THE USAGE OF MACHINE LEARNING TECHNIQUES TO PREDICT THE PROBABLITY OF DEFAULT OF CREDIT CARD CLIENTS.

FINAL REPORT

G3-PGP DSE FT CHN CAPSTONE PROJECT – FINAL REPORT

BATCH DETAILS	PGPDSE-FT-CHENNAI SEP21
TEAM MEMBERS	1. GREASH K, 2. JISMY JOHN, 3. PRUTHIV RAJAN K, 4. SHIBIKANNAN TM, 5. VIGNESH PRABAKARAN
DOMAIN OF PROJECT	FINANCE
PROPOSED PROJECT TITLE	THE USAGE OF MACHINE LEARNING TECHNIQUES TO PREDICT THE PROBABLITY OF DEFAULT OF CREDIT CARD CLIENTS.
GROUP NUMBER	3
TEAM LEADER	SHIBIKANNAN TM
MENTOR NAME	P.V.SUBRAMANIAN

TEAM LEADER SIGN

MENTOR SIGN

P.V. Subramanian

TABLE OF CONTENTS

CHAPTER - 1

1.1 INDUSTRY REVIEW

1.1.1 CURRENT PRACTICES AND BACKGROUND RESEARCH:

1. **Title: Credit card industry analysis.**
   Link:
   https://corporatefinanceinstitute.com/resources/knowledge/credit/credit-card-industry-analysis
   Summary:
   Credit card industry analysis helps determine the current state of credit card companies and their latest products and services. The credit card industry relies on constant innovations in marketing and technology, which has resulted in increasing competition among credit card companies. Credit card companies invest billions of dollars in marketing activities to acquire new customers and expand their customer base. Innovative marketing activities, such as rewards programs, discounts, loyalty points and zero interest, are implemented with the aim of attracting more customers to the program.
2. **Title: Changes landscape of credit's card industry.**
   Link:
   https://www.pwc.in/industries/financial-services/fintech/dp/the-changing-landscape-of-indias-credit-industry.html
   Summary:
   India has historically been a debit card market. However, the boom in credit score card issuance within side the remaining decade has modified this narrative and credit score playing cards are getting used prominently. This boom is similarly extended via way of means of the numerous services and products being presented via way of means of FIs, and such merchandise are being an increasing number of utilized by customers, particularly the millennial population. Credit card issuance has grown considerably in India at a compound annual boom charge (CAGR) of 20% within side the remaining 4 years. The quantity of credit score cardholders multiplied from 29 million in March 2017 to sixty two million in March 2021. It has similarly grown via way of means of 26% and 23% respectively in 2019 and 2020. However, the COVID-19 pandemic affected the boom charge of India`s credit score card enterprise and it grew via way of means of most effective 7% in 2020–21. The boom charge is predicted to enhance marginally in FY21–22 however will continue to be gradual because of the regulations on card issuance via way of means of a few massive banks and bills networks. Similarly, credit score card transactions had been developing at a CAGR of 16% until 2019–20 however went again to the 2018–19 tiers in FY20–21, as depicted within side the determine above. The boom charge became low all through the primary 1/2 of of 2020–21 aleven though it received momentum all through the second one 1/2 of.

3. **Title: Machine-learning algorithms for credit-card applications.**
   Link:
   https://academic.oup.com/imaman/articleabstract/4/1/43/656001?redirectedFrom=PDF
   Summary:
   Credit checks include forecasting applicant credibility and profitability. The purpose of this paper is to apply a set of algorithms to credit card scoring. Little is known about the strengths and weaknesses of their comparisons, despite the fact that many numbers and connectionist learning algorithms address the same problem of learning from classified examples. An experiment comparing top-down guided learning algorithms (G & T and ID3) with perceptron, pocket, and back propagation multi-layer neural learning algorithms is an approved Scottish Bank credit card whose decision-making process is primarily credit. Implemented using a set of applications. Rating system. Overall, they all work with the same level of classification accuracy, but training the neural algorithm takes much longer. This white paper describes the motivations for using machine learning algorithms for credit card scoring, details the algorithms, and compares the performance of these algorithms in terms of accuracy.

1.1 LITERATURE SURVEY

1.1.1 PUBLICATIONS:

1. **Title: Credit scoring using the hybrid neural discriminant technique.**
Source:
Lee, T. S., Chiu, C. C., Lu, C. J., & Chen, I. F. (2002). Credit scoring using the hybrid neural discriminant technique. Expert Systems with Applications, 23(3), 245–254.
Summary:
Credit scoring has become a very important task as the credit industry has experienced double-digit growth over the last few decades. Artificial neural networks are becoming a very popular alternative to credit scoring models due to their associated memory properties and generalisability. However, decisions about network topology, the importance of potential input variables, and the lengthy training process have long been criticised, limiting their application to the handling of credit scoring issues. The purpose of the proposed study is to investigate the performance of credit scoring by integrating neural networks and back propagation into the traditional approach of discriminate analysis. Including the credit score results from the discriminate analysis simplifies the network structure and improves the accuracy of the credit score of the designed neural network model. Credit score against the bank's credit card dataset. The task will be executed. As the results show, the proposed hybrid approach converges much faster than traditional neural network models. In addition, the accuracy of the credit score associated with the proposed methodology has improved, which is superior to traditional discriminant analysis and logistic regression approaches.

2. **Title: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients.**
Source:
Lee, Y. S., Yen, S. J., Lin, C. H., Tseng, Y. N., Ma, L. Y. (2004). A data mining approach to constructing probability of default scoring model. In Proceedings of 10th conference on information management and implementation (pp. 1799–1813).
Summary:
This study covers the default cases of customer payments in Taiwan and compares the prediction accuracy of default probabilities among six data mining methods. From a risk management perspective, the default estimation probability prediction accuracy results are more valuable than the binary results that classify trusted or untrusted customers. Since the actual failure probability is unknown, this study presented a new "sort smoothing method" to estimate the actual failure probability. Using the actual failure probability (Y) as the response variable and the predicted failure probability (X) as the independent variable, the results of simple linear regression (Y = A + BX) show an artificially created prediction model. I am. The coefficient of determination of the neural network is the highest. Its regression intercept (A) is close to zero and its regression coefficient (B) is close to 1. Therefore, of the six data mining techniques, only artificial neural networks can accurately estimate the probability of actual failure.

4. **Title: Using neural network rule extraction and decision tables for credit-risk evaluation.**
   Source: Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. Management Science, 49(3), 312–329.
   Summary:
   Due to their universal approximation properties, neural networks have recently received considerable interest in the development of credit risk assessment models. However, most of this work is primarily focused on developing predictive networks without trying to explain how classification is done. In application domains such as credit risk assessment, it is imperative that credit risk managers have a compact and easy-to-understand set of rules. In this article, we evaluated and contrasted three neural network rule extraction methods (Neurorule, Trepan, and Nefclass) for credit risk assessment. The propositional rules derived from Neurorule were particularly concise and very easy to understand. We also explained how to represent the rules extracted using DT. DT displays rules in an intuitive graphic format that is easy for human experts to see. In addition, it enables simple and user-friendly advice in daily work. The rules and tree DTs extracted by Neurorule and Trepan have shown to be compact and powerful. In conclusion, extracting rules from neural networks and DTs is an effective and powerful management tool that can create sophisticated and user-friendly decision support systems for credit risk assessment. In addition, it would be interesting to apply the proposed approach to other interesting issues in business science. B. Forecast churn, customer retention and bankruptcy.

5. **Title: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research.**
   Source:
   Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 54(6), 627–635.
   Summary:
   Interest in predicting scorecard development is uninterrupted. However, there have been some advances, such as new learning methods, performance measurements, and techniques for reliably comparing different classifiers that are not considered in the credit assessment literature. To fill the gap in these studies, we will update the study by Baesens et al. Compare some new classification algorithms with the state of your credit score. In addition, we will investigate how the evaluation of alternative scorecards differs between established and new indicators of predictive accuracy. Finally, find out if a more accurate classifier makes sense for management. Our research provides professionals and scholars with valuable insights into credit scoring. This helps practitioners keep up with the technological advances in predictive modelling. From an academic point of view, this study provides an independent assessment of modern scoring methods and provides a new basis for comparing future approaches.

6. **Title: The art and science of customer relationship management.**
   Source:
   Berry, M., & Linoff, G. (2000). Mastering data mining: The art and science of customer relationship management. New York: John Wiley & Sons, Inc.
   Summary:
   They cover more advanced topics such as preparing data for analysis and creating the necessary infrastructure for data mining at your company. Features significant updates since the previous edition and updates you on best practices for using data mining methods and techniques for solving common business problems. Covers a new data mining technique in every chapter along with clear, concise explanations on how to apply each technique immediately. Touches on core data mining techniques, including decision trees, neural networks, collaborative filtering, association rules, link analysis, survival analysis, and more. Provides best practices for performing data mining using simple tools such as Excel.

1.1.2 APPLICATION PAST AND UNDERGOING RESEARCH:

6. **Title: Cash and credit card crisis in Taiwan.**
   Source: Chou, M. (2006). Cash and credit card crisis in Taiwan. Business Weekly, 24–27.
   Summary:
   In the past, banks issued credit and debit cards to students even if they were unemployed. As a result, many banks sent credit card vendors to college campuses, persuaded students to apply for credit and debit cards, and seduced them at low interest rates. Due to the lack of work experience and financial skills of these students, young people did not know how much interest they would have to pay by using credit and debit cards vigorously. The main function of AMC is to accept bad debts from banks. By taking on some of the bad debts, AMC helps manage the risk of financial activity and improve the bank's balance sheet. In 2000, the Taiwan Parliament passed the Financial Institution Merger Act. Under the law, foreign companies such as Lone Star, Merrill Lynch and Lehman Brothers are allowed to fund AMC. At the same time, many Taiwanese banks jointly funded several AMCs.

7. **Title: Analysis of financial credit risk using machine learning**
   Source:
   https://arxiv.org/ftp/arxiv/papers/1802/1802.05326.pdf
   Summary:
   Bankruptcy can have a devastating effect on the economy. Bankruptcy of multinational corporations can disrupt the global financial ecosystem as more companies expand abroad and harness foreign resources. Recent advances in communications and information technology have made it increasingly difficult to collect and store business-related data. Using published datasets, we applied a variety of machine learning techniques to determine the relationship between the company's current state and its near-future fate. The results show that predictions with an accuracy of over 95% can be achieved with any machine learning technique when using useful features such as expert scoring. However, the correlation is not very strong when using pure financial factors to predict whether a company will go bankrupt. More features are needed to better explain the data, but this is a higher dimension where data from thousands of public companies is not enough to fill this space with sufficient density. Leads to problems. Due to this "curse of dimensionality", flexible nonlinear models tend to overfit the training sample and therefore cannot be generalised to invisible data. For high-dimensional Polish bankruptcy datasets, simpler models such as logistic regression can predict a company's bankruptcy one year later with an accuracy of 66.4.

8. **Title: The Importance of Credit Risk Management in Banking**
   Source:
   https://blog.crifhighmark.com/the-importance-of-credit-risk-management-in-banking/#:~:text=They%20need%20to%20manage%20their,reserves%20at%20any%20given%20time
   Summary:
   Following were the advantages of credit card risk management learnt from the article. It helps in predicting and/ or measuring the risk factor of any transaction. t helps in planning ahead with strategies to tackle a negative outcome. It helps in setting up credit models which can act as a valuable tool to determine the level of risk while lending.

9. **Title: The importance of machine leaning in risk management.**
   Source:
   https://www.cqf.com/blog/importance-of-machine-learning-for-risk-management#:~:text=One%20prominent%20use%20case%20for,is%20fraud%20detection%20and%20prevention.&text=As%20in%20the%20case%20of,questionable%20behavior%20has%20been%20discerned
   Summary:
   For those seeking a broad and deep background in quant finance, from models and methods to machine learning in today's financial and economic environment, the CQF program is timely, flexible and high quality with a focus on wealth management and risk management. Provides professional training in data science and machine learning. For those on this journey, CQF provides the foundation for pursuing the most interesting and challenging opportunities in the current and future quantitative financial industry.

10. **Title: Credit Card Risk Assessment Based on Machine Learning.**
    Source:
    https://www.researchgate.net/publication/333871592_Credit_Card_Risk_Assessment_Based_on_Machine_Learning
    Summary:
    This article provides an example of a construction bench in Beijing. SSMTE is used to oversample the data. Handling outliers and missing values. And I standardized the variables. Make sure that the range of values for is within the same range of Finally, logistic regression and GridSearchCV regression are set up. The simulation also compares the search rates of logistic regression and GridSearchCV regression. The feasibility and effectiveness of the logistic algorithm is checked. determine whether the the bigger the number of positive samples is the better.

CHAPTER – 2

Domain: Finance

Aim :

This research aimed at the case of customers default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods.

2.1 DATA DICTIONARY:

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23 variables as explanatory variables:

X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.

X2: Gender (1 = male; 2 = female).

X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).

X4: Marital status (1 = married; 2 = single; 3 = others).

X5: Age (year).

X6 - X11: History of past payment. We tracked the past monthly payment records (from April to September, 2005) as follows:

X6: the repayment status in September, 2005.

X7: the repayment status in August, 2005;

X8: the repayment status in July , 2005;

X9: the repayment status in June, 2005;

X10: the repayment status in May, 2005;

X11:the repayment status in April, 2005.

The measurement scale for the repayment status is:

-1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; 3 = payment delay for three months. 4 = payment delay for four months; 5 = payment delay for five months; 6 = payment delay for six months; 7= payment delay for seven months ;8 = payment delay for eight months; 9 = payment delay for nine months and above.

X12-X17: Amount of bill statement (NT dollar).

X12: amount of bill statement in September, 2005;

X13: amount of bill statement in August, 2005;

X14: amount of bill statement in July, 2005;

X15: amount of bill statement in June, 2005;

X16: amount of bill statement in May, 2005;

X17: amount of bill statement in April, 2005;

X18-X23: Amount of previous payment (NT dollar).

X18: amount paid in September, 2005;

X19 :amount paid in August, 2005;

X20: amount paid in July, 2005;

X21: amount paid in June, 2005;

X22: amount paid in May, 2005;

X23: amount paid in April, 2005.

2.2 DATA SET INFORMATION:

This research aimed at the case of customers default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. Because the real probability of default is unknown, this study presented the novel Sorting Smoothing Method to estimate the real probability of default. With the real probability of default as the response variable (Y), and the predictive probability of default as the independent variable (X), the simple linear regression result (Y = A + BX) shows that the forecasting model produced by artificial neural network has the highest coefficient of determination; its regression intercept (A) is close to zero, and regression coefficient (B) to one. Therefore, among the six data mining techniques, artificial neural network is the only one that can accurately estimate the real probability of default.

2.3 VARIABLE CATEGORIZATION (COUNT OF NUMERIC AND CATEGORICAL):

A variable can be defined as something that is subject to change. There are two types of variables, namely:

- Numeric
- Categorical

Numeric variables:
The numbers of numeric features are: 14

X1,X5,X12,X13,X14,X15,X16,X17,X18,X19,X20,X21,X22,X23

Categorical variables:
The numbers of categorical features are: 9
The numerical features are: X2,X3,X4,X6,X7X8,X9,X10,X11

2.1PROJECT JUSTIFICATION

Aim :

This research aimed at the case of customers default payments in Taiwan and compares the predictive accuracy of probability of default among six data mining methods.

Complexity involved:

- Fixing outlier,
- understanding the nature of the clients,
- To avoid false clients detections.

Outcomes:

- This will be a business value problem to avoid the loss for the client.
- The significant factors affecting the failure of repayment of credit card payment is analyzed based on the dataset and is used to create a model to understand the probable number of defaulters.
- This will also help the card issuer to decrease the credit limit amount , hence reduce the credit risk in the coming months.

CHAPTER – 3

3.1 RELATIONSHIP BETWEEN VARIBALES:
UNDERSTANDING DATA BEFORE EDA:

UNIVARIATE ANALYSIS:

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn’t deal with causes or relationships (unlike regression) and it’s major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.

Education :

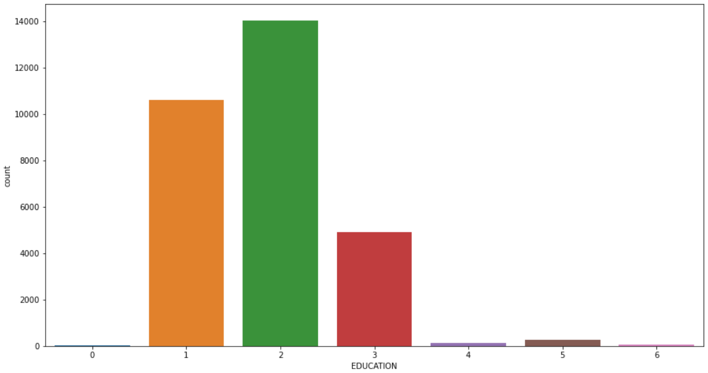


Fig 3.1. Bar graph of education

INFERENCE: According to the data descriptions it clearly stated that 1,2,3 are the representation of class, where as 0,5,6 are the noise in the data.

Marriage :

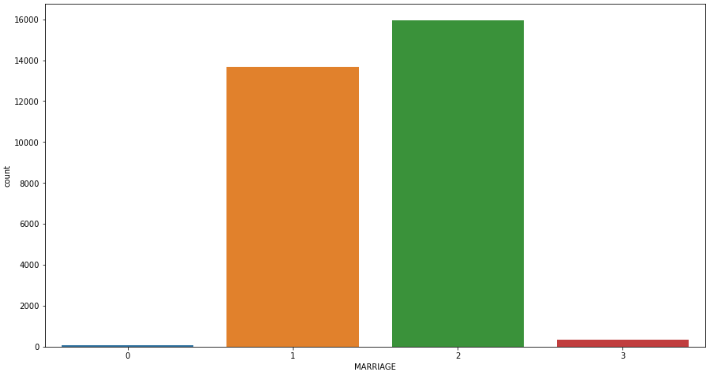


Fig 3.2. Bar graph of mariage

INFERENCE: According to the data descriptions it clearly stated that 1,2,3 are the representation of class, where as 0 is the noise in the data.

Bill amount (September - July) Vs default credit (target)

Bill amount 1 (September):

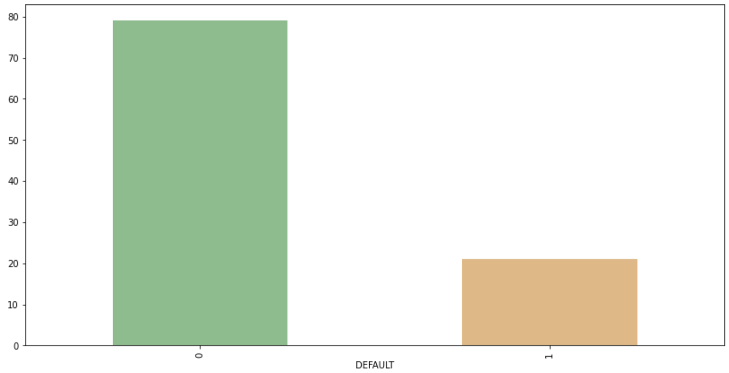


Fig 3.3. Bar graph of Bill amount 1 (September) vs default credit

Bill amount 2 (August):

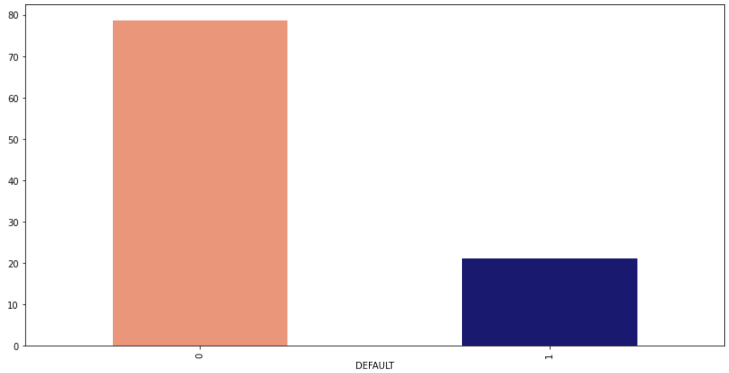


Fig 3.4. Bar graph of Bill amount 2 (August) vs default credit

Bill amount 3 (July):

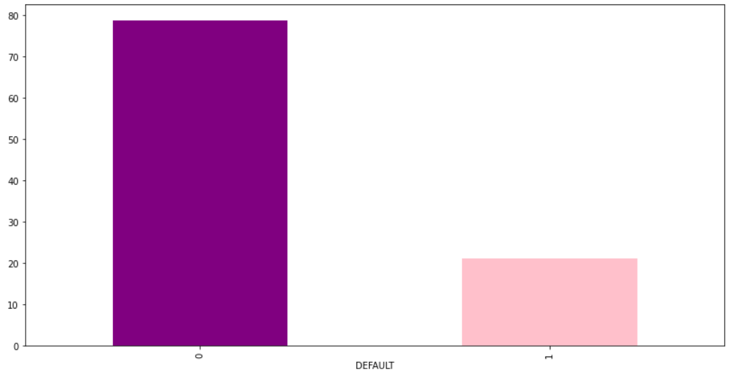


Fig 3.5. Bar graph of Bill amount 3 (July) vs default credit

Bill amount 4 (June):

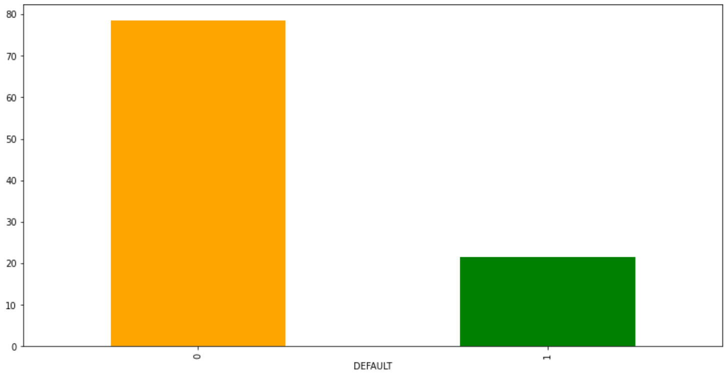


Fig3.6. Bar graph of Bill amount 4 (June) vs default credit

Bill amount 5 (May):

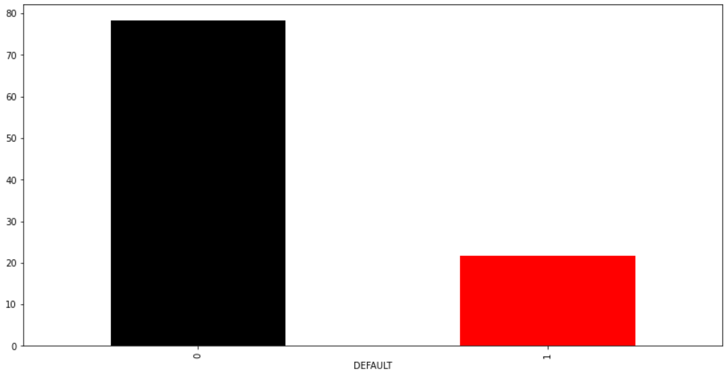


Fig3.7. Bar graph of Bill amount 5 (May) vs default credit

Bill amount 6 (April):

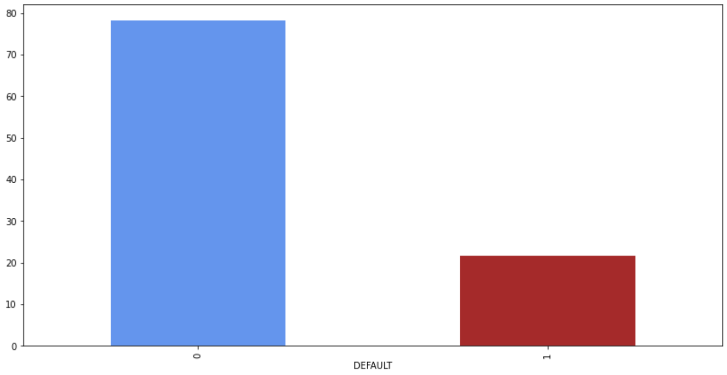


Fig3.8. Bar graph of Bill amount 6 (April) vs default credit

Inference: From the above figure 3 to 8 we could conclude that there is a regular pattern with target corresponding to bill amounts.

3.1.1 BIVARIATE ANALYSIS:

Bivariate analysis is one of the simplest forms of quantitative (statistical) analysis. It involves the analysis of two variables (often denoted as X, Y), for the purpose of determining the empirical relationship between them. ... It is the analysis of the relationship between the two variables.

Credit card Vs Sex :

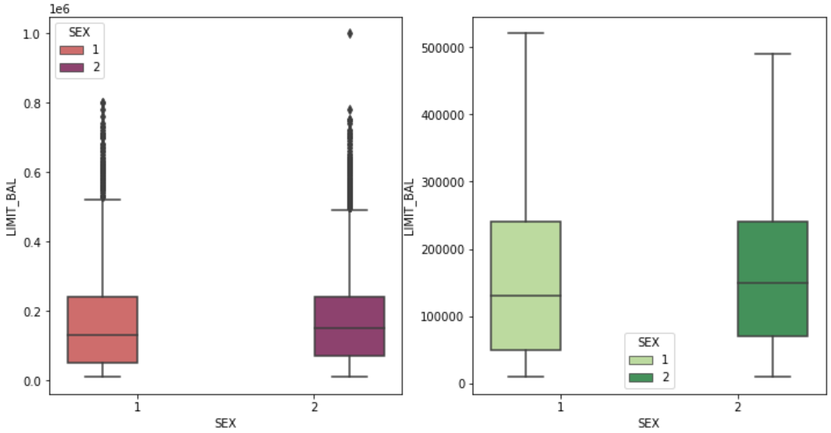


Fig3.9. Box plot of credit card vs sex

INFERENCE: According to box plot we conclude that there is order between them.

Age Vs marriage:

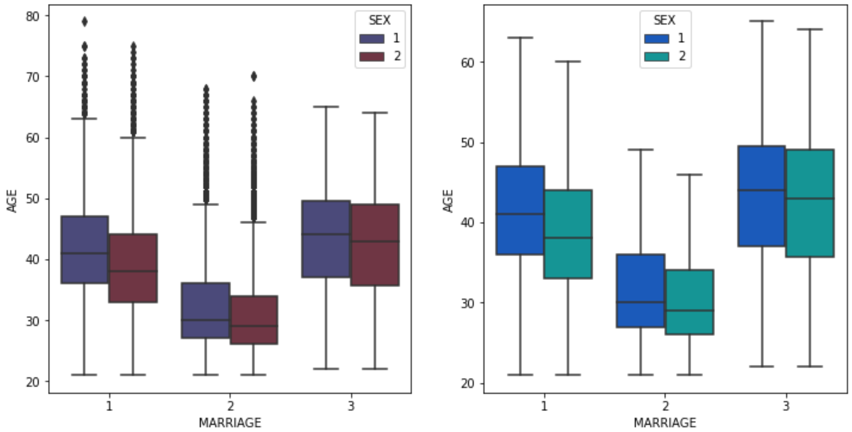


Fig3.10. Box plot of age vs marriage

INFERENCE: According to box plot we conclude that there is relationship between them.

Education Vs age:

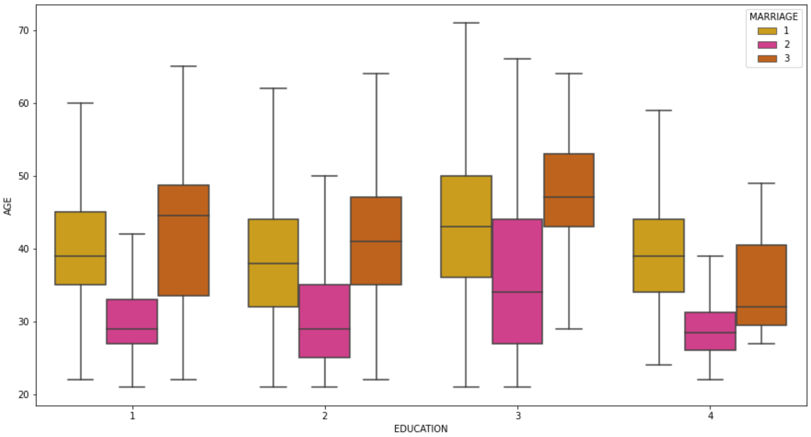


Fig3.11. Box plot of education vs age

INFERENCE: According to box plot we conclude that there is relationship between them.

Age Vs limit balance:

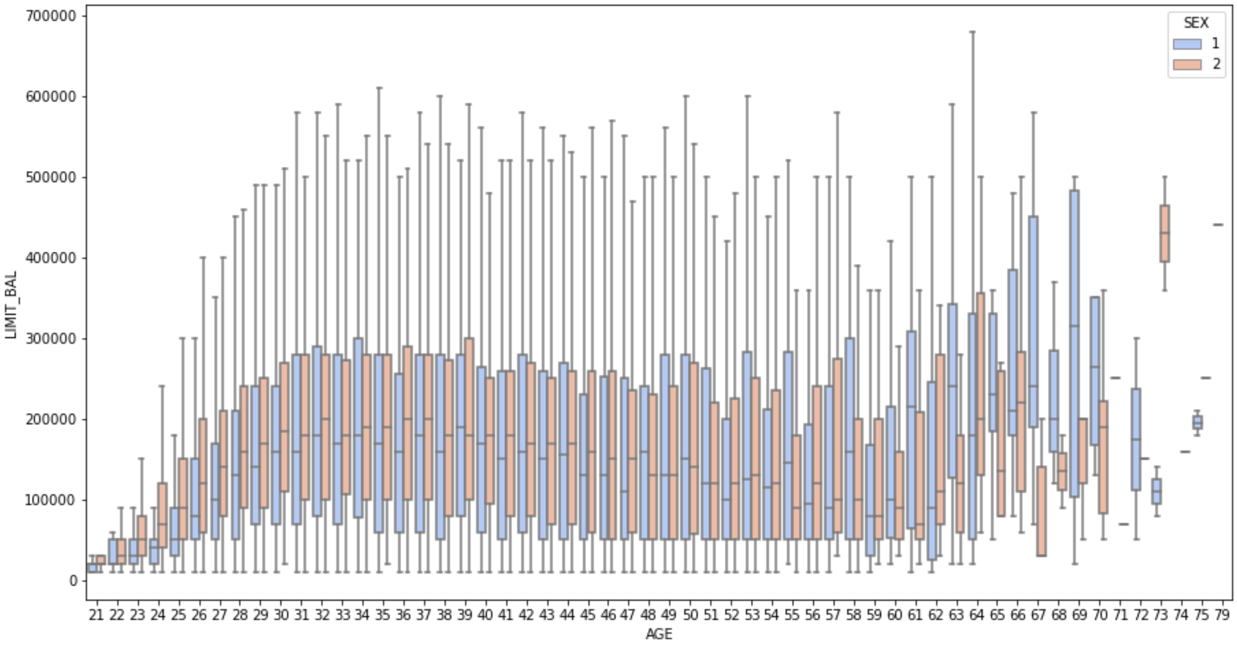


Fig3.12. Box plot of age vs limit balance.

INFERENCE: According to box plot we conclude that there is relationship between them.

Marriage with limit balance:

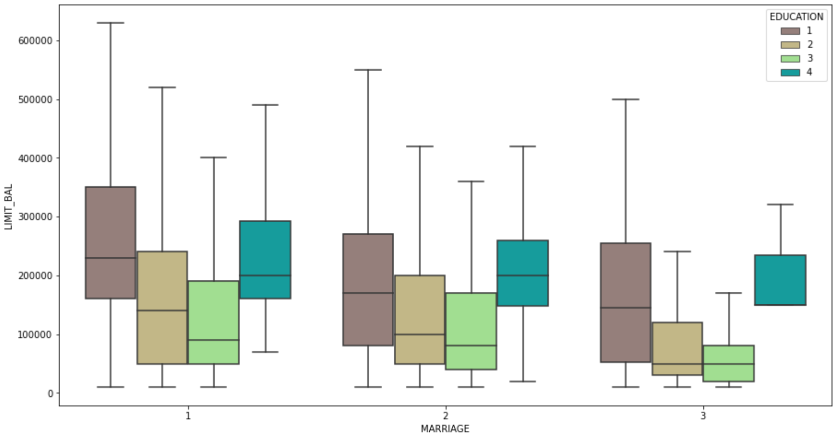


Fig3.13. Box plot of marriage vs limit balance.

INFERENCE: According to box plot we conclude that there is relationship between them.

Maximum limit of credit card limit amount:

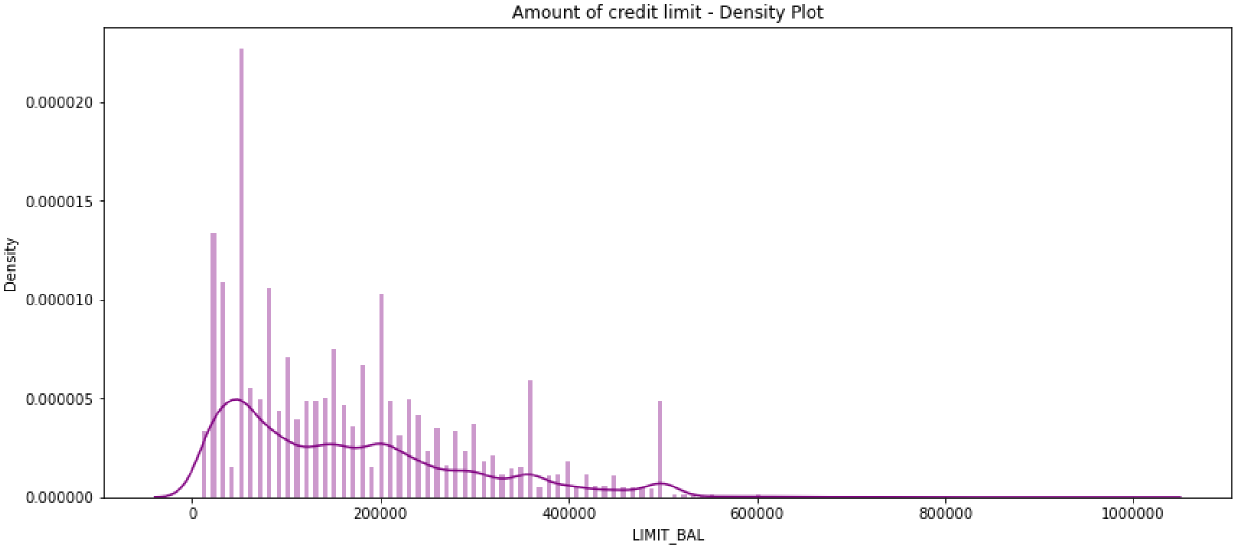


Fig3.14. Density plot of maximum limit of Credit card limit amount.

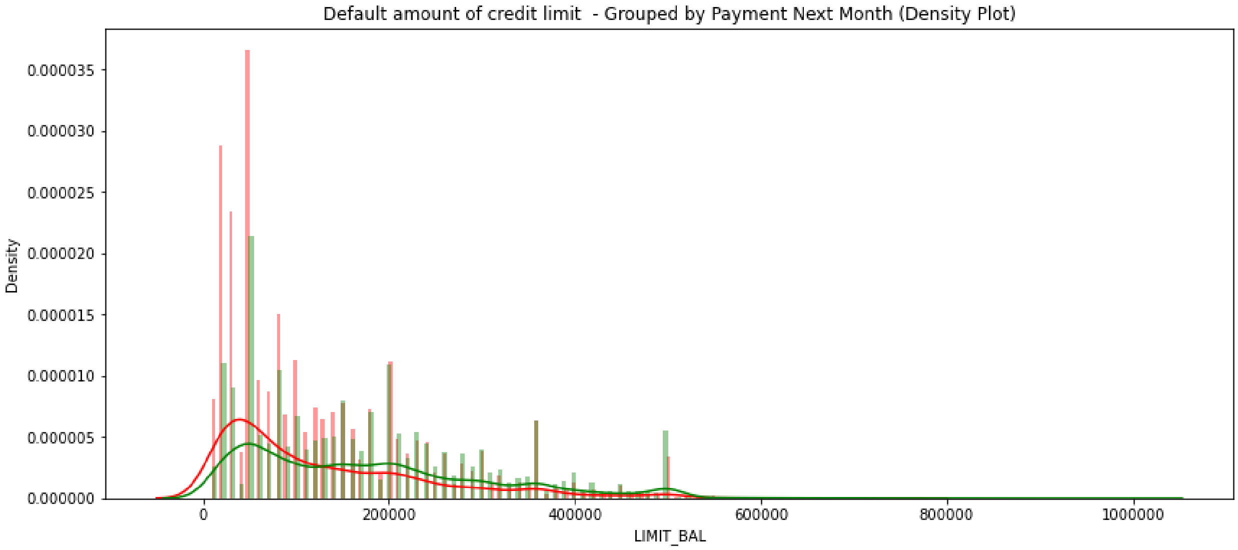


Fig3.15. Density plot of default amount of credit limit.

RELATIONSHIP BETWEEN INDEPENDENT AND TARGET VARIABLE:

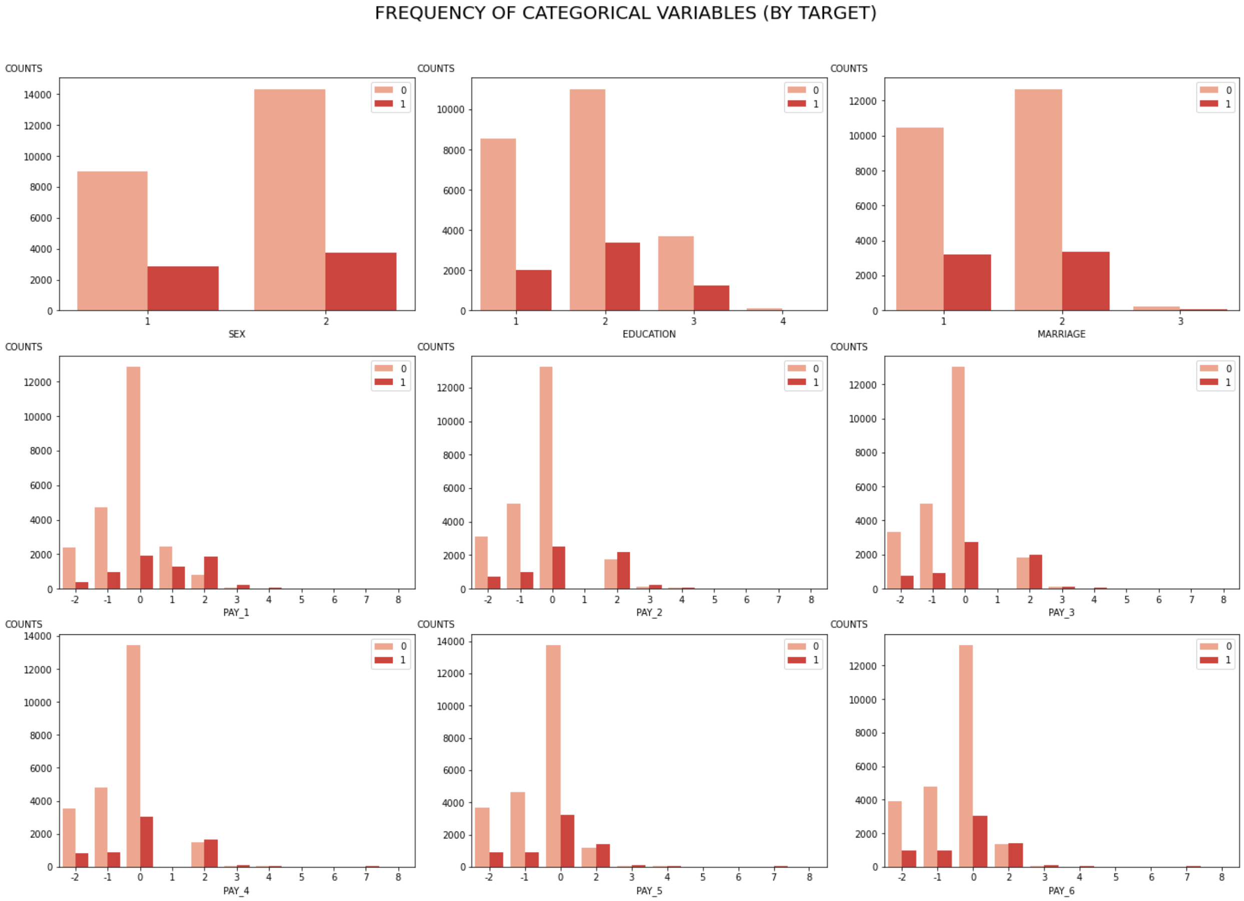


Fig3.16. Relationship between independent and target variable

Frequency distribution:

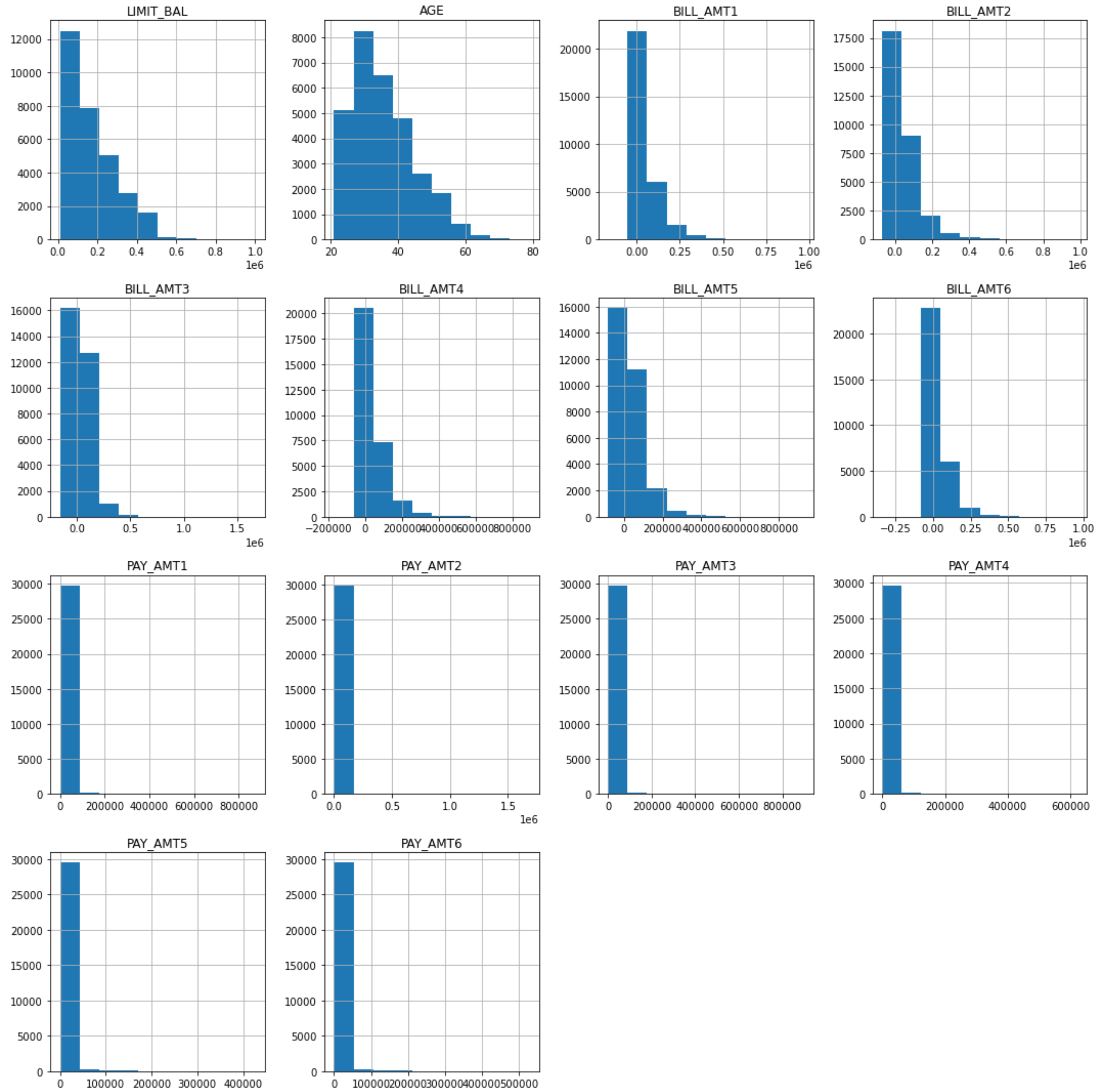
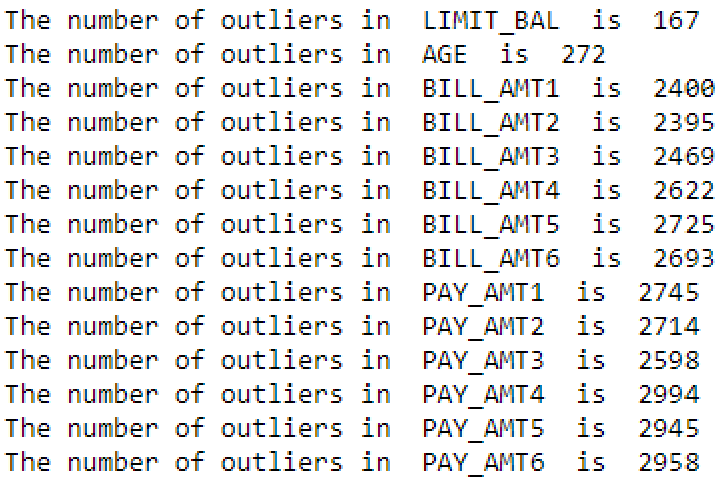


Fig3.17. Relationship between independent and target variable

3.2**OUTLIER:**

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal)objects. They can be caused by measurement or execution errors. ... There are many ways to detect the outliers, and the removal process is the data frame same as removing a data item from the panda's data frame.



```
The number of outliers in  LIMIT_BAL  is  167
The number of outliers in  AGE  is  272
The number of outliers in  BILL_AMT1  is  2400
The number of outliers in  BILL_AMT2  is  2395
The number of outliers in  BILL_AMT3  is  2469
The number of outliers in  BILL_AMT4  is  2622
The number of outliers in  BILL_AMT5  is  2725
The number of outliers in  BILL_AMT6  is  2693
The number of outliers in  PAY_AMT1  is  2745
The number of outliers in  PAY_AMT2  is  2714
The number of outliers in  PAY_AMT3  is  2598
The number of outliers in  PAY_AMT4  is  2994
The number of outliers in  PAY_AMT5  is  2945
The number of outliers in  PAY_AMT6  is  2958
```

Fig 18: Outlier

INFERENCE: Even though there are large number of outliers, we cannot treat the outliers as they are significant and according to the domain, it is possible to have outliers in the bill amount and payment amount.

3.3 STATISTICAL TESTING:

3.3.1.Chi-square contingency:

This function computes the chi-square statistic and p-value for the hypothesis test of independence of the observed frequencies in the contingency table [1] observed. The expected frequencies are computed based on the marginal sums under the assumption of independence

Hypothesis Formation:
Null Hypothesis (Ho): SEX and DEFAULT are independent
Alternate Hypothesis (Ha): SEX and DEFAULT are dependent

Statistical Significance of relationship between SEX and DEFAULT:
Test Statistics: 47.70879689062111
pValue: 4.944678999412044e-12
Degrees of freedom: 1

****

Hypothesis Formation:
Null Hypothesis (Ho): EDUCATION and DEFAULT are independent
Alternate Hypothesis (Ha): EDUCATION and DEFAULT are dependent

Statistical Significance of relationship between EDUCATION and DEFAULT:
Test Statistics: 109.30136242385805
pValue: 1.5512571274062487e-23
Degrees of freedom: 3

****

Hypothesis Formation:
Null Hypothesis (Ho): MARRIAGE and DEFAULT are independent
Alternate Hypothesis (Ha): MARRIAGE and DEFAULT are dependent

Statistical Significance of relationship between MARRIAGE and DEFAULT:
Test Statistics: 31.408475800840222
pValue: 1.5126419390778658e-07
Degrees of freedom: 2

****

Hypothesis Formation:
Null Hypothesis (Ho): PAY_1 and DEFAULT are independent
Alternate Hypothesis (Ha): PAY_1 and DEFAULT are dependent

```
Statistical Significance of relationship between PAY_1 and DEFAULT:
Test Statistics:  5365.964977413581
pValue:  0.0
Degrees of freedom:  10

**************************************************************************************

Hypothesis Formation:
Null Hypothesis (Ho): PAY_2 and DEFAULT are independent
Alternate Hypothesis (Ha): PAY_2 and DEFAULT are dependent

Statistical Significance of relationship between PAY_2 and DEFAULT:
Test Statistics:  3474.4667904168564
pValue:  0.0
Degrees of freedom:  10

**************************************************************************************

Hypothesis Formation:
Null Hypothesis (Ho): PAY_3 and DEFAULT are independent
Alternate Hypothesis (Ha): PAY_3 and DEFAULT are dependent

Statistical Significance of relationship between PAY_3 and DEFAULT:
Test Statistics:  2622.4621276828025
pValue:  0.0
Degrees of freedom:  10

**************************************************************************************

Hypothesis Formation:
Null Hypothesis (Ho): PAY_4 and DEFAULT are independent
Alternate Hypothesis (Ha): PAY_4 and DEFAULT are dependent

Statistical Significance of relationship between PAY_4 and DEFAULT:
Test Statistics:  2341.469945438205
pValue:  0.0
Degrees of freedom:  10

**************************************************************************************

Hypothesis Formation:
Null Hypothesis (Ho): PAY_5 and DEFAULT are independent
Alternate Hypothesis (Ha): PAY_5 and DEFAULT are dependent
```

Statistical Significance of relationship between PAY_5 and DEFAULT:
Test Statistics: 2197.694900930992
pValue: 0.0
Degrees of freedom: 9

****

Hypothesis Formation:
Null Hypothesis (Ho): PAY_6 and DEFAULT are independent
Alternate Hypothesis (Ha): PAY_6 and DEFAULT are dependent

Statistical Significance of relationship between PAY_6 and DEFAULT:
Test Statistics: 1886.835309001187
pValue: 0.0
Degrees of freedom: 9

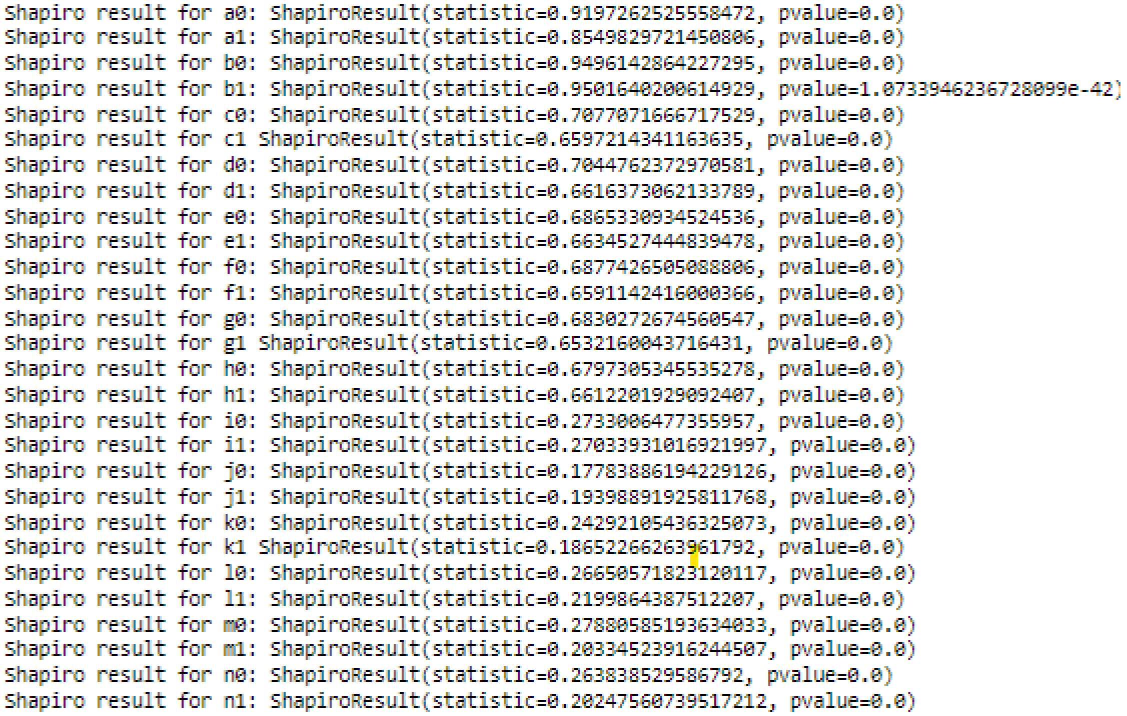
****

INFERENCE: From the results of statistical significance analysis of independent categorical variables with target using Chi-Square Test for Independence, we could see the pValue from all the statistical analysis is less than the significance level of 5% (0.05).

Hence Null hypothesis (Ho) is rejected and Alternate Hypothesis (Ha) can be selected. Thus, it is evident that all the independent categorical variables have significant relationship with the target variable.

3.3.2.ANOVA TEST:

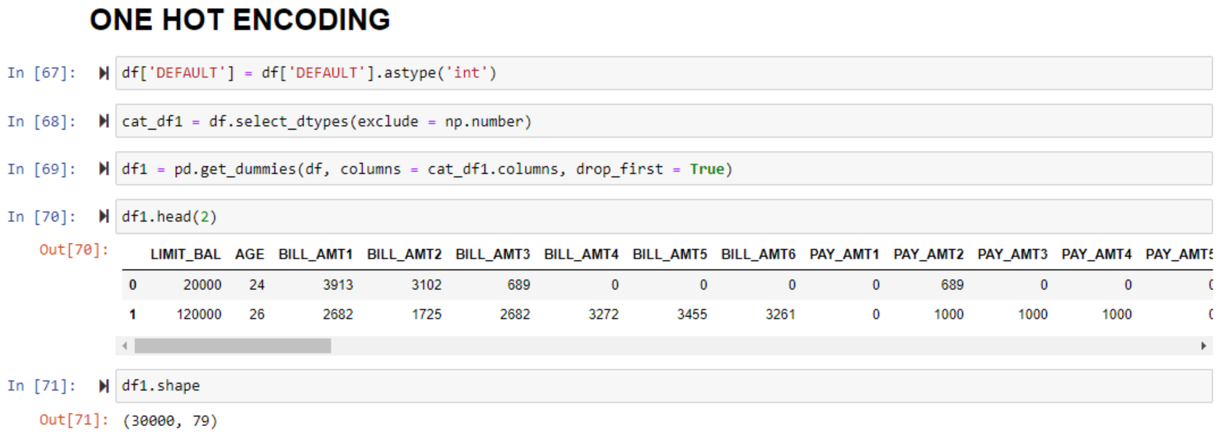
Analysis of variance, or ANOVA, is a statistical method that separates observed variance data into different components to use for additional tests. A one-way ANOVA is used for three or more groups of data, to gain information about the relationship between the dependent and independent variables.



```
Shapiro result for a0: ShapiroResult(statistic=0.9197262525558472, pvalue=0.0)
Shapiro result for a1: ShapiroResult(statistic=0.8549829721450806, pvalue=0.0)
Shapiro result for b0: ShapiroResult(statistic=0.9496142864227295, pvalue=0.0)
Shapiro result for b1: ShapiroResult(statistic=0.9501640200614929, pvalue=1.0733946236728099e-42)
Shapiro result for c0: ShapiroResult(statistic=0.7077071666717529, pvalue=0.0)
Shapiro result for c1 ShapiroResult(statistic=0.6597214341163635, pvalue=0.0)
Shapiro result for d0: ShapiroResult(statistic=0.7044762372970581, pvalue=0.0)
Shapiro result for d1: ShapiroResult(statistic=0.6616373062133789, pvalue=0.0)
Shapiro result for e0: ShapiroResult(statistic=0.6865330934524536, pvalue=0.0)
Shapiro result for e1: ShapiroResult(statistic=0.6634527444839478, pvalue=0.0)
Shapiro result for f0: ShapiroResult(statistic=0.6877426505088806, pvalue=0.0)
Shapiro result for f1: ShapiroResult(statistic=0.6591142416000366, pvalue=0.0)
Shapiro result for g0: ShapiroResult(statistic=0.6830272674560547, pvalue=0.0)
Shapiro result for g1 ShapiroResult(statistic=0.6532160043716431, pvalue=0.0)
Shapiro result for h0: ShapiroResult(statistic=0.6797305345535278, pvalue=0.0)
Shapiro result for h1: ShapiroResult(statistic=0.6612201929092407, pvalue=0.0)
Shapiro result for i0: ShapiroResult(statistic=0.2733006477355957, pvalue=0.0)
Shapiro result for i1: ShapiroResult(statistic=0.27033931016921997, pvalue=0.0)
Shapiro result for j0: ShapiroResult(statistic=0.17783886194229126, pvalue=0.0)
Shapiro result for j1: ShapiroResult(statistic=0.19398891925811768, pvalue=0.0)
Shapiro result for k0: ShapiroResult(statistic=0.24292105436325073, pvalue=0.0)
Shapiro result for k1 ShapiroResult(statistic=0.18652266263961792, pvalue=0.0)
Shapiro result for l0: ShapiroResult(statistic=0.26650571823120117, pvalue=0.0)
Shapiro result for l1: ShapiroResult(statistic=0.2199864387512207, pvalue=0.0)
Shapiro result for m0: ShapiroResult(statistic=0.27880585193634033, pvalue=0.0)
Shapiro result for m1: ShapiroResult(statistic=0.20334523916244507, pvalue=0.0)
Shapiro result for n0: ShapiroResult(statistic=0.263838529586792, pvalue=0.0)
Shapiro result for n1: ShapiroResult(statistic=0.20247560739517212, pvalue=0.0)
```

INFRENCE: pValue of Shapiro Result for scores of different adverse effects < 0.05 (sig. lvl). Hence, Ho is rejected and so data is not normal .

3.4.ENCODING:



ONE HOT ENCODING

```
In [67]:  df['DEFAULT'] = df['DEFAULT'].astype('int')

In [68]:  cat_df1 = df.select_dtypes(exclude = np.number)

In [69]:  df1 = pd.get_dummies(df, columns = cat_df1.columns, drop_first = True)

In [70]:  df1.head(2)
Out[70]:
```

	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5
0	20000	24	3913	3102	689	0	0	0	0	689	0	0	0
1	120000	26	2682	1725	2682	3272	3455	3261	0	1000	1000	1000	0

```
In [71]:  df1.shape
Out[71]:  (30000, 79)
```

CHAPTER – 4

4.1 FEATURE SELECTION:

Feature correlation:

Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling.

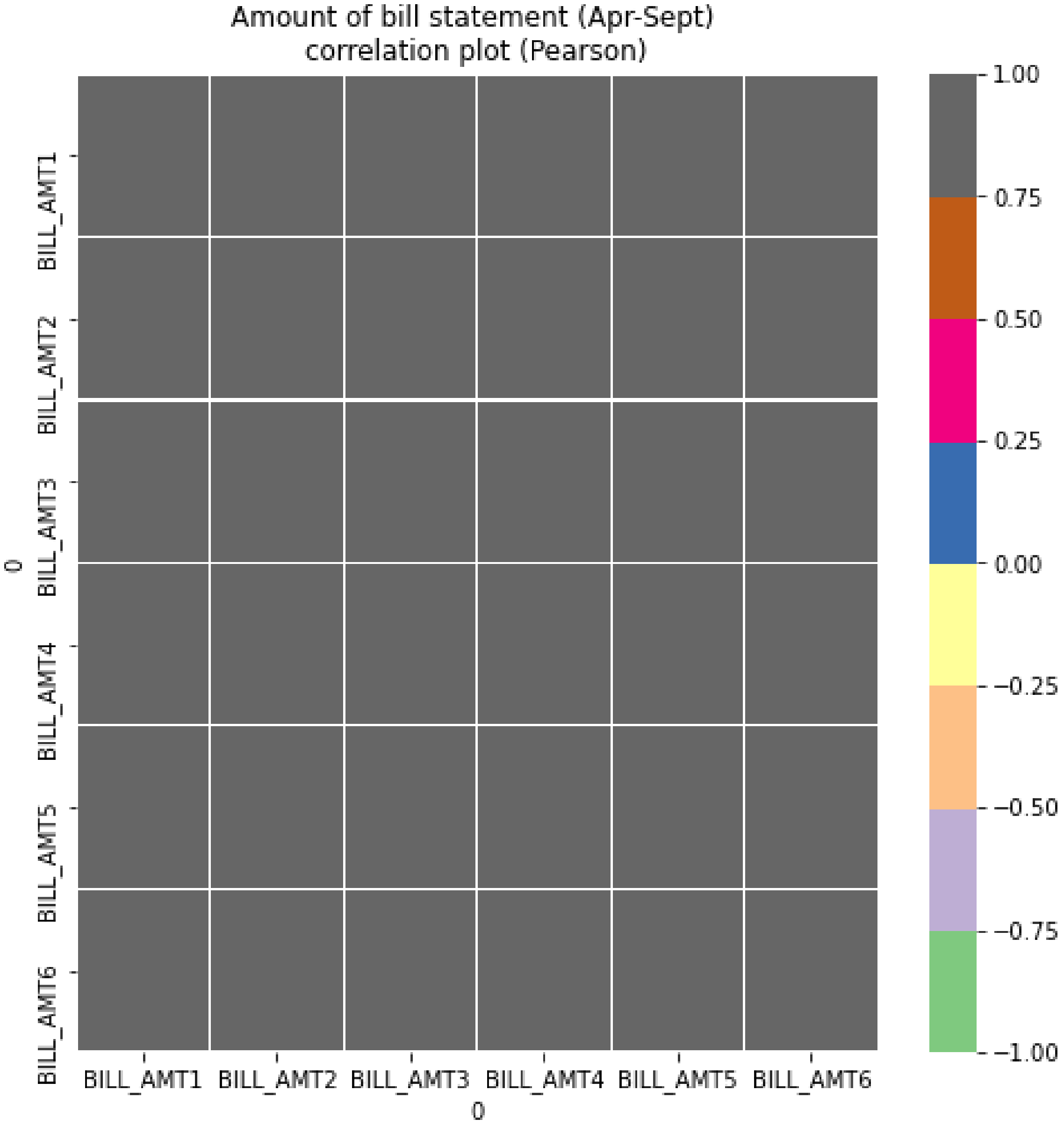


Fig 18: Heatmap of correlation plot

INFFRENCE: Highly correlated

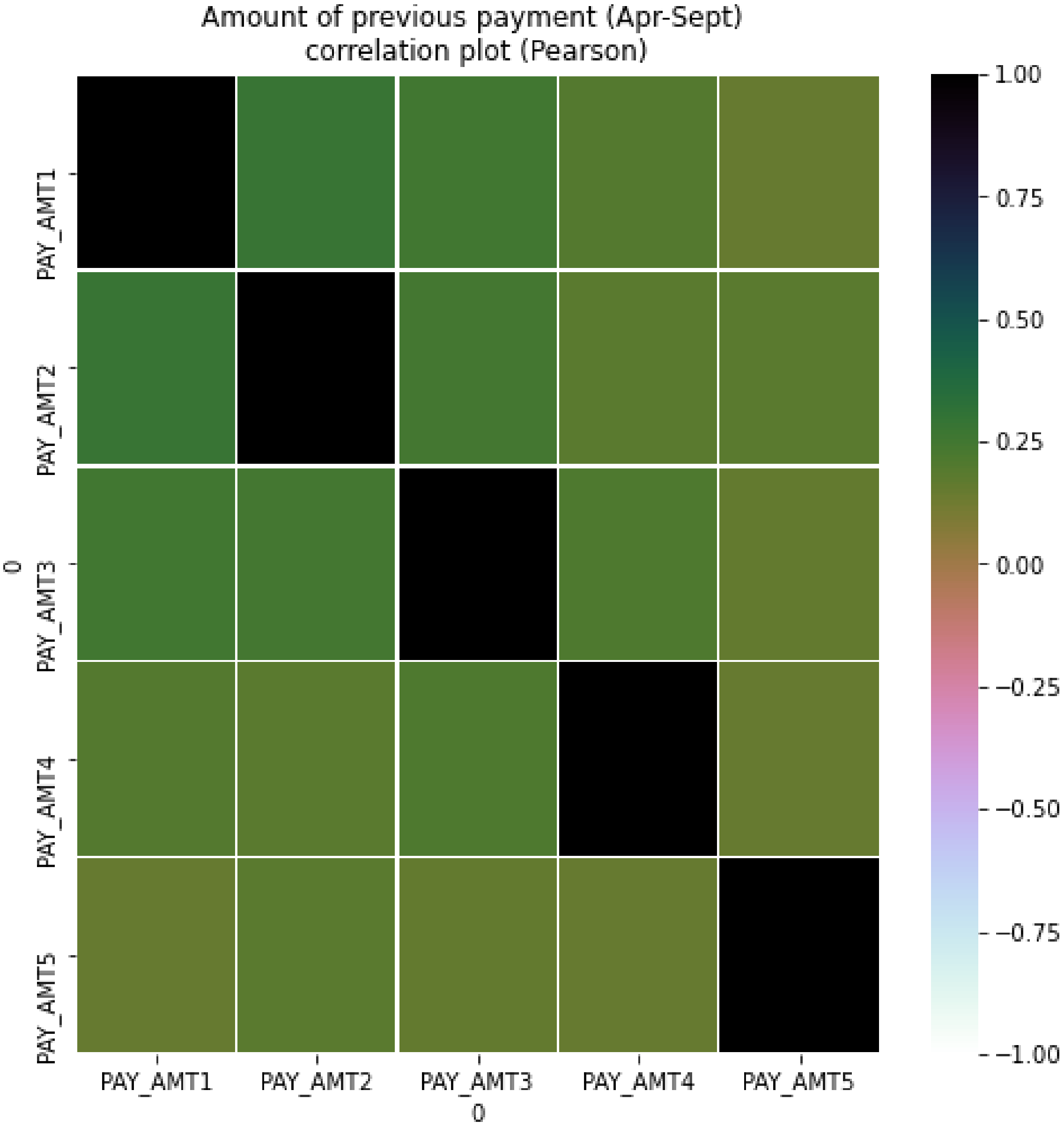
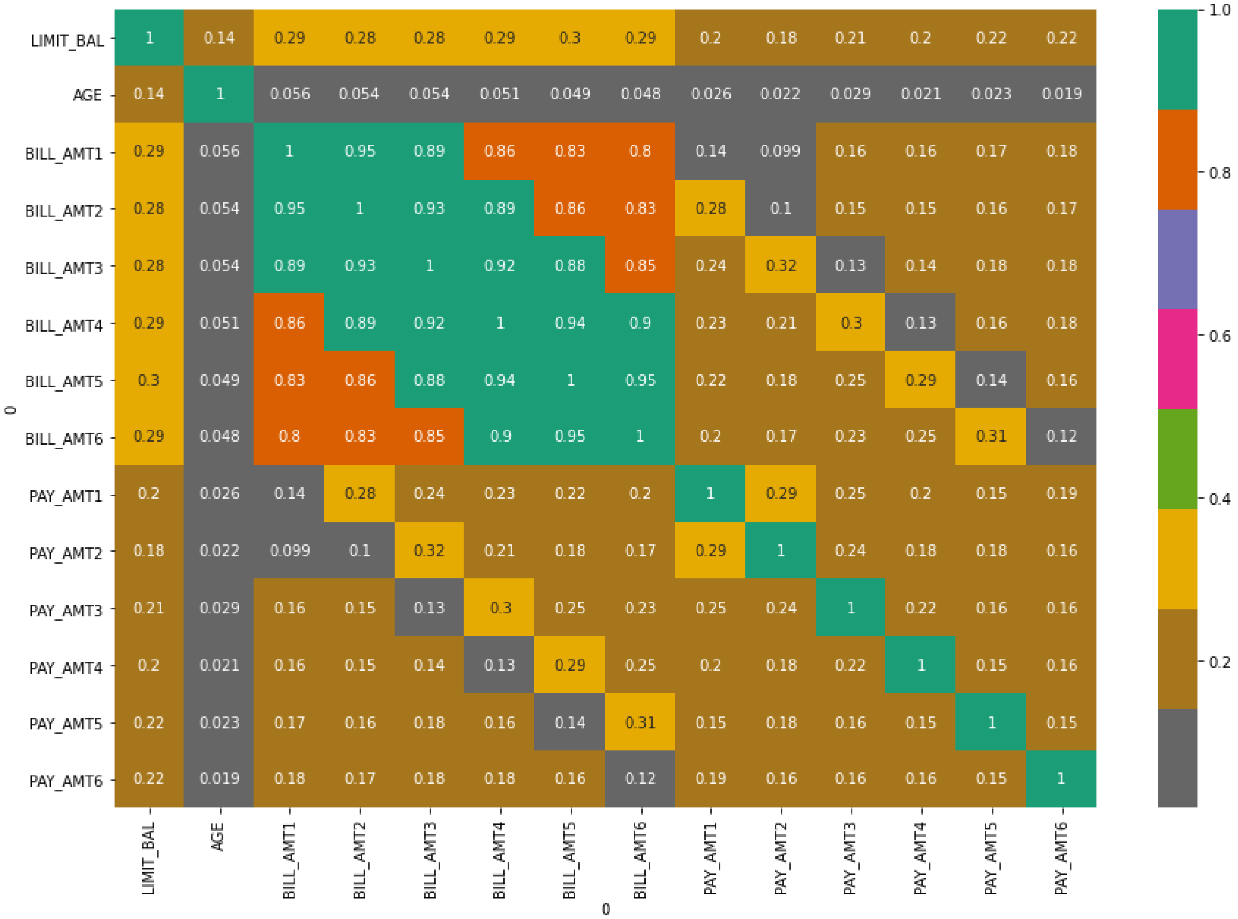


Fig 19: Heatmap of correlation plot - pearson



	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6
LIMIT_BAL	1	0.14	0.29	0.28	0.28	0.29	0.3	0.29	0.2	0.18	0.21	0.2	0.22	0.22
AGE	0.14	1	0.056	0.054	0.054	0.051	0.049	0.048	0.026	0.022	0.029	0.021	0.023	0.019
BILL_AMT1	0.29	0.056	1	0.95	0.89	0.86	0.83	0.8	0.14	0.099	0.16	0.16	0.17	0.18
BILL_AMT2	0.28	0.054	0.95	1	0.93	0.89	0.86	0.83	0.28	0.1	0.15	0.15	0.16	0.17
BILL_AMT3	0.28	0.054	0.89	0.93	1	0.92	0.88	0.85	0.24	0.32	0.13	0.14	0.18	0.18
BILL_AMT4	0.29	0.051	0.86	0.89	0.92	1	0.94	0.9	0.23	0.21	0.3	0.13	0.16	0.18
BILL_AMT5	0.3	0.049	0.83	0.86	0.88	0.94	1	0.95	0.22	0.18	0.25	0.29	0.14	0.16
BILL_AMT6	0.29	0.048	0.8	0.83	0.85	0.9	0.95	1	0.2	0.17	0.23	0.25	0.31	0.12
PAY_AMT1	0.2	0.026	0.14	0.28	0.24	0.23	0.22	0.2	1	0.29	0.25	0.2	0.15	0.19
PAY_AMT2	0.18	0.022	0.099	0.1	0.32	0.21	0.18	0.17	0.29	1	0.24	0.18	0.18	0.16
PAY_AMT3	0.21	0.029	0.16	0.15	0.13	0.3	0.25	0.23	0.25	0.24	1	0.22	0.16	0.16
PAY_AMT4	0.2	0.021	0.16	0.15	0.14	0.13	0.29	0.25	0.2	0.18	0.22	1	0.15	0.16
PAY_AMT5	0.22	0.023	0.17	0.16	0.18	0.16	0.14	0.31	0.15	0.18	0.16	0.15	1	0.15
PAY_AMT6	0.22	0.019	0.18	0.17	0.18	0.18	0.16	0.12	0.19	0.16	0.16	0.16	0.15	1

Fig 20: Heatmap of correlation plot.

4.2.Class imbalance:

When observation in one class is higher than the observation in other classes then there exists a class imbalance.

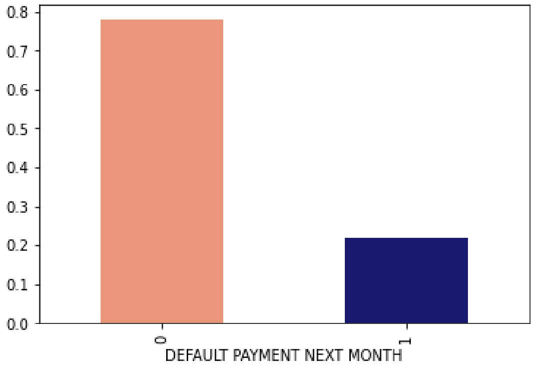
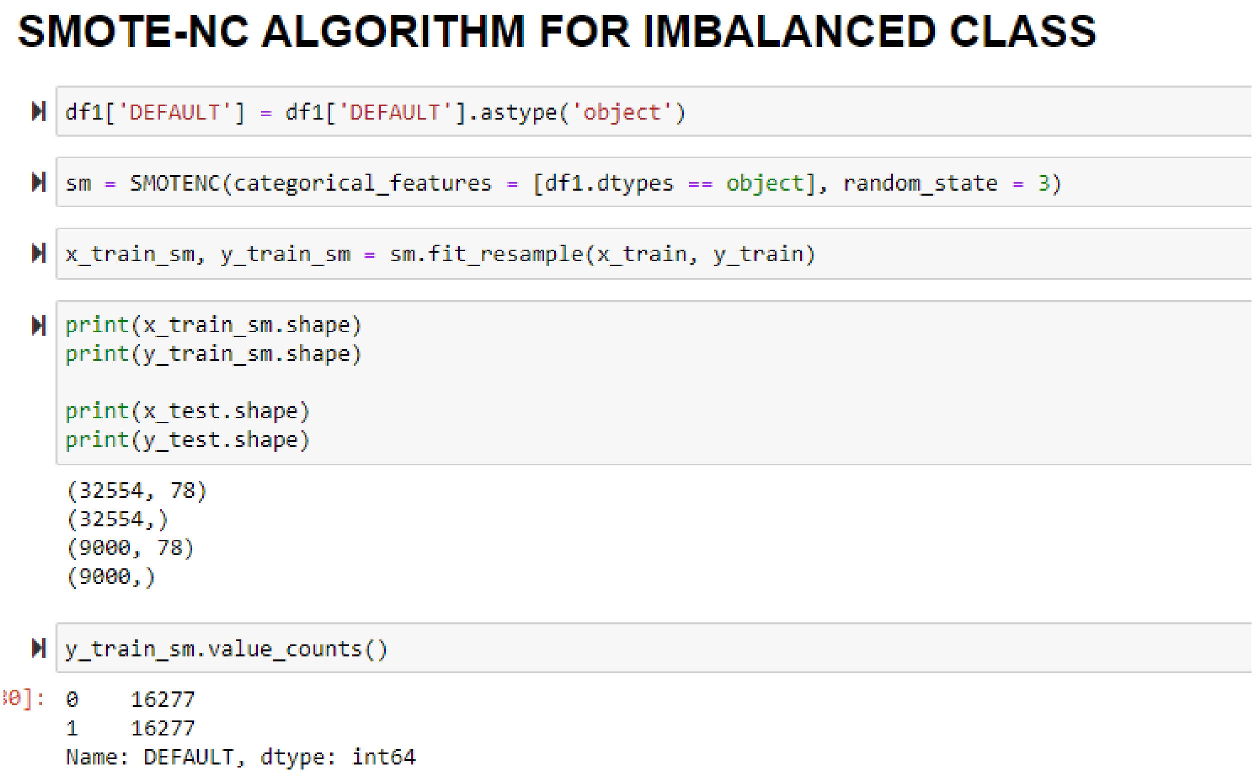


Fig 21: Barplot of target..

4.3.SMOTE:

SMOTE-NC works only when data is a mixed of numerical and categorical features. However, it is not designed to work with only categorical features. The parameters that can be tuned are k-neighbours, which allow to determine the number of nearest neighbours to create the new sample, and sampling strategy, which allows to indicate how many new samples to create.

It is important to remember to only apply it to the training dataset in order to avoid introducing bias to the model. Since our data consists of both numerical and categorical features SMOTE-NC is used instead of SMOTE and SMOTE-N. In SMOTE-NC, the median of standard deviations of all continuous features of the minority class are calculated. When new data points are about to be created, if the nominal features differ when compared with the nearest neighbours, this median is added to the Euclidean distances. The mechanism penalizes differences in nominal features more effectively, which should in theory result in more accurate synthetic nominal features. In our dataset, the class imbalance is as follows:



SMOTE-NC ALGORITHM FOR IMBALANCED CLASS

```
df1['DEFAULT'] = df1['DEFAULT'].astype('object')
```

```
sm = SMOTENC(categorical_features = [df1.dtypes == object], random_state = 3)
```

```
x_train_sm, y_train_sm = sm.fit_resample(x_train, y_train)
```

```
print(x_train_sm.shape)
print(y_train_sm.shape)

print(x_test.shape)
print(y_test.shape)
```

```
(32554, 78)
(32554,)
(9000, 78)
(9000,)
```

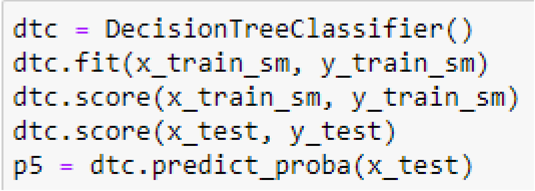
```
y_train_sm.value_counts()
```

```
0    16277
1    16277
Name: DEFAULT, dtype: int64
```

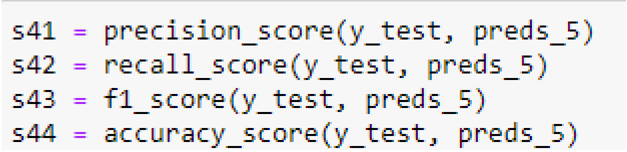
CHAPTER – 5

5.1.BASE LINE MODEL BUILDING- DECISION TREE CLASSIFIER

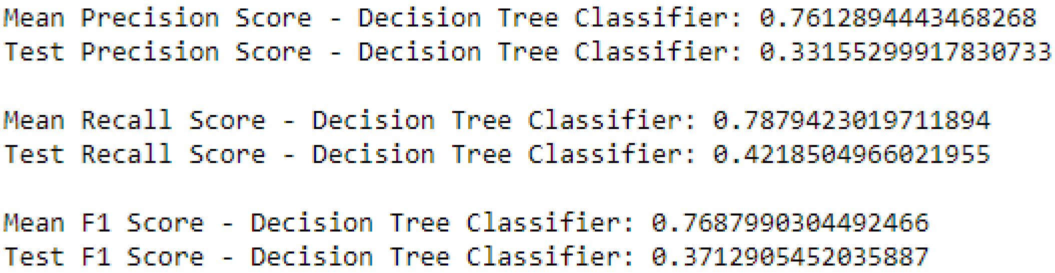
The baseline model for our project Decision Tree Classifier. The reason for selecting Decision Tree is that it is not sensitive to outliers since outliers never cause much reduction in Residual Sum of Squares (RSS) because they are never involved in the split. In our project we did not remove or cap the outliers because every data point is important for prediction and cannot treat the outliers until the client allows us to do the same. There is no requirement of feature scaling techniques such as standardization and normalization as it uses a rule-based approach instead of calculation of distances. Decision Trees are a class of very powerful Machine Learning models cable of achieving high accuracy in many tasks while being highly interpretable. What makes decision trees special in the realm of ML models is really their clarity of information representation.



```
dtc = DecisionTreeClassifier()
dtc.fit(x_train_sm, y_train_sm)
dtc.score(x_train_sm, y_train_sm)
dtc.score(x_test, y_test)
p5 = dtc.predict_proba(x_test)
```

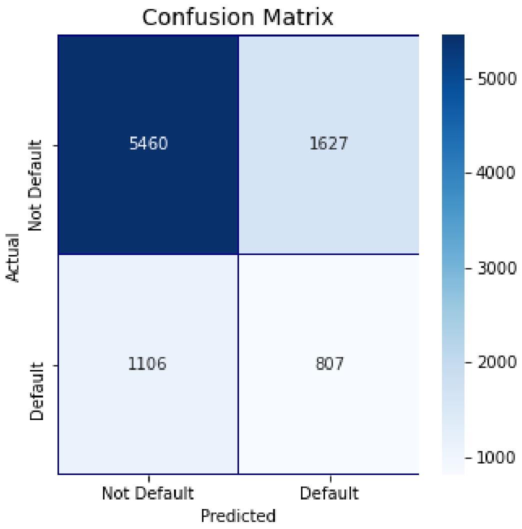


```
s41 = precision_score(y_test, preds_5)
s42 = recall_score(y_test, preds_5)
s43 = f1_score(y_test, preds_5)
s44 = accuracy_score(y_test, preds_5)
```

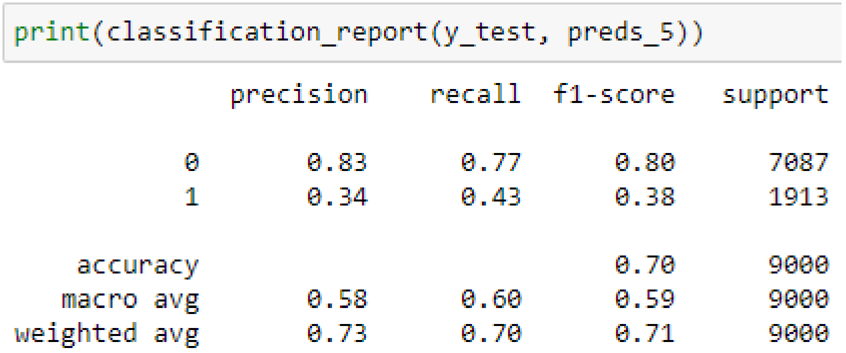


```
Mean Precision Score - Decision Tree Classifier: 0.7612894443468268
Test Precision Score - Decision Tree Classifier: 0.33155299917830733

Mean Recall Score - Decision Tree Classifier: 0.7879423019711894
Test Recall Score - Decision Tree Classifier: 0.4218504966021955

Mean F1 Score - Decision Tree Classifier: 0.7687990304492466
Test F1 Score - Decision Tree Classifier: 0.3712905452035887
```

Model Performance Evaluation Metrics



5.2. Classification Report:



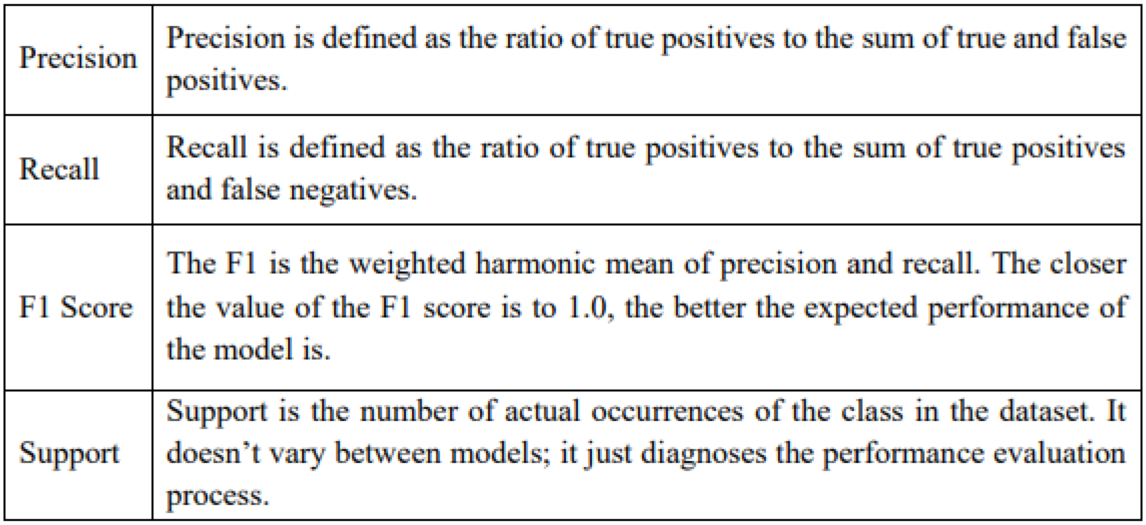
```
print(classification_report(y_test, preds_5))
```

```
              precision    recall  f1-score   support

           0       0.83      0.77      0.80      7087
           1       0.34      0.43      0.38      1913

    accuracy                           0.70      9000
   macro avg       0.58      0.60      0.59      9000
weighted avg       0.73      0.70      0.71      9000
```

A classification report is used to measure the quality of predictions from a classification algorithm. It shows how many predictions are true and how many are false. More specifically, True positives, True negatives, false positives and false negatives.

4 ways to check if the prediction is right or wrong

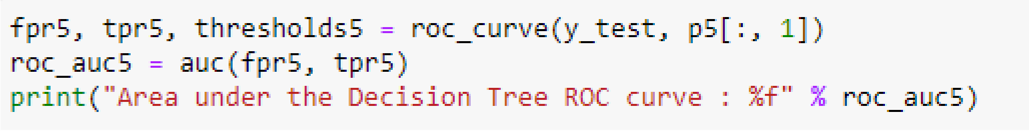
- True Negative: The model correctly predicts the negative class
- False Negative: The model incorrectly predicts the negative class.
- True Positives: The model correctly predicts the positive class
- False Positives: The model incorrectly predicts the positive class.



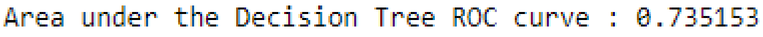
Precision	Precision is defined as the ratio of true positives to the sum of true and false positives.
Recall	Recall is defined as the ratio of true positives to the sum of true positives and false negatives.
F1 Score	The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance of the model is.
Support	Support is the number of actual occurrences of the class in the dataset. It doesn't vary between models; it just diagnoses the performance evaluation process.

From the confusion matrix we can see the precision, recall and f1 score of both positive and negative classes. Among the 3, f1 score is more useful than precision and recall because, it is the weighted average of both precision and recall. In most real-life classification problems, imbalanced class distribution exists and thus f1 score is a better metric to evaluate the model. The f1 score of negative class is 0.80 and that of positive class is 0.38.

5.3. AUC -Area Under the curve:



```
fpr5, tpr5, thresholds5 = roc_curve(y_test, p5[:, 1])
roc_auc5 = auc(fpr5, tpr5)
print("Area under the Decision Tree ROC curve : %f" % roc_auc5)
```

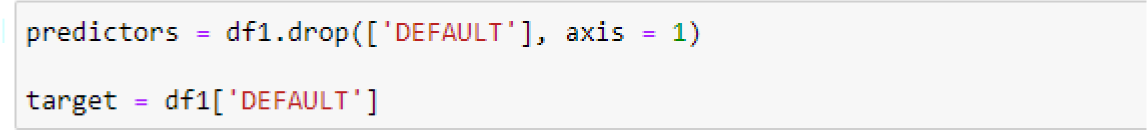


```
Area under the Decision Tree ROC curve : 0.735153
```

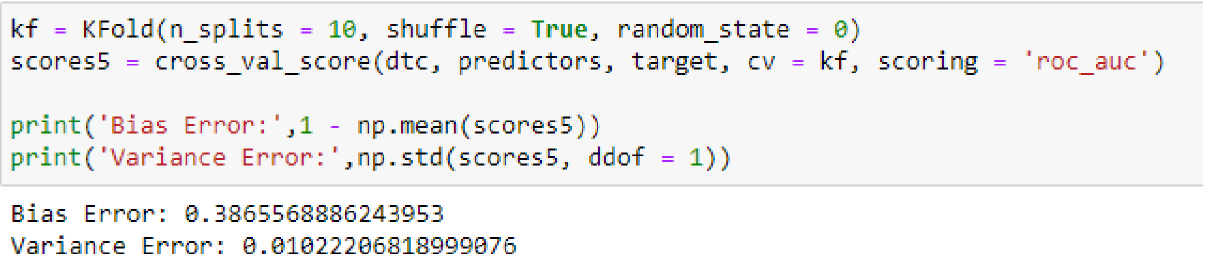
AUC is the one of the most widely used metrics for evaluation and is the best scoring for binary classification. The AUC is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

The AUC score is 0.73. When the AUC is greater than 0.5, there is a high chance that the classifier will be able to distinguish the positive and negative classes and further we should try to maximize the AUC score.

5.3. KFold Cross Validation- Variance and Bias Error:



```
predictors = df1.drop(['DEFAULT'], axis = 1)

target = df1['DEFAULT']
```



```
kf = KFold(n_splits = 10, shuffle = True, random_state = 0)
scores5 = cross_val_score(dtc, predictors, target, cv = kf, scoring = 'roc_auc')

print('Bias Error:',1 - np.mean(scores5))
print('Variance Error:',np.std(scores5, ddof = 1))
```

```
Bias Error: 0.3865568886243953
Variance Error: 0.01022206818999076
```

Cross_val_score splits the data into different folds and gives you scores for each fold from which u can calculate the mean and variance. K fold can help with overfitting because you are essentially split the data into various different train test splits compared to doing it once. A model is said to be good machine learning model if it generalizes any new input data from the problem domain in a proper way that means low variance error and bias error. Overfitting and underfitting are majorly responsible for the poor performance of the ML model. A ML model is said to have underfitting when it cannot capture the underlying trend of the data. A ML model is said to be overfitted when we train it with lot of data and it starts learning from the noise and inaccurate data entries in the data set. In a nutshell, Underfitting is high bias error and overfitting is high variance error. Cross Validation was done on the data (model fitted for first with KFold set up with n splits = 10 and scoring = ‘roc_auc’ as it is binary classification analysis. The Bias error and Variance error for each model is obtained to make the comparative analysis based on the performance of each model. We should try to reduce both variance and bias error to get a good model.

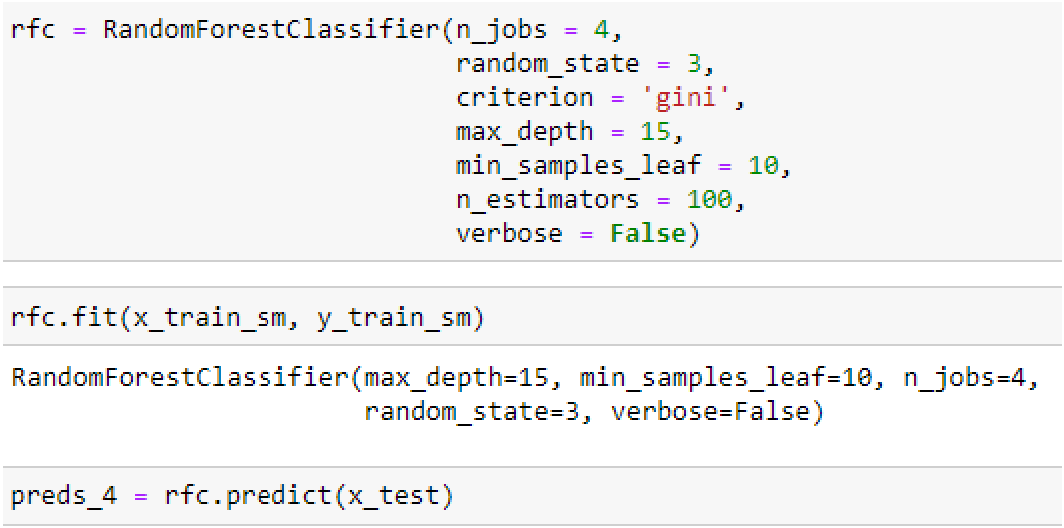
CHAPTER – 6

MODEL BUILDING

6.1.Random Forest Classifier

It is a supervised machine learning algorithm that is widely used in classification problems that consists of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than any individual tree.

Random forest generally reduces the bias error by allocating the features and reduce the variance error by bootstrap sampling or bagging technique. This bootstrapping procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bootstrap sampling is a way of de-correlating the trees by showing them different training sets.



```
rfc = RandomForestClassifier(n_jobs = 4,
                             random_state = 3,
                             criterion = 'gini',
                             max_depth = 15,
                             min_samples_leaf = 10,
                             n_estimators = 100,
                             verbose = False)
```

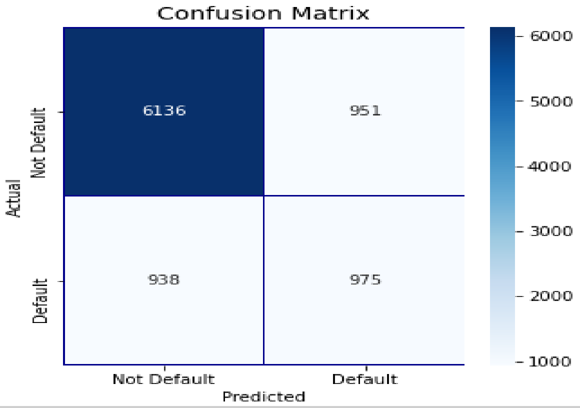
```
rfc.fit(x_train_sm, y_train_sm)
```

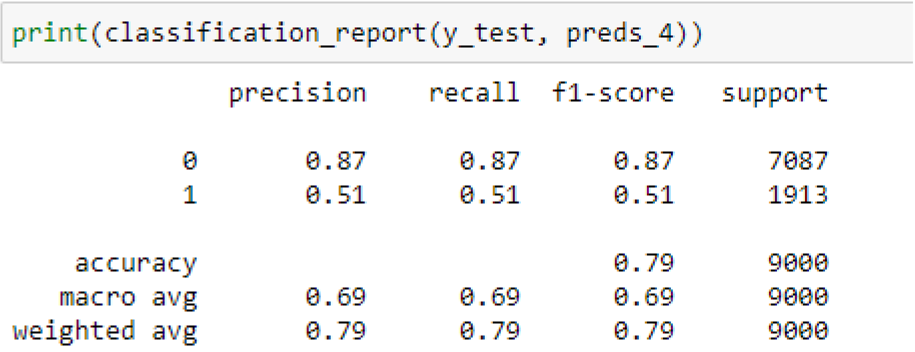
```
RandomForestClassifier(max_depth=15, min_samples_leaf=10, n_jobs=4,
                       random_state=3, verbose=False)
```

```
preds_4 = rfc.predict(x_test)
```

Model Performance Evaluation Metrics

6.1.1.Classification Report:



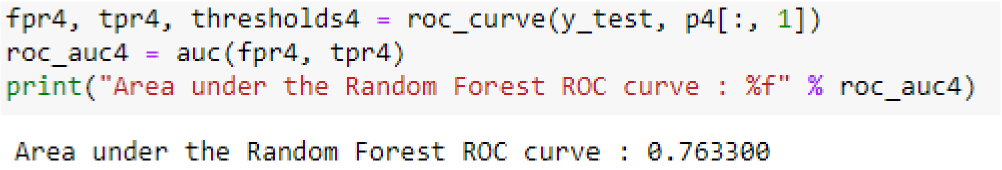


```
print(classification_report(y_test, preds_4))
```

```
              precision    recall  f1-score   support

           0       0.87      0.87      0.87      7087
           1       0.51      0.51      0.51      1913

    accuracy                           0.79      9000
   macro avg       0.69      0.69      0.69      9000
weighted avg       0.79      0.79      0.79      9000
```

From the confusion matrix we can see the precision, recall and f1 score of both positive and negative classes. Among the 3, f1 score is more useful than precision and recall because, it is the weighted average of both precision and recall. In most real-life classification problems, imbalanced class distribution exists and thus f1 score is a better metric to evaluate the model. The f1 score of negative class is 0.87 and that of positive class is 0.51. We can see that random forest gives a better f1 score when compared to decision tree model/baseline model. We can also see that precision and recall has also increased when compared to decision

6.1.2.AUC -Area Under the Curve

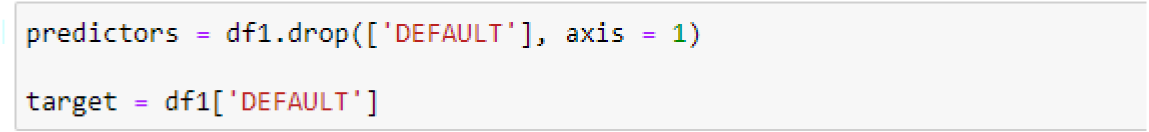


```
fpr4, tpr4, thresholds4 = roc_curve(y_test, p4[:, 1])
roc_auc4 = auc(fpr4, tpr4)
print("Area under the Random Forest ROC curve : %f" % roc_auc4)
```

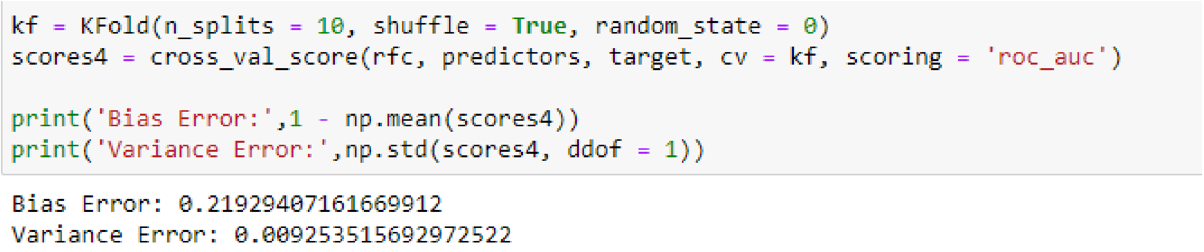
```
Area under the Random Forest ROC curve : 0.763300
```

The AUC score is 0.763300. When the AUC is greater than 0.5, there is a high chance that the classifier will be able to distinguish the positive and negative classes and further we should try to maximize the AUC score.

6.1.3. KFold Cross Validation- Variance and Bias Error:



```
predictors = df1.drop(['DEFAULT'], axis = 1)

target = df1['DEFAULT']
```



```
kf = KFold(n_splits = 10, shuffle = True, random_state = 0)
scores4 = cross_val_score(rfc, predictors, target, cv = kf, scoring = 'roc_auc')

print('Bias Error:',1 - np.mean(scores4))
print('Variance Error:',np.std(scores4, ddof = 1))
```

```
Bias Error: 0.21929407161669912
Variance Error: 0.009253515692972522
```

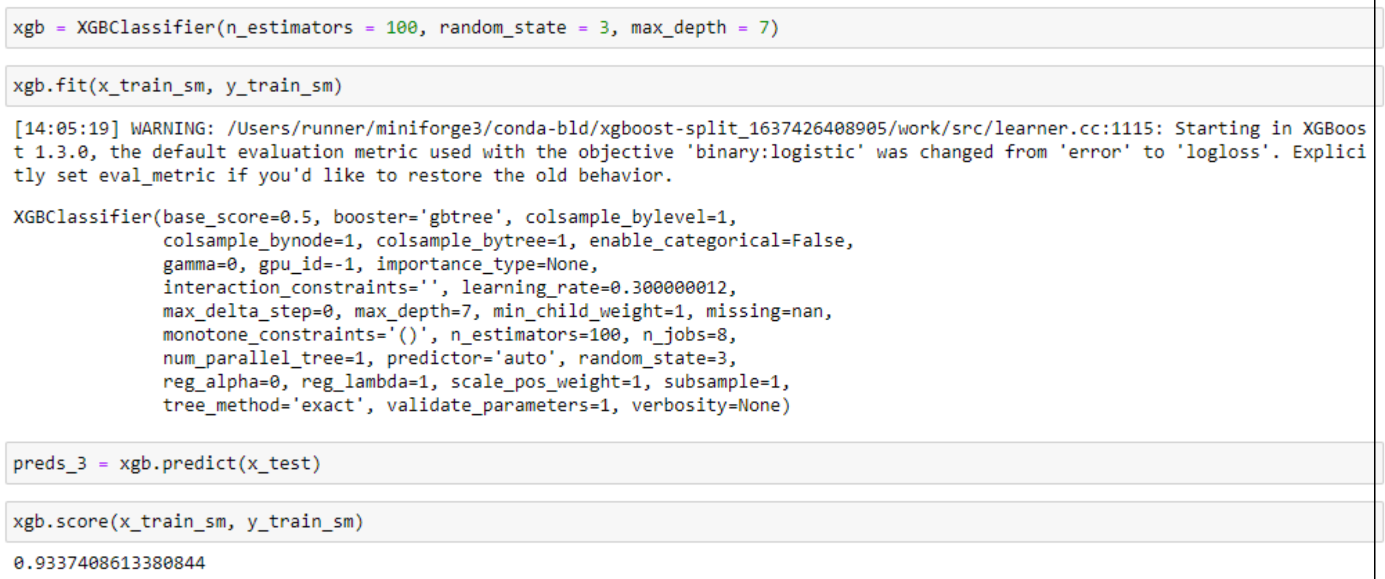
After k fold cross validation, we can see that the bias error is 0.21 and variance error is 0. 009.The variance and bias error in random forest lesser when compared to decision tree.

6.2.XGBOOST CLASSIFIER

XGBoost and Gradient Boosting Machines (GBMs) are both ensemble tree methods that apply the principle of boosting weak learners (CARTs generally) using the gradient descent architecture. However, XGBoost improves upon the base GBM framework through systems optimization and algorithmic enhancements.

Parallelization: XGBoost approaches the process of sequential tree building using parallelized implementation. This is possible due to the interchangeable nature of loops used for building base learners; the outer loop that enumerates the leaf nodes of a tree, and the second inner loop that calculates the features. This nesting of loops limits parallelization because without completing the inner loop (more computationally demanding of the two), the outer loop cannot be started. Therefore, to improve run time, the order of loops is interchanged using initialization through a global scan of all instances and sorting using parallel threads. This switch improves algorithmic performance by offsetting any parallelization overheads in computation.

Tree Pruning: The stopping criterion for tree splitting within GBM framework is greedy in nature and depends on the negative loss criterion at the point of split. XGBoost uses 'max_depth' parameter as specified instead of criterion first, and starts pruning trees backward. This 'depth-first' approach improves computational performance significantly.



```
xgb = XGBClassifier(n_estimators = 100, random_state = 3, max_depth = 7)
```

```
xgb.fit(x_train_sm, y_train_sm)
```

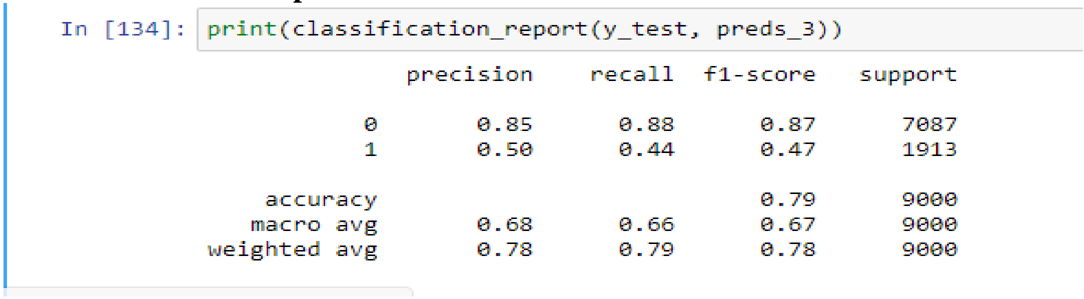
```
[14:05:19] WARNING: /Users/runner/miniforge3/conda-bld/xgboost-split_1637426408905/work/src/learner.cc:1115: Starting in XGBoos
t 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explici
tly set eval_metric if you'd like to restore the old behavior.

XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=7, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=8,
              num_parallel_tree=1, predictor='auto', random_state=3,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
```

```
preds_3 = xgb.predict(x_test)
```

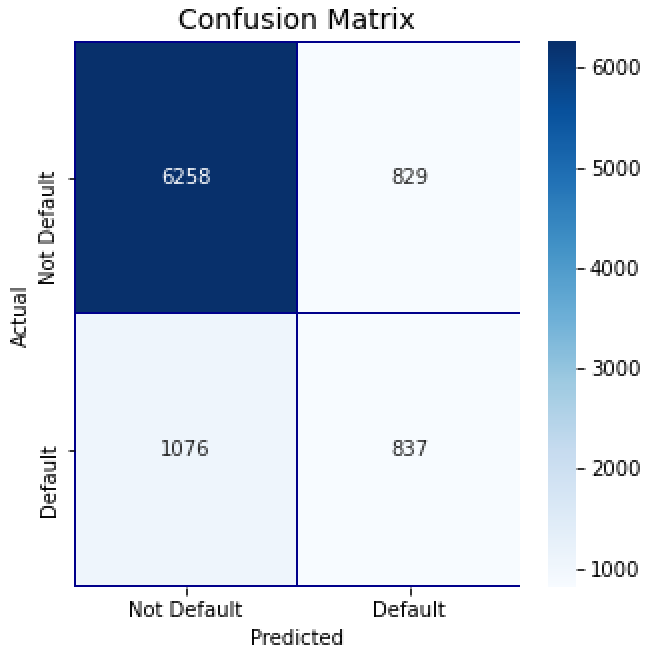
```
xgb.score(x_train_sm, y_train_sm)
```

```
0.9337408613380844
```

6.2.1.Classification report

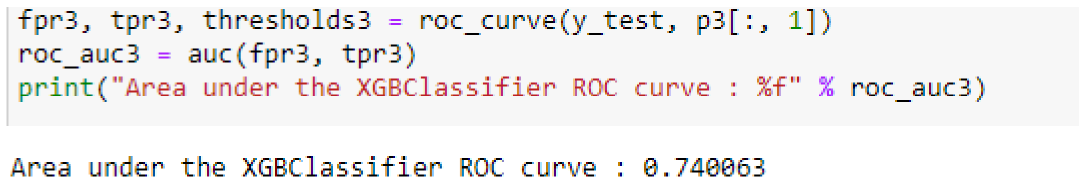


```
In [134]: print(classification_report(y_test, preds_3))

                        precision    recall  f1-score   support

                   0       0.85      0.88      0.87      7087
                   1       0.50      0.44      0.47      1913

            accuracy                           0.79      9000
           macro avg       0.68      0.66      0.67      9000
        weighted avg       0.78      0.79      0.78      9000
```



From the confusion matrix we can see the precision, recall and f1 score of both positive and negative classes. Among the 3, f1 score is more useful than precision and recall because, it is the weighted average of both precision and recall. In most real-life classification problems, imbalanced class distribution exists and thus f1 score is a better metric to evaluate the model. The f1 score of negative class is 0.87 and that of positive class is 0.47.

6.2.2. AUC -Area Under the Curve

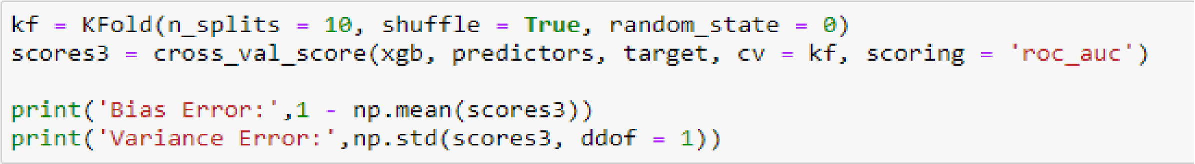


```
fpr3, tpr3, thresholds3 = roc_curve(y_test, p3[:, 1])
roc_auc3 = auc(fpr3, tpr3)
print("Area under the XGBClassifier ROC curve : %f" % roc_auc3)
```

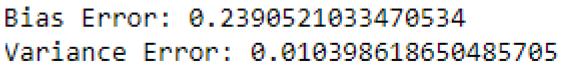
```
Area under the XGBClassifier ROC curve : 0.740063
```

The AUC score is 0.740063.

6.2.3. KFold Cross Validation- Variance and Bias Error:



```
kf = KFold(n_splits = 10, shuffle = True, random_state = 0)
scores3 = cross_val_score(xgb, predictors, target, cv = kf, scoring = 'roc_auc')

print('Bias Error:',1 - np.mean(scores3))
print('Variance Error:',np.std(scores3, ddof = 1))
```



```
Bias Error: 0.2390521033470534
Variance Error: 0.010398618650485705
```

After k fold cross validation, we can see that the bias error is 0.23 and variance error is 0.010.

6.3.ADABOOST CLASSIFIER

Ada-boost or Adaptive Boosting is one of ensemble boosting classifier proposed by Yoav Freund and Robert Schapire in 1996. It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier. The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations. Any machine learning algorithm can be used as base classifier if it accepts weights on the training set. Adaboost should meet two conditions:

The classifier should be trained interactively on various weighed training examples.

In each iteration, it tries to provide an excellent fit for these examples by minimizing training error.

It works in the following steps:

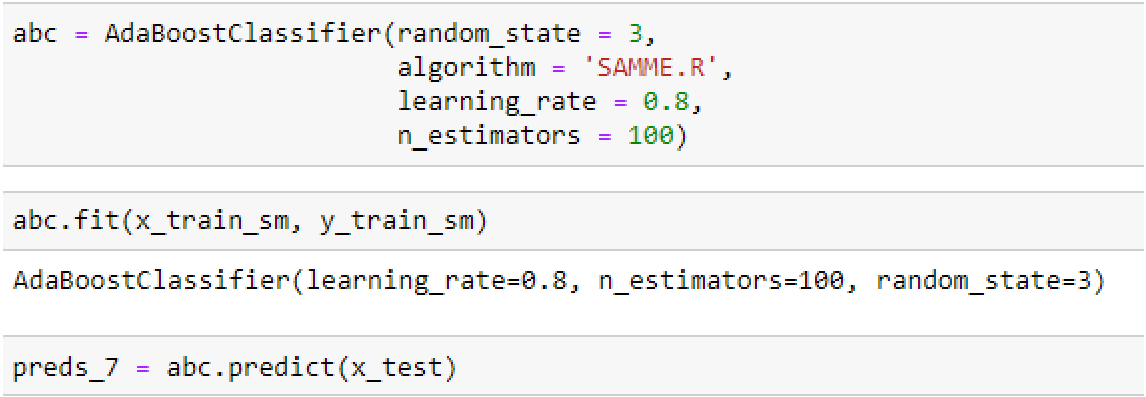
a. Initially, Adaboost selects a training subset randomly.

b. It iteratively trains the AdaBoost machine learning model by selecting the training set based on the accurate prediction of the last training.

c. It assigns the higher weight to wrong classified observations so that in the next iteration these observations will get the high probability for classification.

d. Also, it assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier. The more accurate classifier will get high weight.

e. This process iterates until the complete training data fits without any error or until reached to the specified maximum number of estimators.



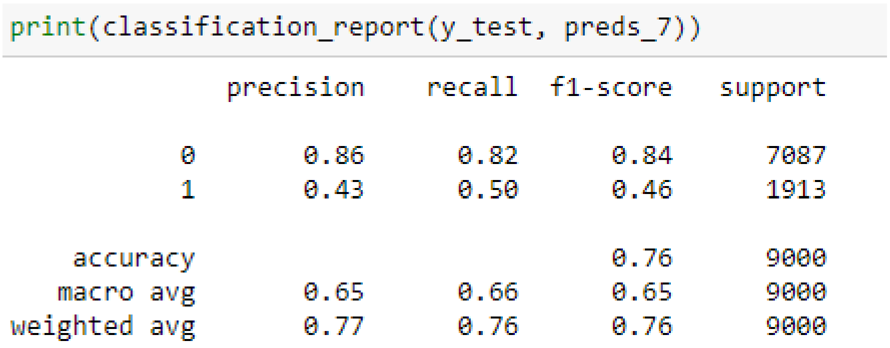
```
abc = AdaBoostClassifier(random_state = 3,
                         algorithm = 'SAMME.R',
                         learning_rate = 0.8,
                         n_estimators = 100)
```

```
abc.fit(x_train_sm, y_train_sm)
```

```
AdaBoostClassifier(learning_rate=0.8, n_estimators=100, random_state=3)
```

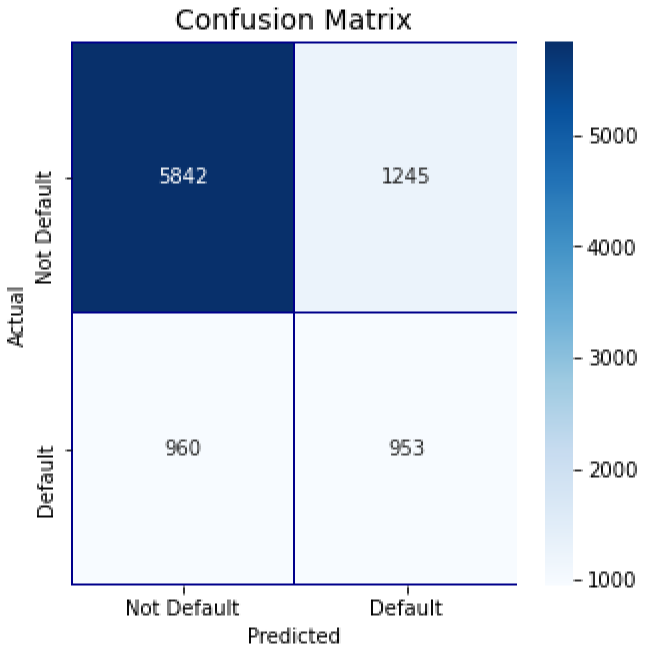
```
preds_7 = abc.predict(x_test)
```

6.3.1.Classification Report



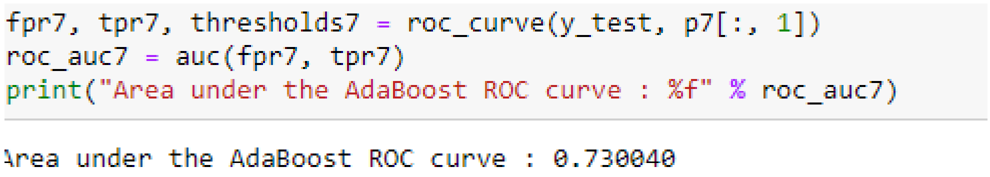
```
print(classification_report(y_test, preds_7))
```

```
              precision    recall  f1-score   support

           0       0.86      0.82      0.84      7087
           1       0.43      0.50      0.46      1913

    accuracy                           0.76      9000
   macro avg       0.65      0.66      0.65      9000
weighted avg       0.77      0.76      0.76      9000
```



From the confusion matrix we can see the precision, recall and f1 score of both positive and negative classes. Among the 3, f1 score is more useful than precision and recall because, it is the weighted average of both precision and recall. In most real-life classification problems, imbalanced class distribution exists and thus f1 score is a better metric to evaluate the model. The f1 score of negative class is 0.84 and that of positive class is 0.46.

6.3.2.AUC -Area Under the Curve

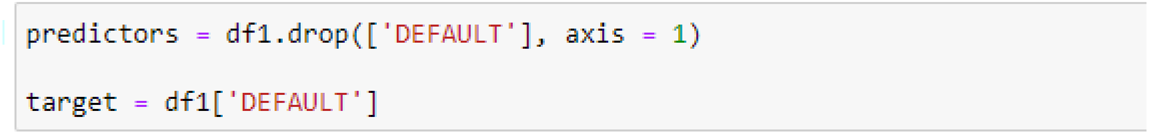


```
fpr7, tpr7, thresholds7 = roc_curve(y_test, p7[:, 1])
roc_auc7 = auc(fpr7, tpr7)
print("Area under the AdaBoost ROC curve : %f" % roc_auc7)
```

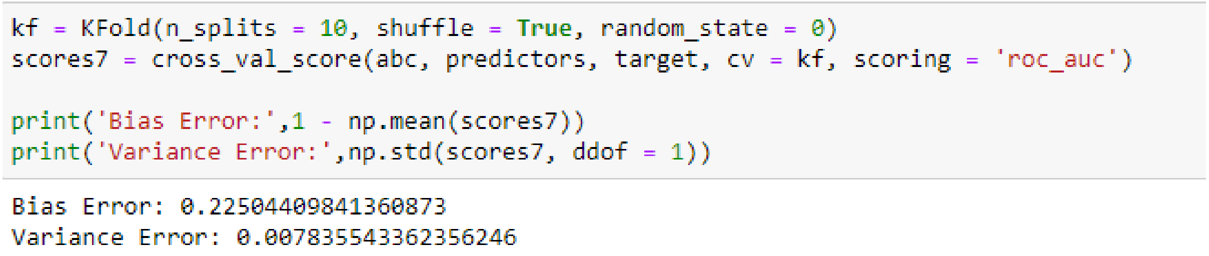
```
Area under the AdaBoost ROC curve : 0.730040
```

The AUC score is 0.730040

6.3.3. KFold Cross Validation- Variance and Bias Error:



```
predictors = df1.drop(['DEFAULT'], axis = 1)

target = df1['DEFAULT']
```



```
kf = KFold(n_splits = 10, shuffle = True, random_state = 0)
scores7 = cross_val_score(abc, predictors, target, cv = kf, scoring = 'roc_auc')

print('Bias Error:',1 - np.mean(scores7))
print('Variance Error:',np.std(scores7, ddof = 1))
```

```
Bias Error: 0.22504409841360873
Variance Error: 0.007835543362356246
```

After k fold cross validation, we can see that the bias error is 0.225 and variance error is 0.0078.

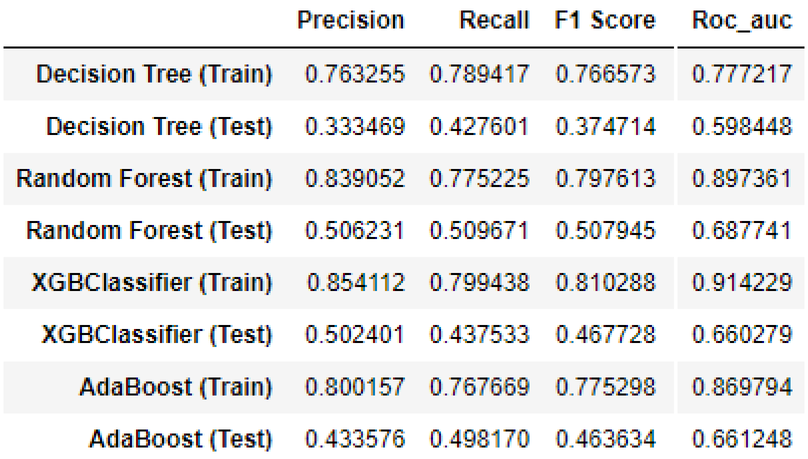
.

CHAPTER – 7

MODEL EVALUATION

7.1.PRECISION, RECALL, F1-SCORE AND ACCURACY

PRECISION	Precision is defined as the ratio of true positives to the sum of true and false positives.
RECALL	Recall is defined as the ratio of true positives to the sum of true positives and false negatives
F1-SCORE	The F1 is the weighted harmonic mean of precision and recall. The closer the value of the F1 score is to 1.0, the better the expected performance of the model is.
ACCURACY	Accuracy is the most intuitive performance measure and it is the ratio of correctly predicted observations to the total observations



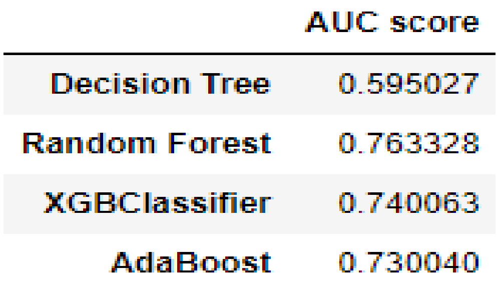
	Precision	Recall	F1 Score	Roc_auc
Decision Tree (Train)	0.763255	0.789417	0.766573	0.777217
Decision Tree (Test)	0.333469	0.427601	0.374714	0.598448
Random Forest (Train)	0.839052	0.775225	0.797613	0.897361
Random Forest (Test)	0.506231	0.509671	0.507945	0.687741
XGBClassifier (Train)	0.854112	0.799438	0.810288	0.914229
XGBClassifier (Test)	0.502401	0.437533	0.467728	0.660279
AdaBoost (Train)	0.800157	0.767669	0.775298	0.869794
AdaBoost (Test)	0.433576	0.498170	0.463634	0.661248

When comparing different models, we can see that random forest is least overfit and the precision, recall, f1 score is better for Random Forest.

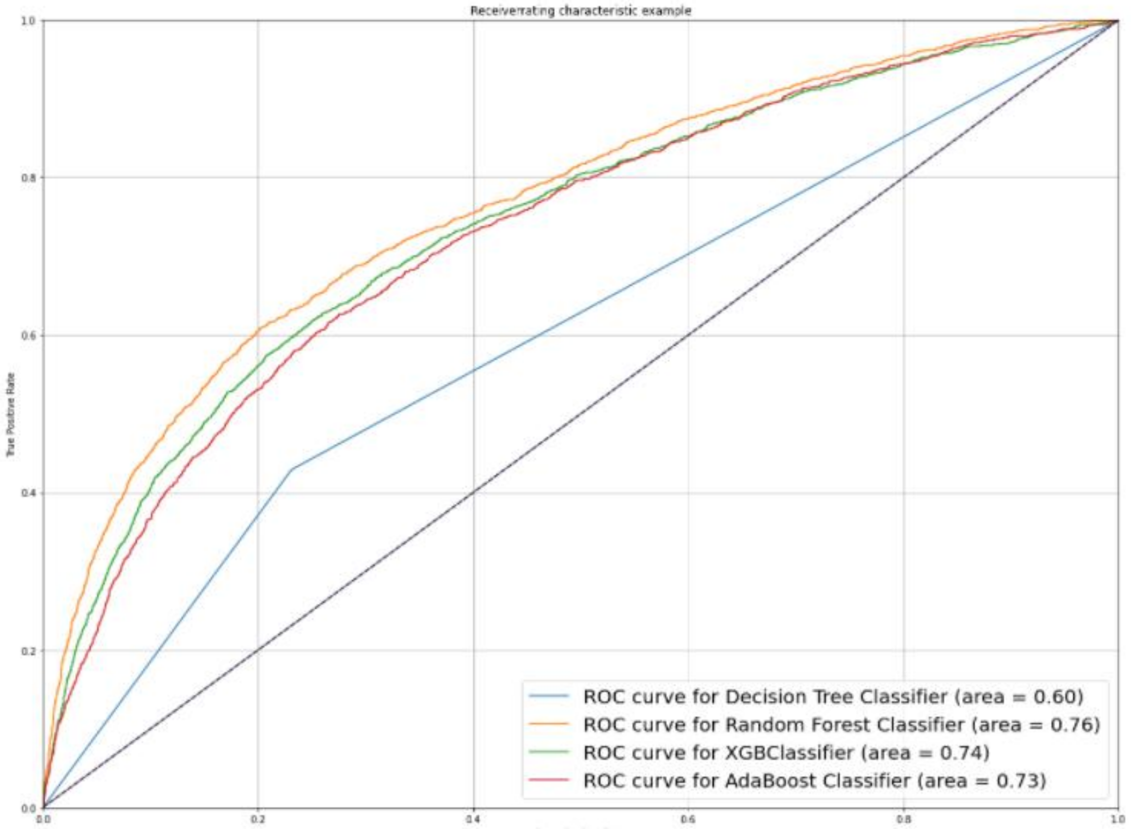
7.2.ROC-AUC SCORE:

An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate and False Positive Rate.

AUC is the one of the most widely used metrics for evaluation and is the best scoring for binary classification. The AUC is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.



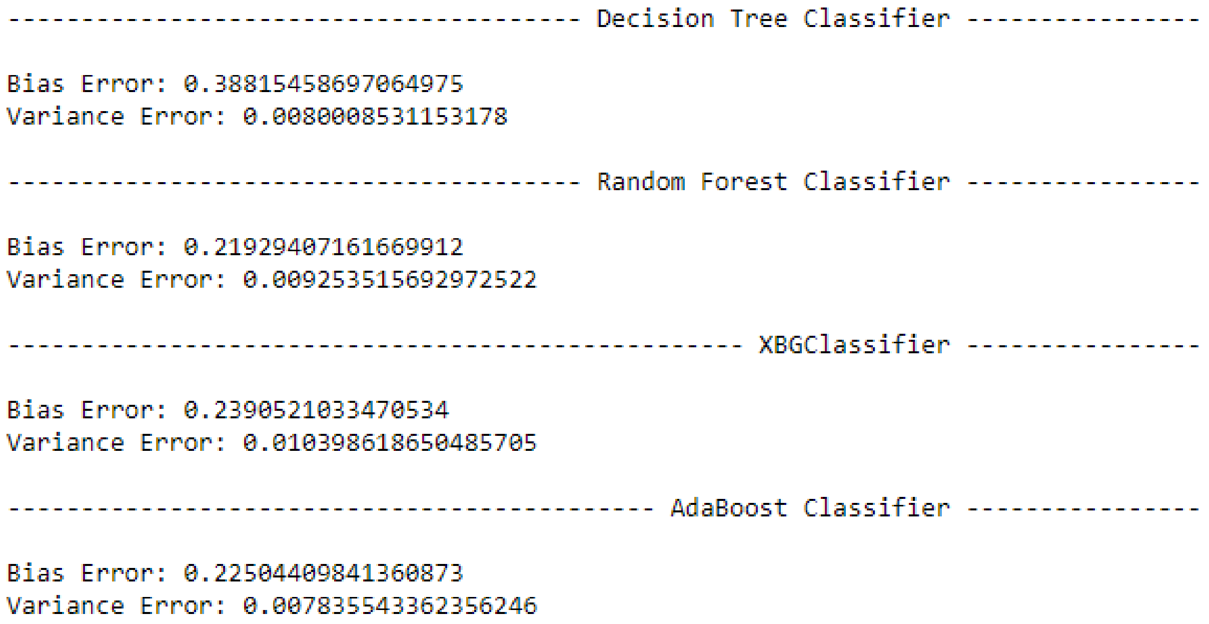
	AUC score
Decision Tree	0.595027
Random Forest	0.763328
XGBClassifier	0.740063
AdaBoost	0.730040



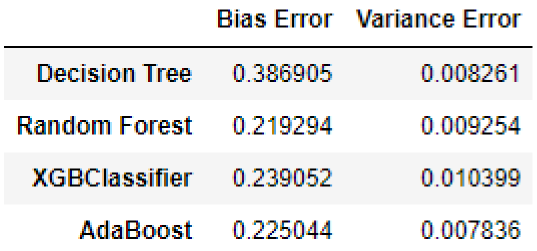
From the above graph, we can see that random forest has the highest AUC score followed by XGBclassifier, AdaBoost Classifier and Decision Tree classifier.

7.3.BIAS AND VARIANCE ERROR:

A model is said to be good machine learning model if it generalizes any new input data from the problem domain in a proper way that means low variance error and bias error. Overfitting and underfitting are majorly responsible for the poor performance of the ML model. A ML model is said to have underfitting when it cannot capture the underlying trend of the data. A ML model is said to be overfitted when we train it with lot of data and it starts learning from the noise and inaccurate data entries in the data set. In a nutshell, Underfitting is high bias error and overfitting is high variance error.



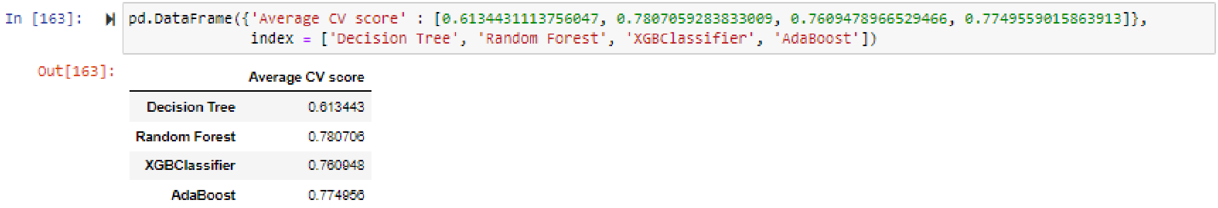
```
--------------------------------------- Decision Tree Classifier ----------------

Bias Error: 0.38815458697064975
Variance Error: 0.0080008531153178

--------------------------------------- Random Forest Classifier ----------------

Bias Error: 0.21929407161669912
Variance Error: 0.009253515692972522

--------------------------------------------------- XBGClassifier ----------------

Bias Error: 0.2390521033470534
Variance Error: 0.010398618650485705

-------------------------------------------- AdaBoost Classifier ----------------

Bias Error: 0.22504409841360873
Variance Error: 0.007835543362356246
```



	Bias Error	Variance Error
Decision Tree	0.386905	0.008261
Random Forest	0.219294	0.009254
XGBClassifier	0.239052	0.010399
AdaBoost	0.225044	0.007836

From the above table, we see the variance and bias error of both AdaBoost and Random Forest is close to each other,

7.4. CROSS - VALIDATION SCORE



```
In [163]:  pd.DataFrame({'Average CV score' : [0.6134431113756047, 0.7807059283833009, 0.7609478966529466, 0.7749559015863913]},
                          index = ['Decision Tree', 'Random Forest', 'XGBClassifier', 'AdaBoost'])
```

Out[163]:

	Average CV score
Decision Tree	0.613443
Random Forest	0.780706
XGBClassifier	0.760948
AdaBoost	0.774956

IMPLICATIONS

Since F1 score, AUC score of random forest is better than any other model and due to comparatively low variance and bias error, we consider Random Forest as the final model.

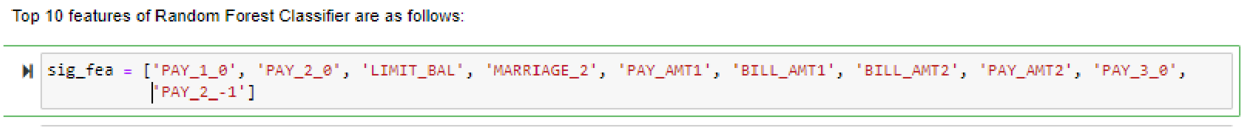
After considering the features using feature selection, we are going to create final model using Random Forest Classifier.

CHAPTER – 8

8.1 FINAL MODEL-RANDOM FOREST CLASSIFIER

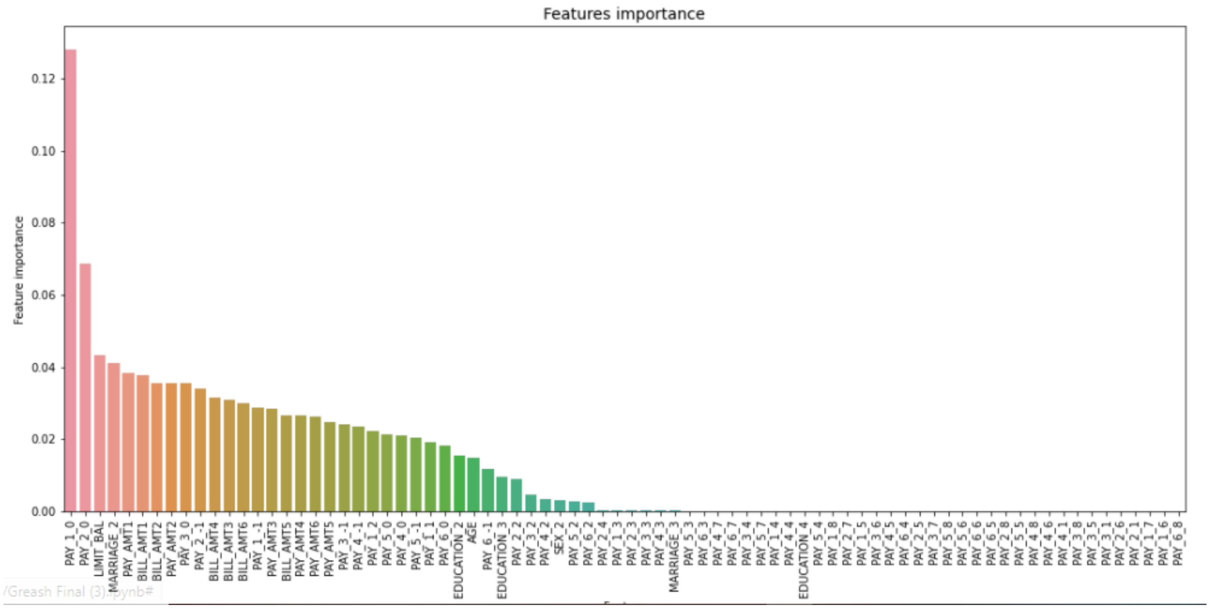
It is a supervised machine learning algorithm that is widely used in classification problems that consists of many decision trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than any individual tree.

VARIABLE IMPORTANCE PLOT:

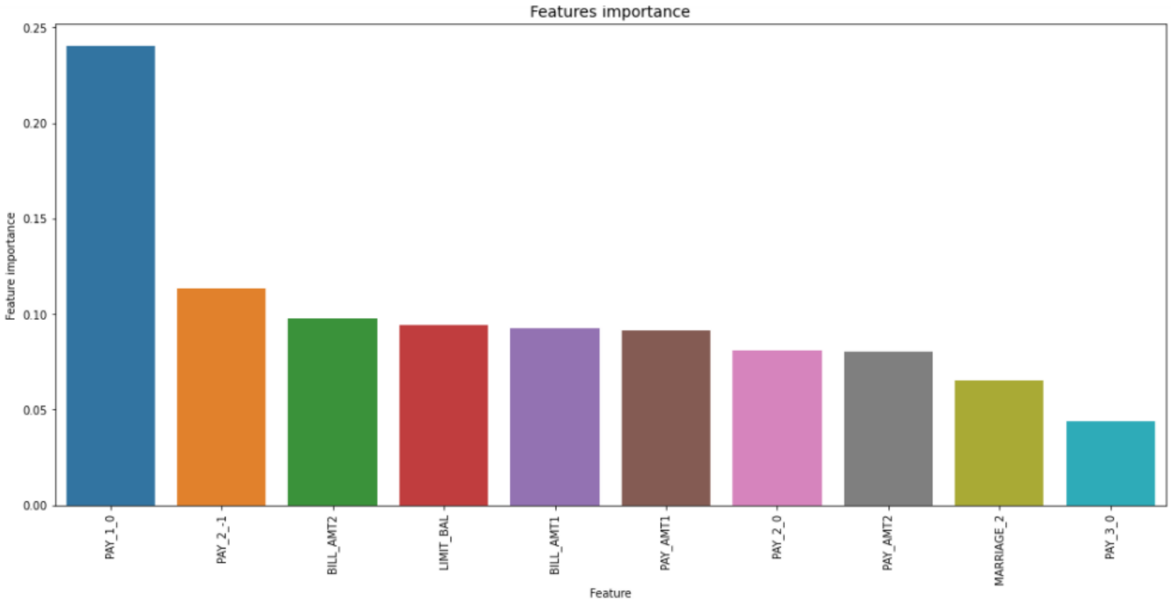


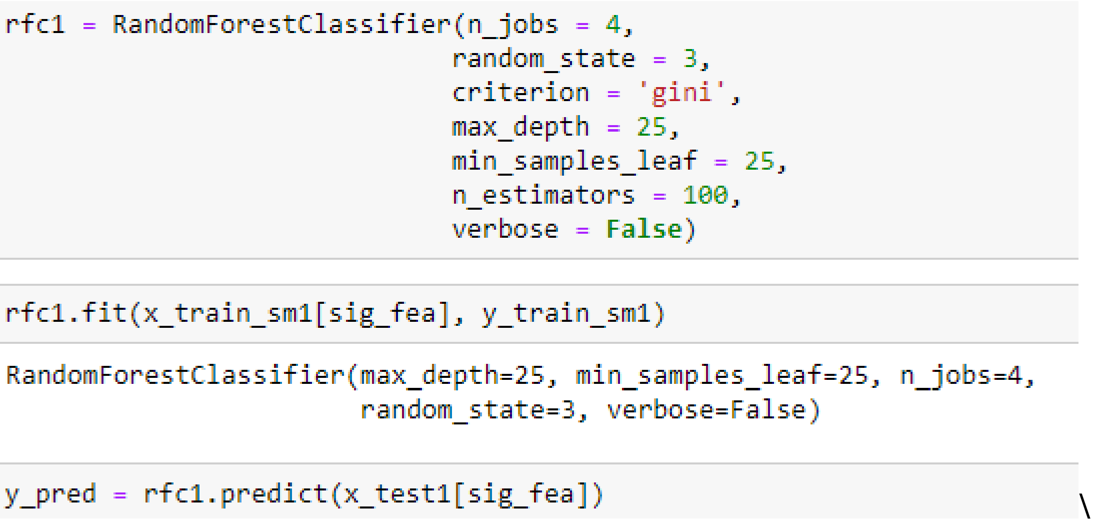
Top 10 features of Random Forest Classifier are as follows:

```
sig_fea = ['PAY_1_0', 'PAY_2_0', 'LIMIT_BAL', 'MARRIAGE_2', 'PAY_AMT1', 'BILL_AMT1', 'BILL_AMT2', 'PAY_AMT2', 'PAY_3_0',
           'PAY_2_-1']
```



The important features using Random Forest Classifier are PAY_1_0, PAY_2_0, LIMIT_BAL, MARRIAGE_2, BILL_AMT1 and PAY_AMT1.





```
rfc1 = RandomForestClassifier(n_jobs = 4,
                              random_state = 3,
                              criterion = 'gini',
                              max_depth = 25,
                              min_samples_leaf = 25,
                              n_estimators = 100,
                              verbose = False)
```

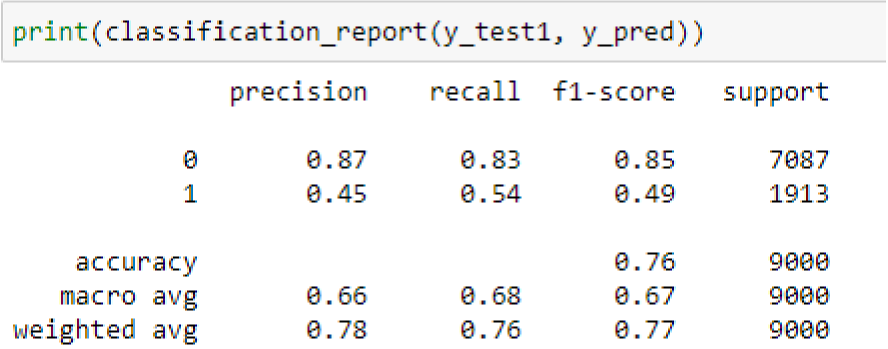
```
rfc1.fit(x_train_sm1[sig_fea], y_train_sm1)
```

```
RandomForestClassifier(max_depth=25, min_samples_leaf=25, n_jobs=4,
                       random_state=3, verbose=False)
```

```
y_pred = rfc1.predict(x_test1[sig_fea])
```

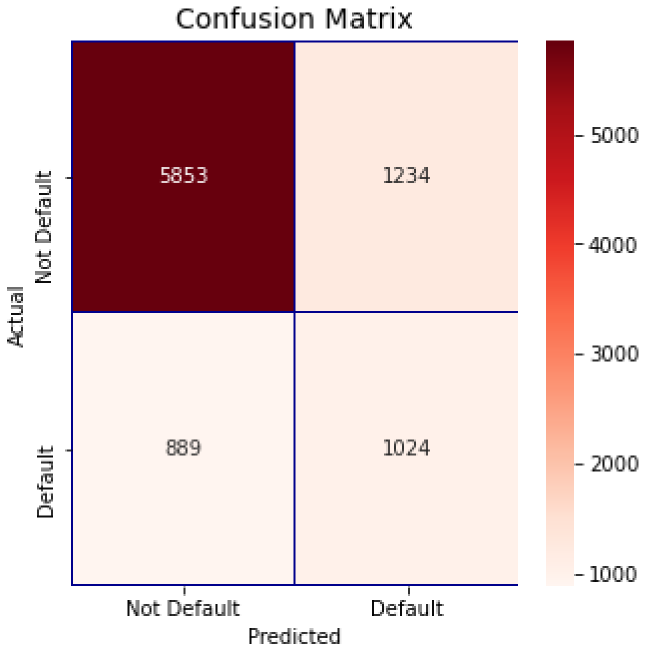
\

8.1.Classification report:

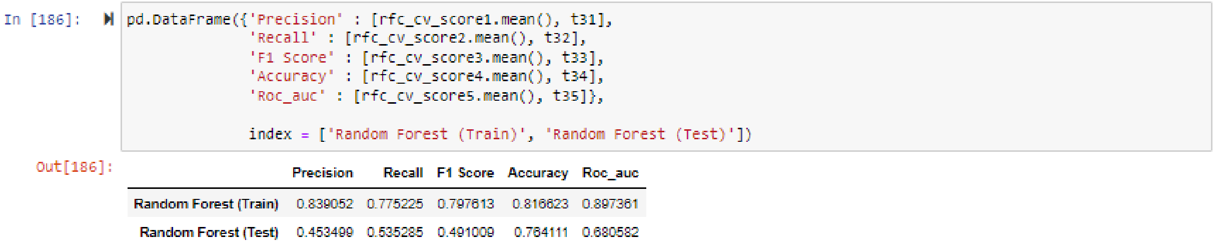


```
print(classification_report(y_test1, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.87      0.83      0.85      7087
           1       0.45      0.54      0.49      1913

    accuracy                           0.76      9000
   macro avg       0.66      0.68      0.67      9000
weighted avg       0.78      0.76      0.77      9000
```



The f1 score of negative class is 0.85 and that of positive class is 0.49.

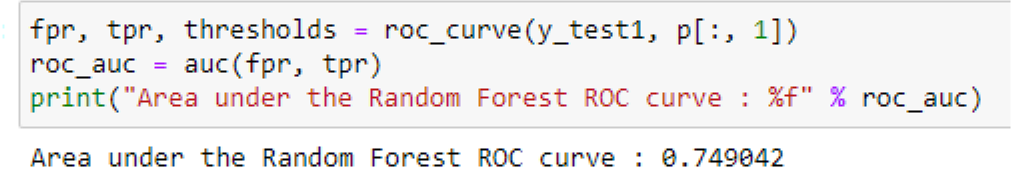


```
In [186]:  pd.DataFrame({'Precision' : [rfc_cv_score1.mean(), t31],
                         'Recall' : [rfc_cv_score2.mean(), t32],
                         'F1 Score' : [rfc_cv_score3.mean(), t33],
                         'Accuracy' : [rfc_cv_score4.mean(), t34],
                         'Roc_auc' : [rfc_cv_score5.mean(), t35]},

                         index = ['Random Forest (Train)', 'Random Forest (Test)'])
```

Out[186]:

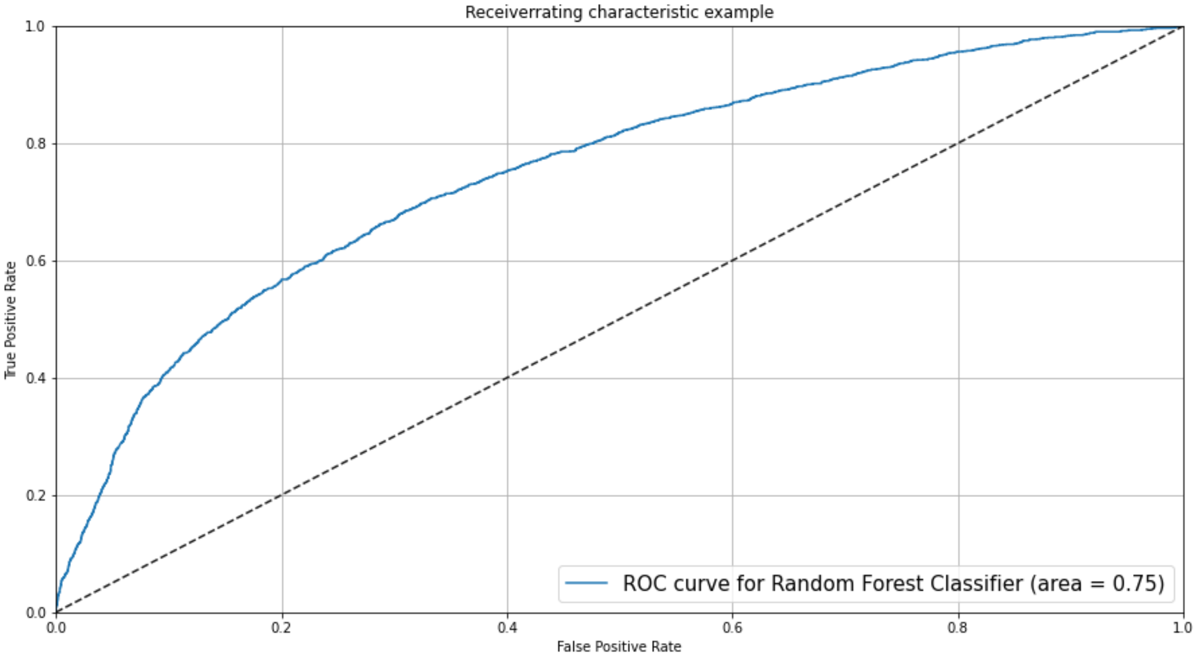
	Precision	Recall	F1 Score	Accuracy	Roc_auc
Random Forest (Train)	0.839052	0.775225	0.797613	0.816623	0.897361
Random Forest (Test)	0.453499	0.535285	0.491009	0.764111	0.680582

8.2.ROC-AUC score



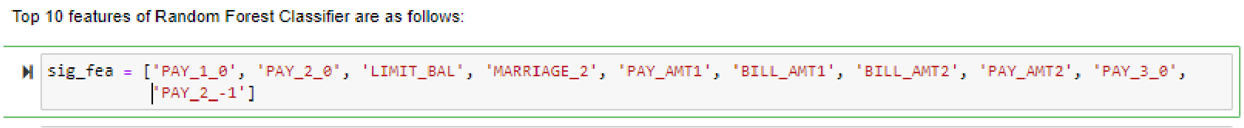
```
fpr, tpr, thresholds = roc_curve(y_test1, p[:, 1])
roc_auc = auc(fpr, tpr)
print("Area under the Random Forest ROC curve : %f" % roc_auc)
```

```
Area under the Random Forest ROC curve : 0.749042
```



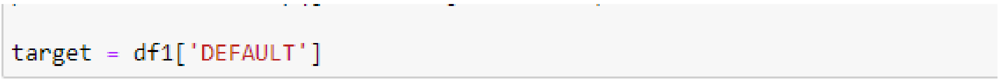
The AUC score is 75 for the final model. When the AUC is greater than 0.5, there is a high chance that the classifier will be able to distinguish the positive and negative classes and further we should try to maximize the AUC score.

8.3.BIAS AND VARIANCE ERROR

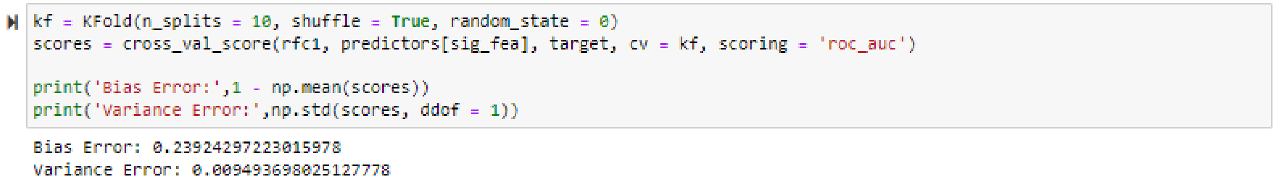


Top 10 features of Random Forest Classifier are as follows:

```
sig_fea = ['PAY_1_0', 'PAY_2_0', 'LIMIT_BAL', 'MARRIAGE_2', 'PAY_AMT1', 'BILL_AMT1', 'BILL_AMT2', 'PAY_AMT2', 'PAY_3_0',
           'PAY_2_-1']
```

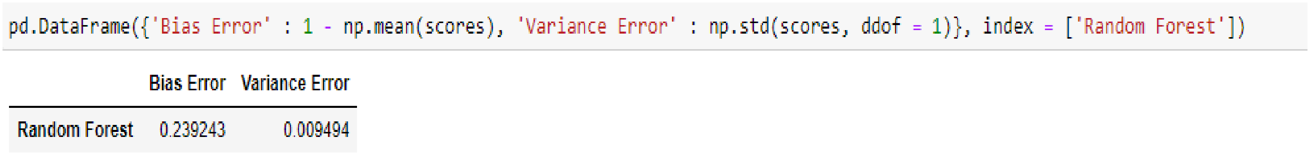


```
target = df1['DEFAULT']
```



```
kf = KFold(n_splits = 10, shuffle = True, random_state = 0)
scores = cross_val_score(rfc1, predictors[sig_fea], target, cv = kf, scoring = 'roc_auc')

print('Bias Error:',1 - np.mean(scores))
print('Variance Error:',np.std(scores, ddof = 1))
```

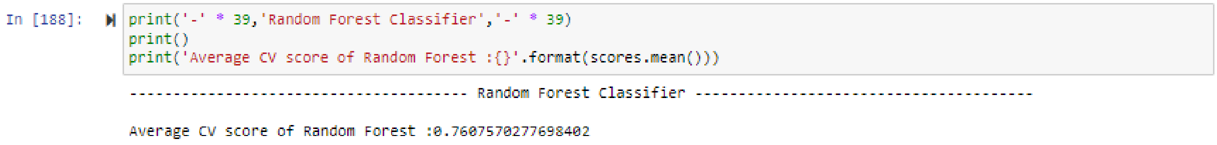
```
Bias Error: 0.23924297223015978
Variance Error: 0.009493698025127778
```



```
pd.DataFrame({'Bias Error' : 1 - np.mean(scores), 'Variance Error' : np.std(scores, ddof = 1)}, index = ['Random Forest'])
```

	Bias Error	Variance Error
Random Forest	0.239243	0.009494

8.4. CROSS-VALIDATION SCORE



```
In [188]: print('-' * 39,'Random Forest Classifier','-' * 39)
          print()
          print('Average CV score of Random Forest :{}'.format(scores.mean()))
```

```
--------------------------------------- Random Forest Classifier ---------------------------------------

Average CV score of Random Forest :0.7607570277698402
```

The bias error is 0.23 and variance error is 0.009 which shows that it is not highly overfit or underfit. The goal of any supervised Machine learning model is to achieve low bias and low variance. Here we have done feature selection which can be viewed as a variance reduction method that trades off the benefits of decreased variance with the harm of increased bias. This is the reason why there is a slight increase in bias error of random forest model done with feature selection .Overall , we can say the final model as best separation model as it has high f1 score, AUC score , low variance and low bias.

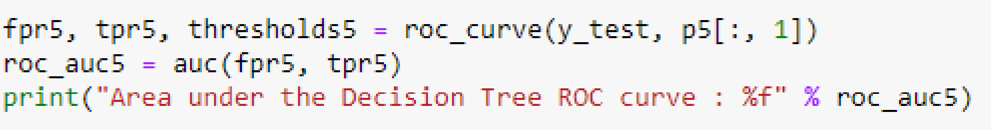
CHAPTER – 9

COMPARISON TO BENCHMARK

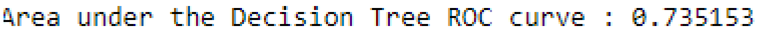
When we compare the final model with the benchmark, we can undoubtedly say that the final model has improved to a great extent. To prove the same, we would like summarize and compare scores of both baseline model (decision tree) and final model (Random Forest).

1.AUC ROC SCORE

a. Baseline model:

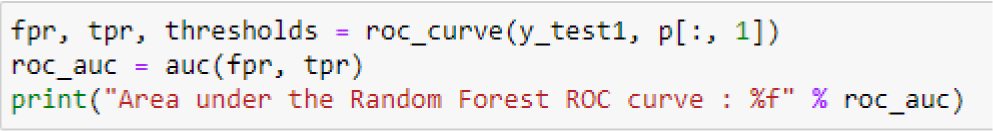


```
fpr5, tpr5, thresholds5 = roc_curve(y_test, p5[:, 1])
roc_auc5 = auc(fpr5, tpr5)
print("Area under the Decision Tree ROC curve : %f" % roc_auc5)
```

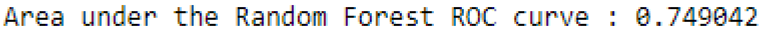


```
Area under the Decision Tree ROC curve : 0.735153
```

b. Final model:



```
fpr, tpr, thresholds = roc_curve(y_test1, p[:, 1])
roc_auc = auc(fpr, tpr)
print("Area under the Random Forest ROC curve : %f" % roc_auc)
```

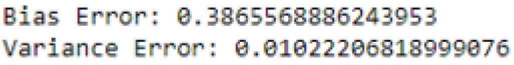


```
Area under the Random Forest ROC curve : 0.749042
```

Here we can see AUC score of final model is better and greater than baseline model.

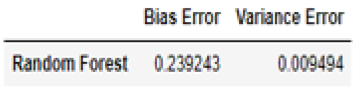
2.VARIANCE AND BIAS ERROR

a. Baseline model:



```
Bias Error: 0.3865568886243953
Variance Error: 0.01022206818999076
```

b. Final model:



	Bias Error	Variance Error
Random Forest	0.239243	0.009494

Here we see that, bias error is reduced to large extent when compared to baseline model. The variance error also decreased when compared to base model.

Due, to all these factors, we can say that the final model has improved from the benchmark model.

CHAPTER – 10

The project was about finding the best separation between the default and non-default clients of the credit card. The project is highly useful for banks and credit card issuing companies as it will help the banks to prevent loss by providing the customer with alternative options such as forbearance or debt consolidation etc. Forbearance refers to the temporary postponement of loan payments granted by lenders. Lenders and other creditors grant forbearance as an alternative to forcing a property into foreclosure or leaving the borrower to default on the loan.

Banks generally performs a charge-off on delinquent credit cards and eats the losses. If only there was a way to predict which customers had the highest probability of defaulting, so that it may be prevented. In this project we used different machine learning models for analysis. The final model is created using random forest classifier and was quite successful in bringing out good model with low bias and low variance.