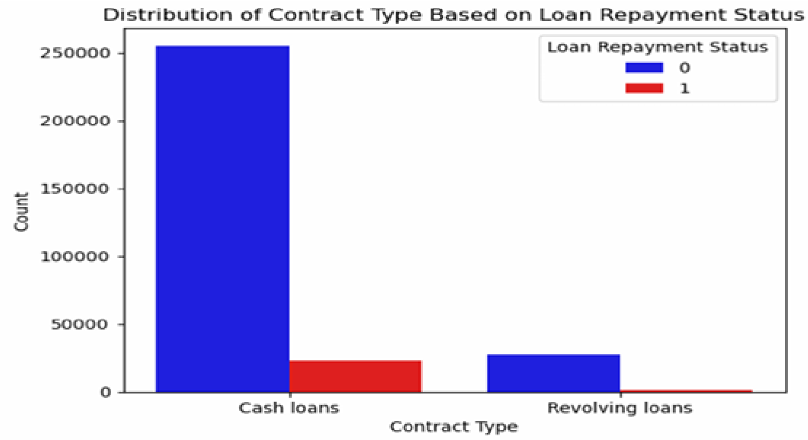


Fig 5- Cash loans have a significantly higher count, with successful repayments (blue) far outnumbering defaults (red). Defaults (red) are much fewer in both categories, indicating that the majority of loans are being repaid successfully, with cash loans being the most prevalent type. Revolving loans also show more successful repayments than defaults, but the overall count is much lower compared to cash loans.

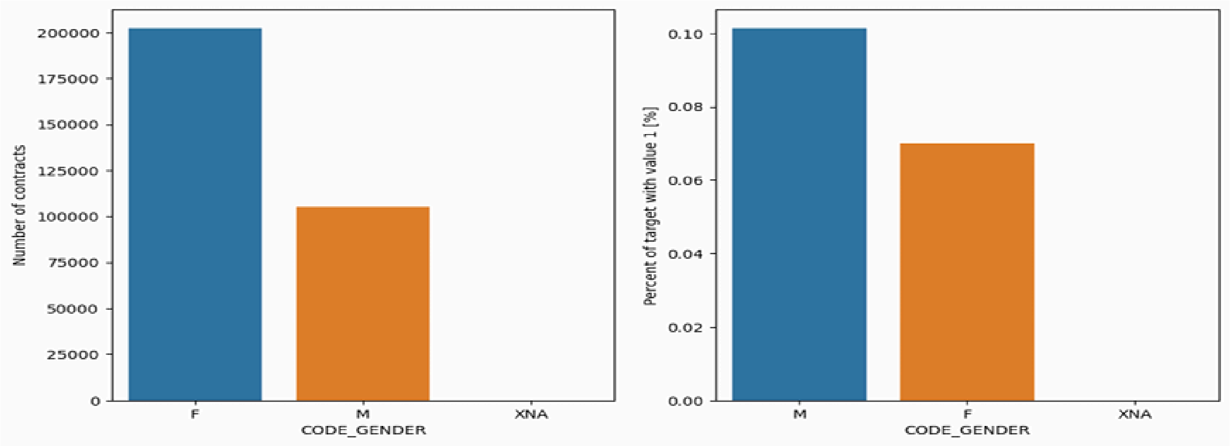


Fig 6- From the left subplot, we observe that more females (coded as "F") have taken loans than males (coded as "M"). The number of contracts taken by females is nearly double that of males. From the right subplot, the default rate for males is higher than that of females. Around 10% of male borrowers default, while approximately 7% of female borrowers default. The higher number of female borrowers may be due to factors such as gender-focused financial inclusion initiatives or women being more risk-averse in managing loans. However, males may have a higher default rate because they may take larger or riskier loans, leading to a higher probability of default.

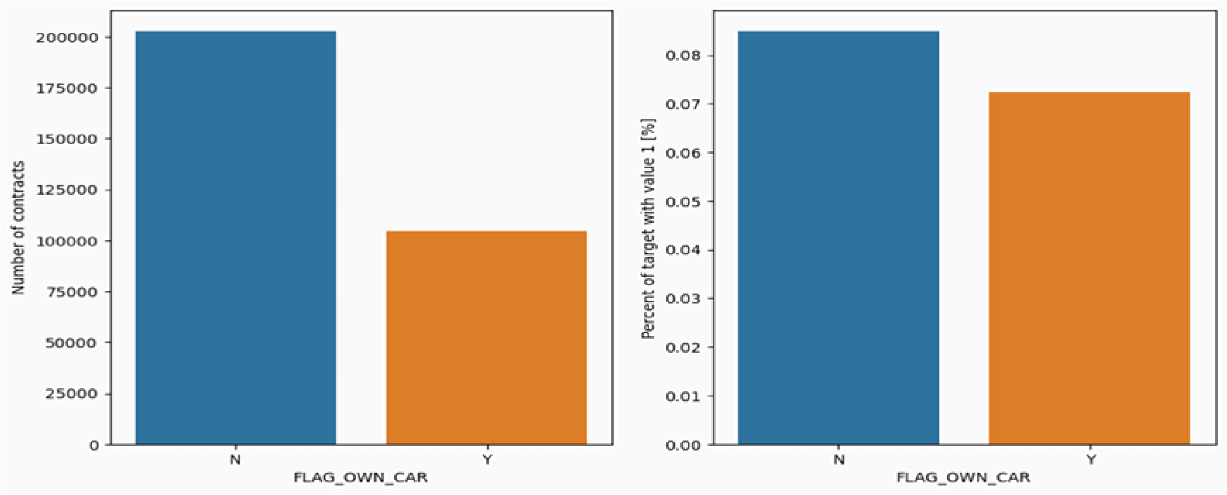


Fig 7- From the first subplot ,we observe that a larger number of individuals without a car have more contracts, roughly double, compared to those who own a car In the second subplot ,we can see that the percentage of defaulters (target with value 1) is higher among those who do not own a car.The default rate for car non-owners is slightly above 8%, while for car owners, it is closer to 7.5%. Higher contracts for non-car owners: The higher number of contracts for non-car owners might suggest that individuals without a car either need more financial assistance or might be more likely to apply for loans. Lower default rate among car owners: Car ownership could indicate better financial stability, which might explain the lower default rate among car owners.

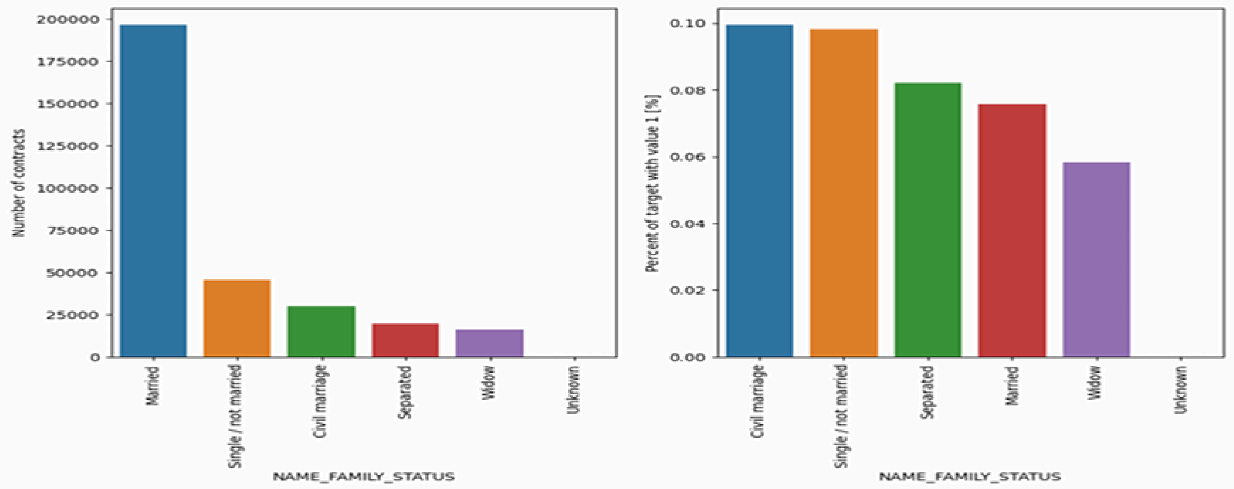
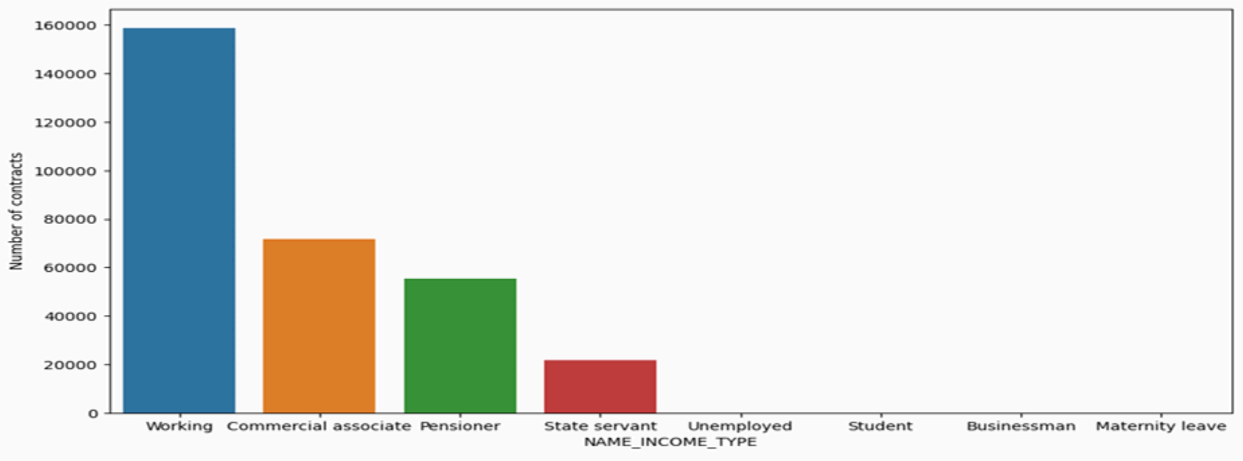


Fig 8- In the first subplot (left), the majority of contracts are held by married individuals, with single/not married individuals holding significantly fewer contracts. Civil marriage, separated, widow, and unknown statuses have much lower numbers. The second subplot (right) shows that civil marriage and single/not married individuals have the highest default rates, close to 10%. Widows have the lowest default rate, followed by married and separated individuals. Married individuals are the most common loan holders, suggesting they may seek loans for family or financial planning. Default risk is higher among those in civil marriages and single/not married individuals, possibly due to varying financial responsibilities or support systems. Widows show the least likelihood of default.



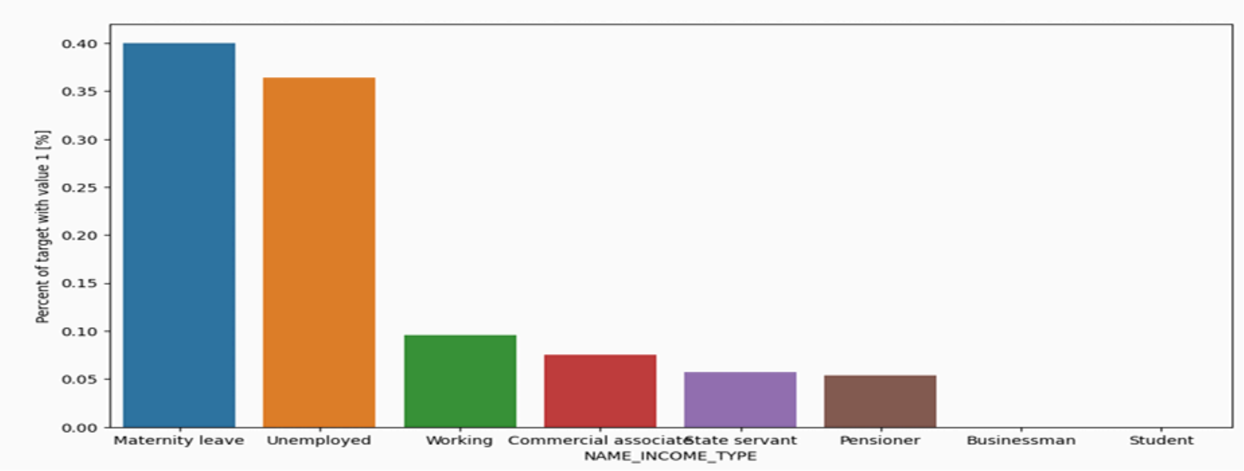


Fig 9- In the first subplot (top), most contracts are held by individuals who are working, followed by commercial associates and pensioners. Unemployed, students, businessmen, and those on maternity leave have the lowest contract numbers. The second subplot (bottom) shows that individuals on maternity leave and the unemployed have the highest default rates, with maternity leave nearing 40% and the unemployed close to 35%. The working population has a much lower default rate, followed by commercial associates and state servants. Working individuals take loans to cover major expenses, leverage opportunities, handle emergencies, consolidate debt, or because their stable income makes credit easily accessible. Default rates are significantly higher for those on maternity leave and the unemployed, possibly reflecting unstable or limited income during these periods. Working individuals show a lower default risk, indicating more financial security

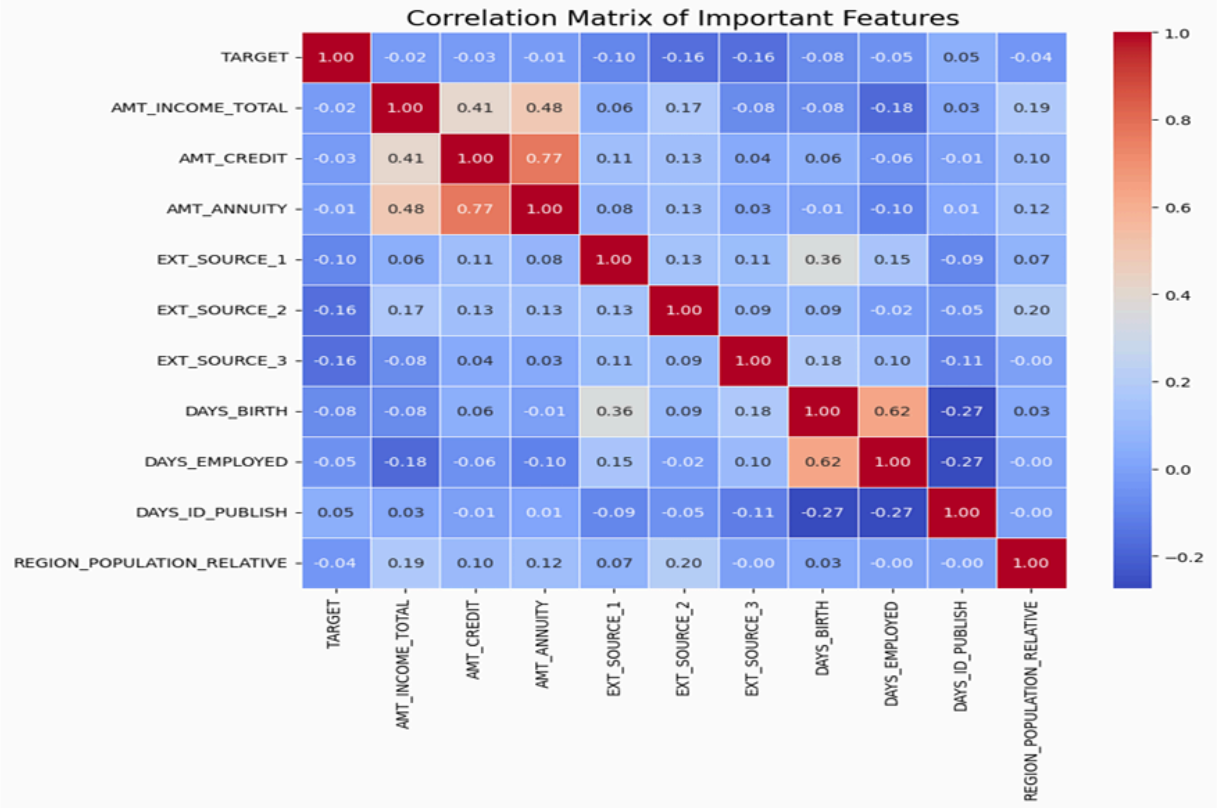


Fig 10- **Negative Correlation with TARGET**- DAYS_BIRTH(age) also has a slight negative correlation with the TARGET (-0.08), suggesting that older individuals are slightly less likely to default. **Positive Correlation**- AMT_CREDITisstrongly correlated with AMT_ANNUITY (0.77), indicating that as the credit amount increases, the annuity (loan payments) tends to increase proportionally. **Low Correlation with TARGET**- Most Features, such as AMT_INCOME_TOTAL, AMT_CREDIT, DAYS_EMPLOYED, and REGION_POPULATION_RELATIVE, have very low correlations with TARGET (near zero), meaning these features have little to no linear relationship with default risk.

Key findings include:

Demographic factors: Age, gender, and family status are associated with default risk. Younger borrowers, females, and married individuals tend to have lower default rates.

Income And Employment: Income Levels and employment stability are crucial factors. Lower income, unemployment, and maternity leave are linked to higher default risk.

Loan Characteristics: Loan type, credit amount, and contract duration influence default risk. Cash loans have higher default rates than revolving loans, and larger credit amounts are associated with increased risk.

2. **Data Preprocessing-** The goal is to ensure the data is in a format suitable for machine learning models.

Memory Optimization:

- The reduce_memory_usage function optimizes the memory usage of numerical columns by downcasting them to smaller data types (e.g., float64 → float32 or int64 → int8). This is useful when working with large datasets to reduce memory footprint.

Outliers:

- **Interquartile Range (IQR) Method** to identify potential outliers

Handling Large Data with Dask:

- The code uses **Dask** to handle large datasets in parallel and out-of-core processing.

- app_train, credit_card_balance, and app_test are converted to Dask DataFrames, enabling operations on chunks of data without loading the entire dataset into memory.

Merging Datasets:

- Data from multiple tables (app_train, credit_card_balance, and app_test) is merged on a common key (SK_ID_CURR). This ensures that all related information is consolidated into a single DataFrame for modeling.

Handling Missing Values:

- The SimpleImputer is used to replace missing values in numerical columns (AMT_BALANCE and SK_DPD) with the mean or median of those columns.
- Imputing missing values ensures that the machine learning model doesn't encounter errors due to NaN values.

Mapping Binary Variables:

- Binary columns like FLAG_OWN_CAR and CODE_GENDER are mapped to numerical values (Y/N, M/F) to make them suitable for machine learning.

Scaling Numerical Features:

- Numerical columns like AMT_INCOME_TOTAL, AMT_CREDIT, and SK_DPD are scaled using MinMaxScaler to normalize the data to a range between 0 and 1. This ensures that features with large ranges don't dominate those with smaller ranges.

One-Hot Encoding for Categorical Variables:

- Categorical variables such as CODE_GENDER, NAME_CONTRACT_TYPE, and NAME_HOUSING_TYPE are transformed into one-hot encoded columns. This creates binary indicators for each category, making them usable by machine learning models.

3. **Feature Engineering:-** Feature engineering involves transforming raw data into features that improve model performance. It ensures that the data is represented in a way that the algorithm can learn from effectively.

Defining Numerical and Categorical Features: Features are classified into numerical (numerical_features) and categorical (categorical_features) groups to apply appropriate preprocessing techniques.

Numerical Feature Preprocessing: A pipeline is defined to handle numerical features: Missing values are imputed using the mean. The data is standardized using StandardScaler, which scales features to have a mean of 0 and a standard deviation of 1.

Categorical Feature Encoding: Categorical features are one-hot encoded using OneHotEncoder. This converts categorical variables into a binary matrix suitable for machine learning algorithms.

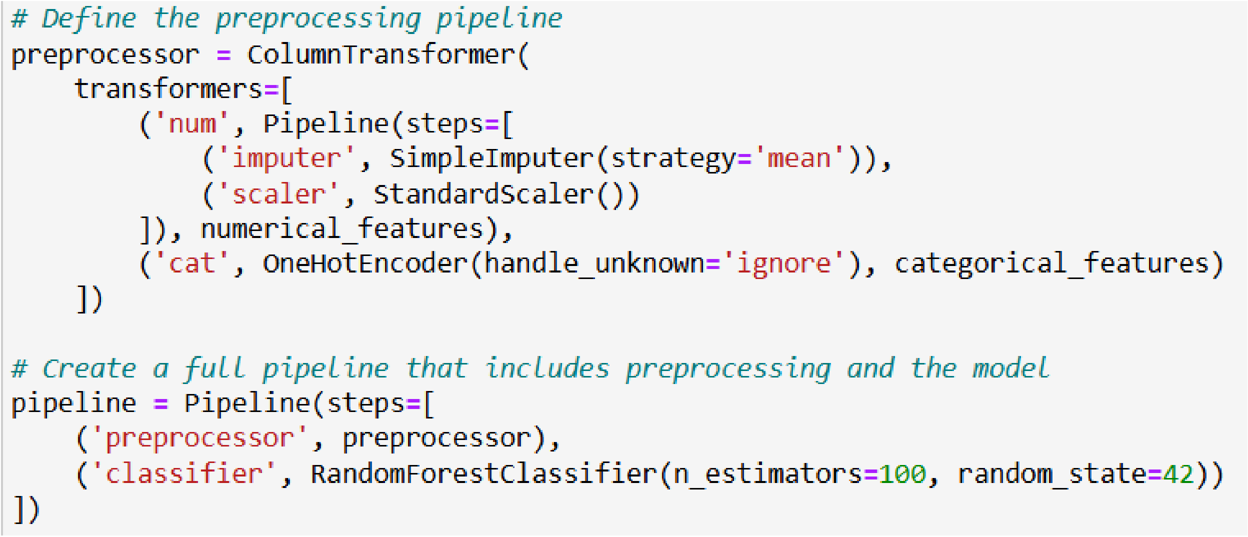
Combining Preprocessing Steps: The ColumnTransformer combines separate preprocessing pipelines for numerical and categorical features, ensuring a unified transformation process.

Pipeline Integration

The code integrates preprocessing and feature engineering into a single pipeline using Pipeline:

Preprocessor: Combines all preprocessing steps for numerical and categorical features.

Model: A RandomForestClassifier is added as the final step, enabling training directly on the preprocessed data.



```
# Define the preprocessing pipeline
preprocessor = ColumnTransformer(
    transformers=[
        ('num', Pipeline(steps=[
            ('imputer', SimpleImputer(strategy='mean')),
            ('scaler', StandardScaler())
        ]), numerical_features),
        ('cat', OneHotEncoder(handle_unknown='ignore'), categorical_features)
    ])

# Create a full pipeline that includes preprocessing and the model
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(n_estimators=100, random_state=42))
])
```

4. **Model Development:-** The provided code initializes and configures multiple machine learning models to predict binary outcomes (e.g., loan default). Each model is encapsulated in a pipeline that integrates preprocessing and classification steps, ensuring streamlined data handling and training.

Preprocessing Pipelines:

Imputation: Missing values are handled using SimpleImputer with strategies like mean or median.**Scaling**: Continuous features are scaled using StandardScaler for models like SVM, KNN, and Neural Networks, which are sensitive to feature magnitudes. **One-Hot Encoding**: Categorical variables are converted into numerical formats using OneHotEncoder for compatibility with machine learning algorithms.

Model Initialization:

A range of algorithms is included, such as:

- Tree-based: Random Forest, Decision Tree, XGBoost, CatBoost, LightGBM, and AdaBoost.
- Linear models: Logistic Regression.
- Non-linear models: SVM, K-Nearest Neighbors, Neural Networks.
- Probabilistic models: Naive Bayes.

A deep neural network (DNN) is defined separately for binary classification tasks using TensorFlow/Keras. It includes layers for feature transformation and a final sigmoid layer for binary output probabilities.

5. **Model Training:-**
   Each model is trained on the preprocessed dataset (X_train and y_train) to learn patterns in the data:

Training Pipeline Models: The .fit() method is applied to train each pipeline model on the training dataset. Predictions are generated on the test set (X_test) using .predict().

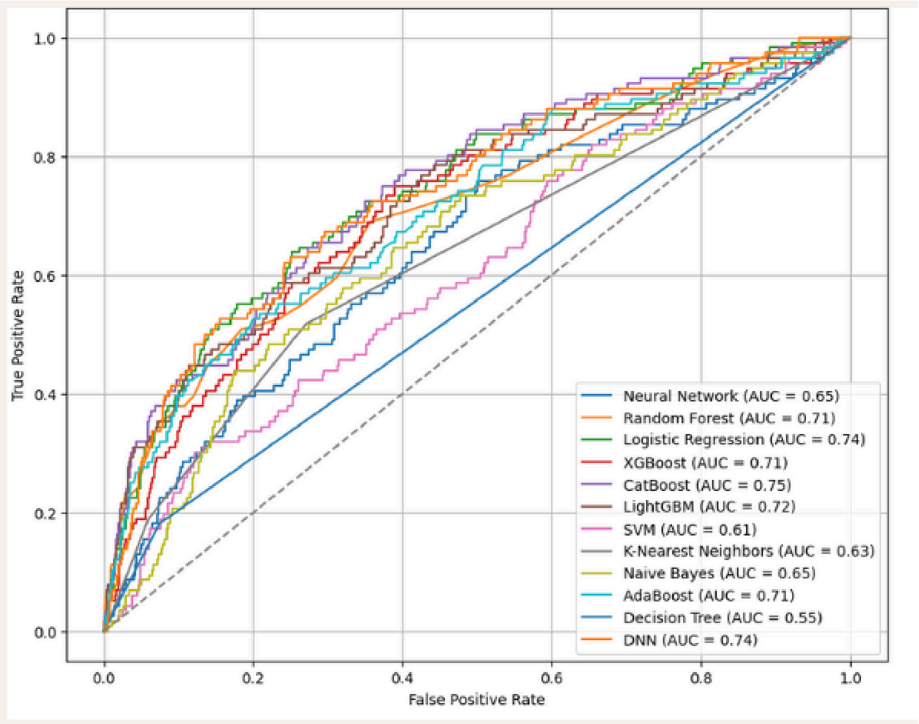
Training the DNN: The DNN is trained using the .fit() method with specified epochs and batch sizes. Predictions are computed using the .predict() method, with probabilities converted into binary labels for evaluation.

Evaluation
For each model, the following metrics are computed:

- **Accuracy**: Measures overall correctness.
- **F1 Score**: Balances precision and recall.

- **Confusion Matrix**: Captures true positives, true negatives, false positives, and false negatives.
- **AUC-ROC**: For models supporting probability outputs, an AUC-ROC curve is generated to assess the tradeoff between true positive and false positive rates.

ML Algorithm	**F1 Score**	**Accuracy**	**AUC**
Neural Network	0.1156	90.36	0.65
CatBoost	0.0496	92.75	0.75
Logistic Regression	0.0168	92.68	0.74
DNN	0	92.69	0.74
LightGBM	0.1085	92.75	0.72
Random Forest	0.0333	92.69	0.71
XGBoost	0.0896	92.31	0.71
AdaBoost	0.1037	92.38	0.71
Naive Bayes	0.1372	8.88	0.65
KNN	0.0451	92	0.63
SVM	0	92.69	0.61
Decision Tree	0.1707	87.15	0.55



CatBoost and LightGBM emerge as top contenders, with CatBoost being slightly better in AUC, while LightGBM offers a higher F1 score. Either could work well, especially if you apply class imbalance handling techniques. So, we have chosen CatBoost for our project.

Special Handling:

- For the DNN, predictions are thresholded at 0.5 to convert probabilities into binary classifications.
- Only models supporting probability outputs compute AUC-ROC metrics.

Frontend and Connectivity:-

Our Flask application provides an API endpoint that integrates a machine learning model (saved as a pipeline) for credit default prediction. The application accepts input from a frontend (via a form in index.html), processes the data, makes a prediction, and returns the results in JSON format.

The Flask app is created using the Flask class. The model pipeline is loaded from a file using joblib.load().
The calculate_credit_score() function computes a credit score based on various financial features (e.g., payment history, credit utilization, credit history length, etc.).

These features are provided in the input data, and the function outputs a calculated score out of 850.

- The determine_fico_range() function maps the calculated credit score to a FICO range. The score is classified into categories such as "Exceptional", "Very Good", "Good", etc.

/ route: Renders the index.html page (frontend).

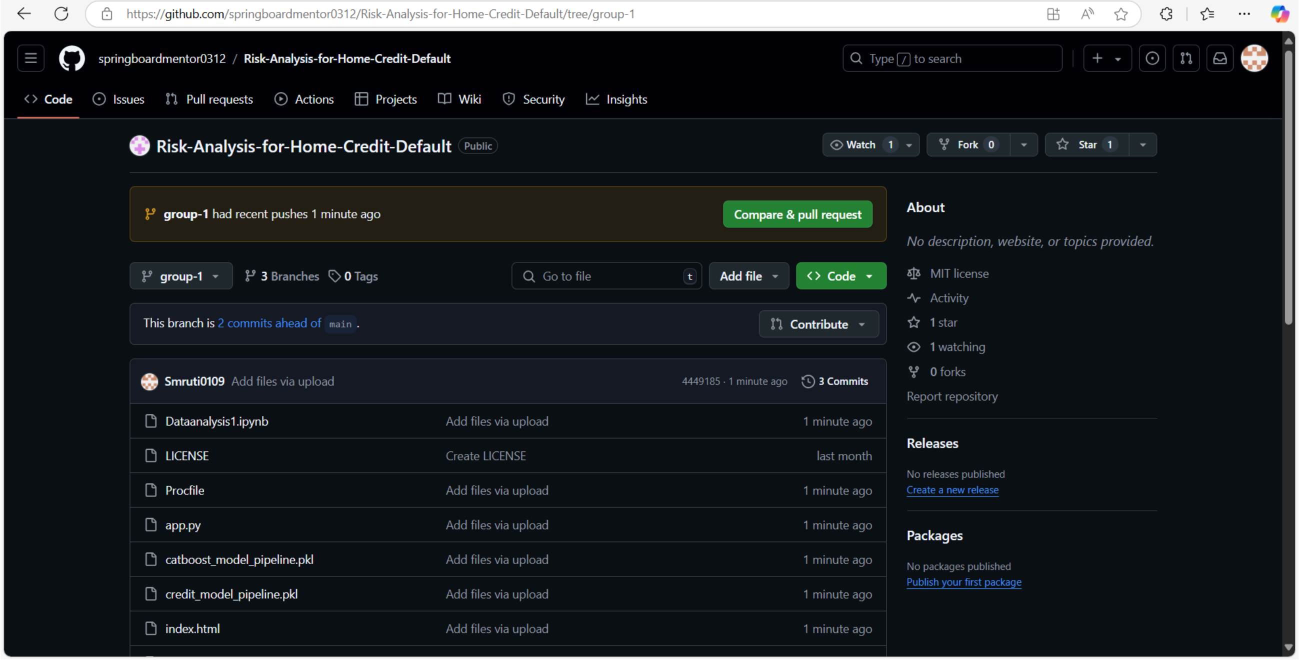
/predict route: Handles POST requests where the frontend sends form data. The data is converted to a dataframe, passed through the model pipeline for prediction, and also processed for credit score calculation.

The input_data is gathered from the form in index.html, where each form input is either converted to a float (for numerical values) or left as a string (for categorical values).
After extracting and preparing the input data, the input_df (pandas dataframe) is passed into the pre-trained pipeline for prediction.
The prediction (1 for defaulter, 0 for non-defaulter) is returned by the model.

The results of the prediction, credit score, and FICO score range are bundled into a dictionary and returned as a JSON response.
Any errors during the prediction process (such as missing or invalid input data) are caught by the try-except block, and an error message is returned as a JSON response.

Deployment :-

For deployment, we used Render, a cloud platform that simplifies web app hosting. My Flask application, along with its dependencies specified in a requirements.txt file, was deployed directly from the Git repository. The project is structured to include a templates/ folder for HTML files, but for this deployment, we adjusted the app to locate the index.html file appropriately. Render automatically builds and deploys the app, providing a live URL for access.


https://github.com/springboardmentor0312/Risk-Analysis-for-Home-Credit-Default/tree/group-1
springboardmentor0312 / Risk-Analysis-for-Home-Credit-Default
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Risk-Analysis-for-Home-Credit-Default Public
Watch 1
Fork 0
Star 1
Compare & pull request
group-1
3 Branches
0 Tags
Go to file
Add file
Code
Contribute

group-1 had recent pushes 1 minute ago

This branch is 2 commits ahead of `main` .

Smruti0109 Add files via upload — 4449185 · 1 minute ago — 3 Commits

Dataanalysis1.ipynb	Add files via upload	1 minute ago
LICENSE	Create LICENSE	last month
Procfile	Add files via upload	1 minute ago
app.py	Add files via upload	1 minute ago
catboost_model_pipeline.pkl	Add files via upload	1 minute ago
credit_model_pipeline.pkl	Add files via upload	1 minute ago
index.html	Add files via upload	1 minute ago

About

No description, website, or topics provided.

- MIT license
- Activity
- 1 star
- 1 watching
- 0 forks

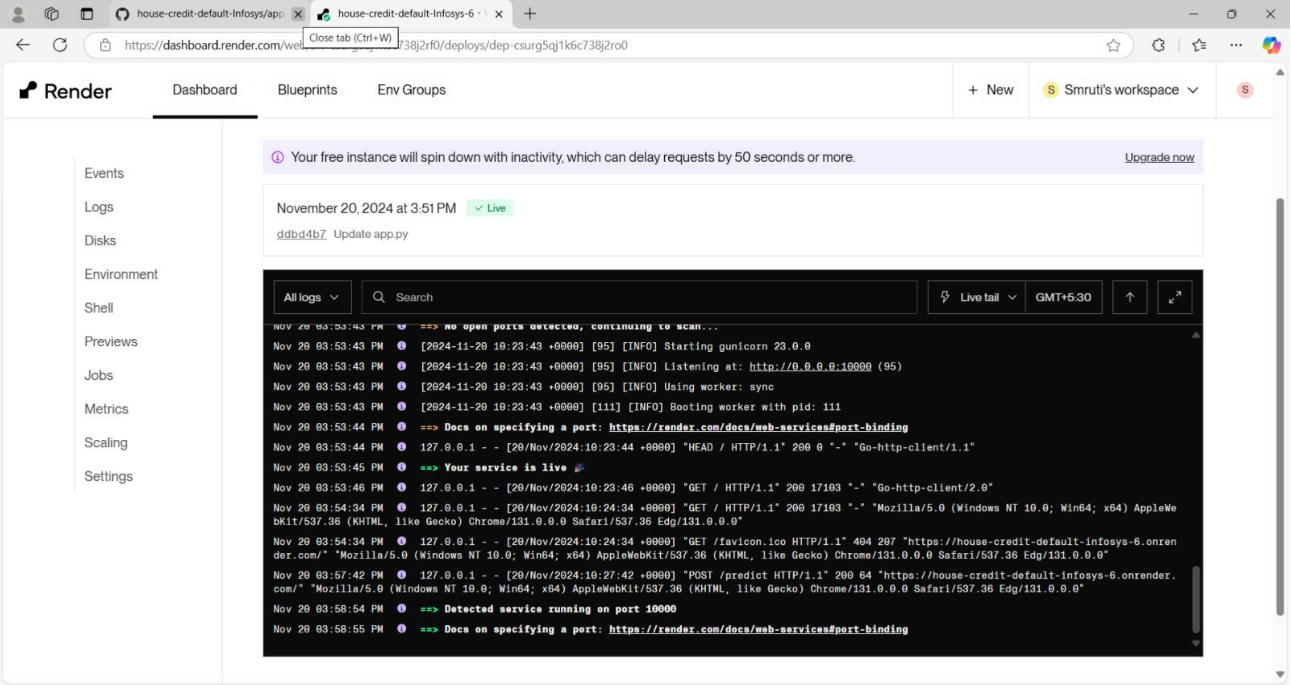
Report repository

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package



Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

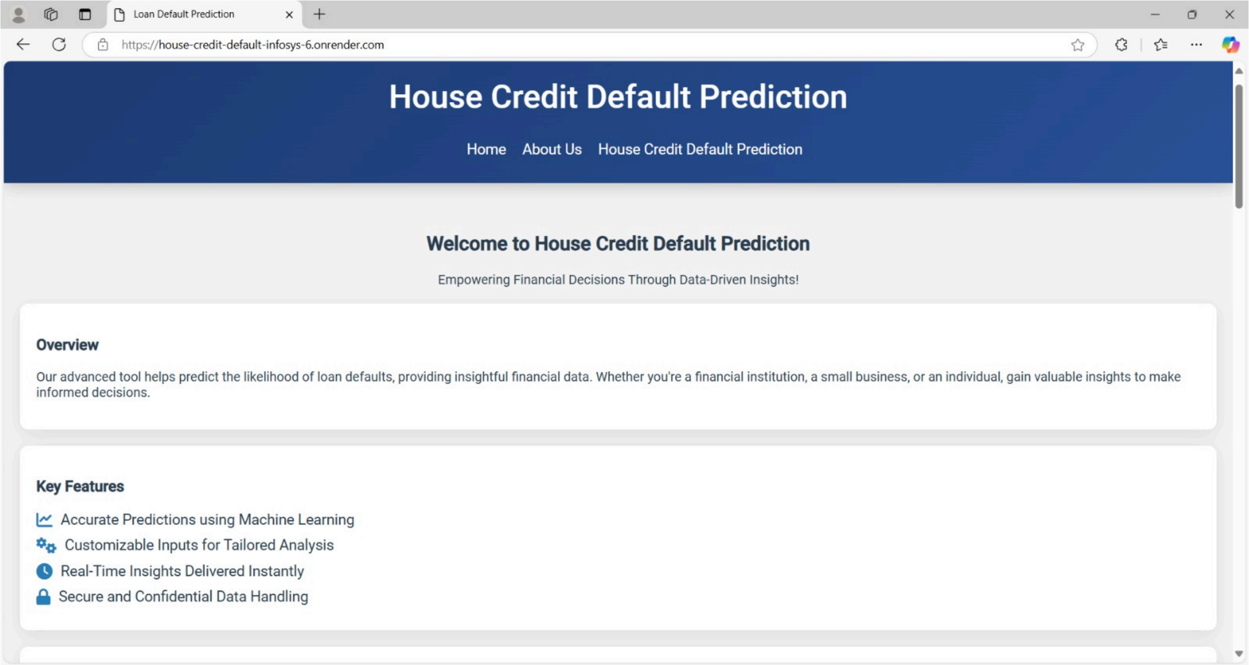
November 20, 2024 at 3:51 PM ✓ Live

ddbd4b7 Update app.py

```
Nov 20 03:53:43 PM  ==> No open ports detected, continuing to scan...
Nov 20 03:53:43 PM  [2024-11-20 10:23:43 +0000] [95] [INFO] Starting gunicorn 23.0.0
Nov 20 03:53:43 PM  [2024-11-20 10:23:43 +0000] [95] [INFO] Listening at: http://0.0.0.0:10000 (95)
Nov 20 03:53:43 PM  [2024-11-20 10:23:43 +0000] [95] [INFO] Using worker: sync
Nov 20 03:53:43 PM  [2024-11-20 10:23:43 +0000] [111] [INFO] Booting worker with pid: 111
Nov 20 03:53:44 PM  ==> Docs on specifying a port: https://render.com/docs/web-services#port-binding
Nov 20 03:53:44 PM  127.0.0.1 - - [20/Nov/2024:10:23:44 +0000] "HEAD / HTTP/1.1" 200 0 "-" "Go-http-client/1.1"
Nov 20 03:53:45 PM  ==> Your service is live 🎉
Nov 20 03:53:46 PM  127.0.0.1 - - [20/Nov/2024:10:23:46 +0000] "GET / HTTP/1.1" 200 17103 "-" "Go-http-client/2.0"
Nov 20 03:54:34 PM  127.0.0.1 - - [20/Nov/2024:10:24:34 +0000] "GET / HTTP/1.1" 200 17103 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWe
bKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
Nov 20 03:54:34 PM  127.0.0.1 - - [20/Nov/2024:10:24:34 +0000] "GET /favicon.ico HTTP/1.1" 404 207 "https://house-credit-default-infosys-6.onren
der.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
Nov 20 03:57:42 PM  127.0.0.1 - - [20/Nov/2024:10:27:42 +0000] "POST /predict HTTP/1.1" 200 64 "https://house-credit-default-infosys-6.onrender.
com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36 Edg/131.0.0.0"
Nov 20 03:58:54 PM  ==> Detected service running on port 10000
Nov 20 03:58:55 PM  ==> Docs on specifying a port: https://render.com/docs/web-services#port-binding
```

The URL for our Project:- Loan Default Prediction

RESULTS AND DISCUSSION



House Credit Default Prediction

Home About Us House Credit Default Prediction

Welcome to House Credit Default Prediction

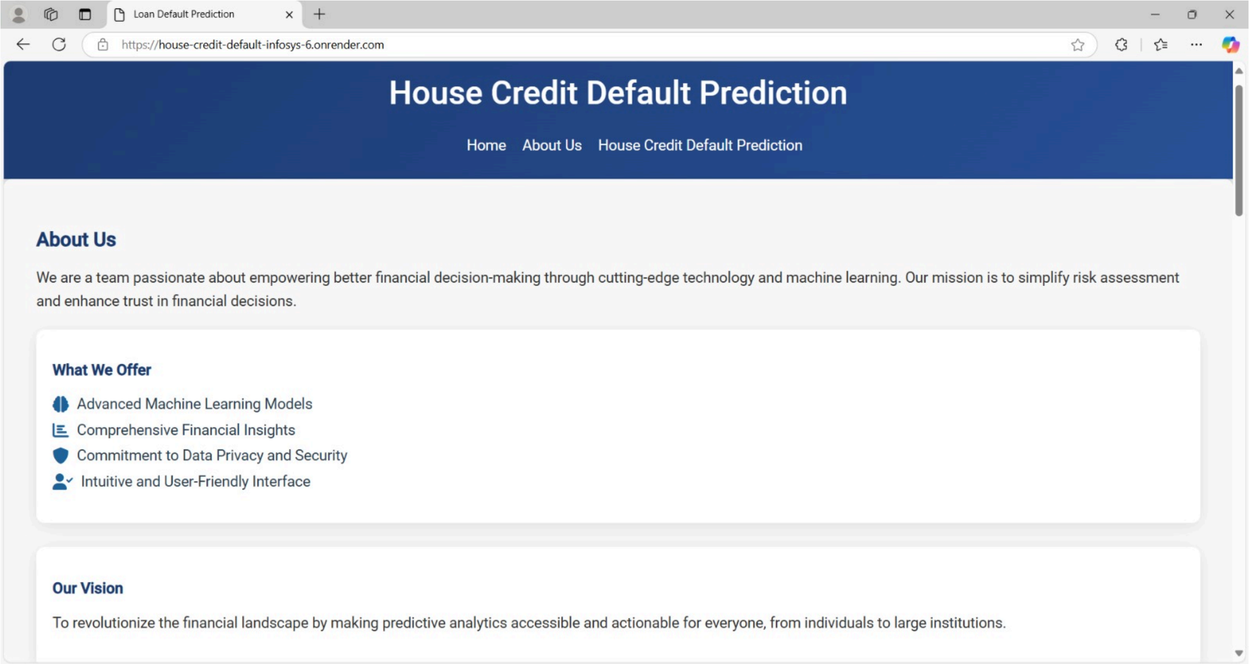
Empowering Financial Decisions Through Data-Driven Insights!

Overview

Our advanced tool helps predict the likelihood of loan defaults, providing insightful financial data. Whether you're a financial institution, a small business, or an individual, gain valuable insights to make informed decisions.

Key Features

- Accurate Predictions using Machine Learning
- Customizable Inputs for Tailored Analysis
- Real-Time Insights Delivered Instantly
- Secure and Confidential Data Handling



House Credit Default Prediction

Home About Us House Credit Default Prediction

About Us

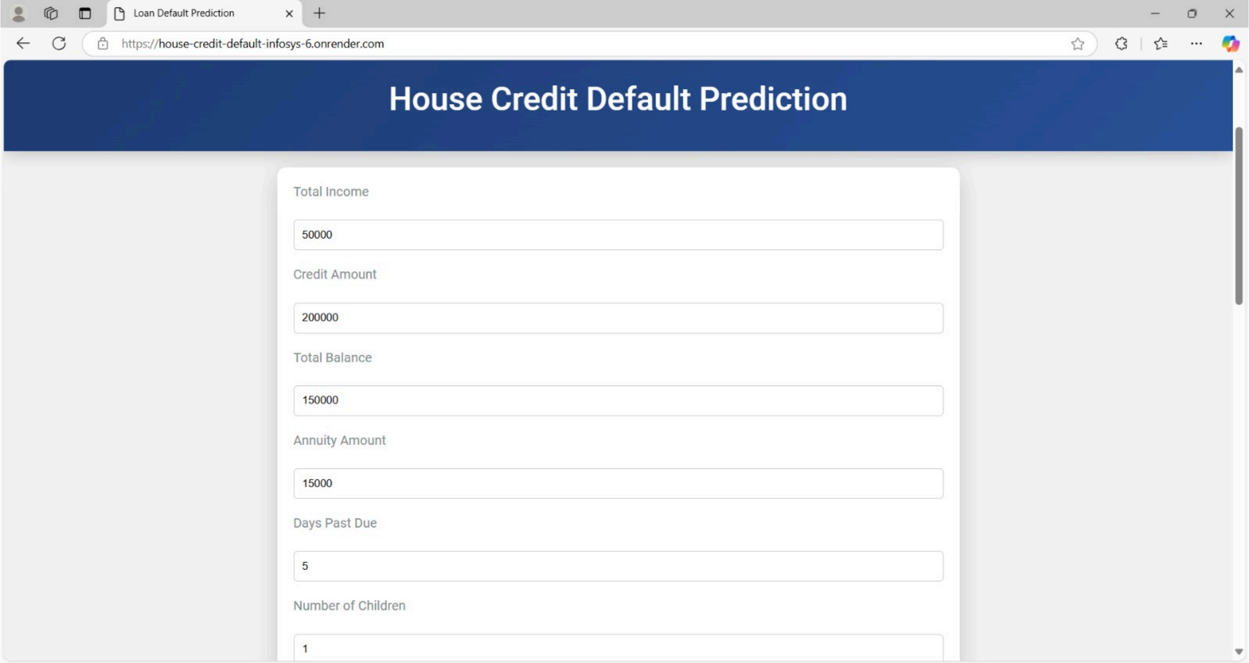
We are a team passionate about empowering better financial decision-making through cutting-edge technology and machine learning. Our mission is to simplify risk assessment and enhance trust in financial decisions.

What We Offer

- Advanced Machine Learning Models
- Comprehensive Financial Insights
- Commitment to Data Privacy and Security
- Intuitive and User-Friendly Interface

Our Vision

To revolutionize the financial landscape by making predictive analytics accessible and actionable for everyone, from individuals to large institutions.



House Credit Default Prediction

Total Income

50000

Credit Amount

200000

Total Balance

150000

Annuity Amount

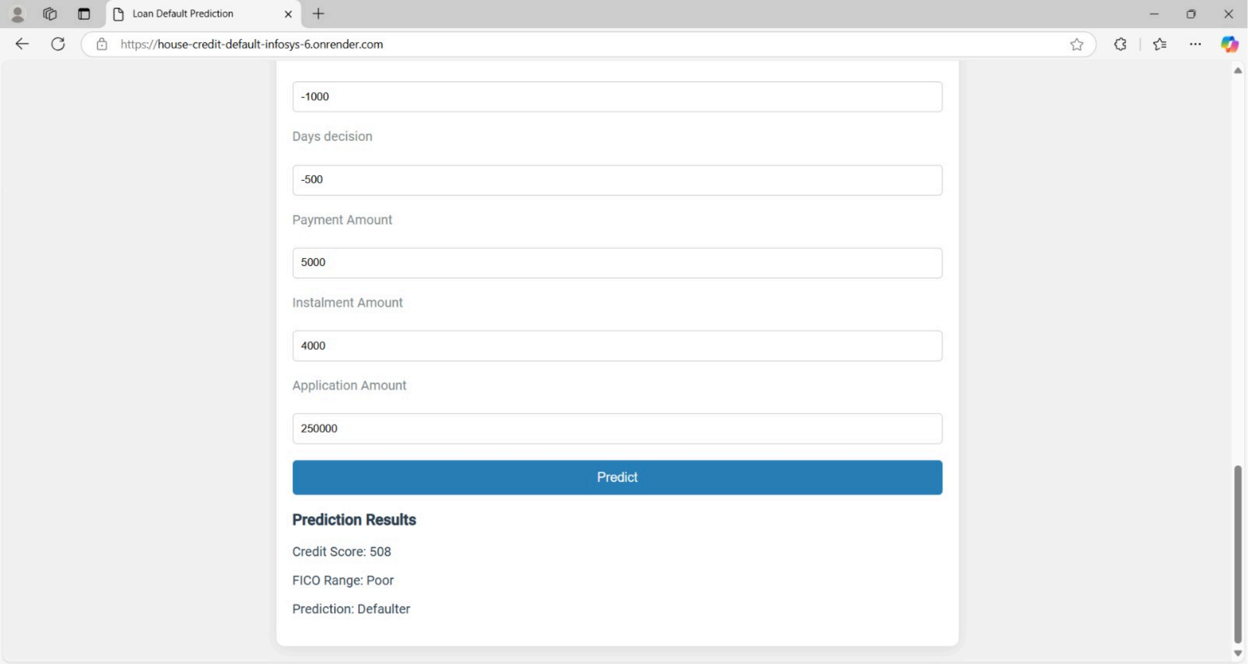
15000

Days Past Due

5

Number of Children

1



-1000

Days decision

-500

Payment Amount

5000

Instalment Amount

4000

Application Amount

250000

Prediction Results

Credit Score: 508

FICO Range: Poor

Prediction: Defaulter

The output of the loan prediction form provides the following key results:

1. Prediction (Defaulter or Non-Defaulter):

Based on the model's prediction, it classifies the applicant as a Defaulter (prediction = 1) or Non-Defaulter (prediction = 0), indicating whether the individual is likely to default on the loan.The prediction is influenced by a combination of factors, including the applicant’s financial history and behavior, which are processed through the machine learning model.

2. Credit Score:

A numeric credit score is calculated using specific financial details from the form, such as payment history, credit balance, and other financial metrics. The score ranges from 0 to 850, with higher scores representing a more creditworthy applicant.

3. FICO Range:

The calculated credit score is mapped to a FICO range (e.g., Exceptional, Good or Poor), which categorizes the applicant's creditworthiness. This helps provide a more understandable context to the credit score and shows the general level of financial reliability.

The combination of these outputs helps the user understand not just whether they are likely to default on the loan but also their creditworthiness based on detailed financial metrics.

CONCLUSION

The Risk Analysis for Home Credit Default project successfully lays the foundation for building a robust predictive model to assess the likelihood of loan defaults. Through thorough exploratory data analysis (EDA), we identified key patterns and characteristics of the dataset, leading to better feature selection and data preprocessing strategies.

By detecting and handling missing data, addressing outliers, and performing feature engineering, we ensured a cleaner, more reliable dataset for further analysis. Visualizations helped us uncover trends and relationships between variables, allowing for more informed decision-making.

As the project progresses into the modeling phase, machine learning algorithms will be applied to predict default probabilities, using the insights gathered during EDA to improve model accuracy. The ultimate goal is to deliver actionable recommendations that help mitigate credit risk, providing financial institutions with a valuable tool for making informed lending decisions.

FUTURE SCOPES

Creating Transparent Reports for Stakeholders: Develop dashboards or summary reports that explain the model's predictions for different customer profiles in a way that non-technical stakeholders can understand.

Scalable Architecture: Consider deploying the model in a distributed environment with scalable resources (e.g., Apache Spark or Dask on a cloud platform).

Real-Time Prediction: Transition to a real-time inference system that can make instant predictions as new customer data arrives, enabling banks to make real-time decisions on loan approvals.

API Integration: Develop RESTful APIs or gRPC services for embedding the model into broader financial systems or applications

GDPR and Privacy Compliance: Implement mechanisms to protect sensitive data, ensuring that your model complies with data protection regulations.

TEAM CONTRIBUTION

The members of **Group 1** have collaborated effectively to ensure the successful completion of the project, with each member contributing their expertise in different areas. **S. Chaitanya Deepthi** focused on conducting **Exploratory Data Analysis (EDA)**, performing data cleaning and uncovering insights to prepare the dataset for further processing. **Sk. Shakila** played a key role in the **training phase**, where she worked on selecting, tuning, and training machine learning models to achieve optimal performance. **Smruti Deshpande** took responsibility for **modeling, connectivity, deployment**, and **documentation**, including designing and implementing the machine learning models, integrating backend systems, deploying the final model in a production environment, and creating detailed project documentation. Meanwhile, **C. Sahi** specialized in **frontend development, connectivity**, and **documentation**, developing the user interface, integrating it with the backend for a seamless experience, and contributing to the documentation process. Together, the team ensured the project's success through their dedicated efforts and collaborative approach.

REFERENCES

[1] B. A. Çallı and E. Coşkun, "A Longitudinal Systematic Review of Credit Risk Assessment and Credit Default Predictors," *SAGE Open*, vol. 11, no. 4, Oct. 2021. [Online]. Available: **https://journals.sagepub.com/doi/full/10.1177/21582440211061333.**

[2] W. Koehrsen, "Start Here: A Gentle Introduction," *Kaggle*. [Online]. Available: **https://www.kaggle.com/code/willkoehrsen/start-here-a-gentle-introduction.**

[3] "Featured Prediction Competition," *Kaggle*, Home Credit Default Risk. [Online]. Available: **https://www.kaggle.com/competitions/home-credit-default-risk/discussion/59347.**

[4]. L. Zhu, D. Qiu, D. Ergu, C. Ying, and K. Liu, "A study on predicting loan default based on the random forest algorithm," *Procedia Computer Science*, vol. 162, pp. 503-513, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050919320277.

[5]. U. Aslam, H. I. T. Aziz, A. Sohail, and N. K. Batcha, "An Empirical Study on Loan Default Prediction Models," *Journal of Computational and Theoretical Nanoscience*, vol. 16, no. 8, pp. 3483–3488, Aug. 2019, doi: 10.1166/jctn.2019.8312. [Online]. Available: https://www.researchgate.net/publication/335966806.

[6]. M. Madaan, A. Kumar, C. Keshri, R. Jain, and P. Nagrath, "Loan default prediction using decision trees and random forest: A comparative study," *IOP Conference Series: Materials Science and Engineering*, vol. 1022, no. 1, p. 012042, 2021. doi: 10.1088/1757-899X/1022/1/012042. [Online]. Available: https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012042/pdf.

[7]. D. Dansana, S. G. K. Patro, B. K. Mishra, V. Prasad, A. Razak, and A. W. Wodajo, "Analyzing the impact of loan features on bank loan prediction using Random Forest algorithm," *Engineering Reports*, vol. 4, no. 7, p. e12707, July 2022. doi: 10.1002/eng2.12707. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/10.1002/eng2.12707.

[8].W. Wu, "Machine Learning Approaches to Predict Loan Default," *Intelligent Information Management*, vol. 14, no. 5, pp. 157–164, Jan. 2022, doi: 10.4236/iim.2022.145011. [Online]. Available: **https://www.researchgate.net/publication/363868916.**