Credit EDA

Pavan Manikanta Bellam, Rohith Chinnamallappagari and Manisha Chandramaully

March 27, 2024

CREDIT EDA

BELLAM PAVAN MANIKANTA
Department of Computer Science Engineering
Parul Institute of Engineering and Technology
Vadodara, India
200303124143@paruluniversity.ac.in

CHINNAMALLAPPAGARI ROHITH
Department of Computer Science Engineering
Parul Institute of Engineering and Technology
Vadodara, India
200303124175@paruluniversity.ac.in

MANISHA CHANDRAMAULLY
Assistant Professor,
Department of Computer Science & Engineering,
Parul Institute of Engineering and Technology
manisha.chandramaully29321@paruluniversity.ac.in

***Abstract*— This research paper delves into the realm of credit analysis through an in-depth exploration of two distinct datasets related to client loan applications. The first dataset encompasses a comprehensive array of client information recorded at the time of loan application, while the second dataset provides insights into the client's historical interactions with the loan application process. Our methodology comprised separate analyses of each dataset, followed by a meticulous integration process aimed at facilitating a holistic examination of credit-related trends and patterns. By employing a diverse set of exploratory data analysis techniques, including descriptive statistics, data visualization, and correlation analysis, we unearthed intricate relationships within each dataset. This approach enabled us to gain a nuanced understanding of client creditworthiness and the factors influencing loan approval outcomes. Moreover, the amalgamation of findings from both datasets enriched our insights, revealing critical connections between application attributes and historical application outcomes. This paper contributes to the evolving landscape of credit analysis by emphasizing the importance of leveraging diverse datasets for a comprehensive understanding of client credit profiles and enhancing decision-making processes in the financial domain.**

Keywords—Exploratory Data Analysis (EDA), credit-based datasets, Credit analysis, loan approval, payment difficulties, client history, data integration, data visualization, creditworthiness, financial decision-making.

I. INTRODUCTION

Credit assessment lies at the heart of financial decision-making, influencing lending practices and risk management strategies. In this project, we delve into the realm of Exploratory Data Analysis (EDA) applied to credit-based datasets. These datasets encapsulate crucial information about clients' financial profiles and historical loan data, offering insights into payment difficulties and loan approval statuses. By combining and analyzing these datasets, we aim to unravel patterns, trends, and relationships that inform creditworthiness and risk assessment. Through meticulous preprocessing and analysis, our project endeavours to contribute to the understanding of credit analysis methodologies, empowering stakeholders with actionable insights for informed financial decision-making.

A. Problem Statement

This project aims to conduct Exploratory Data Analysis (EDA) on loan applicant data to identify patterns and mitigate financial risks for loan providing companies. By analysing applicant profiles, the goal is to differentiate between individuals capable of repayment and potential defaulters, ensuring sound loan approval decisions and minimizing business losses.

B. Scope

The scope of this research project encompasses a comprehensive exploration of credit analysis using two primary datasets: 'application_data.csv' and 'previous_application.csv'. These datasets serve as the cornerstone for understanding client loan applications and their historical interactions with the loan application process. The project primarily employs exploratory data analysis (EDA) techniques, including descriptive statistics, data visualization, and correlation analysis, to unravel intricate patterns, trends, and relationships embedded within the datasets. A pivotal aspect of the project involves integrating the two datasets to facilitate a holistic examination of credit-related phenomena. Through this integration, the project aims to enrich insights by elucidating connections between application attributes and historical application outcomes. The overarching goal is to generate actionable insights into client creditworthiness and the factors influencing loan approval outcomes. While the project strives to offer valuable insights, it acknowledges inherent limitations such as data availability, time constraints, and computational resources. Despite these constraints, the project endeavours to provide a structured analysis of credit-based datasets, contributing to the ongoing discourse surrounding credit assessment practices and informing future research endeavours in the domain.

C. Aim and Objective

The aim of this project is to comprehensively analyze credit-related data to gain insights into client behaviors and loan outcomes. Through meticulous examination of payment behaviors and loan histories, the project seeks to identify patterns and trends within the data. By exploring relationships between client characteristics and loan outcomes, the research aims to contribute to the enhancement of credit assessment methodologies and decision-making processes in financial contexts. Through these objectives, the project endeavors to provide valuable insights that can inform and improve credit risk assessment practices, ultimately facilitating more informed financial decision-making.

II. LITERATURE SURVEY

Matthieu Komorowski [1] has described about the most common tools available for exploring a dataset, which is essential in order to gain a good understanding of the features and potential issues of a dataset, as well as helping in hypothesis generation.

Jitendra Pramanik [2] emphasizes the importance of data analysis in making informed decisions, citing examples like recommendation systems and product purchase predictions. Exploratory Data Analysis (EDA) using Python, with libraries like pandas and matplotlib, was employed to interpret Amazon electronic item review datasets. Python's object-oriented, interpreted, and interactive nature facilitated comprehensive analysis of the data.

V.P. Sumathi [3] explores the surge in loan applications in India and the challenges banks face in predicting repayment capabilities. Utilizing exploratory data analysis, the paper reveals a preference for short-term loans, often sought for debt consolidation. Graphical representations assist bankers in comprehending client behavior for informed decision-making.

Sudhamathy G. [4] focuses on mitigating bank loan risks through data mining techniques. The paper proposes a decision tree-based model to assess loan default probabilities, utilizing pre-processed datasets for efficient predictions. Experimental results validate the model's efficacy in risk management strategies.

Rory M. Leith [5] utilizes exploratory data analysis to detect trends and statistical characteristics in nine streamflow time series, presenting results graphically and through relevant statistical tests. The approach not only identifies trends but also contextualizes responses against observed values, revealing periods of unusual flow conditions and non-normal behaviours in flow sequences.

Patricia Jimbo Santana [6] conducts a comparative analysis between optimization-based methods initialized with neural networks and partition algorithms based on trees for extracting credit risk rules. Results from real databases reveal that the former yields rules with reduced cardinality and acceptable classification precision, making it desirable for financial institutions making face-to-face credit approval decisions. This approach enables easier training of bank employees in selecting optimal customers, facilitating retail customer interactions.

III. METHODOLOGY

Exploratory Data Analysis

Exploratory Data Analysis (EDA) involves initial steps in data analysis to understand the main characteristics of a dataset. It includes processes such as Dataset and Data Exploration, Data Cleaning, and various visualization techniques like histograms, scatterplots, and boxplots to uncover patterns, trends, and relationships within the data. The following are the steps I have followed to perform this project.

A. Dataset and Data Exploration

Dataset and Data Exploration involves examining the structure and content of a dataset before analysis. It includes tasks such as checking the dimensions of the dataset, identifying variable types, and understanding distributions of data through summary statistics and visualizations.

B. Data cleaning

Data Cleaning is the process of identifying and correcting errors, inconsistencies, or missing values in a dataset to improve its quality and reliability for analysis. It includes tasks such as removing duplicates, imputing missing values, and identifying and handling outliers.

C. Binning values

Binning Values is a data preprocessing technique used to group continuous or categorical data into discrete intervals or bins. This simplifies the data and can help identify patterns or trends that may not be apparent when analyzing individual values. Binning values can be useful for reducing the complexity of datasets, creating categorical variables from continuous data, or preparing data for analysis techniques that require discrete inputs.

D. Univariate Analysis

Univariate Analysis focuses on analyzing and summarizing the characteristics of a single variable in a dataset. It involves examining the distribution of values, calculating descriptive statistics, and visualizing data with plots like histograms, distribution plots, boxplots to understand its behavior in isolation.

- Histogram:

  A histogram is a graphical representation of the distribution of numerical data, divided into bins or intervals. It displays the frequency or count of observations falling within each bin, allowing for the visualization of data distribution and identifying patterns such as skewness or peaks.
- Distplot:

  A distplot, or distribution plot, is a graphical representation of the distribution of a univariate dataset. It combines a histogram with a kernel density estimate (KDE) plot, showing the frequency distribution of data along with an estimated probability density function. Distplots help visualize the central tendency, spread, and shape of the distribution of data.
- Boxplot:

  A boxplot, or box-and-whisker plot, is a graphical summary of the distribution of numerical data through quartiles. It displays the median, quartiles, and range of the dataset, as well as identifying outliers. Boxplots are useful for comparing the distributions of different variables or groups within a dataset.

E. *Bivariate Analysis*

Bivariate Analysis involves analyzing the relationship between two variables in a dataset. It includes examining correlations, associations, or dependencies between variables using statistical measures and visualizations like scatterplots, pair plots to understand their interactions.

- Scatterplot:

  A scatterplot is a graphical representation of data points plotted on a Cartesian plane, with one variable on the x-axis and another on the y-axis. It shows the relationship between two variables, allowing for visual examination of patterns, trends, and correlations. Scatterplots are useful for identifying relationships and outliers in datasets.

- Pair plot:

  A pair plot is a grid of scatterplots and histograms that allows for the visualization of pairwise relationships between multiple variables in a dataset. It displays scatterplots for numerical variables and histograms for distributions along the diagonal. Pair plots are useful for identifying patterns and correlations between variables.

F. *Multivariate Analysis*

Multivariate Analysis involves exploring relationships among three or more variables simultaneously. It extends beyond bivariate analysis to uncover complex patterns and interactions within the data. Techniques such as heatmap visualization, multiple regression, principal component analysis (PCA), and cluster analysis are commonly used in multivariate analysis.

- Heatmap:

  A heatmap is a graphical representation of data where values in a matrix are represented as colors. It is often used to visualize correlations or relationships between variables in a dataset, with higher values indicated by warmer colors (e.g., red) and lower values by cooler colors (e.g., blue). Heatmaps help identify patterns and trends in complex datasets.

IV. EDA IN PYTHON

Python's ease of use, large library, and strong data handling make it a popular choice for exploratory data analysis (EDA). Python is an open-source language that is available on several platforms and provides adaptability and compatibility with third-party tools. Its comprehensibility and readability enable developers to comprehend and alter code effectively. Python's abundance of libraries makes for smooth visualization, which helps to produce reports that are both understandable and informative. With Python's robust capabilities and easy-to-use interface, analysts can quickly and easily gain insight into data using EDA tasks.

Pandas is a leading package for data analysis, renowned for its versatility and robust capabilities. With Pandas, users can efficiently clean, transform, and analyze datasets, facilitating seamless data manipulation tasks. It supports various data formats, including CSV, enabling easy storage and retrieval of data on computers. Moreover, Pandas offers functionalities for data cleaning, visualization, and storage, streamlining the entire data analysis process. Leveraging the underlying power of the NumPy package, Pandas enhances data processing efficiency and performance. Additionally, Pandas seamlessly integrates with plotting functions from Matplotlib and Seaborn, enabling users to create insightful visualizations to further analyze and interpret data trends effectively.

Jupyter Notebook is a powerful tool for executing code in a cell-by-cell manner, offering a convenient console-based computing approach. Its web-based application process allows for seamless interaction with code, facilitating input and output of computations in an intuitive manner. Additionally, Jupyter Notebook provides rich media representations of objects, enhancing the visual presentation of data analysis results.

V. WORKING ON THE DATASETS

It's time to investigate and learn more about the data. The information we are utilizing is from the credit-based datasets where the primary datasets are 'application_data.csv' which contains all the information of the client at the time of application and it is about whether a client has payment difficulties.
and 'previous_application.csv' contains information about the client’s previous loan data. It contains the data whether the previous application had been Approved, Cancelled, Refused or Unused offer. We will examine the data and consider our alternatives.

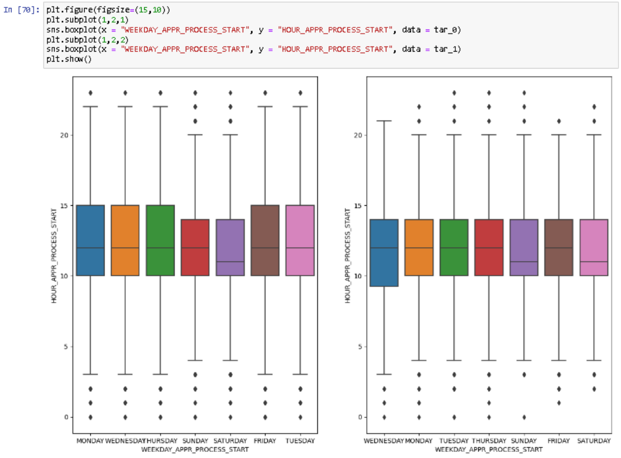
1. Now here we import all the necessary libraries like pandas, NumPy, matplotlib and seaborn.
2. Later we have imported one the dataset which is a csv file named as application_data.csv which contains 307511 rows and 122 columns as data frame df.
3. Later we have imported one the dataset which is a csv file named as application_data.csv which contains 307511 rows and 122 columns as data frame df. We have used head( ) method to get top 5 rows of the data frame.
4. Next, we have explored the dataset with info() method provides information about the data frame, including the number of rows, column names, data types, and memory usage and describe() method computes summary statistics for numerical columns such as count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum.
5. Next, we started to check and handle null values. primarily, we drop columns with more than 47% of null values and then named the new data frame as app_df which contains columns with null percentage less than 47%. we achieved this using dropna() method.
6. Next, we find the null percentage of each column in app_df using isnull().mean()*100 method to handle the remaining columns with null values.
7. Next filled the missing values with mean, median, mode or with new values based on the type of the column using a method fillna().
8. Next, we aimed to understand the unique values in a column of the dataset, along with their corresponding percentages in the distribution of column values. We

used the `value_counts(normalize=True) * 100` method for this purpose.

9. Next, we used the boxplot to visualize the distribution of values in a column.
10. Following that, we introduced several new columns derived from binning certain columns within the 'app_df' data frame. Initially, we utilized a lambda function to ensure that the data remained non-negative and absolute. Subsequently, employing the 'cut()' method, we categorized the column data into distinct intervals, with these categories serving as the values for the newly created columns.
11. Following the creation of the newly formed columns, we utilized bar graphs and pie charts to visually represent each column's distribution. This approach allowed us to gain insights into the proportion of data within each category.
12. Next, we executed this "app_df.TARGET.value_counts(normalize=True)*100" to calculate the percentage distribution of values in the "TARGET" column of the Data Frame "app_df". It provides the normalized counts of each unique value in the "TARGET" column, expressed as percentages.
13. Next, In the dataset there is a column named as TARGET which has only two values 0 and 1. where, 1 indicates client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 indicates all other cases. Now we are dividing the app_df data frames into two data frames tar_0 with all the data where TARGET column with value 0 and tar_1 with all the data where TARGET column with value 1.
14. Next we performed univariate analysis by initially dividing the whole columns into categorical columns and numerical columns and used `value_counts(normalize=True) method to understand the unique values in a column of the categorical values, along with their corresponding percentages in the distribution of column values and plotted the categorical columns with pie chart. Now, with numerical columns we used describe() method which computes summary statistics for numerical columns such as count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum and plotted them using boxplot to understand the distribution of values in that column. And also plotted distplot to get visual comparison of the distribution of a column between two groups, 'tar_0' and 'tar_1', representing clients with and without payment difficulties, respectively.
15. Next, we conducted bivariate analysis on two sets of variables: 'WEEKDAY_APPR_PROCESS_START' and 'HOUR_APPR_PROCESS_START' for the 'tar_0' and 'tar_1' groups, and 'AGE_CATEGORY' and 'AMT_CREDIT' for the same groups. The first set of boxplots compares the timing of loan applications across weekdays for both groups, while the second set compares the amount of credit granted across different age categories.
16. Next, we have plotted pairplot on AMT_INCOME_TOTAL, AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE to understand the relationship among these variables.
17. Next we performed multivariate analysis on AMT_INCOME_TOTAL, AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE, YEARS_BIRTH, YEARS_EMPLOYED, YEARS_REGISTRATION, YEARS_ID_PUBLISH, YEARS_LAST_PHONE_CHANGE by generating correlation matrix using corr() method and plotting heatmap of the correlation matrix.
18. Next we performed all the above steps on "previous_data.csv" to analysis this dataset in the same way by importing data file as papp_df data frame.
19. Next merged both the data frames based on the common column SK_ID_CURR using the method merge() method. And used head(), info(), shape(0 method to know about the merged data frame.
20. Here, we have made the pivot table on required columns using pivot_table() method and plot the pivot table using heat map.

VI. RESULT

Here are some of the results and conclusions that we had made as the result.

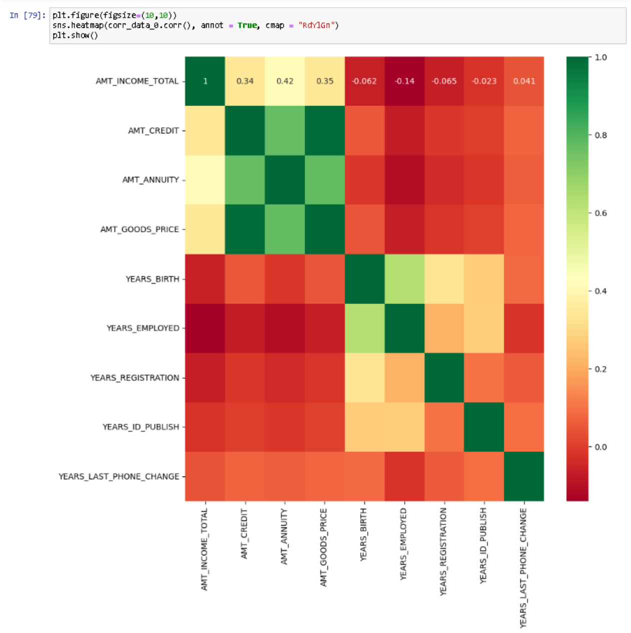


```
In [70]: plt.figure(figsize=(15,10))
         plt.subplot(1,2,1)
         sns.boxplot(x = "WEEKDAY_APPR_PROCESS_START", y = "HOUR_APPR_PROCESS_START", data = tar_0)
         plt.subplot(1,2,2)
         sns.boxplot(x = "WEEKDAY_APPR_PROCESS_START", y = "HOUR_APPR_PROCESS_START", data = tar_1)
         plt.show()
```

Based on the above graph the conclusions we have made are:

- The Bank operates between 10am to 3pm except for Saturday and Sunday Its between 10am to 2pm.
- We can observe that around 11:30am to 12pm around 50% of Customers visit the branch for loan application on all the days except for Saturday where the time is between 10am to llam for both Target 0 and 1
- The loan defaulters have applied for the loan between 9:30am-10am and 2pm where as the applicants who repay the loan on time have applied for the loan between loam to 3pm

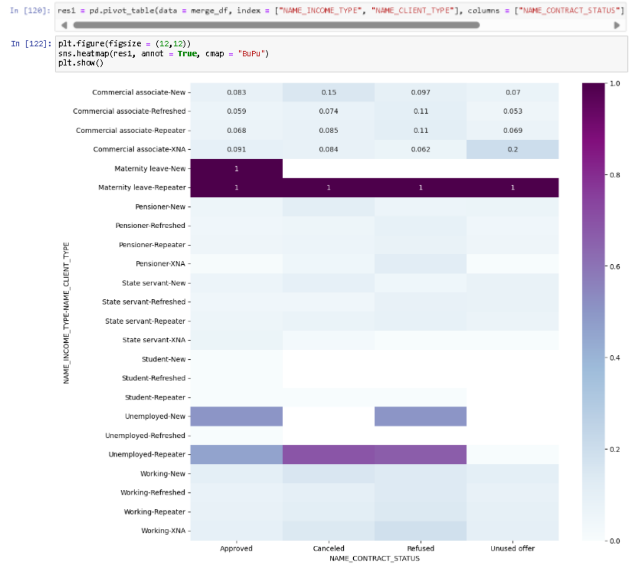
```
In [77]: corr_data_0 = tar_0[["AMT_INCOME_TOTAL", "AMT_CREDIT", "AMT_ANNUITY", "AMT_GOODS_PRICE", "YEARS_BIRTH", "YEARS_EMPLOYED", "Y
         corr_data_0.head()
```



```
In [79]: plt.figure(figsize=(10,10))
         sns.heatmap(corr_data_0.corr(), annot = True, cmap = "RdYlGn")
         plt.show()
```

Based on the above graph the conclusions we have made are:

1. AMT_INCOME_TOTAL - It is less correlated with AMT_CREDIT, AMT_ANNUITY, AMT_GOODS_PRICE respectively
2. AMT_CREDIT - It has a strong positive correlation index of 0.98, 0.75 with AMT_GOODS_PRICE, AMT_ANNUITY respectively and also positive correlation With other Year Columns
3. AMT_ANNUITY - It has positive correlation Index ot 0.75 with AMT_CREDIT and AMT_GOODS_PRICE and Negative With YEAR_EMPLOYED and YEAR_REGISTRATION
4. AMT_GOODS_PRICE - It has a strong positive correlation index 0.98 with AMT_ANNUITY and AMT_CREDIT and weak positive correlation with other Year columns.



```
In [120]: res1 = pd.pivot_table(data = merge_df, index = ["NAME_INCOME_TYPE", "NAME_CLIENT_TYPE"], columns = ["NAME_CONTRACT_STATUS"]
```

```
In [122]: plt.figure(figsize = (12,12))
          sns.heatmap(res1, annot = True, cmap = "BuPu")
          plt.show()
```

Based on the above graph the conclusions we have made are:

1. Applicants with income type maternity leave and client type new are having more chances of getting the loan approved
2. Applicants with income type maternity leave, unemployed and client type Repeater are having getting the loan cancelled
3. Applicants with income type maternity leave, Unemployed and client type Repeater are having getting the loan refused
4. Applicants with income type maternity leave and client type Repeater, Working and client type New are not able to utilize the Bank's offer.

VII. CONCLUSION

In this study, we utilized Exploratory Data Analysis (EDA) methods to decipher critical insights within credit datasets. Conducted within the Jupyter Notebook environment using Python, alongside essential libraries such as NumPy, Pandas, Matplotlib, and Seaborn, our analysis delved into the intricacies of client financial data. Looking ahead, we plan to expand our investigation by incorporating additional datasets and leveraging advanced analytical techniques to gain deeper insights into exploratory data analysis methodologies.

REFERENCES

[1] Matthieu Komorowski, Dominic C. Marshall, Justin D. Salciccioli and Yves Crutain, 2016, Exploratory Data Analysis

[2] Kabita Sahoo, Abhaya Kumar Samal, Jitendra Pramanik, Subhendu Kumar Pani, 2019, Exploratory Data Analysis using Python

[3] X.Francis Jency, V.P.Sumathi, Janani Shiva Sri, 2018, An Exploratory Data Analysis for Loan Prediction Based on Nature of the Clients

[4] Sudhamathy G., 2016, Credit Risk Analysis and Prediction Modelling of Bank Loans Using R

[5] Rory M. Leith, Keith W. Hipel & Herman Goertz, 2013, EXPLORATORY DATA ANALYSIS

[6] Patricia Jimbo Santana, Augusto Villa Monte, Enzo Rucci Laura Lanzarini, Aurelio F. Bariviera, 2016, An exploratory analysis of methods for extracting credit risk rules

[7] Exploratory data analysis – From Wikipedia, the free encyclopedia [Online], Available: https://en.wikipedia.org/wiki/Exploratory_data_analysis

[8] Exploratory Data Analysis in Python – From Geeks for Geeks [online], Available: https://www.geeksforgeeks.org/exploratory-data-analysis-in-python/