Machine Learning Project – Predicting Credit Score Performance

CONTENTS

TEAM MEMBERS

1. Praneeth Devanabanda
2. Arti Khanchandani
3. Prajwal H V
4. Latikesh Bari
5. Sudeepan Mandal

INTRODUCTION TO PROBLEM STATEMENT

In the realm of global finance, the ability to accurately assess creditworthiness is paramount. A credit score, a numerical representation of an individual's creditworthiness, plays a pivotal role in financial decision-making. Leveraging the wealth of data encompassing basic bank details and professional information, this project endeavors to construct an intelligent system capable of categorizing individuals into credit score brackets. With a dataset comprising both categorical and numerical variables, alongside missing values and anomalies, the journey unfolds to uncover patterns inherent in credit profiles.

Tasked as data scientists within a prominent finance corporation, our mandate is clear: streamline the credit assessment process through predictive analytics. By delving into historical banking and professional records, we aim to discern the attributes that correlate with varying credit scores. Our ultimate objective is to develop a robust machine learning model adept at classifying individuals into distinct credit score categories.

Divided into Train and Test datasets, our methodology hinges on utilizing the former to construct and validate our predictive model. Subsequently, the predictive labels for the Test dataset will be generated and amalgamated with respective IDs. This synthesized output will be encapsulated within a structured dataframe and submitted alongside the solution file in a compressed zip folder.

Through this initiative, we aspire not only to mitigate manual efforts but also to enhance the efficiency and accuracy of credit assessment processes, thereby facilitating informed decision-making within the realm of global finance.

DATA DESCRIPTION

The dataset consists of all the basic bank details of the customers, and the dataset has categorical and numerical variables.

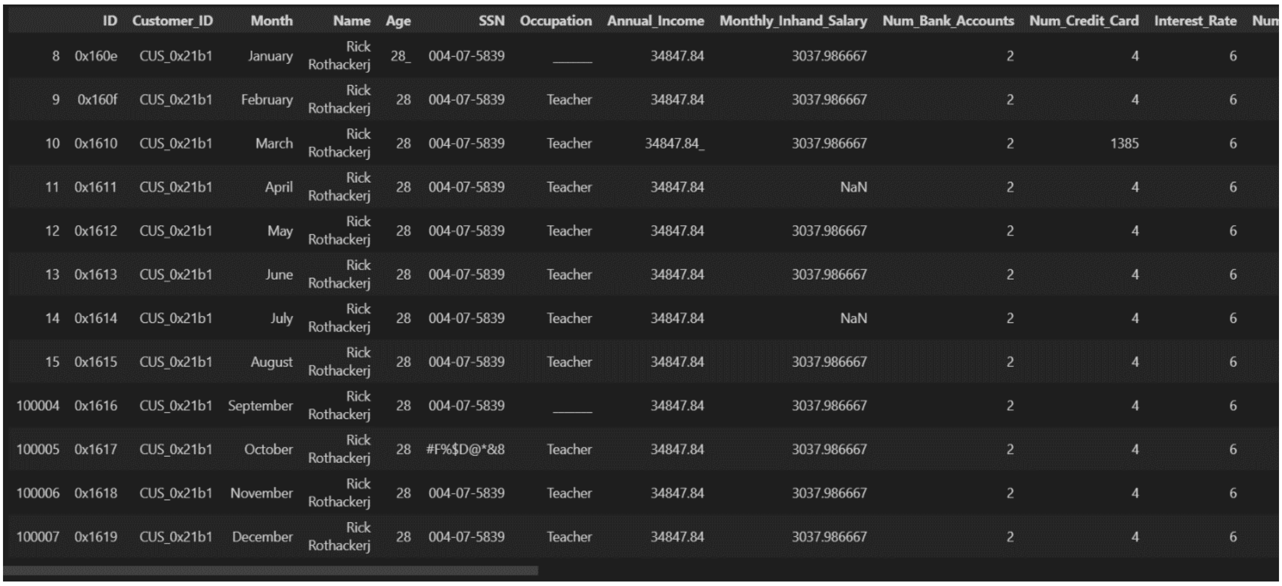
	Column Name	Description
1.	ID	Represents a unique identification of an entry.
2.	CUSTOMER ID	This represents the unique identification of a person.
3.	MONTH	Represents the month of the year.
4.	NAME	Represents the name of a person.
5.	AGE	Represents the age of the person.
6.	SSN	Represents the social security number of the person.
7.	OCCUPATION	Represents the occupation of the person.
8.	ANNUAL INCOME	Represents the yearly income of the person.
9.	MONTHLY IN-HAND SALARY	Represents the monthly base salary of a person.
10.	NUM BANK ACCOUNTS	This represents the number of bank accounts a person holds.
11.	NUM CREDIT CARD	This represents the number of other credit cards held by the person.
12.	INTEREST Rate	This represents the interest rate on a credit card.
13.	NUM OF LOAN	Represents the number of loans taken from the bank.
14.	TYPE OF LOAN	Represents the type of loan taken by the person.
15.	DELAY FROM DUE DATE	Represents the average number of days delayed from the payment date.
16.	NUM OF DELAYED PAYMENT	Represents the average number of payments delayed by a person.
17.	CHANGED CREDIT LIMIT	This represents the percentage change in the credit card limit.
18.	NUM CREDIT INQUIRIES	Represents the number of credit card inquiries.
19.	CREDIT MIX	This represents the classification of the mix of credits.
20.	OUTSTANDING DEBT	This represents the remaining debt to be paid(in USD).
21.	CREDIT UTILIZATION RATIO	This represents the utilization ratio of credit cards.
22.	CREDIT HISTORY AGE	This represents the age of the credit history of the person.
23.	PAYMENT OF MIN AMOUNT	Represents whether only the minimum amount was paid by the person.
24.	TOTAL EMI PER MONTH	Represents the monthly EMI payments(in USD).
25.	AMOUNT INVESTED MONTHLY	Represents the monthly amount invested by the customer(in USD)
26.	PAYMENT BEHAVIOUR	Represents the payment behavior of the customer
27.	MONTHLY BALANCE	Represents the monthly amount of the customer (in USD).

PROBLEM SOLVING STEPS

PRE-PROCESSING

→ After examining the provided train and test datasets, we've found that each customer has 12 records, with each record representing their monthly activity with the bank. The training set comprises 8 months of a customer's data, while the test set contains the remaining 4 months. We combined the train and test sets and performed preprocessing and cleaning on the entire dataset, organized by customer.

→ Upon concatenation, the 12 records for one customer are as follows:



	ID	Customer_ID	Month	Name	Age	SSN	Occupation	Annual_Income	Monthly_Inhand_Salary	Num_Bank_Accounts	Num_Credit_Card	Interest_Rate	Num
8	0x160e	CUS_0x21b1	January	Rick Rothackerj	28_	004-07-5839	_______	34847.84	3037.986667	2	4	6	
9	0x160f	CUS_0x21b1	February	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	3037.986667	2	4	6	
10	0x1610	CUS_0x21b1	March	Rick Rothackerj	28	004-07-5839	Teacher	34847.84_	3037.986667	2	1385	6	
11	0x1611	CUS_0x21b1	April	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	NaN	2	4	6	
12	0x1612	CUS_0x21b1	May	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	3037.986667	2	4	6	
13	0x1613	CUS_0x21b1	June	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	3037.986667	2	4	6	
14	0x1614	CUS_0x21b1	July	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	NaN	2	4	6	
15	0x1615	CUS_0x21b1	August	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	3037.986667	2	4	6	
100004	0x1616	CUS_0x21b1	September	Rick Rothackerj	28	004-07-5839	_______	34847.84	3037.986667	2	4	6	
100005	0x1617	CUS_0x21b1	October	Rick Rothackerj	28	#F%$D@*&8	Teacher	34847.84	3037.986667	2	4	6	
100006	0x1618	CUS_0x21b1	November	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	3037.986667	2	4	6	
100007	0x1619	CUS_0x21b1	December	Rick Rothackerj	28	004-07-5839	Teacher	34847.84	3037.986667	2	4	6	

→ Observing the data, we noticed null values in certain records and non-numeric characters in numerical features. Subsequently, we initiated preprocessing on a per-customer basis, which involved selecting and processing the 12 records of each customer individually. Following this, we concatenated the processed data row-wise into a new dataframe and sorted the indices to separate the train and test sets.

→ For reference, here's the code snippet marking the beginning and end of the loop for customer-wise cleaning:

```python
X1 = pd.DataFrame(columns=X.columns)

for i in tqdm(X['Customer_ID'].unique()):
    df = X[X['Customer_ID'] == i]

    # Converting 'Month' names to numbers
    df['Month_Num'] = df['Month'].map(month_dict)
    df['Month_Num'] = df['Month_Num'].astype('category')

    # All the necessary pre-processing

    X1 = pd.concat([X1, df], axis=0)

X = X1.sort_index()
```

→ Our approach involved addressing various issues, including the replacement of non-numeric characters in numerical features, the imputation of missing values using median or mode, adjustments to data types following necessary imputations or changes, and rectification of inconsistencies, among others.

→ Preprocessing constituted a significant portion of our efforts, accounting for approximately 70% of the team's time, owing to the extensive scrutiny required for numerous columns. Upon completing customer-wise iterations, we made the decision to drop certain features such as 'Name' and 'SSN', which were deemed irrelevant for credit score prediction.

→ Additionally, columns like 'Month', 'Occupation', 'Payment_Behaviour', and other binary features were designated as 'category' data types. Subsequently, these categorical columns were encoded into numerical values using Label Encoder.

→ The target variable classes were encoded as follows: 'poor' as 0, 'Standard' as 1, and 'Good' as 2.

→ Following the completion of all cleaning procedures, we partitioned the concatenated train and test datasets, resulting in the creation of two new CSV files. Specifically, 'Cleaned Train.csv' contained X_train and y_train, while 'Cleaned Test.csv' contained X-test.

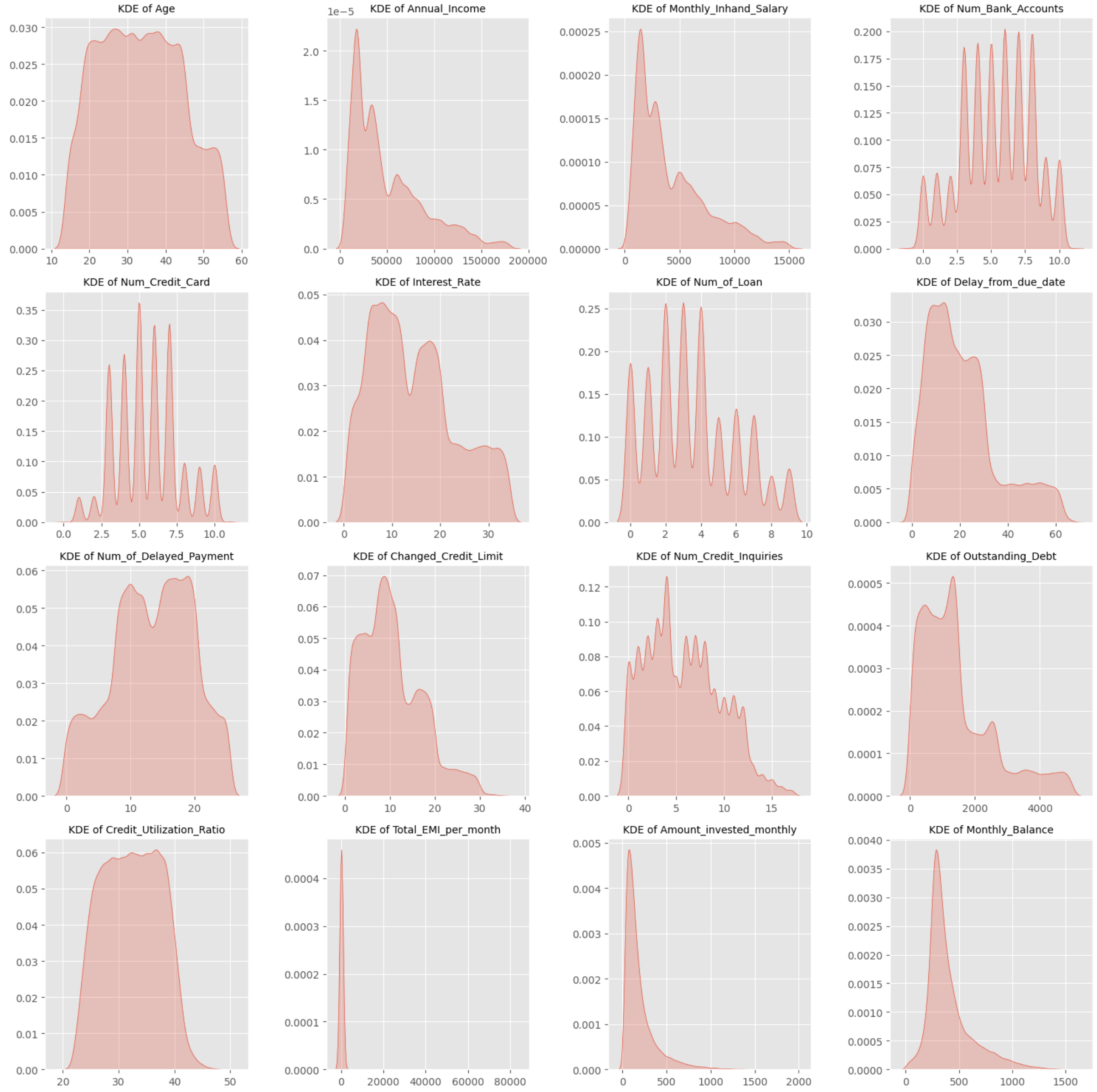
ANALYSIS & STATISTICS

EDA

→ Upon importing 'Cleaned Train.csv', we initiated Exploratory Data Analysis (EDA). Initially, we segregated the target class, 'Credit_Score', from the remaining columns, assigning them the labels 'y' and 'X', respectively.

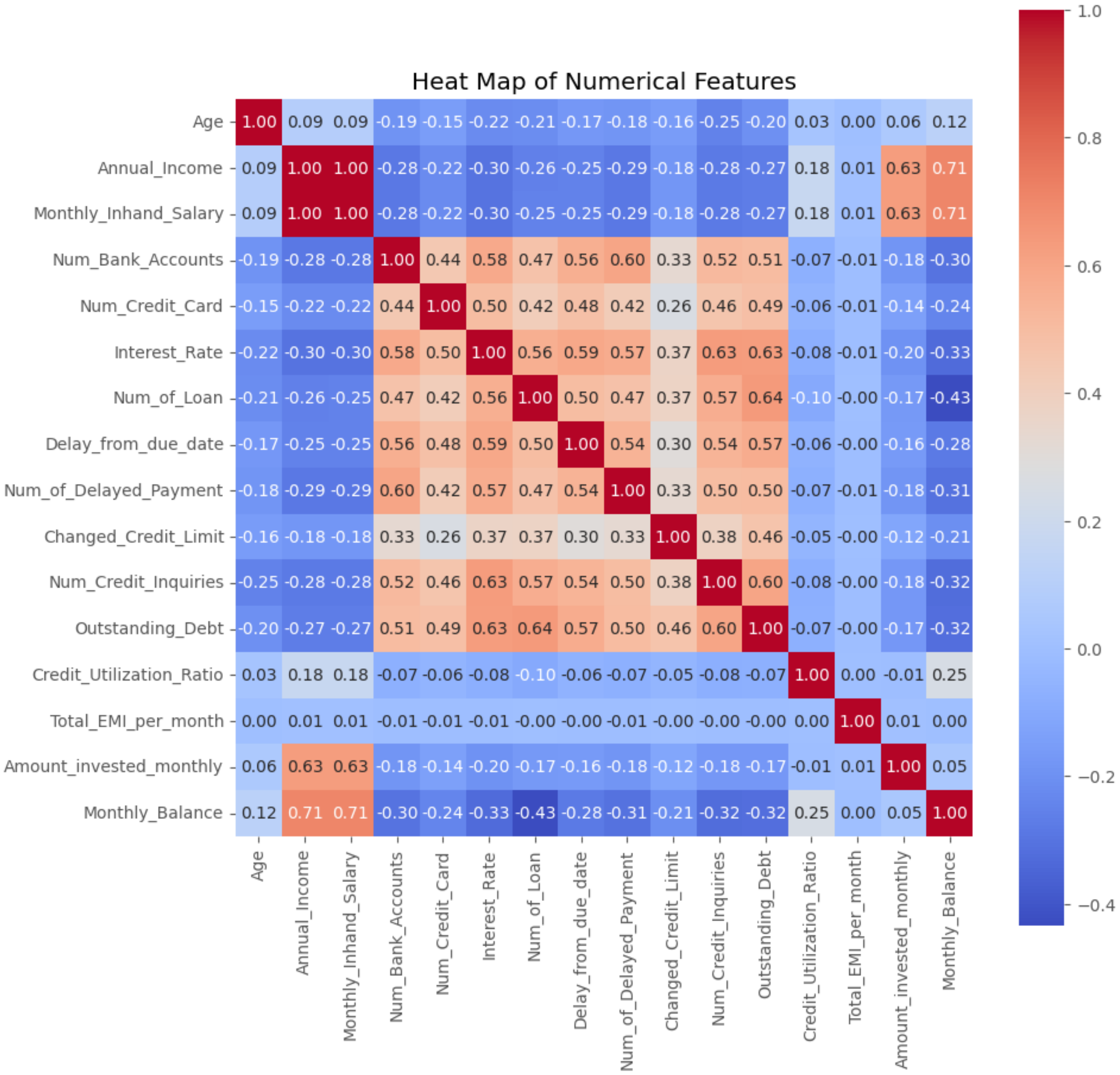
→ Our first focus during EDA was on examining the distribution and boxplots of numerical features. Due to their large number, we have provided only the

Kernel Density Estimates (KDEs) of the numerical features below:



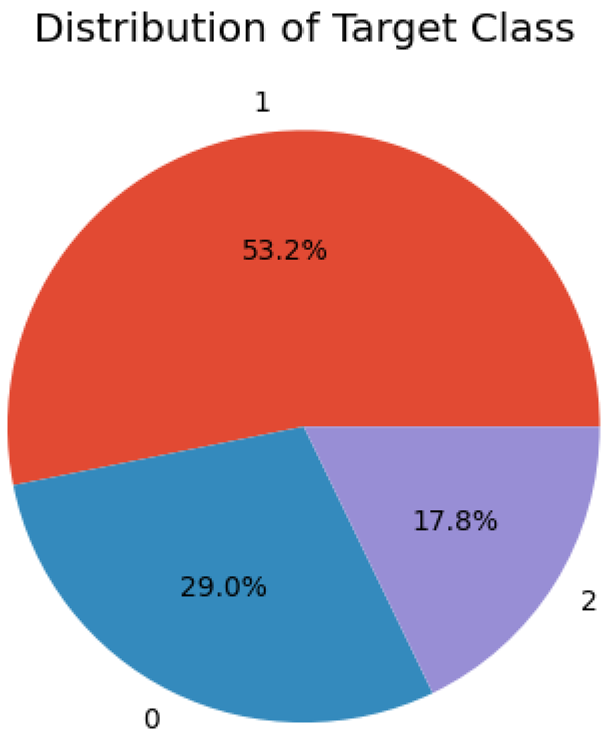
→ Upon reviewing the distributions and boxplots of the numerical features, we observed that none of them adhere to a normal distribution, and there are outliers that require attention. However, as per project guidelines, we refrained from dropping any records.

→ The Heat Map of Correlation matrix for numerical columns is as follows:

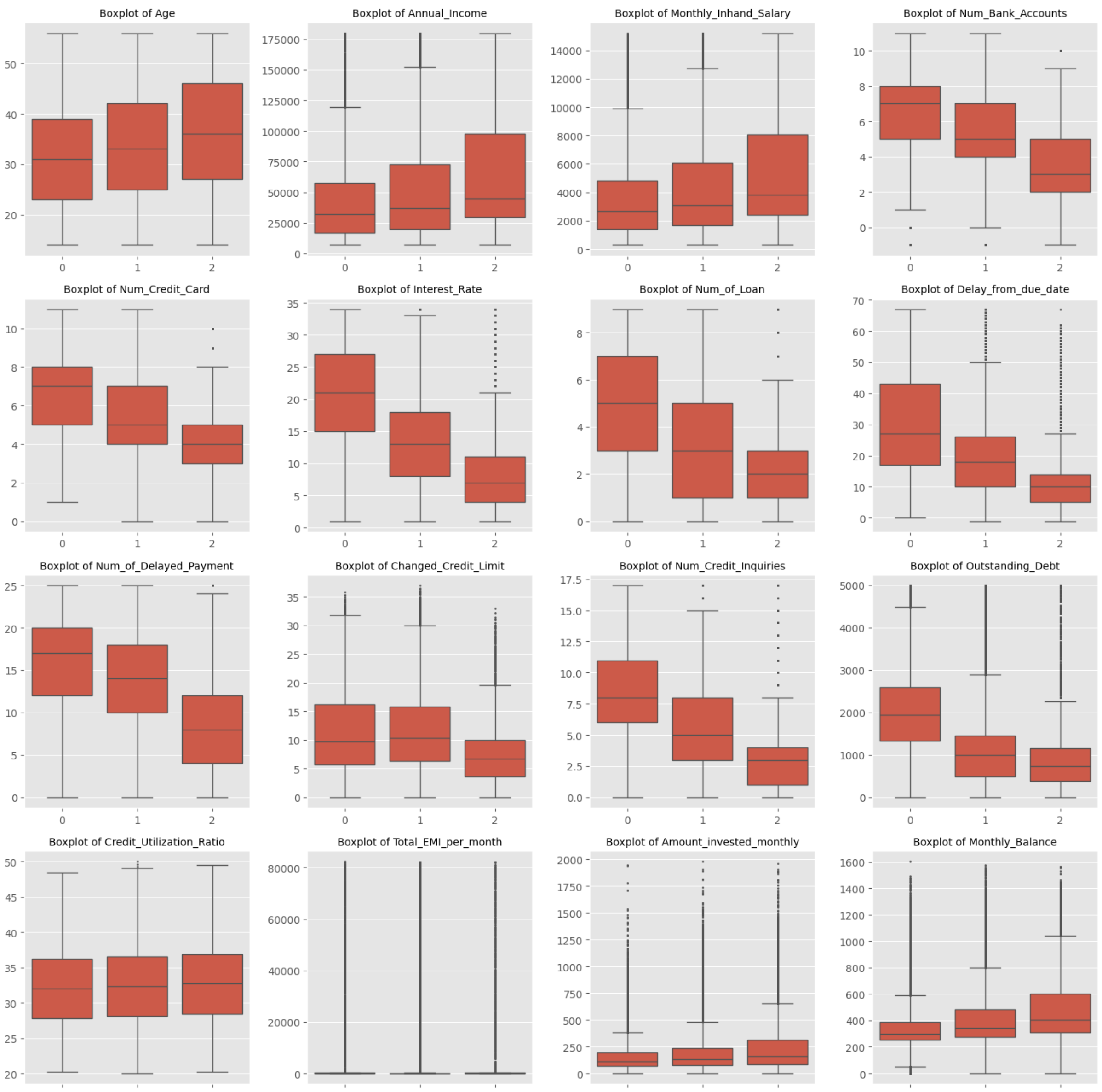


→ The highest correlation is observed between 'Monthly_Inhand_Salary' and 'Monthly Balance', which is expected since a higher monthly salary typically corresponds to a higher monthly balance, and vice versa. On the other hand, the lowest correlation is observed between 'Monthly Balance' and 'Num_of_Loan', which is also understandable. A higher monthly balance implies a higher salary and suggests fewer loans taken.

→ Regarding the distribution of the target class, the majority of records indicate a Standard credit score, followed by Poor credit score.



→ We have then looked into credit score wise boxplots and stripplots of numerical features to check if any feature is able to differentiate the credit scores. Boxplots are as follows:



→ Strip plots didn't offer useful insights, but boxplots did. They reveal that certain variables like 'Num_Credit_Card', 'Interest_Rate', 'Num_of_Loan', 'Delay_from_due_date', 'Num_of_Dealyed_Payment', and 'Num_Credit_Inquiries' exhibit higher values for 'Poor' credit scores, which is expected. Supervised Learning Models should account for this to make accurate predictions. Moreover, the boxplot for 'Monthly_Balance' indicates that individuals with 'Good' credit scores tend to have higher monthly balances, suggesting they are typically higher earners compared to others.

INFERENTIAL STATISTICS

After conducting exploratory data analysis (EDA), we decided to perform hypothesis testing on the data. Here are the summarized questions and findings from four tests:

1. One-way ANOVA was initially attempted to assess if Annual Income is consistent across all target variables with a 5% significance level. However, ANOVA assumptions of normality and Levene's test were violated. Subsequently, we employed non-parametric Kruskal-Wallis test followed by post-hoc Dunn’s test, revealing that Annual Income varies significantly across all target variables.

2. Similarly, we applied the same procedure to examine if the median values of 'Credit_Utilization_Ratio' differ significantly among the target variable classes. Once again, the assumptions were not met, and p-values were nearly zero for pair-wise comparisons, indicating substantial differences in credit utilization ratios among the groups.

3. A chi-square test was conducted to investigate the independence between Occupation and Credit Score. With a p-value close to zero, we concluded that there exists a dependence between these two features.

4. Likewise, a chi-square test was performed to assess the relationship between Payment Behavior and Credit Score. The results indicated a significant dependence between these two variables.

MODELS AND PERFORMANCE

→ We have implemented a total of 6 models to check which model is better at finding intricate patterns in data to make accurate predictions. The models are:

1. Decision Trees
2. K Nearest Neighbors
3. Random Forests
4. XG Boosting Classifier
5. Ada Boosting Classifier
6. Gradient Boosting Classifier

→ After fitting the training data to a model, we observed that it took considerable time due to the large dataset of 100,000 records and computational constraints. To address this, we split the train set into sub-train and sub-test sets using the train_test_split function in an 85:15 ratio. We then hypertuned and trained the models with the sub-train set and evaluated their performance on the sub-test set as well as cross-validation scores on the entire train set. Here's a summary of the performance of different models:

Models	Cross Validation Score before Hypertuning Parameters	Cross Validation Score after Hypertuning Parameters	Performance on Sub-Test Set
Decision Tree	70.77%	70.79%	71%
KNN	64.21%	65.62%	-
Ada Boosting	65.03%	63.42%	71%
XG Boosting	69.96%	71.21%	-
Random Forest	69.73%	71.06%	71%
Gradient Boosting	70.45%	70%	81%

→ Few points to mention:

1. For simpler models like Decision Tree and KNN, we tuned parameters using GridSearchCV. However, for Boosting Classifiers, we used RandomizedSearchCV with 100 iterations due to computational constraints.

2. During RandomizedSearchCV, cross-validation was performed using ShuffleSplit with 1 split and an 80:20 ratio. This significantly reduced time and computation.
3. All the scores were based on ‘Accuracy’ only.

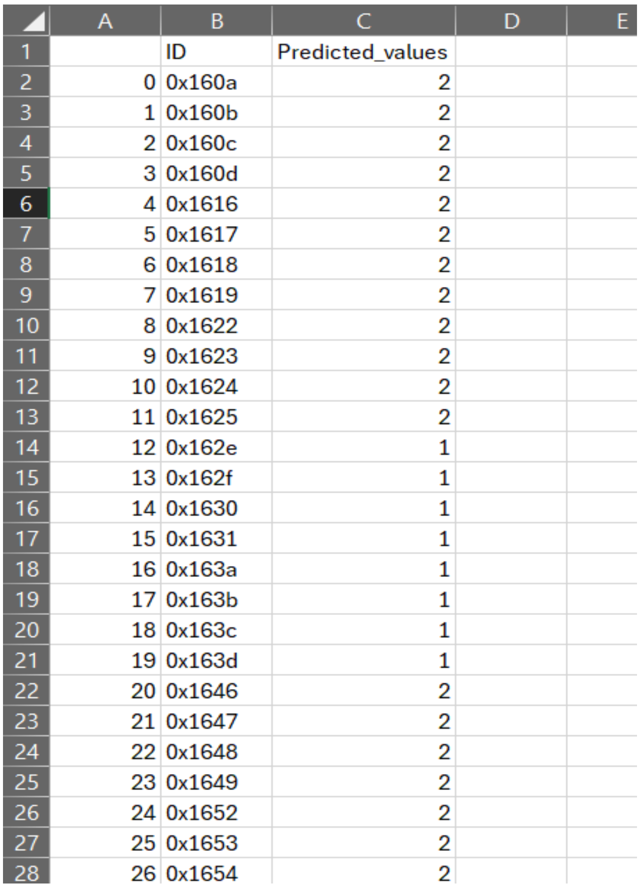
→ From the Performance Summary table, it's evident that Gradient Boosting achieved much higher accuracy on the sub-test set compared to others. Therefore, we selected this model for making predictions on the 'Cleaned Test.csv' data.

FINAL MODEL AND PREDICTIONS

```
# Training Gradient Boost Model with best parameters deduced from
hypertuning
model = GradientBoostingClassifier(validation_fraction=0.1,
n_estimators=200, min_samples_leaf=8, max_features=21, max_depth=9,
criterion='friedman_mse')

model.fit(X_train, y_train)
```

→ We have generated credit score predictions and saved them in the 'Output.csv' file. Here are the first few rows:



	A	B	C	D	E
1		ID	Predicted_values		
2	0	0x160a	2		
3	1	0x160b	2		
4	2	0x160c	2		
5	3	0x160d	2		
6	4	0x1616	2		
7	5	0x1617	2		
8	6	0x1618	2		
9	7	0x1619	2		
10	8	0x1622	2		
11	9	0x1623	2		
12	10	0x1624	2		
13	11	0x1625	2		
14	12	0x162e	1		
15	13	0x162f	1		
16	14	0x1630	1		
17	15	0x1631	1		
18	16	0x163a	1		
19	17	0x163b	1		
20	18	0x163c	1		
21	19	0x163d	1		
22	20	0x1646	2		
23	21	0x1647	2		
24	22	0x1648	2		
25	23	0x1649	2		
26	24	0x1652	2		
27	25	0x1653	2		
28	26	0x1654	2		

WHAT COULD HAVE BEEN DONE BETTER

→ While our focus was primarily on boosting and bagging algorithms due to the assumption of their superior performance, we acknowledge that implementing additional multiclass classifiers could have provided a more comprehensive understanding of performance variations across different models.

→ Additionally, we recognize the potential benefit of incorporating more advanced ensemble techniques, such as the Stacking Method, into our analysis. Stacking involves training multiple base classifiers and then using a meta-learner to combine their predictions, leveraging the diverse perspectives offered by individual models to potentially enhance overall predictive accuracy and robustness. By embracing such advanced methodologies, we could have enriched our predictive modeling capabilities and gained deeper insights into the intricacies of the dataset, potentially uncovering even more nuanced patterns and relationships for credit score prediction.

TAKEAWAYS AND CONCLUSION

→ The thorough preprocessing conducted enabled us to explore new approaches for handling and cleaning datasets. This step was instrumental in identifying and addressing data inconsistencies, outliers, and missing values, thereby laying a solid foundation for subsequent analysis.

→ Following preprocessing, in-depth exploratory data analysis (EDA) uncovered hidden patterns and relationships within the dataset. This comprehensive exploration facilitated the identification of features that exhibited significant variations across different target classes, providing valuable insights for model building and prediction tasks.

→ In summary, this analysis provided valuable insights into model selection, parameter tuning, efficient cross-validation strategies, and the practical implications of classifier performance in real-world applications, particularly in credit score prediction tasks. These learnings can significantly benefit future workspace endeavors, guiding the adoption of effective machine learning techniques and strategies for optimal performance and deployment.