NOVA IMS Information Management School

(2023-2024)

Report

Credit Risk Modeling

Group 11

Realized by:

Filipe Francisco 20230903

Luís Silvano 20201479

Maria Martins 20230243

Nilda Neto 20230247

Nuno Duarte 20230262

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Index

Figure’s Index

Table's Index

1. Credit Risk Modeling Abstract

Credit risk modeling is an important component of finance that plays a relevant role in assessing and managing the possible risks connected with lending and financial transactions. It evaluates the likelihood of a borrower defaulting on a loan or failing to meet their financial obligations using statistical approaches, mathematical models, and data analysis. Credit risk modeling entails collecting and analyzing a large amount of data, such as the borrower's credit history, financial records, economic indicators, and other relevant information. This information is then utilized to create predictive models that evaluate the creditworthiness of individuals, businesses, and other entities. The models aim to identify patterns, correlations, and risk indicators that can be used to assess the likelihood of default and the severity of losses. Furthermore, credit risk modeling is a key discipline that enables financial institutions to make sound lending decisions, efficiently manage risk, and sustain the financial ecosystem's stability. As technology progresses and financial markets evolve, the field's approaches and models are refined to address the difficulties of an ever-changing landscape.

2. Credit Risk Introduction

Credit scoring can be described as a variety of statistical analyses that can provide its users with a reasonable estimate of the probability that a loan applicant or any counterparty will default or become delinquent (World Bank, 2019). For this academic curricular unit Predictive Analytics in Finance, we were proposed to carry out a project related to credit Risk Modeling. Often, risk scoring involves the usage of predictive models that provide statistical probabilities that any given applicant will be good or bad in terms of grades (Siddiqi, 2005). To meet the requirements of the curricular unit of Predictive Analytics in Finance, taught by Professor Afshin Ashofteh, we were proposed to carry out a project where we would process data based on the material taught in class and, subsequently, take conclusions about this same data, strengthening our problem formulation and solving skills, research, cooperation, and communication. **Firstly**, our team **explored a set of data provided**, analyzing clients' characteristics, whether clients are eligible to receive a loan, and associated risks. We observed the training and the predictive datasets, to understand which values needed to be handled (evaluate their structure, and the datatypes of each variable and recognize the missing values and outliers of each variable). **Secondly**, we **start by preprocessing** the raw data in each dataset (treating missing values and outliers, transforming categorical data into numerical and into dummies, we also created new variables, evaluating, and checking data coherence, applying correlation, ANOVA test, and comparing the importance of models as a way of performing feature selection, and scaling variables), and finally, we prepared both the validation dataset and the test dataset in the same way as we did the training. **Thirdly**, we **implemented different models** in our project, such as Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and Naïve Bayes all regarding classification problems, this found out which model was more competent for predicting the target. After these classifier models, we made a final analysis of the Deep Model where we used the library (TensorFlow) neural network model in which the objective is to find patterns in the complex data and monitor their respective performance.

3. Primary Statistical Analysis

After importing the respective packages and the respective datasets in Python, we started with Exploration Criteria, we aimed to explore which values we had and which attributes we would concern from the training dataset and the predictive dataset (the first look at the datasets). We understand that when we applied the code to get the shape of the table, we had 20,600 clients and 27 attributes in the predictive dataset, and we had 310,704 clients and 29 attributes in train validation to work with (- Datatypes of our variables). After realizing the structure of our dataset, our team decided to read the types of each attribute, noticing some categorical data that would need to be treated to use some of the models to understand data accuracy. Next, we compared the frequency of missing values from each training data set (- Percentage of Missing Values each Variable). In general terms, we identified 179 duplicate rows only in the predictive dataset, which represented 0.87% of the overall data set, and regarding the train validation we did not have any duplicates. Then, we want to evaluate the clients from their loan grade, employment length, home ownership, income source verification, loan's purpose, address state, and the capability of paying the loan. So, for that firstly, we had to **define our target variable** so we could **understand when we have a good loan** (non-default loan), or a bad loan (default loan) according to the respective evaluation, in both datasets. In train validation we just had **to replace the values** Charged Off', 'Default',' Does not meet the credit policy. Status: Charged Off', and 'Late (31-120 days)' for default and the others by non-default, and in the predictive **dataset we had to create a new variable customer default status** where the good loans were when their verification status was verified of the respective loan and the other's will be non-defaults because the loans were not verified.

To **address outliers**, we first carried out box-plot graphs, so we could identify which exact variables were presented. After recognizing their presence in the main variables that we consider critical for our analysis, we treat them by changing the outliers by their median when the variables have less than 1% of outliers according to all datasets in predictive and train validation datasets, and we used another method called Winsorization to treat the outliers when the variables have more than 3% of outliers; this method replaces that specific outlier from the values that are closest to the quartile one or quartile three according to the significance level that we select. We used these two approaches because we didn't want to lose information from our dataset if we removed the rows, and we also didn't want to have different scales in the respective variables if we transformed the data. Finally, we didn't want to use the mean to replace the outliers because the value of the mean was already influenced by the outliers. After understanding our dataset better, we did an individual analysis of each consumer loan data. We may conclude from Loan Grade **that customers classified as default are mostly from class B, C, and A, and now** according to the non-default most clients are classified as grade C, with 33% of these grades being non-default, indicating that they are strong possibilities for a loan. It also has a lot of clients classified as B, with a percentage of 28% Non-Default, and D, with a percentage of 16% non-default regarding the predictive dataset. Now in the train validation dataset, we conclude from Loan Grade **that customers classified as default are mostly from class B, C, and A, and now** according to the non-default most clients are classified as grade B, with 30% of these grades. It also has a lot of clients classified as C, with a percentage of 29% Non-Default, and A, with a percentage of 17%. This Client's Loan Grade chart refers to the credit quality assigned to a borrower's loan based on various factors such as credit history, financial stability, the quality of the guarantee, the probability of repayment, and other relevant indicators. Lenders use loan grades to assess the risk associated with lending money to a particular individual or business. The classification consists of assigning a grade depending on your default risk. Thus, on the x-axis, we can visualize the number of people (clients) whose loan was classified with a grade, that is, the grades represented on the y-axis. The Client Employment Length graph shows us the time taken by clients to eventually grant credit. On the y-axis, we have the years of work and on the x-axis, we have the number of people who have worked for n years. According to the predictive dataset, we realized that the higher number of individuals who non-default is 4,408 individuals and they have been working for more than 10 years while the lower of non-defaults loans was 465 clients who have been working for 7 years for the loan be approved. For last, we can note as well, that we have more non-defaults than defaults in all years. Now regarding the train validation dataset, we have more defaults than non-defaults in all years, the higher number of clients of non-defaults is 32,528 individuals and they have been working for more than 10 years while the lower is 3,773 individuals and have been working for 9 years. In the Homeownership Status provided by the borrower during registration and the status of the loan quality, that is, if they were non-default or default. Based on the predictive dataset, we can infer that the majority of borrowers disclosed their mortgage status; there are 6,262 individuals who are in non-default and around 3,716 clients who are in default. In-home ownership status rent we had 5960 individuals' non-defaults and 2,099 defaults, in home ownership status own we had 1,587 non-defaults and 794 defaults, and for last, we had any 2 non-defaults and 0 defaults. Attending to the train validation dataset, we have more defaults than non-defaults in all homeownership statuses, we have in mortgage 47,749 non-defaults and 106,580 defaults, in rent, we have 37,872 non-defaults and 82,666 defaults, and in own we have 10,733 non-defaults and 24,849 defaults, for last, in any we have 1 non-default and 254 defaults. In the predictive dataset, the Client's Income Source Verification graph shows us the status of procedures related to verifying the source of income of which client. We can conclude that the main part of clients had their source of income verified, 13,811 loans were verified and 6,609 were not verified. Concerning the train validation dataset, we have more loans verified than not verified, 203,660 verified loans and 107,044 loans not verified. In the predictive dataset and train validation dataset the Client's Purpose Loan, although there are a number of reasons for obtaining credit, the client's purpose for the loan is shown in the predictive dataset and the train validation dataset. This graph allows us to draw the conclusion that the majority of clients seek loans for credit card debt consolidation and debt consolidation; the only areas where we have fewer loans are for weddings and renewable energy. In the predictive dataset the Client's Address state, we observe that most clients are in the state of California, with 3,015 clients, followed by the states of New York 1,889 clients, Texas 1,802 clients, and Florida 1,573 clients. In contrast, the states that have the fewest clients are Idaho 24 clients, Maine with 33 clients, Vermont 34 clients, and, lastly, Alaska with 36 clients. In the train validation dataset, most clients are in the state of California with 43,550 clients, followed by Texas with 26,531 clients, then New York with 24,608 clients, and Florida with 23,290 clients. And the lower value of clients asking for loans is in Vermont with only 622 clients. We saw the Client's Loan Amounts and were able to establish that the middle 50% of customers have between 8,275 and 20,800 dollars, where the minimum and maximum amount of loan obtained by a client is 1,000 and 40,000 dollars, respectively. We can conclude that depending on the Amount of Credit, the minimum and maximum total payments are 0 and 59,808.26 respectively, and with an average that varies from 5,783.21 and 18,592.57 dollars. Overall, we can see a highly skewed observation to the right. We can see each Client's Loan Payment history on the final graph. Then, we can see that 67.7% of the loans in the prediction dataset were paid back, and around 32.3% were not; meanwhile, 58.4% of the loans in the train validation dataset were paid back, and approximately 41.6% were not.

3.1 Preprocessing Credit Analysis

In the Prerocessing Criteria, the objective was to take care of cleaning, transforming, and reducing errors in the raw data of the respective datasets (Data Preparation, Data Cleaning, Feature Engineering, and Scaling Data). We decided to treat the outliers and missing values instead of deleting them as we mentioned before, we changed the data types only in the predictive dataset so we could do extensive analysis with different models that don't allow non-numerical data, we also reviewed the coherence of the data in both datasets, we performed a feature selection, with Univariate Variables for understand if we have any variable that was 0 that was always constant but we did not have any value like that, we did the Pearson and Spearman correlation test for understand the specific relationships of the variables and we found that loan amount and found amount they were high correlated and then the interest rate variable so we must have sure if remove them was a good option so for confirm we also used the ANOVA test for understand what were the variables more important for our target population for could understand better what was their influence in the target variable, and then we used Decision Trees and Random Forest for evaluate the importance of the respective variables again, after understanding that they were giving very similar results and very correlated we decided to remove them, and then lastly, we scaled the data, by adjusting the values of the variables so that they fall within a similar numerical range. So, starting with the preprocessing itself, we made a copy of our initial table of the training dataset, and we dropped columns from the dataset that we believed would not be worthen using, for our analysis: 'id', 'total_acc', 'open_acc', 'inq_last_6mths', 'funded_amnt_inv', 'out_prncp', 'revol_util', 'pub_rec', 'issue_d', 'earliest_cr_line', ' risk', followed by the creation and transformation of new columns, to understand the Payment Progress of the Client Loans. The next move was to treat missing values, first, we confirmed their position and their percentage. Then, to fill in missing values, from all the columns from 'train_f' that are objects, we fill in that respective column with the mode, and all the columns that are integer, will fill in that respective column with their mean. Then we had to check the coherence of our dataset, as noticed before, we had some issues with the variables Verification Status, Employment Length, and Term. In the first one, we had two categories “Verified” and "Source Verified”, and after checking the variable this did not make sense, so after some consideration, we decided to get these two categories together. In the second one, we removed the ‘+’ in this variable, replaced the whole string “less than 1 year” with the string “0”, and removed the “years” and “year” in 2 categories. For the variable “Term”, we removed the months in the categories, leaving just “36” and “60”. Despite this, we still changed the decimal class of the numerical variables, rounding to 2 decimal places in Loan Amounts, Annual Income, and total_pymnt. Now, we have decided to treat the outliers. Firstly, we checked the median of our variables (50%). The method that we will use for treating the outliers is to change by their median and Winsorization because if we remove them, we will lose information. We cannot change by their mean because the value is already affected by the outliers, so we will just replace by their median in the variables that we have low outliers and regarding, where we have more outliers 3%. We believe that replacing the Winsorization is the best course of action because it will replace the specific outliers from the values that are closest to Q3 and Q1, and it won't have an impact on the variable's scale. Using the median is not the best option because it is also impacted by outliers.

4. Methodology

For our group to understand whether the client had loans that were good (non-default) or that had loans that were bad (default). Before we could utilize the particular libraries and features on datasets, we had to import the corresponding datasets into Python and the corresponding packages. Following import, we choose to examine the relevant datasets to gain a better grasp of the subject matter by utilizing the dataset’s variables, structures, and graph-based data visualization. As a result, following the exploration, we began organizing the relevant datasets, cleaning them, and choosing the best variables for feature selection (Univariate Variables to understand if we have any variable that was 0, we did the Pearson and Spearman correlation test for understand the specific relationships of the variables, we did ANOVA test for understand what were the variables more important for our target population for have an accurate analysis, and we used Decision Trees and Random Forest), and then scaling the respective data for after could apply the respective algorithms for understanding the accuracy of the data, after treating all the data, so finishing the preprocessing, we created the respective train, validation, and the test for we could see the final results by applying the models, how good was our data, their accuracy. After creating the train, validation, and test we started applying the models and then saw their respective results, and how accurate they were, and in the end, we applied neural networks in the Deep Model in a way to see the results of the accuracy.

5. Discussion

5.1 Modeling Assessment

To develop and assess our estimations for a credit scoring model we created a logistic regression model (LRM), and then we carried out confusion matrices so we could understand if the results computed were right. Commonly, a Confusion Matrix (CM) allows us to have a hint of how well-fitted a model can be in comparison with another one. At first, we divided our data set into train and validation, then we created a logistic regression model for both data set types, and ultimately, we calculated the confusion matrices for each model developed. Results suggest that our Training data set was better fitted to the validation data set and therefore achieved more trustworthy results. Our confusion matrices demonstrate that with the Training Logistic regression, we had 35,195 True Positives (TP) and 152,015 True Negatives (TN). Conversely, we had 4,227 TP and just 19,115 TN for the validation logistic regression. Additionally, we plotted a graph to observe the different matrices we could have applied various thresholds to get our proportions of TP and FP (False Positives) rate, which we call Receiver Operator Characteristic Curve, or simply ROC Curve, whereas the Area under the curve (AUC) would give us which model present better predictions, considering the matrices. The same reasoning, we applied for the rest of the models, and we ended up getting five different predictive models with different prediction results to compare one another and finally choosing the one that better fit. In this sense, our training data set produced better prediction results in the Random Forest, Gradient Boosting Classifier (GBC), and Naive Bayes model. Therefore, given the different ROC curves plotted under the models developed, we would have better results if we used the GBM models amongst others because they contain the greater AUC value. Nonetheless, we can also analyze the quality of our predictions using three main tools: Precision, which gives us the total TP estimates we had in a universe of TP and FP, Recall, which gives us an idea of the total number of TP estimates we got correctly among the universe of TP, and finally the F1 that kind of plays the role of an average ratio between these last two mentioned. Hence, for evaluating the quality of our models we will stick to the F1, and for instance, in the case of LRM the train data set had a little higher F1 in predicting False Positive (FP), specifically train data set had 0.53 of F1 score, and the validation had 0.52 of F1 score, however, they scored the same accuracy, roughly 0.75. On the other hand, Naive Bayes Algorithm scored equal results for F1 regarding accuracy, and so on. In general, this study uncovers a high level of True Negatives Rate across the different models. As it is, this measure represents that the bank can trust in the reliability of its model to correctly predict the probability of default in loans. A True Negative rate plays an important part in credit scoring in a way that can give fair predictions about the present and future loan clients for each loan product portfolio. With a fair estimate of the probability of default, loan managers can have more trustworthy estimations for issuing reserves in the balance sheet and complying with different legal requirements of risk management. Overall, our F1 score (right side on the right side in the brackets of the table), indicates a high rate across all the models. On the other hand, AUC, also in Table 1., shows similar performances between Random Forest and GBC models, and slightly below levels for Logistic and Naïve

Bayes models. Finally, the Gradient Boosting Classifier achieves the best performance both on the training and validation data set.

6. Deep Modell learning machine

6.1 Accuracy and loss for training and validation data set

In terms of accuracy of the deep model learning, we applied 100 epoch size for both the [train]() and [validation data]() set, and results generally suggest that as the number of epochs grow across both data sets our losses tend to reduce, and the accuracy goes into the opposite direction. That means as our deep learning machine for credit scoring increases in experience and understanding we are steadily getting better predictions. An interesting fact is that from epoch 27 up to 79 we assist those losses in the validation data set increasing, which might represent a phase where the model encounters challenges or fluctuates in its learning process, then they go down from 80 epoch to the end, suggesting a positive adjustment or improved learning in the latter part of the training process. Overall, the pattern shows that as the number of epochs increases, the deep learning model improves accuracy and decreases losses by improving its comprehension of the credit scoring problem. The validation data exhibits irregularities during specific epochs that may be attributed to several variables, including but not limited to the model design, data properties, or optimization tactics. To achieve the best credit score predictions, the model's performance should be continuously monitored, analyzed, and adjusted as necessary.

7. Conclusion

In conclusion, in the given problem to predict the probability of default depending on their behavior, we observed the dataset and did an exploration of it and we also saw that we had to make some adjustments to have a more precise and correct model. So then, we treated missing values and outliers, checked the coherence of the dataset, created some new variables, did a feature selection, and then did a preparation of both validation and test datasets. After this phase was time to write different algorithms for the different models and adjust them with different parameters to improve them and get better results. While the training dataset exhibits a commendable accuracy of approximately 1, the other one exhibits an accuracy of 0,75, the F1 scores. The results suggest that the Gradient Boosting Classifier, particularly on the training dataset, showcases a robust performance in credit scoring, providing the best valuable insights for decision-making in lending scenarios. With this, we obtained different outcomes so after some analysis and comparison, we evaluated each one of them, and based on the values we decided to get the results of the dataset. So, we believe that the model we implemented into our project will be a good solution to predict whether a client will be chosen to get the loan or not.

8. Annex

8.1 Unseen Dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20600 entries, 0 to 20599
Data columns (total 27 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   id                   20600 non-null  object
 1   loan_amnt            20420 non-null  float64
 2   funded_amnt          20420 non-null  float64
 3   funded_amnt_inv      20420 non-null  float64
 4   term                 20420 non-null  object
 5   int_rate             20420 non-null  float64
 6   installment          20420 non-null  float64
 7   grade                20420 non-null  object
 8   emp_title            19196 non-null  object
 9   emp_length           19210 non-null  object
 10  home_ownership       20420 non-null  object
 11  annual_inc           20420 non-null  float64
 12  verification_status  20420 non-null  object
 13  issue_d              20420 non-null  float64
 14  purpose              20420 non-null  object
 15  addr_state           20420 non-null  object
 16  dti                  20402 non-null  float64
 17  delinq_2yrs          20420 non-null  float64
 18  earliest_cr_line     20420 non-null  float64
 19  inq_last_6mths       20420 non-null  float64
 20  open_acc             20420 non-null  float64
 21  pub_rec              20420 non-null  float64
 22  revol_bal            20420 non-null  float64
 23  revol_util           20403 non-null  float64
 24  total_acc            20420 non-null  float64
 25  out_prncp            20420 non-null  float64
 26  total_pymnt;         20420 non-null  object
dtypes: float64(17), object(10)
memory usage: 4.2+ MB
```

Figure 1 - Datatypes of our dataset

```
loan_amnt              0.873786
funded_amnt            0.873786
funded_amnt_inv        0.873786
term                   0.873786
int_rate               0.873786
installment            0.873786
grade                  0.873786
emp_title              6.815534
emp_length             6.747573
home_ownership         0.873786
annual_inc             0.873786
verification_status    0.873786
issue_d                0.873786
purpose                0.873786
addr_state             0.873786
dti                    0.961165
delinq_2yrs            0.873786
earliest_cr_line       0.873786
inq_last_6mths         0.873786
```

Figure 1.1 - Percentage of Missing Values each Variable

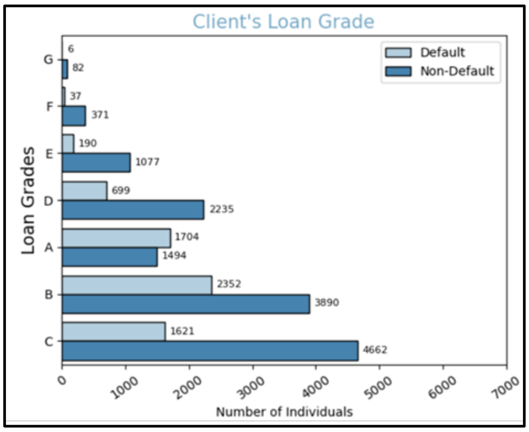

Figure 2 - Client's Loan Grade

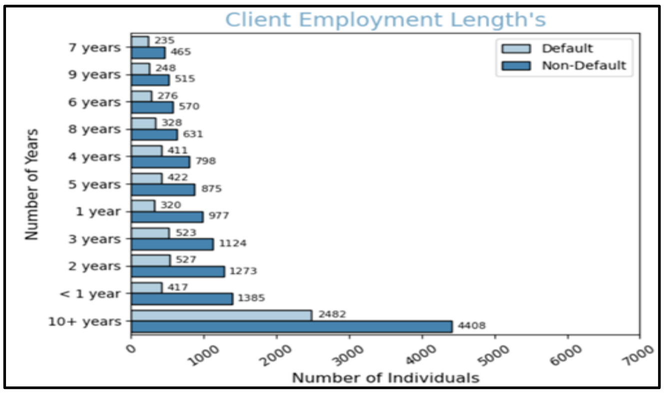

Figure 3 - Client's Employment

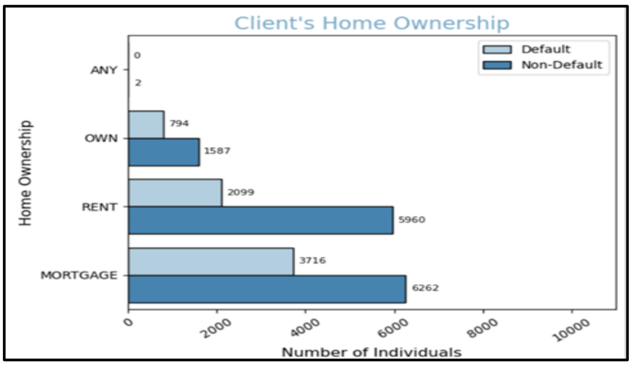

Figure 4 - Client's Home Ownership

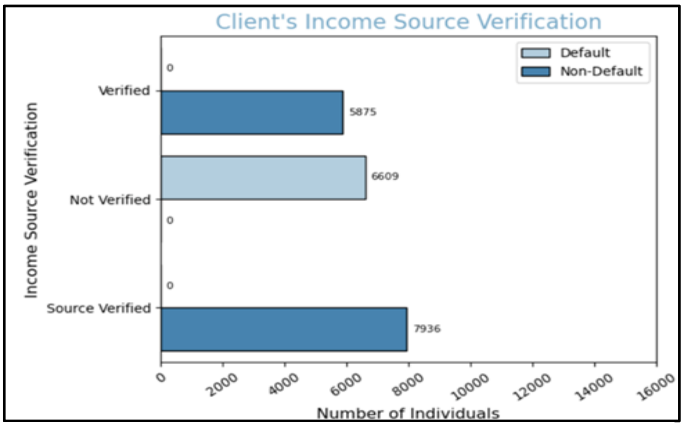

Figure 5 – Client's Income Source

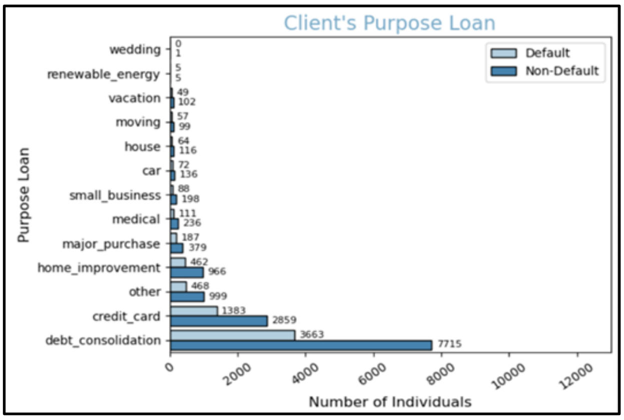

Figure 6 – Client's Purpose Loan

Figure 7 – Client's Address

Figure 8 – Client's Loan Amounts Box-plot

Figure 9 – Client's Total Credit Revolving Balance Box-Plot

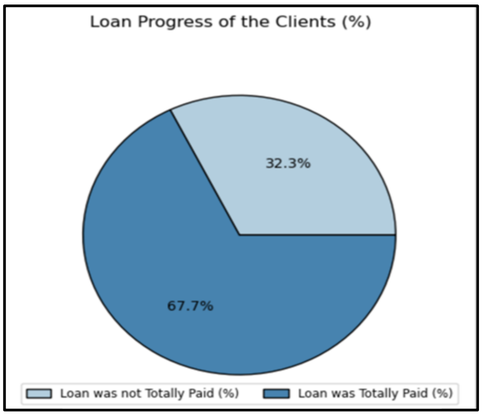

Figure 10 – Loan Progress of Clients

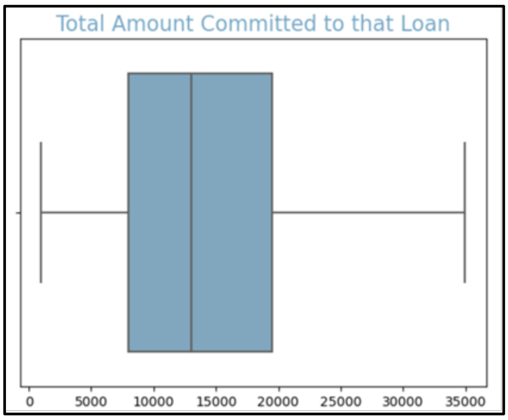

Figure 11 – Total Amount Committed to that Loan

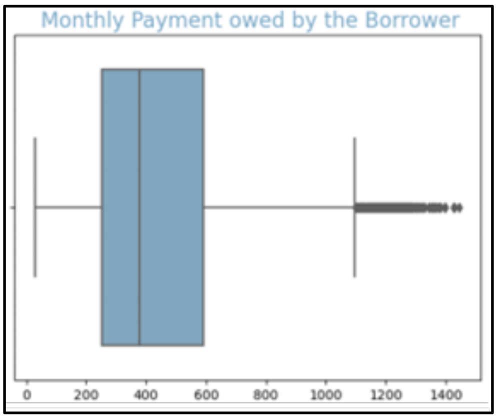

Figure 12 – Monthly Payment Owed by the Borrower

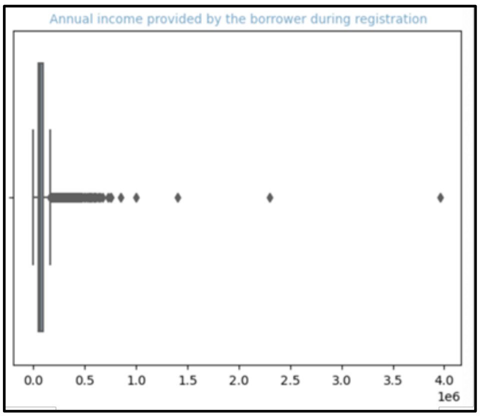

Figure13 – Annual Income Provided by the Borrower during Registration

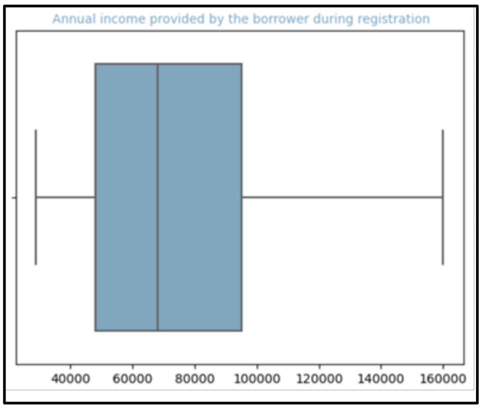

Figure 13.1 – Annual Income Provided by the Borrower during registration

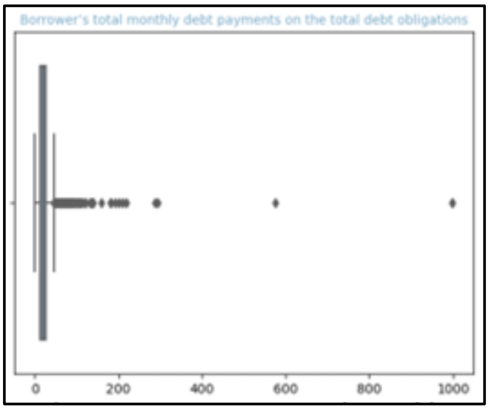

Figure 14 – Borrower's Total Monthly Debt Payments on the Total Debt Obligations

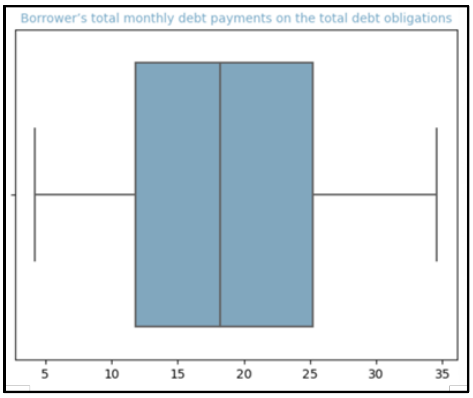

Figure 14.1 – Borrower's Total Monthly Debt Payments on the Total Debt

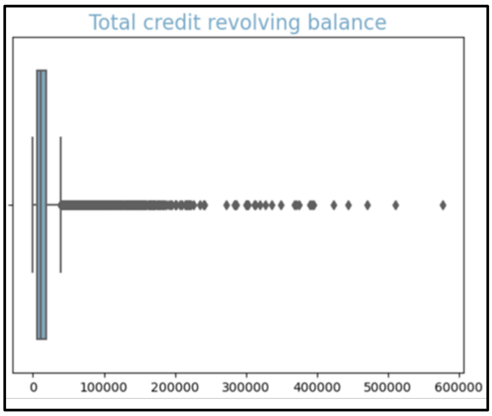

Figure 15 – Total Credit Revolving Balance

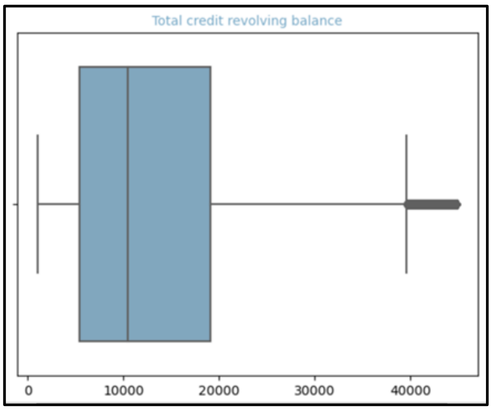

Figure 15.1 – Total Credit Revolving

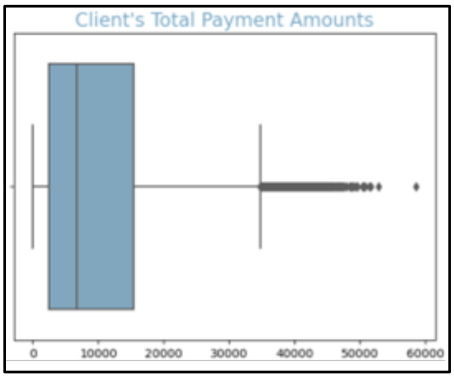

Figure 16 – Client's Total Payment

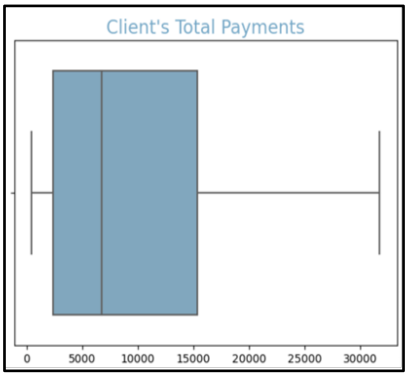

Figure 17 – Client's Total Payments without Outliers

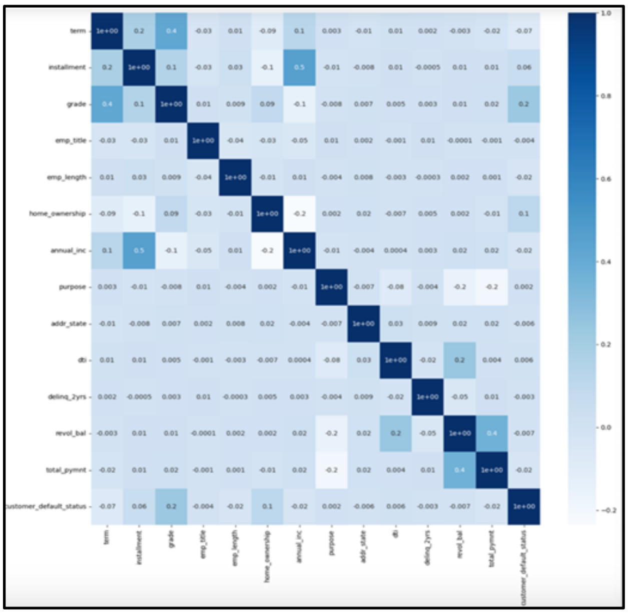

Figure 18 – Pearson Correlation Matrix

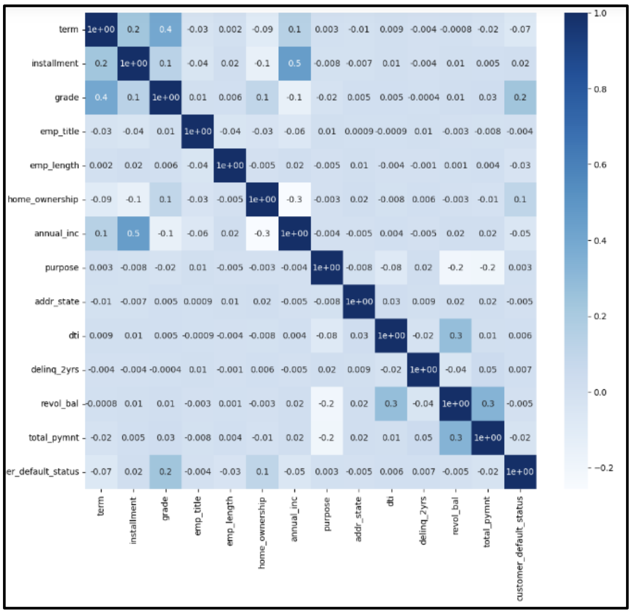

Figure 19 – Spearman Correlation

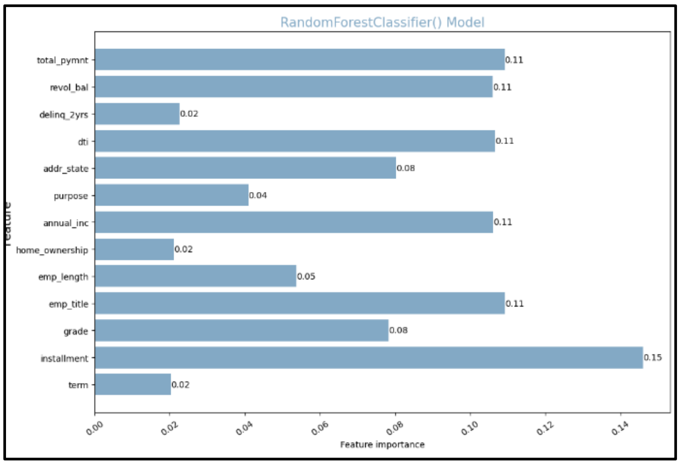

Figure 20 – Random Forest Classifier

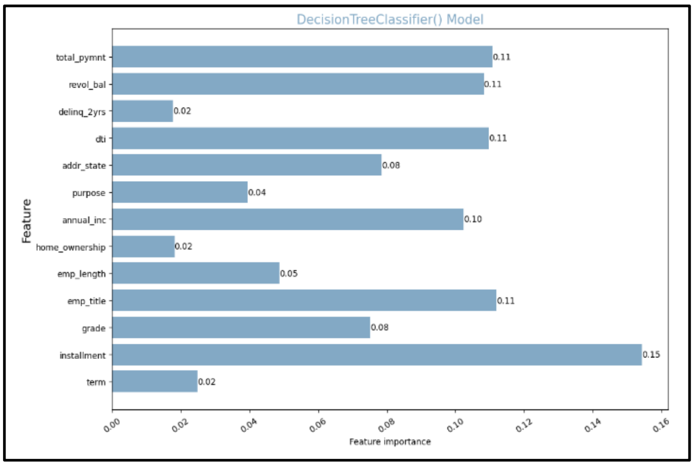

Figure 21 – Decision Tree Classifier

term	True
installment	True
grade	True
emp_title	False
emp_length	True
home_ownership	True
annual_inc	False
purpose	False
addr_state	False
dti	False
delinq_2yrs	False
revol_bal	False
total_pymnt	True
dtype: bool	

Figure 22 – ANOVA Test

TRAIN	precision	recall	f1-score	support
0	0.59	0.26	0.36	5344
1	0.72	0.91	0.81	11136
accuracy			0.70	16480
macro avg	0.66	0.59	0.58	16480
weighted avg	0.68	0.70	0.66	16480
VALIDATION	precision	recall	f1-score	support
0	0.59	0.26	0.36	668
1	0.72	0.92	0.81	1392
accuracy			0.70	2060
macro avg	0.66	0.59	0.58	2060
weighted avg	0.68	0.70	0.66	2060

RESULTS
Train: 0.8058413804020896
Validation: 0.8059418457648546

Figure 220 – Logistic Regression

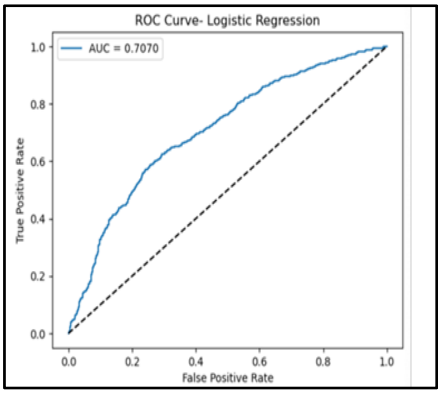

Figure 23.1 – ROC Curve Logistic Regression

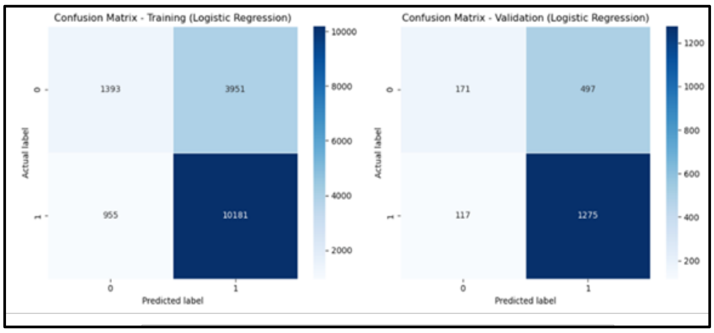

Figure 23.2 – Confusion Matrix Logistic

TRAIN	precision	recall	f1-score	support
0	1.00	1.00	1.00	5344
1	1.00	1.00	1.00	11136
accuracy			1.00	16480
macro avg	1.00	1.00	1.00	16480
weighted avg	1.00	1.00	1.00	16480
VALIDATION	precision	recall	f1-score	support
0	0.56	0.35	0.43	668
1	0.74	0.87	0.80	1392
accuracy			0.70	2060
macro avg	0.65	0.61	0.61	2060
weighted avg	0.68	0.70	0.68	2060

RESULTS
Train: 0.9998786407766991
Validation: 0.7014563106796117

Figure 24 – Random Forest

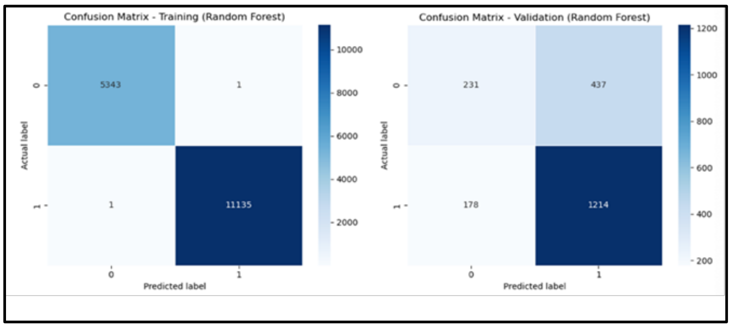

Figure 24.1 – Confusion Matrix Random Forest

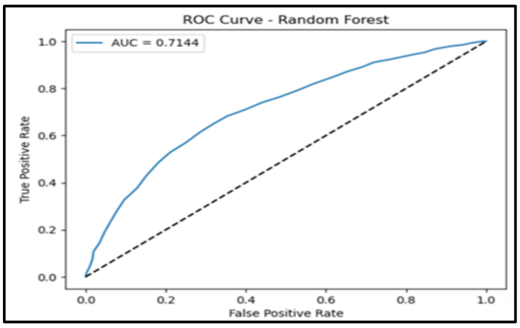

Figure 24.2 – ROC Curve Random

TRAIN	precision	recall	f1-score	support
0	0.64	0.30	0.41	5344
1	0.73	0.92	0.82	11136
accuracy			0.72	16480
macro avg	0.69	0.61	0.61	16480
weighted avg	0.70	0.72	0.68	16480
VALIDATION	precision	recall	f1-score	support
0	0.63	0.27	0.38	668
1	0.72	0.93	0.81	1392
accuracy			0.71	2060
macro avg	0.68	0.60	0.59	2060
weighted avg	0.70	0.71	0.67	2060

RESULTS
Train: 0.7185679611650485
Validation: 0.712135922330097

Figure 25 – Gradient Boosting Classifier

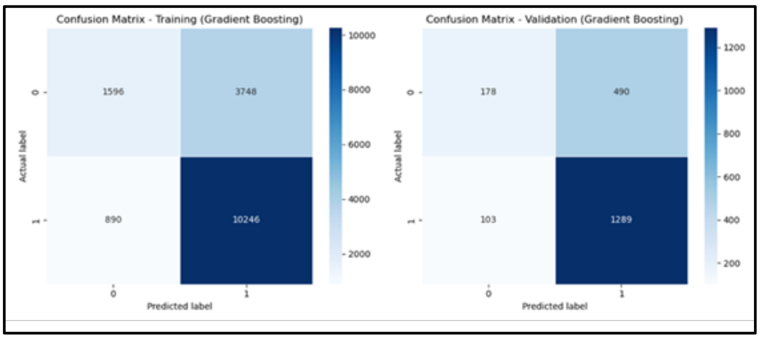

Figure 25.1 – Confusion Matrix

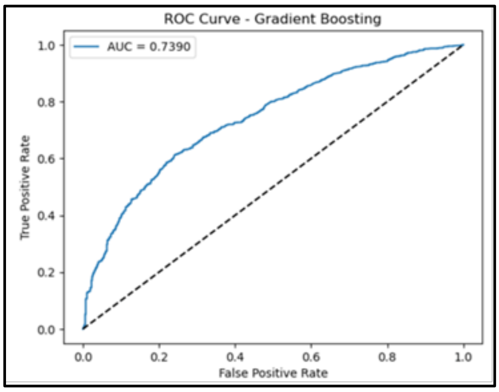

Figure 25.2 – ROC Curve

```
----------------------- TRAIN -----------------------
              precision    recall  f1-score   support

           0       0.58      0.25      0.35      5344
           1       0.72      0.91      0.80     11136

    accuracy                           0.70     16480
   macro avg       0.65      0.58      0.58     16480
weighted avg       0.67      0.70      0.66     16480

-------------------- VALIDATION ---------------------
              precision    recall  f1-score   support

           0       0.56      0.24      0.34       668
           1       0.71      0.91      0.80      1392

    accuracy                           0.69      2060
   macro avg       0.64      0.58      0.57      2060
weighted avg       0.66      0.69      0.65      2060

---------------------- RESULTS ---------------------
Train: 0.6972087378640777
Validation: 0.691747572815534
```

Figure 26 – Naïve Bayes Algorithm

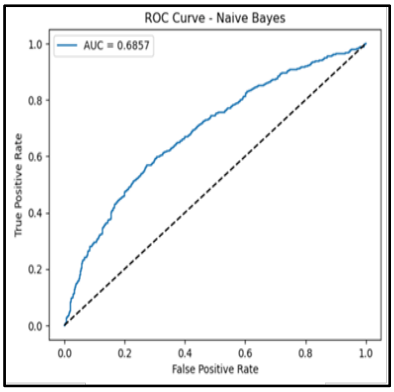

Figure 26.1 – ROC Curve

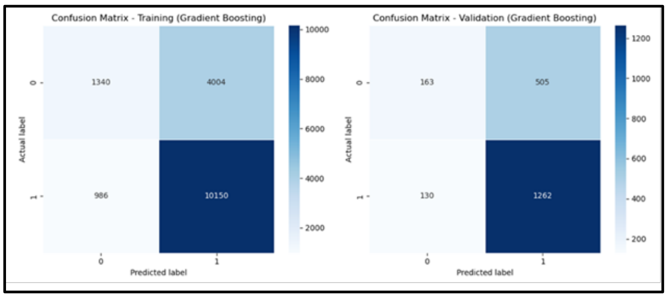

Figure 26.2 – Confusion Matrix

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 310704 entries, 0 to 310703
Data columns (total 29 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   id                   310704 non-null  int64
 1   loan_amnt            310704 non-null  int64
 2   funded_amnt          310704 non-null  int64
 3   funded_amnt_inv      310704 non-null  float64
 4   term                 310704 non-null  object
 5   int_rate             310704 non-null  float64
 6   installment          310704 non-null  float64
 7   grade                310704 non-null  object
 8   emp_title            281139 non-null  object
 9   emp_length           288089 non-null  object
 10  home_ownership       310704 non-null  object
 11  annual_inc           310704 non-null  float64
 12  verification_status  310704 non-null  object
 13  issue_d              310704 non-null  object
 14  purpose              310704 non-null  object
 15  addr_state           310704 non-null  object
 16  dti                  310556 non-null  float64
 17  delinq_2yrs          310704 non-null  int64
 18  earliest_cr_line     310704 non-null  object
 19  inq_last_6mths       310703 non-null  float64
 20  open_acc             310704 non-null  int64
 21  pub_rec              310704 non-null  int64
 22  revol_bal            310704 non-null  int64
 23  revol_util           310491 non-null  float64
 24  total_acc            310704 non-null  int64
 25  out_prncp            310704 non-null  float64
 26  total_pymnt          310704 non-null  float64
 27  loan_status          310704 non-null  object
 28  risk                 310704 non-null  int64
dtypes: float64(9), int64(9), object(11)
memory usage: 68.7+ MB
```

Figure 27 – Datashape & Datatypes of our variables

```
id                   0    id                 0.0
loan_amnt            0    loan_amnt          0.0
funded_amnt          0    funded_amnt        0.0
funded_amnt_inv      0    funded_amnt_inv    0.0
term                 0    term               0.0
                    ..                       ...
total_acc            0    total_acc          0.0
out_prncp            0    out_prncp          0.0
total_pymnt          0    total_pymnt        0.0
loan_status          0    loan_status        0.0
risk                 0    risk               0.0
Length: 29, dtype: int64 Length: 29, dtype: float64
```

Figure 28 – Missing Values

```
Epoch 1/100
165/165 - 1s - loss: 0.5927 - accuracy: 0.6828 - val_loss: 0.5715 -
val_accuracy: 0.7015 - 1s/epoch - 8ms/step
Epoch 2/100
165/165 - 0s - loss: 0.5646 - accuracy: 0.7064 - val_loss: 0.5667 -
val_accuracy: 0.6990 - 287ms/epoch - 2ms/step
Epoch 3/100
165/165 - 0s - loss: 0.5592 - accuracy: 0.7121 - val_loss: 0.5613 -
val_accuracy: 0.7063 - 296ms/epoch - 2ms/step
Epoch 4/100
165/165 - 0s - loss: 0.5548 - accuracy: 0.7154 - val_loss: 0.5659 -
val_accuracy: 0.7024 - 301ms/epoch - 2ms/step
Epoch 5/100
165/165 - 0s - loss: 0.5525 - accuracy: 0.7170 - val_loss: 0.5635 -
val_accuracy: 0.7029 - 272ms/epoch - 2ms/step
```

Figure 29- Deep Modelling – Accuracy and loss

```
65/65 [==============================] - 0s 1ms/step - loss: 0.5673
- accuracy: 0.7150

Test loss: 0.57. Test accuracy: 71.50%
```

Figure 29.1 - Accuracy and Loss test

8.2 Train Validation Dataset

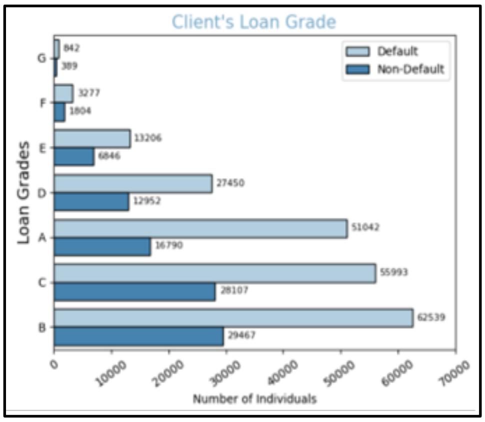

Figure 30 – Client's Loan

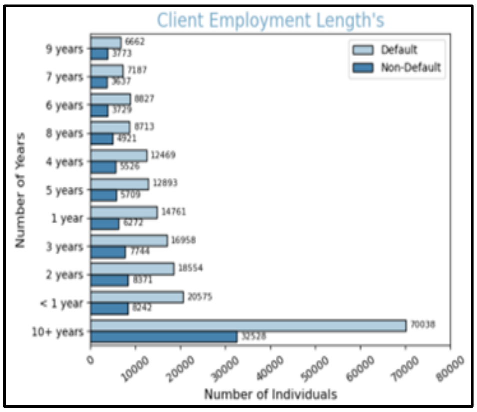

Figure 31 – Client's Employment Length's

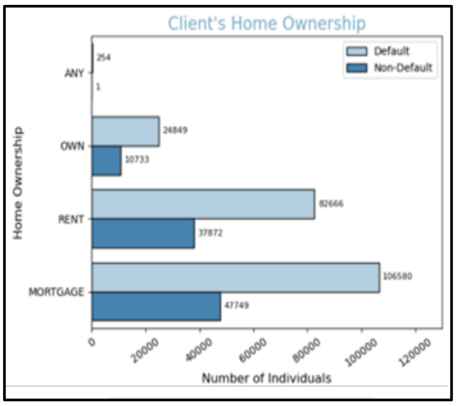

Figure 32 – Client's Home Ownership

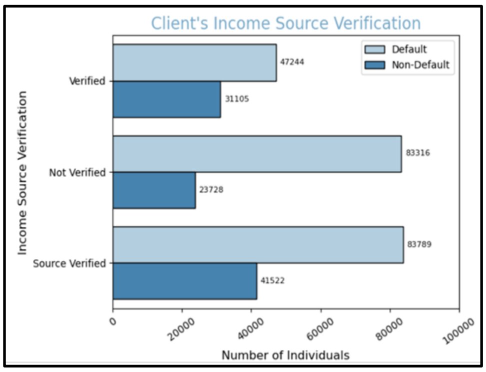

Figure 33 – Client's Income Source

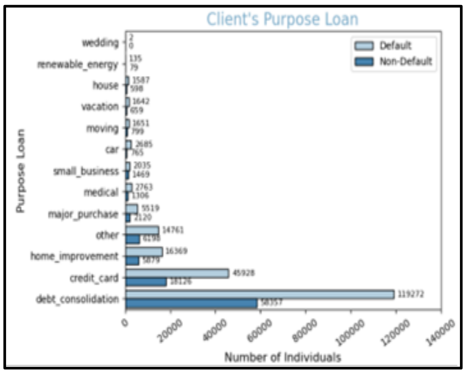

Figure 34 – Client's Purpose Loan

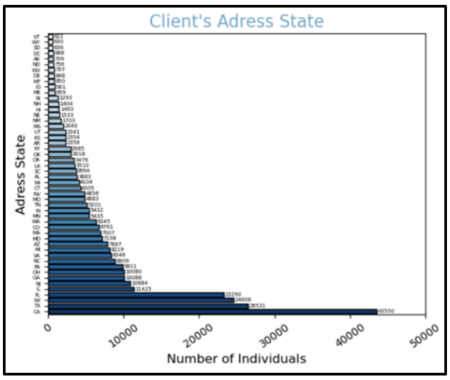

Figure 35 – Client's Address State

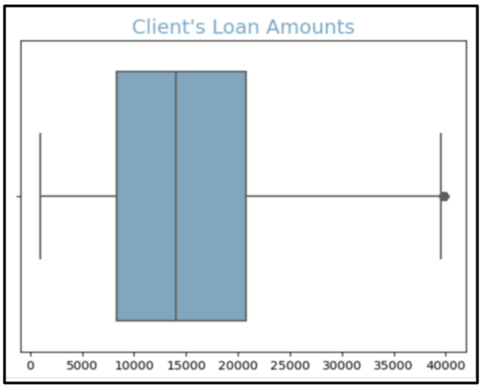

Figure 36 – Client's Loan Amounts

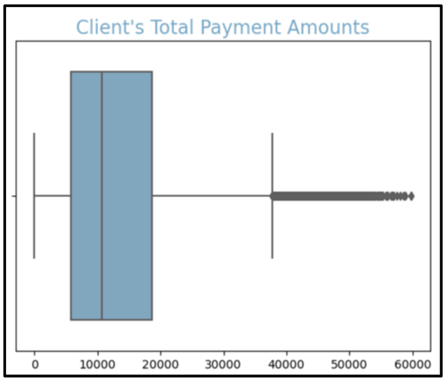

Figure 37 – Client's Total Payment Amounts

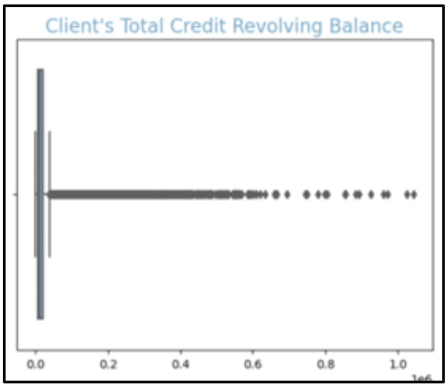

Figure 38 – Client's Total Credit Revolving Balance

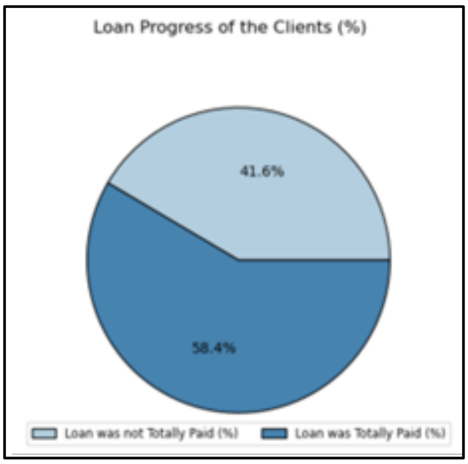

Figure 39 – Loan Progress of the Clients

loan_amnt	0	loan_amnt	0.000000
funded_amnt	0	funded_amnt	0.000000
term	0	term	0.000000
int_rate	0	int_rate	0.000000
installment	0	installment	0.000000
grade	0	grade	0.000000
emp_title	29565	emp_title	9.515487
emp_length	22615	emp_length	7.278632
home_ownership	0	home_ownership	0.000000
annual_inc	0	annual_inc	0.000000
verification_status	0	verification_status	0.000000
purpose	0	purpose	0.000000
addr_state	0	addr_state	0.000000
dti	148	dti	0.047634
delinq_2yrs	0	delinq_2yrs	0.000000
revol_bal	0	revol_bal	0.000000
total_pymnt	0	total_pymnt	0.000000
loan_status	0	loan_status	0.000000
Payment Progress	0	Payment Progress	0.000000
dtype: int64		dtype: float64	

Figure 40 – Data Cleaning to check the Missing Values

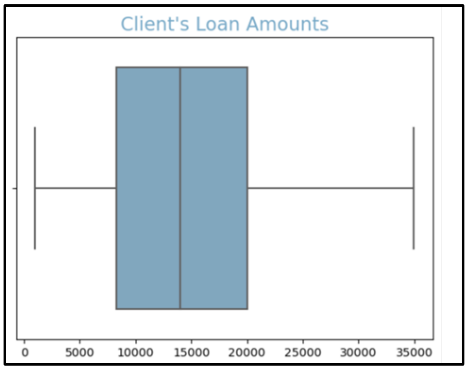

Figure 41 – Client's Loan Amounts without Outliers

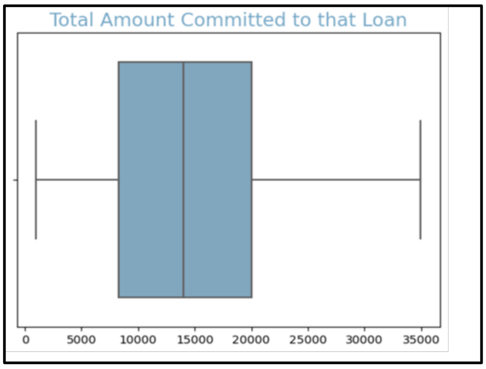

Figure 42 – Total Amount Committed to that Loan

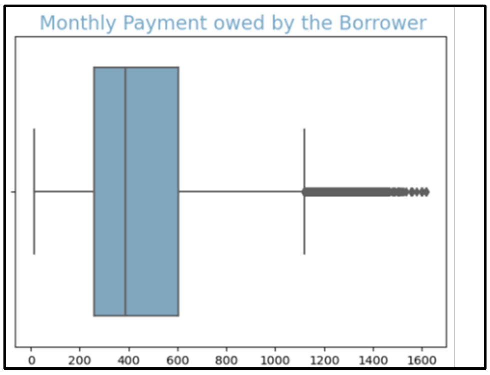

Figure 43 – Monthly Payment Owed by the Borrower

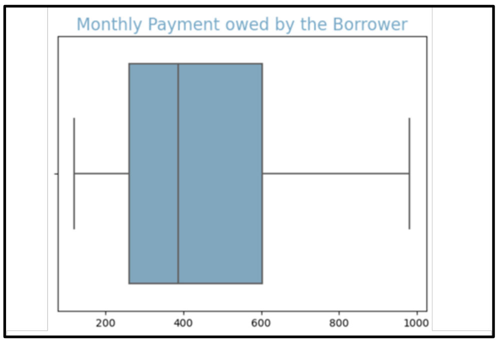

Figure 43.1 – Monthly Payment Owed by the Borrower

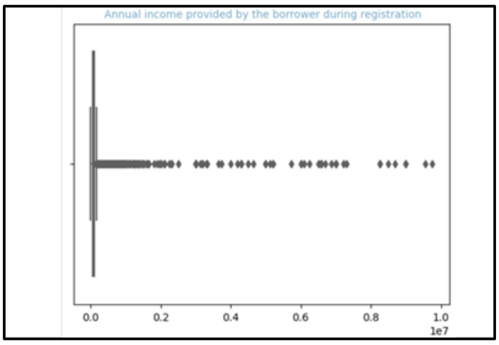

Figure 44 – Annual Income provided by the borrower during registration

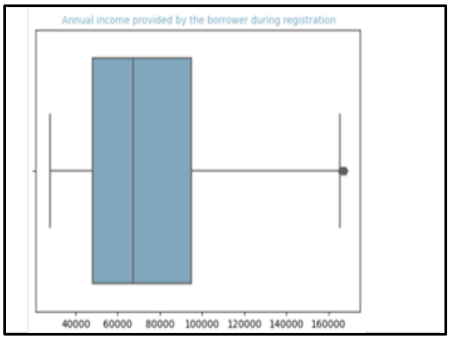

Figure 44.1 – Annual Income provided by the Borrower during Registration

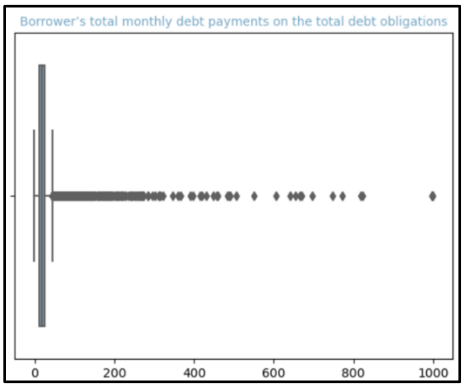

Figure 45 – Borrower's Tot. Monthly Debt Payments on the Tot. Debt Obligations

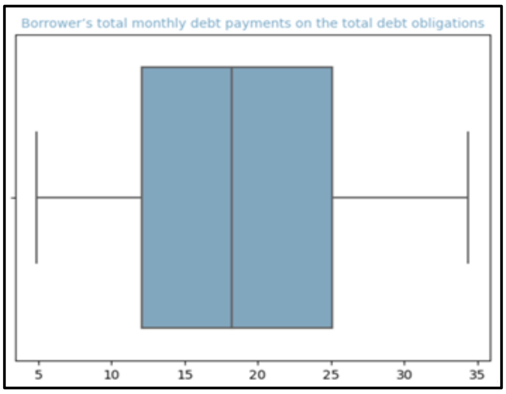

Figure 45.1 – Borrower's Tot. Monthly Debt Payments on the Tot. Debt Obligations

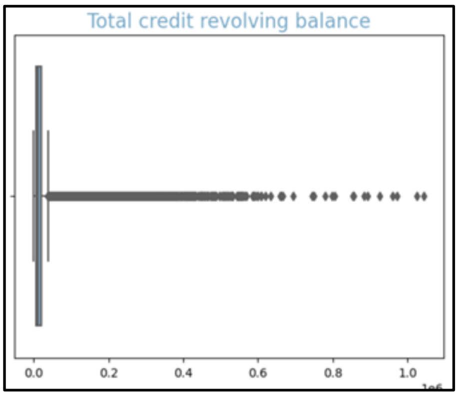

Figure 46 – Total Credit Revolving Balance

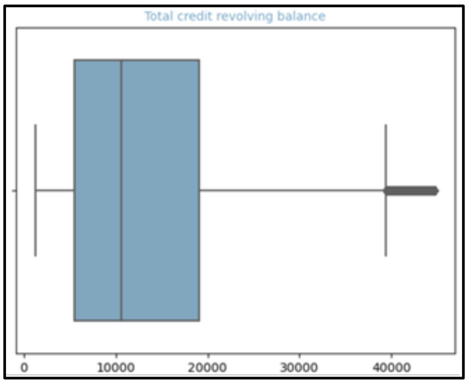

Figure 46.1 – Total Credit Revolving Balance

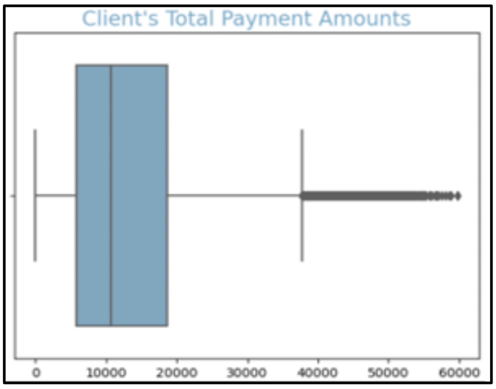

Figure 47 – Client's Total Payment Amounts

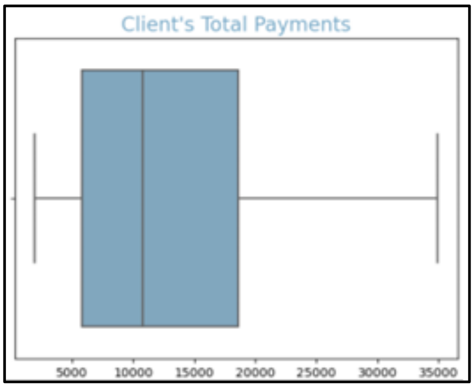

Figure 48 – Client's Total Payments without Outliers

```
term                   2.118542e-01
installment            5.866544e+04
grade                  1.601080e+00
emp_title              6.718425e+08
emp_length             6.947157e+00
home_ownership         8.757567e-01
annual_inc             1.410522e+09
verification_status    2.258270e-01
purpose                4.415369e+00
addr_state             2.153264e+02
dti                    7.136542e+01
delinq_2yrs            8.486201e-01
revol_bal              1.351944e+08
total_pymnt            8.689363e+07
loan_status            2.139456e-01
dtype: float64
```

Figure 49 – Univariate Variables

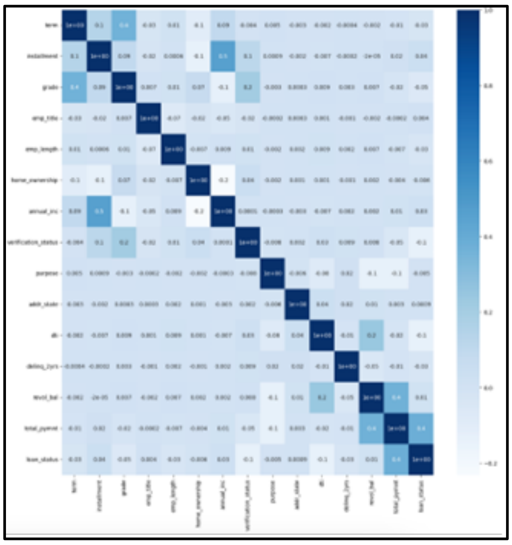

Figure 50 – Pearson & Spearman

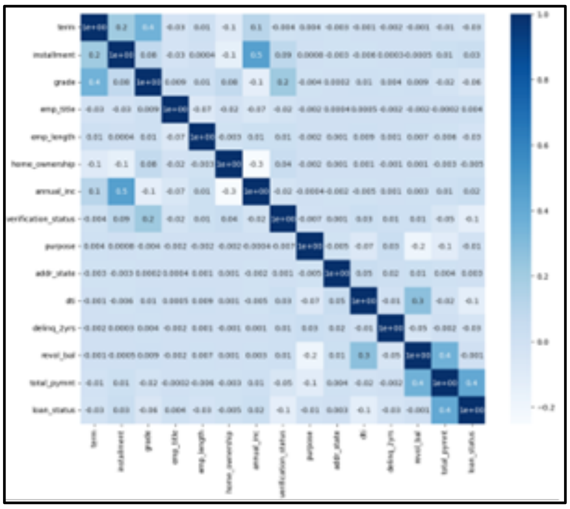

Figure 51 – Spearman Correlation

```
term                   False
installment             True
grade                   True
emp_title              False
emp_length             False
home_ownership         False
annual_inc             False
verification_status     True
purpose                False
addr_state             False
dti                     True
delinq_2yrs             True
revol_bal              False
total_pymnt             True
dtype: bool
```

Figure 52 – ANOVA Test

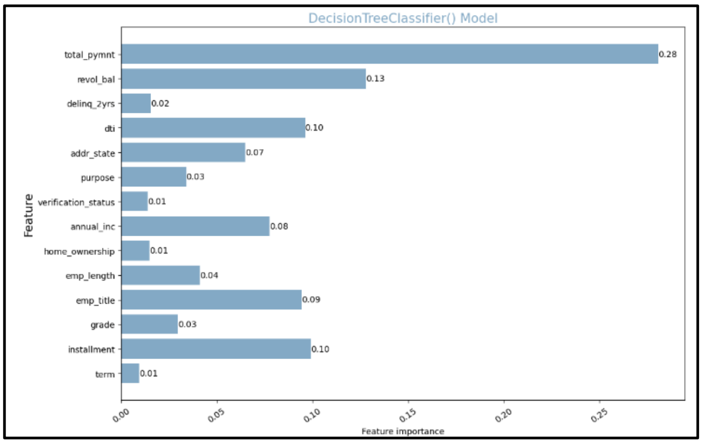

Figure 53 – Decision Tree Classifier Model

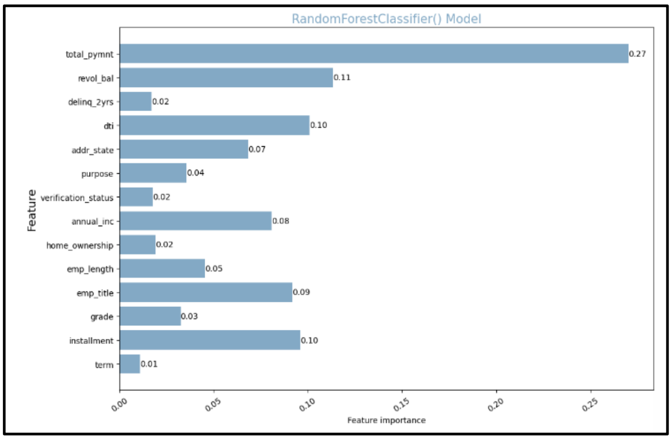

Figure 54 – Random Forest Classifier Model

```
====================== TRAIN ======================
              precision    recall  f1-score   support

           0       0.65      0.46      0.53     77266
           1       0.78      0.89      0.83    171297

    accuracy                           0.75    248563
   macro avg       0.71      0.67      0.68    248563
weighted avg       0.74      0.75      0.74    248563

==================== VALIDATION ====================
              precision    recall  f1-score   support

           0       0.63      0.44      0.52      9523
           1       0.78      0.89      0.83     21547

    accuracy                           0.75     31070
   macro avg       0.71      0.67      0.68     31070
weighted avg       0.74      0.75      0.74     31070
====================== RESULTS ======================
Train: 0.8320857839582027
Validation: 0.831846468514731
```

Figure 55 – Logistic Regression Model

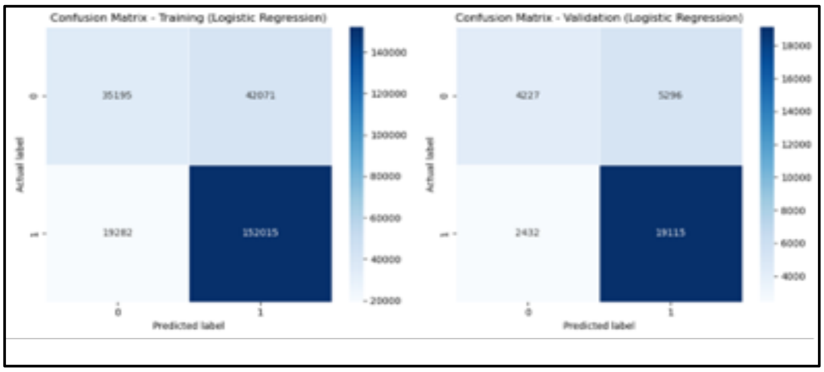

Figure 55.1 – Confusion Matrix LGM

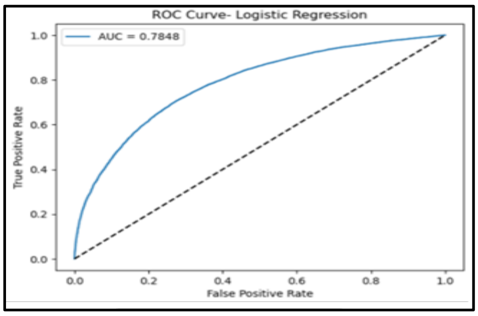

Figure 55.2 – ROC Curve LRM

```
----------------------- TRAIN ------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     77266
           1       1.00      1.00      1.00    171297

    accuracy                           1.00    248563
   macro avg       1.00      1.00      1.00    248563
weighted avg       1.00      1.00      1.00    248563

-------------------- VALIDATION ----------------------
              precision    recall  f1-score   support

           0       0.66      0.51      0.58      9523
           1       0.80      0.88      0.84     21547

    accuracy                           0.77     31070
   macro avg       0.73      0.70      0.71     31070
weighted avg       0.76      0.77      0.76     31070

---------------------- RESULTS ----------------------
Train: 0.9998591906277282
Validation: 0.7694238815577727
```

Figure 56 – Random Forest Model

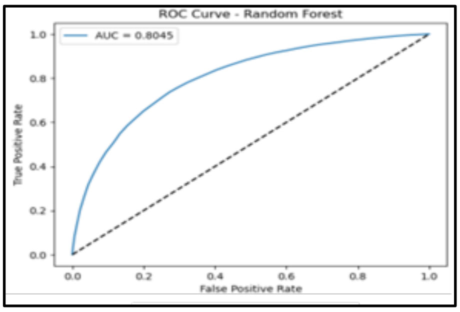

Figure 56.1- ROC Curve RFM

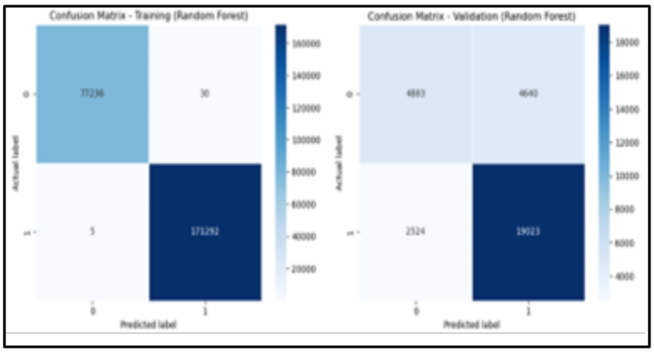

Figure 56.2 – Confusion Matrix RFM

```
----------------------- TRAIN ------------------------
              precision    recall  f1-score   support

           0       0.69      0.47      0.56     77266
           1       0.79      0.90      0.84    171297

    accuracy                           0.77    248563
   macro avg       0.74      0.69      0.70    248563
weighted avg       0.76      0.77      0.76    248563

-------------------- VALIDATION ----------------------
              precision    recall  f1-score   support

           0       0.68      0.47      0.56      9523
           1       0.79      0.90      0.85     21547

    accuracy                           0.77     31070
   macro avg       0.74      0.69      0.70     31070
weighted avg       0.76      0.77      0.76     31070

---------------------- RESULTS ----------------------
Train: 0.7703881913237288
Validation: 0.7711297071129707
```

Figure 57 – Gradient Boosting Classifier

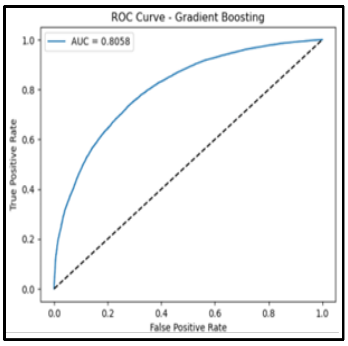

Figure 57.1 – ROC Curve GBC

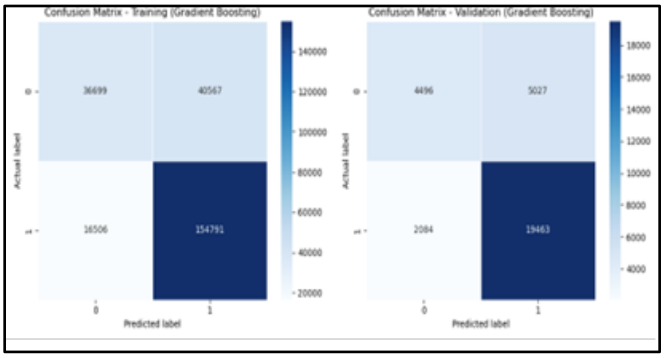

Figure 57.2 – Confusion Matrix GBC

```
----------------------- TRAIN ------------------------
              precision    recall  f1-score   support

           0       0.55      0.52      0.53     77266
           1       0.79      0.81      0.80    171297

    accuracy                           0.72    248563
   macro avg       0.67      0.66      0.67    248563
weighted avg       0.71      0.72      0.72    248563

-------------------- VALIDATION ----------------------
              precision    recall  f1-score   support

           0       0.54      0.51      0.53      9523
           1       0.79      0.81      0.80     21547

    accuracy                           0.72     31070
   macro avg       0.67      0.66      0.66     31070
weighted avg       0.71      0.72      0.72     31070

---------------------- RESULTS ----------------------
Train: 0.7182525154588575
Validation: 0.7178628902478275
```

Figure 58 – Naïve Bayes Algorithm

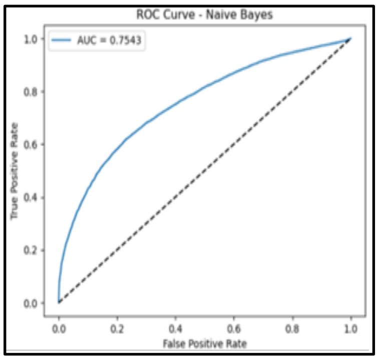

Figure 58.1 – ROC Curve NBA

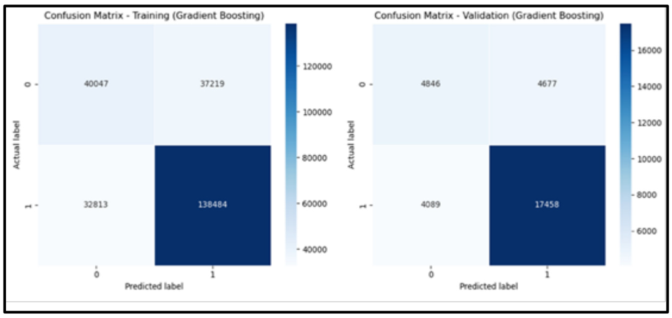

Figure 58.2 – Confusion Matrix GBC

```
Epoch 1/100
2486/2486 - 7s - loss: 0.4922 - accuracy: 0.7583 - val_loss: 0.485
3 - val_accuracy: 0.7640 - 7s/epoch - 3ms/step
Epoch 2/100
2486/2486 - 4s - loss: 0.4820 - accuracy: 0.7656 - val_loss: 0.481
0 - val_accuracy: 0.7654 - 4s/epoch - 1ms/step
Epoch 3/100
2486/2486 - 3s - loss: 0.4786 - accuracy: 0.7680 - val_loss: 0.479
4 - val_accuracy: 0.7666 - 3s/epoch - 1ms/step
Epoch 4/100
2486/2486 - 4s - loss: 0.4765 - accuracy: 0.7695 - val_loss: 0.478
3 - val_accuracy: 0.7686 - 4s/epoch - 2ms/step
Epoch 5/100
2486/2486 - 4s - loss: 0.4745 - accuracy: 0.7711 - val_loss: 0.476
1 - val_accuracy: 0.7691 - 4s/epoch - 1ms/step
```

Figure 59 – Deep Modelling – Accuracy and Loss

```
971/971 [==============================] - 1s 1ms/step - loss: 0.470
4 - accuracy: 0.7726

Test loss: 0.47. Test accuracy: 77.26%
```

Figure 60 – Accuracy and Loss Test

Algorithm	Confusion Matrix Predict Label	AUC value	F1 score
Logistic Regression	[14%,61%] [8%,17%]	0.7848	(0.53; 0.83)
Random Forest	[31%,69] [0%,0%]	0.8045	(1.00; 1.00)
Gradient Boosting Classifier (GBC)	[15%,62%] [7%,16%]	0.8058	(0.56; 0.84)
Naïve Bayes	[16%,56%] [13%,15%]	0.7543	(0.53; 0.80)

Table 1 - Evaluation results of the algorithm credit scoring

9. Bibliography

World Bank. (2019). *Credit Scoring Approaches Guidelines.* Washington: The World Bank

Siddiqi, Naeem. Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring. Copyright © 2005, SAS Institute Inc., Cary, North Carolina, USA

How to plot ROC curve in Python. 2023. *Stack overflow*. Available at: https://stackoverflow.com/questions/25009284/how-to-plot-roc-curve-in-python

tf.keras.callbacks.EarlyStopping. 2023. *TensorFlow*. Available at: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

How can I plot a confusion matrix? [duplicate]. 2016. *Stack overflow*. Available at: https://stackoverflow.com/questions/35572000/how-can-i-plot-a-confusion-matrix

Label Encoding in Python. *Geeks for geeks*. Available at: https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/

Research Methodology Advanced Tools. 2022. *How to Remove Outliers Using Python(outliers)(python)(PYTHON)(Boxplot)(Normality check)*. YouTube. Available at: https://www.youtube.com/watch?v=U1owDs-NrrI&t=646s

Winsorization. *Geeks for geeks*. 2021. Available at: https://www.geeksforgeeks.org/winsorization/

Birkett, Alex. (2023). "Outliers in Statistics: How to Find and Deal with Them in Your Data". Available at:https://cxl.com/blog/outliers/

Mousa, Waleed. (2023). "Feature Engineering". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7141312047220224000/?utm_source=share&utm_medium=member_android

Mousa, Waleed. (2023). "Linear Regression". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7140572955754835968/?utm_source=share&utm_medium=member_android

Aswin, Loga. (2023). "Means clustering". *Linkedin*. Available at:https://www.linkedin.com/feed/update/urn:li:activity:7138620180833738752/?utm_source=share&utm_medium=member_android

Su, Woongsik. (2023). "Pyton for Finance". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7138169593059012609/?utm_source=share&utm_medium=member_android

Chaskar, Prasad. (2023). “Diamond Price Prediction”. *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7137466218105499651/?utm_source=share&utm_medium=member_androidhttps://www.linkedin.com/feed/update/urn:li:activity:7137381364688908288/?utm_source=share&utm_medium=member_android

Mousa, Waleed. (2023). "Data Cleaning". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7137381142860587009/?utm_source=share&utm_medium=member_android

Shaikh, Vasim. (2023). "Univariate Analysis". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7136597230987182080/?utm_source=share&utm_medium=member_android

Rayguru, Saransh. (2023). "Exploratory Data analysis and Prepocessing Pipeline". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7135480567080386560/?utm_source=share&utm_medium=member_android

Sharma, Alok. (2023). "Pandas Cheatsheet". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7135590093695713280/?utm_source=share&utm_medium=member_android

Kemboi, David. (2023). "Data Cleaning". *Linkedin*. Available at: https://www.linkedin.com/feed/update/urn:li:activity:7136644673195905024/?utm_source=share&utm_medium=member_androidhttps://www.linkedin.com/feed/update/urn:li:activity:7141463583460110336/?utm_source=share&utm_medium=member_android

Yellowbrick: Machine Learning Visualization. *Yellowbrick*. 2019. Available at: https://www.scikit-yb.org/en/latest/

Split data into multiple columns. *Microsoft*. 2023. Available at: https://support.microsoft.com/en-us/office/split-data-into-multiple-columns-0dec75cd-4e83-4b39-81a5-9f604be95da0

Seaborn: Statistical Data Visualization. *Michael Waskom*. 2012-2023. Available at: seaborn: statistical data visualization — seaborn 0.13.0 documentation (pydata.org)

Histogram-based Gradient Boosting Classification Tree. *Scikit-Learn Developers.* 2007-2023. Available at: sklearn.ensemble.HistGradientBoostingClassifier — scikit-learn 1.3.2 documentation

Naive Bayes. *Scikit-Learn Developers.* 2007-2023. Available at: sklearn.ensemble.HistGradientBoostingClassifier — scikit-learn 1.3.2 documentation

A random forest classifier. *Scikit-Learn Developers*. 2007-2023. Available at: sklearn.ensemble.HistGradientBoostingClassifier — scikit-learn 1.3.2 documentation

Gradient Boosting for Classification. *Scikit-Learn Developers.* 2007-2023. Available at: sklearn.ensemble.HistGradientBoostingClassifier — scikit-learn 1.3.2 documentation

Logistic Regression (logit, MaxEnt) Classifier. *Scikit-Learn Developers.* 2007-2023. Available at: sklearn.ensemble.HistGradientBoostingClassifier — scikit-learn 1.3.2 documentation

Input/Output. NumFOCUS, Inc. 2023. Available at: Input/output — pandas 2.1.4 documentation (pydata.org)

Tranforming Skewed Data for Machine Learning. *Open Data Science.* 2023. Available at: opendatascience.com/transforming-skewed-data-for-machine-learning/

Top 5 Predictive Analytics Models and Algorithms. *InsightSoftweare*. 2023. Available at: Top 5 Predictive Analytics Models and Algorithms - insightsoftware

How to Perform ANOVA in Python. *Renesh Bedre*. 2023. Available at: How to Perform ANOVA in Python (reneshbedre.com)

Matplotlib Pyplot. *W3 Schools.* 1999-2023. Availabe«le at: Matplotlib Pyplot (w3schools.com)

A decision Tree Classifier. Scikit-Learn Developers. 2007-2023. Available at: sklearn.tree.DecisionTreeClassifier — scikit-learn 1.3.2 documentation