Contents

List of Tables

List of Figures

1 Chapter 1

Introduction

1.1 Introduction

Credit Score

A credit score is a three-digit number that rates your creditworthiness. FICO scores range from 300 to 850. The higher the score, the more likely you are to get approved for loans and for better rates. A credit score is based on your credit history, which includes information like the number accounts, total levels of debt, repayment history, and other factors. Lenders use credit scores to evaluate your credit worthiness, or the likelihood that you will repay loans in a timely manner.

There are three major credit bureaus in the U.S. :

1. **Equifax**
2. **Experian**
3. **TransUnion**

This trio dominates the market for collecting, analyzing, and disbursing information about consumers in the credit markets. The credit score model was created by the Fair Isaac Corp., now known as FICO, and is used by financial institutions. While other credit scoring system exist, the FICO score is by far the most commonly used.

There are a number factors that go into calculating your FICO credit score, including your repayment history, your debt utilization, the length of your credit history, your credit mix, and any new account openings.Lenders use your credit score to determines whether to approve you for products like mortgages, personal loans, and credit cards, and what interest rates you will pay.

How Credit Scores Work

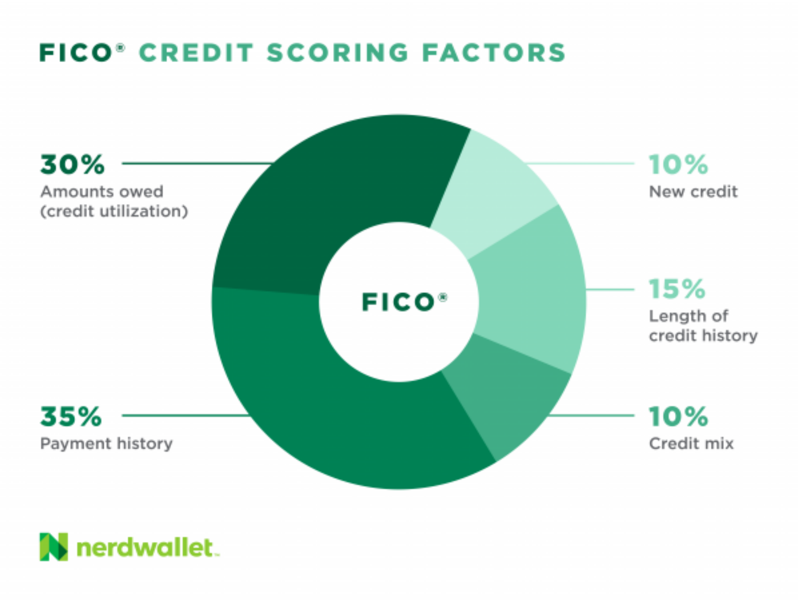
A credit score can significantly affect your financial life. It plays a key role in a lender's decision to offer you credit. Lenders are more likely to approve you for loans when you have a higher credit score, and are more likely to decline your loan applications when you have lower scores. You can also get better interest rates when you have a higher credit score, which can save you money in the long-term. Conversely, a credit score of 700 or higher is generally viewed positively by lenders, and may result in a lower interest rate. Scores greater than 800 are considered excellent. Every creditor defines its own ranges for credit scores and its own criteria for lending. Here are the general ranges for how credit scores are categorized.

1. Excellent: 800–850
2. Very Good: 740–799
3. Good: 670–739
4. Fair: 580–669
5. Poor: 300–579

How Your Credit Score Is Calculated

The three major credit reporting agencies in the U.S. (Equifax, Experian, and TransUnion) report, update, and store consumer's credit histories. While there can be differences in the information collected by the three credit bureaus, five main factors are evaluated when calculating a credit score.

- Payment history (35%)
- Amounts owed (30%)
- Length of credit history (15%)
- Credit mix (10%)
- New credit (10%)



1. Payment history: Your payment history includes whether you've paid your bills on time. It takes into account how many late payments you've had, and how late they were.

2. Amounts owed: Amounts owed is the percentage of credit you've used compared to the credit available to you, which is known as credit utilization.

3. Length of credit history: Longer credit histories are considered less risky, as there is more data to determine payment history.

4. Credit mix: A variety of credit types shows lenders you can manage various types of credit. It can include installment credit, such as car loans or mortgage loans, and revolving credit, such as credit cards.

5. New credit: Lenders view new credit as a potential sign you may be desperate for credit. Too many recent applications for credit can negatively affect your credit score.

1.2 Literature Review

In 2022, Bornvalue Chitambira did the project on credit scoring using the machine learning method. In this project they consider consumer credit scoring instead of corporate credit scoring and they focused is on methods that are currently used in practice by banks such as logistic regression and decision trees and also compare their performance against machine learning method. And also they did the important issues such as dataset imbalance, model overfitting and calibration of model probabilities.they implement algorithms and analyse their performance on credit dataset with 30000 observations from Taiwan, extracted from the University of California Irvine (UCI).

In 2006, Martin VOJTEK did research on credit scoring.The objective of the research was predicting the credit score in the retail segment. They focused on the retail loans how it is effect the credit score. And other objective is to segregate "good borrowers" from "bad borrowers" in terms of their credit worthiness.In This paper they carried out many statistical method and model.

In 2003, Thomas L. paper on Sample Selection Bias in Credit Scoring Models. In this paper they carried out sample selection bias. They analysed the sample selection bias in credit scoring model.

In 2011,Hussein A Abdou did paper on Credit Scoring, Statistical Techniques and Evaluation Criteria.This this project he explained about the How credit scoring has developed in importance , Credit scoring applications etc. The aim of this paper is to carry out the credit scoring applications in various areas, in general, but primarily in finance and banking, in particular. This paper also aims to investigate how credit scoring has developed in importance, and to identify the key determinants in the construction of a scoring model of different statistical techniques and performance evaluation criteria.

In 2005, Bensic et al M., Sarlija, N. and Zekic-Susac they did the research on Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. They did the alogirthm to predict the credit scoring using the statistical techniques. They analysed and implemented the algorithm using advanced techniques like neural networks.

In 1997, Hand, D.J. and Henley, W.E.did project on Statistical Classification Methods in Consumer Credit Scoring. In this project they took the data related consumers and they analysed the their performance in credit score using the statistical techniques. They compared the performance of credit score of consumers and implemented using the machine learning methods.

1.3 Objectives

1. To know which are the factors influencing to the credit score.
2. To determine the effects of credit history age on credit score.
3. To determine the effects on credit score based on the relationship between outstanding debt and minimum number of amount paid.
4. To determine the effects of credit mix on credit score.
5. To predict the credit score based on the performance of the customer.

1.4 Scope of study

Credit score classification helps in the finance sector to make decisions on whether to offer you a mortgage, credit card, auto loan, and other credit products, as well as for tenant screening and insurance on the basis of their transactional activities. They are also used to determine the interest rate and credit limit you receive.

2 Chapter 2

Methodology

2.1 Materials and Methods

2.1.1 About the Data

Credit score classification data is collected from “https://www.kaggle.com/search”. This data contain 1,00,000 rows and 28 columns.

- **ID**: Represents a unique identification of an entry
- **Customer ID**: Represents a unique identification of a person
- **Month**: Represents the month of the year
- **Name**: Represents the name of a person
- **Age**: Represents the age of the person
- **SSN**: Represents the social security number of a person
- **Occupation**: Represents the occupation of the person
- **Annual Income**: Represents the annual income of the person
- **Monthly Inhand Salary**: Represents the monthly base salary of a person
- **Number Bank Accounts**: Represents the number of bank accounts a person holds
- **Number Credit Card**: Represents the number of other credit cards held by a person
- **Interest Rate**: Represents the interest rate on credit card
- **Number of Loan**: Represents the number of loans taken from the bank
- **Delay from due date**: Represents the average number of days delayed from the payment date

- **Number of Delayed Payment** : Represents the average number of payments delayed by a person

- **Changed Credit Limit**: Represents the percentage change in credit card limit

- **Number Credit Inquiries**: Represents the number of credit card inquiries

- **Credit Mix**: Represents the classification of the mix of credits

- **Outstanding Debt**: Represents the remaining debt to be paid (in USD)

- **Credit Utilization Ratio**: Represents the utilization ratio of credit card

- **Credit History Age**: Represents the age of credit history of the person

- **Payment of Min Amount**: Represents whether only the minimum amount was paid by the person (yes or no)

- **Total EMI per month**: Represents the monthly EMI payments (in USD)

- **Amount invested monthly**: Represents the monthly amount invested by the customer (in USD)

- **Payment Behaviour**: Represents the payment behavior of the customer (in USD)

- **Monthly Balance**: Represents the monthly balance amount of the customer (in USD)

- **Credit score**: Represent the score of the credit ('Good' 'Standard' 'Poor')

2.1.2 Statistical Techniques

- **K-nearest neighbours (KNN)**

  K-nearest neighbor (KNN) classifiers are based on learning by analogy whereby given an unknown sample, a KNN classifier carries out a search in the pattern space for the KNN that are closest to the unknown sample in terms of Euclidean distance:

  $$d_{(i)} = ||x_{(i)} - x_{(0)}||$$

  The unknown sample is then assigned the most common class among its KNN For example, given a query point x_0, we find the k training points $x(i)$, $i = 1, ..., k$ closest in distance to x0, and then classify using majority vote among the k neighbors. These classifiers are memory-based, and require no model to be fit. The major advantage of this approach is that it is not required to establish predictive model before classification. The disadvantages are that KNN does not produce a simple classification probability formula and its predictive accuracy is highly affected by the measure of distance and the cardinality k of the neighborhood

- **Multinomial Logistic Regression**

  Multinomial logistic regression is an extension of logistic regression that adds native support for multi-class classification problems.

  Logistic regression, by default, is limited to two-class classification problems. Some extensions like one-vs-rest can allow logistic regression to be used for multi-class classification problems, although they require that the classification problem first be transformed into multiple binary classification problems.

  Instead, the multinomial logistic regression algorithm is an extension to the logistic regression model that involves changing the loss function to cross-entropy loss and predict probability distribution to a multinomial probability distribution to natively support multi-class classification problems

- **Random Forest Classifier**

  The Random forest or Random Decision Forest is a supervised Machine learning algorithm used for classification, regression, and other tasks using decision trees. The Random forest classifier creates a set of decision trees from a randomly selected subset of the training set. It is basically a set of decision trees (DT) from a randomly selected subset of the training set and then It collects the votes from different decision trees to decide the final prediction.

- **Ordinal Regression**

  Ordinal regression is a variant of regression models that normally gets utilized when the data has an ordinal variable. Ordinal variable means a type of variable where the values inside the variable are categorical but in order. We can also find the name of ordinal regression as an ordinal classification because it can be considered a problem between regression and classification.
  To perform ordinal regression we can use a generalized linear model(GLM). GLM has the capability of fitting a coefficient vector and a set of thresholds to data. Let's say in a data set we have observations, represented by length-p vectors X1 through Xn, and against these observations, we have responses Y1 through Yn, in the responses each variable is an ordinal variable. We can think of Y as a nondecreasing vector and apply the length-p coefficient vector and set of thresholds. A set of thresholds is responsible for dividing the real number line into segments, corresponding to the response levels that are similar to the numbers of segments.

  Mathematically we can represent this model as

  $$Pr(yi|x) = (i - w.x)$$

  where,

  - i = inverse link function
  - w = length-p coefficient vector

  - x = set of thresholds with property $\theta_1, \theta_2 \cdot \theta_{k-1}$

- **Backpropagation**

  Backpropagation is the essence of neural net training. It is the practice of fine-tuning the weights of a neural net based on the error rate (i.e. loss) obtained in the previous epoch (i.e. iteration.) Proper tuning of the weights ensures lower error rates, making the model reliable by increasing its generalization.

- **Chi Square Test**

  The distribution of a categorical variable in a sample often needs to be compared with the distribution of a categorical variable in another sample. Here the sample data are displayed in a contingency table. The hypothesis under consideration are given as follows :
  H_0 : The two categorical variables are independent.
  H_1 : The two categorical variables are dependent.
  The test statistic is given as follows :

  $$\chi^2 = \Sigma \left[\frac{(O-E)^2}{E} \right]$$

  where O represents the observed frequency, E is the expected frequency under the null hypothesis and is computed as follows :

  $$E = \frac{row\ total \times column\ total}{sample\ size}$$

  The expected frequency count for each cell of the table must be atleast 5.
  The test procedure is to reject the null hypothesis H_0 if $\chi^2 > \chi^2_{\alpha,(r-1)(c-1)}$, where $\chi^2_{\alpha,(r-1)(c-1)}$ is the upper α^{th} percentile value of the central chi-square distribution with $(r-1)(c-1)$ degrees of freedom, r is the number of rows and c is the number of columns.

- **kruskal-Wallis H Test**

  The Kruskal-Wallis test is one of the non parametric tests that is used as a generalized form of the Mann Whitney U test. It is used to test the null hypothesis which states that 'k' number of samples has been drawn from the same population or the identical population with the same or identical median. If Sj is the population median for the jth group or sample in the Kruskal-Wallis test, then the null hypothesis in mathematical form can be written as $S1 = S2 = \ldots.. = Sk$. Obviously, the alternative hypothesis would be that Si is not equal to Sj. This means that at least one pair of groups or samples has different pairs.

the Kruskal Wallis test formula for comparing medians of more than two groups. Below given is the formula for the test:

$$H = \frac{12}{N(N+1)} \Sigma \frac{R_i^2}{n_i} - 3(N+1)$$

Where,

- K = number of groups used for comparison
- N = total size of the sample
- ni = i^{th} group's sample size
- Ri = total of the ranks related to i^{th} group

- **Analysis Of Variance**

An ANOVA test is a statistical test used to determine if there is a statistically significant difference between two or more categorical groups by testing for differences of means using a variance.
Another key part of ANOVA is that it splits the independent variable into two or more groups.
Assumption On Anova

1. An ANOVA can only be conducted if there is no relationship between the subjects in each sample. This means that subjects in the first group cannot also be in the second group
2. The different groups/levels must have equal sample sizes.
3. An ANOVA can only be conducted if the dependent variable is normally distributed so that the middle scores are the most frequent and the extreme scores are the least frequent.

4. Homogeneity of variance means that the deviation of scores is similar between populations.

- **Box Plot**
  A box and whisker plot—also called a box plot—displays the five-number summary of a set of data. The five-number summary is the minimum, first quartile, median, third quartile, and maximum. In a box plot, we draw a box from the first quartile to the third quartile. A vertical line goes through the box at the median. The whiskers go from each quartile to the minimum or maximum.

- **Violin Plot**
  A violin plot is a hybrid of a box plot and a kernel density plot, which shows peaks in the data. It is used to visualize the distribution of numerical data. Unlike a box plot that can only show summary statistics, violin plots depict summary statistics and the density of each variable.

- **Normal Quantile-Quantile (Q-Q) Plot**
  When the quantiles of two variables are plotted against each other, then the plot obtained is known as quantile – quantile plot or qqplot. This plot provides a summary of whether the distributions of two variables are similar or not with respect to the locations.

- **Validation Curve**
  A Validation Curve is an important diagnostic tool that shows the sensitivity between changes in a Machine Learning model's accuracy with changes in hyperparameters of the model. The validation curve plots the model performance metric (such as accuracy, F1-score, or mean squared error) on the y-axis and a range of hyperparameter values on the x-axis. The hyperparameter values of the models are typically varied on a logarithmic scale, and the model is trained and evaluated using a cross-validation technique for each hyperparameter value.

  Two curves are present in a validation curve – one for the training set score and one for the cross-validation score. By default, the function for validation curve, present in the scikit-learn library performs 3-fold cross-validation. A validation curve is used to evaluate an existing model based on hyper-parameters and is not used to tune a model. This is because, if we tune the model according to the validation score, the model may be biased towards the specific data against which the model is tuned; thereby, not being a good estimate of the generalization of the model.

3 Chapter 3

Results and Discussion :

3.1 Data Cleaning

3.1.1 Filled Missing Value

This data set contains unwanted entries,symbols and missing value. All these data cleaning was done by filling the missing value using statistical method like Mean,Median, Mode etc.

3.1.2 Outlier

Box plot represent summarisation of each numerical variable.

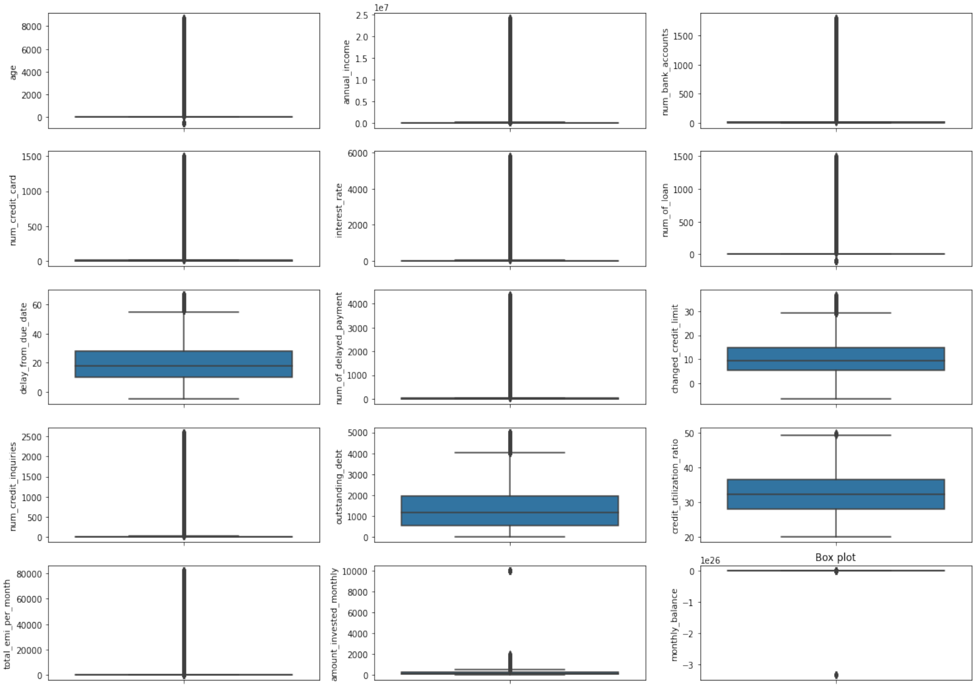


Figure 1: Summarisation numerical variable

From this box plot we observed that these numerical variable contains the outliers. Using inter quartile method, some of the outliers are removed and some of the outliers are filled.

3.2 Bivariate Data Analysis

3.2.1 Check in balance

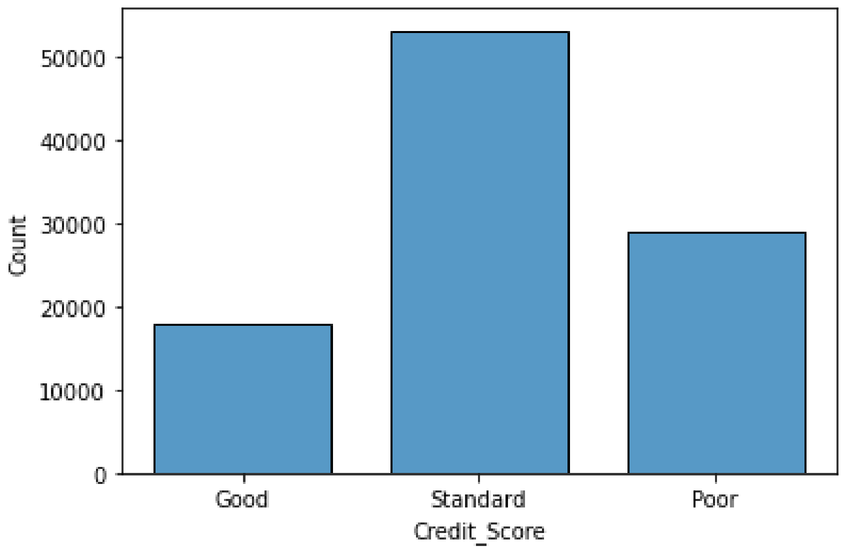
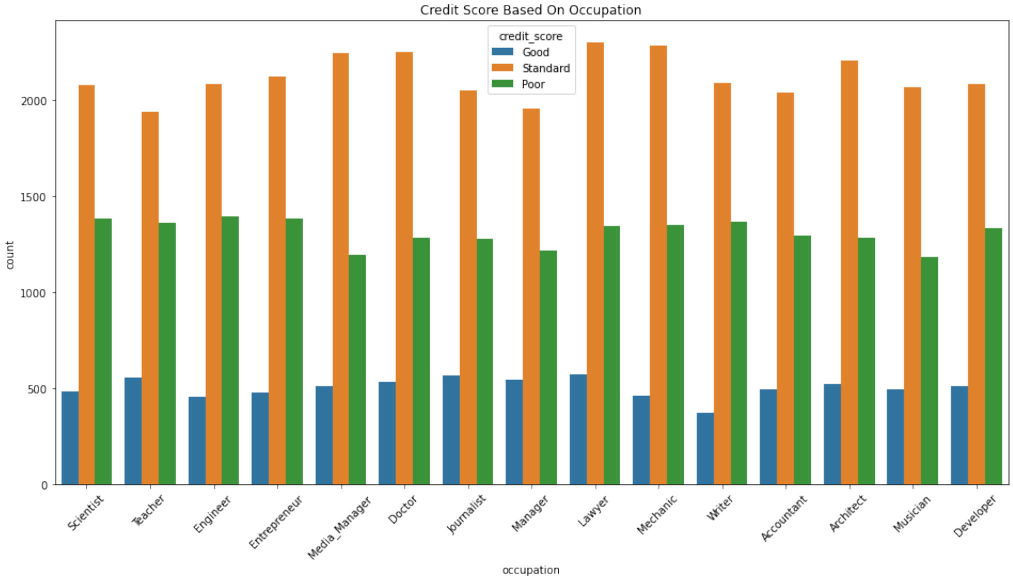


Figure 2: Check the balance

From this above graph, we can conclude that data is not equally balanced .

3.2.2 To know the credit score on different occupation



From above graph we observe that, there's not much difference in the credit scores of all occupations mentioned in the data.

3.2.3 To determine whether the annual income effects the credit score.

Checking the normality

Shapiro Test

H_0: Sample is from the normal distributions.
H_1: Sample is not from the normal distributions.

The value obtained as follows:
P-value: $0.000e$-127
Test Statistic: 0.87737

Here we can observe that, obtained p-value is less than 0.05. Hence we reject H_0 and accept H_1. Therefore we conclude that Sample is not from normal distribution.

Hence alternative Non parametric test Kruskal-Wallis H test used.

Kruskal-Wallis H test

H_0:Median income of three different group of credit score are equal.
H_1:Median income of three different group of credit score are not equal.

The value obtained as follows:
P-value: $1.778e$-305
Test Statistic:1403.4251

Here we can observe that, obtained p-value is less than 0.05. Hence we reject H_0 and accept H_1. Therefore we conclude that median income of three different group of credit score are not equal. Hence the factor income effects the credit score.

3.2.4 To Check whether the credit score effects the number of bank account

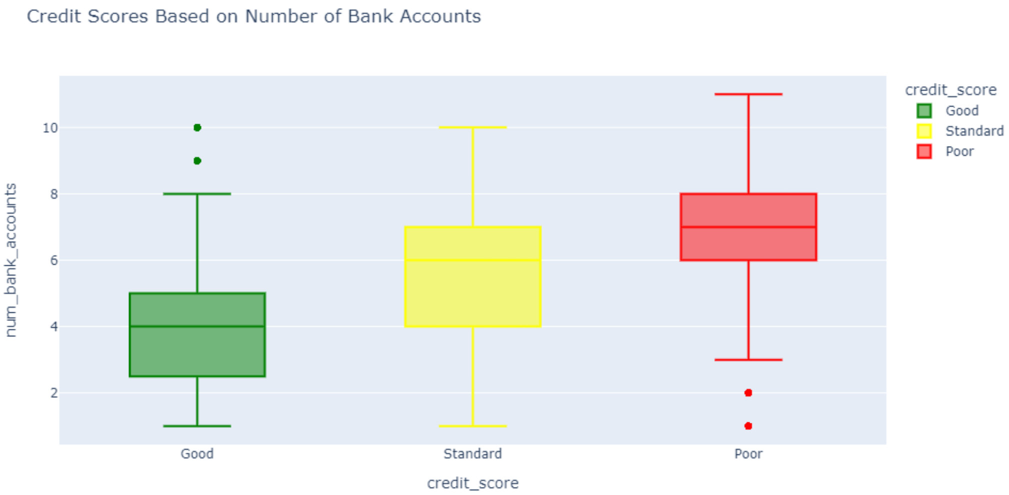


Figure 3: Credit score effects on number of bank account

From this box plot we observed that, who are having the number of bank account between range 3 to 5 their credit score is good. And between range 4 to 7, their credit score is standard. And who are having the more than 6 bank account their credit score is poor.

3.2.5 To Check whether the credit score on effect the number of credit card

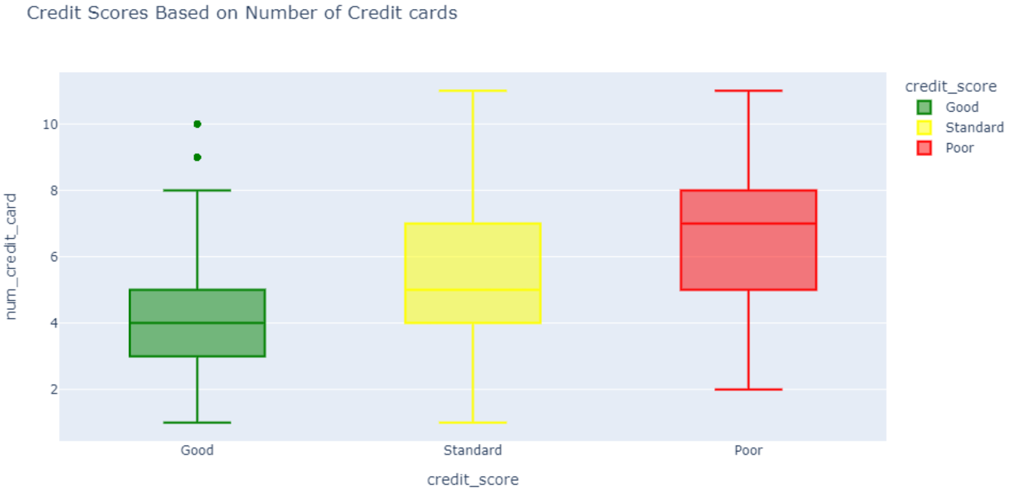


Figure 4: Credit score effects on number of credit card

From this box plot we observe that, who are having the number of credit card between range 3 to 5 their credit score is good. And between range 4 to 7, their credit score is standard. and who are having the more then 7 credit card their credit score is poor.

3.2.6 To determine whether the interest rate effects credit score.

Normal Q-Q Plot

Checking the normality

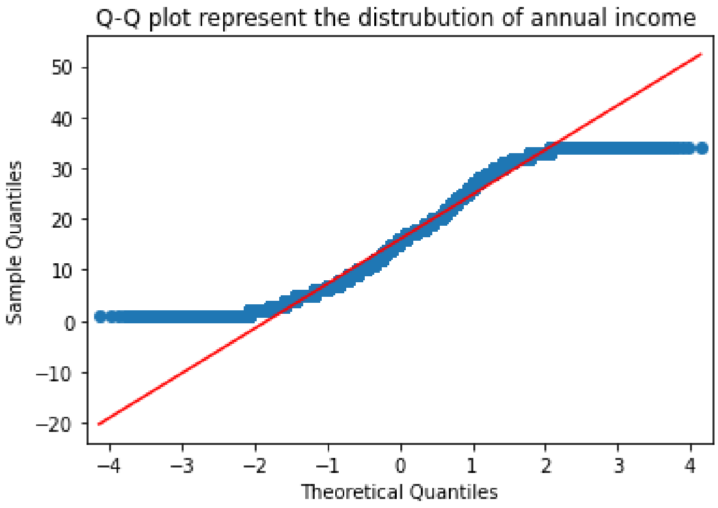


Figure 5: Q-Q plot on factor interest rate

From the Normal Q-Q plot we observe that, data is not distributed normally. Hence Non parametric test Kruskal-Wallis H test used.

Kruskal-Wallis H test

H_0:Median interest rate of three different groups credit score are equal.
H_1:Median interest rate of three different groups credit score are not equal.

The value obtained as follows:
P-value: 0.00
Test Statistic:14595.1822

Here we can observe that the obtained p-value is less than 0.05. Hence we reject H_0 and accept H_1. Therefore we conclude that median of interest rate of three different groups credit score are not equal. Hence factor interest rate effects the credit score.

3.2.7 To Check whether the credit score effects the number of loan

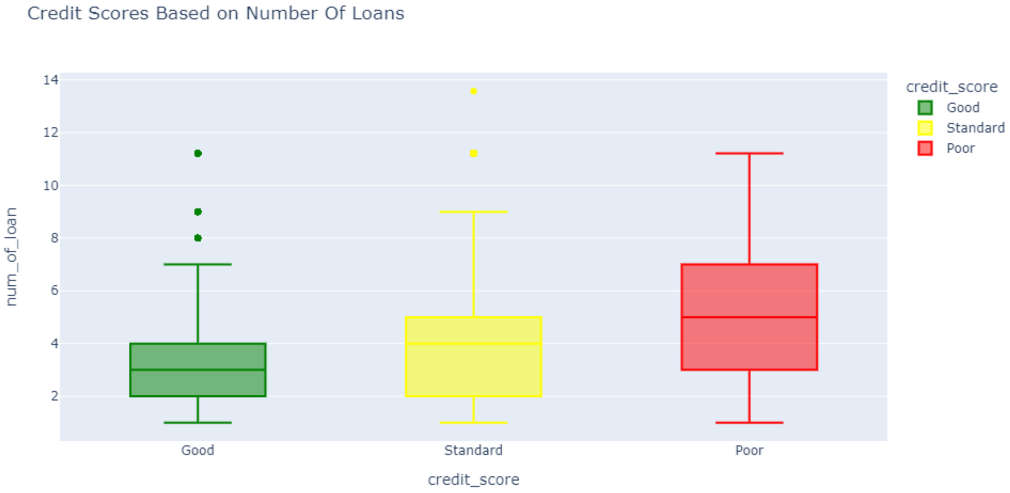


Figure 6: Credit score effects on number of loan

From this box plot we observe that, who are having the number of loan between range 2 to 4 their credit score is good. And between range 4 to 5, their credit score is standard and who are having the more than 5 loans their credit score is poor.

3.2.8 To Check whether the credit score effects the delay from the due date

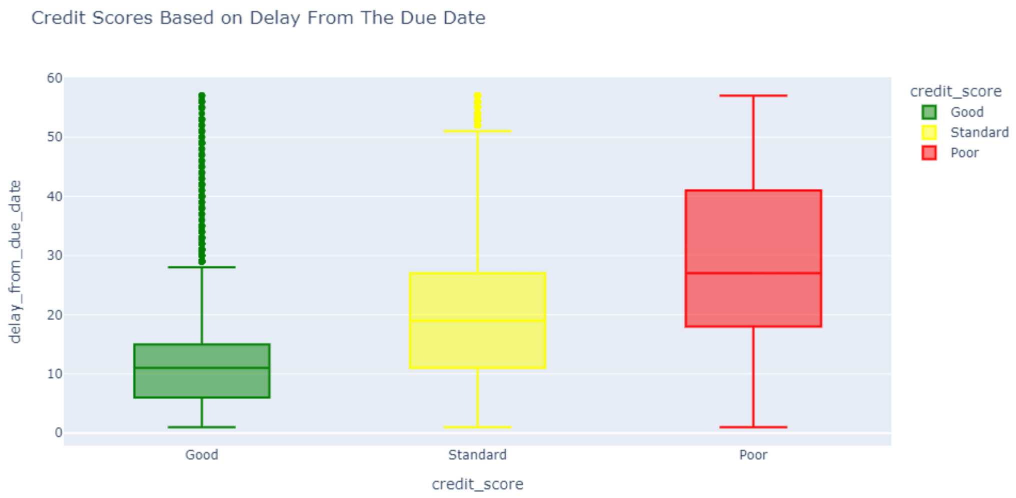


Figure 7: Credit score effects on dealy from due date

From this box plot we observe that, the credit score will be good if the delay from the due date is less and it will be poor if the delay from the due date is more.

3.2.9 To check whether the outstanding debt effects the credit score.

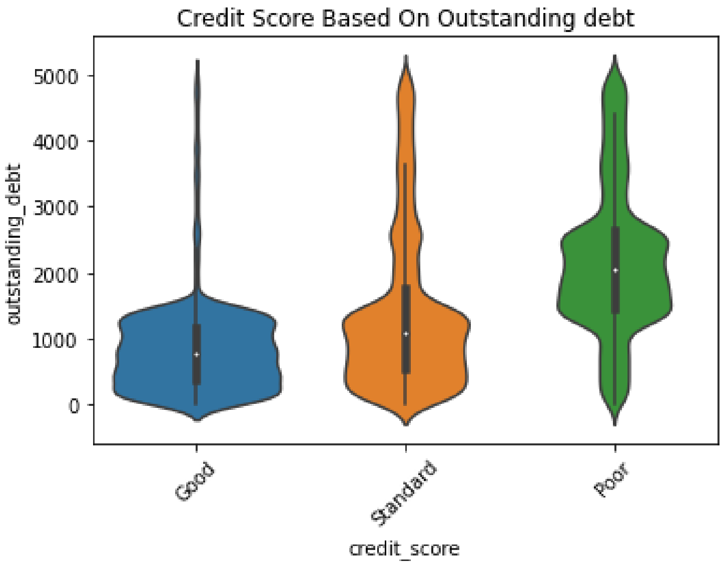


Figure 8: Credit score effects on Outstanding debt

From this violin plot we observe that, lesser the outstanding debt have their good and standard credit score and larger the outstanding debt have their poor credit score.

3.2.10 To determine whether the payment of minimum amount effects the credit score.

Chi Square Test

Table 1: The following table shows the group of credit score on payment of minimum amount.

credit score\payment of min amount	Yes	No	Not Mentioned
Good	1051	5525	992
Standard	20045	7993	3759
Poor	15302	7993	3759

The following hypothesis are:

H_0: There is no association between payment of minimum amount and credit score.
H_1:There is association between payment of minimum amount and credit score.

The value obtained as follows:
P-value: 0.00
Test Statistic:11838.2447

Here we can observe that, obtained p-value is lesser than 0.05. Hence we reject the null hypothesis. There is association between payment of minimum amount and credit score. Hence credit score effect based on the payment of minimum amount.

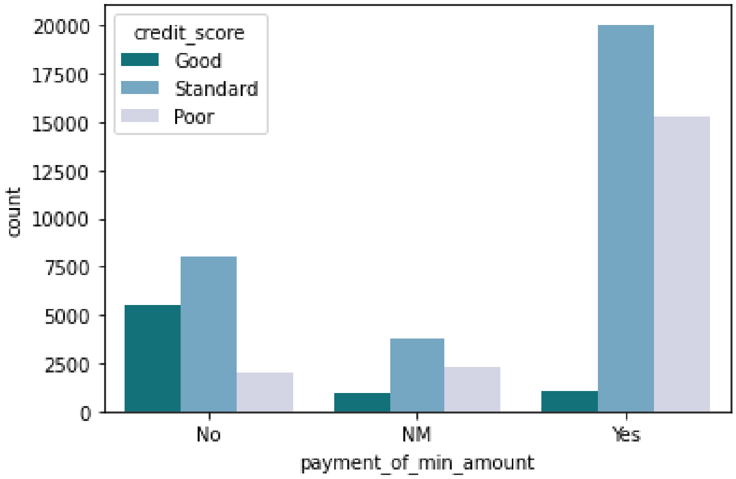


Figure 9: Bar graph representation of payment of minimum amount

From this bar graph we observe that, factor payment of minimum amount is ”yes” then number of good credit score is less but number of standard credit score is high as well as number of poor credit score is high. And factor payment of minimum amount is ”no” then number of good and standard is high as well as number of poor credit score is low.

3.2.11 To know effects on credit score based on the relationship between the outstanding debt and payment of minimum amount.

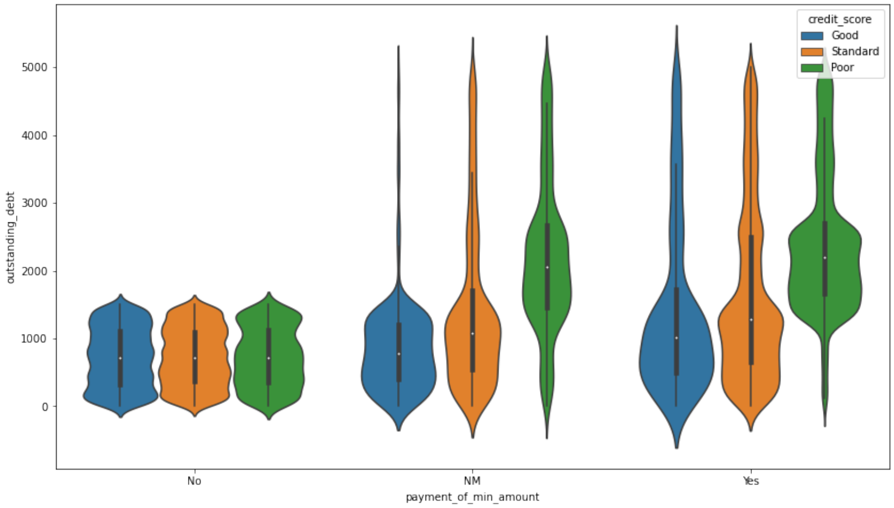


Figure 10: Relationship between outstanding debt and minimum payment amount

From this violin plot we observe that, who are paying the minimum amount on outstanding debt their credit score is poor.And also who are not paying the minimum amount on outstanding debt their credit score is Good.

3.2.12 To check whether the factor emi per month effects the credit score.

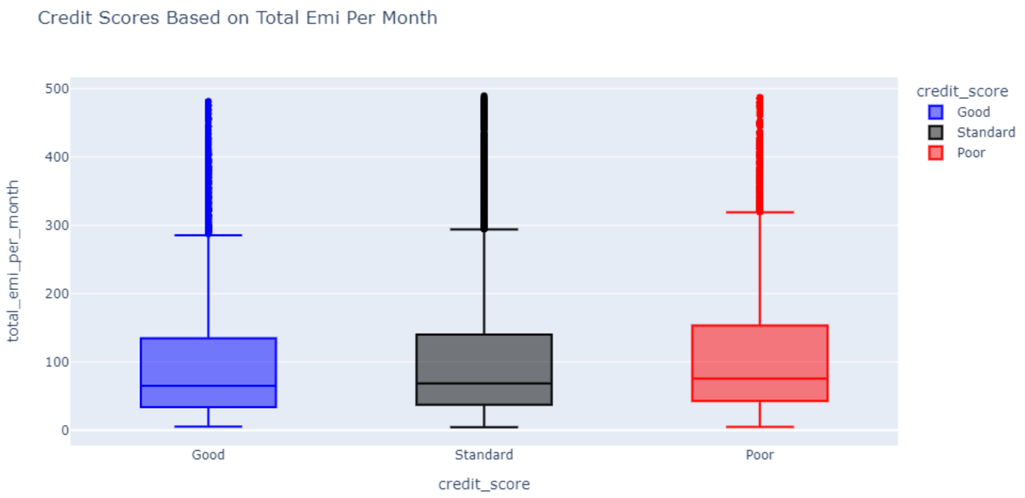


Figure 11: Credit score effects on EMI per month

By visualizing the above graph we observe that, emi per month is not effects the credit score.

3.2.13 To check whether the credit utilization ratio effects the credit score.

Analysis Of Variance

The hypothesis as follows:

H_0: There is no mean difference between the group of credit score based on credit utilization ratio.
H_1: There is mean difference between the group of credit score based on credit utilization ratio.

The value obtained as follows:

P-value: 1.2842e-28
Test Statistic:41.2251

Here we can observe that the obtained p-value is lesser than 0.05. Hence we reject the null hypothesis. There is mean difference between the group of credit score based on credit utilization ratio. Hence credit utilization ratio effect the credit score.

3.2.14 To check whether the factor Number of credit inquiries effects credit score.

Ordinal Regression.

The following hypothesis

H_0:There is no association between the Number of credit inquiries and credit score.

H_1:There is association between the Number of credit inquiries and credit score.

The value obtained as follows: P-value: 0.00
Test Statistic:108.106

Here we can observe that, obtained p-value is lesser than 0.05. Hence we reject the null hypothesis. There is association between Number of credit inquiries and credit score.

Ordered Model analysis, we investigated the relationship between the credit score and three predictor variables, having more credit inquiries and being categorized as Standard/Poor credit status are associated with higher credit scores, while being categorized as Good/Standard credit status is associated with lower credit scores. So the factors influencing the credit score.

3.2.15 To check whether the factor Payment Behaviour effects the credit score.

Chi Square Test

Table 2: The following table shows the group of credit score on payment behaviour.

P.B	high spent	high spent	high spent	low spent	low spent	low spent
/Credit Score	large value	medium value	small value	high value	medium value	small valu
Good	1567	1790	542	445	765	2323
Standard	5011	7629	3380	2570	2306	10901
Poor	2177	4403	1936	1365	1796	7985

The following hypothesis are:

H_0: There is no association between payment of minimum amount and credit score.
H_1:There is association between payment of minimum amount and credit score.

The value obtained as follows:
P-value: 0.00
Test Statistic:11838.2447

Here we can observe that, the obtained p-value is lesser than 0.05. Hence we reject the null hypothesis. There is association between payment behaviour amount and credit score. Hence credit score effect based on the payment behaviour.

3.2.16 To check whether the factor amount invested by monthly effects the credit score.

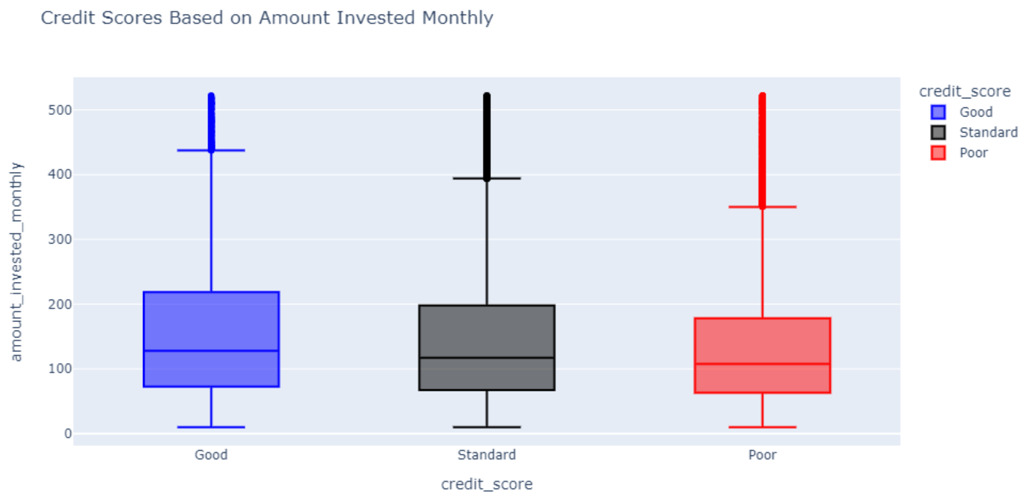


Figure 12: Credit score effects on amount invested by monthly

By visualizing the above graph we observe that, investing the amount by monthly is not effects the credit score.

3.2.17 To know whether the factor Number of delayed payment effects the credit score.

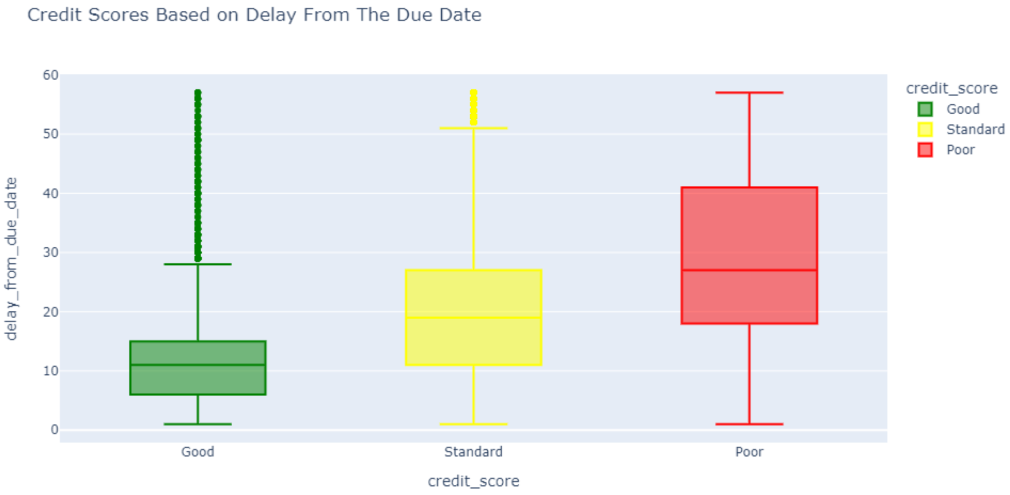


Figure 13: Credit score effects on number of delayed payment

From this box plot observed we that, number of payment delayed is high their credit score is poor and also number of payment delayed is low their credit score is good.

3.2.18 To know the relationship between the monthly balance and credit score.

Ordinal Regression.

The following hypothesis

$H0$:There is no association between the monthly balance and credit score.

$H1$:There is association between the monthly balance and credit score.

The value obtained as follows: P-value: 0.0023e-28
Test Statistic:-94.862

Here we can observe that, obtained p-value is lesser than 0.05. Hence we reject the null hypothesis. There is association between monthly balance and credit score.

3.2.19 To know the relationship between the credit history age and credit score.

Ordinal Regression.

The following hypothesis

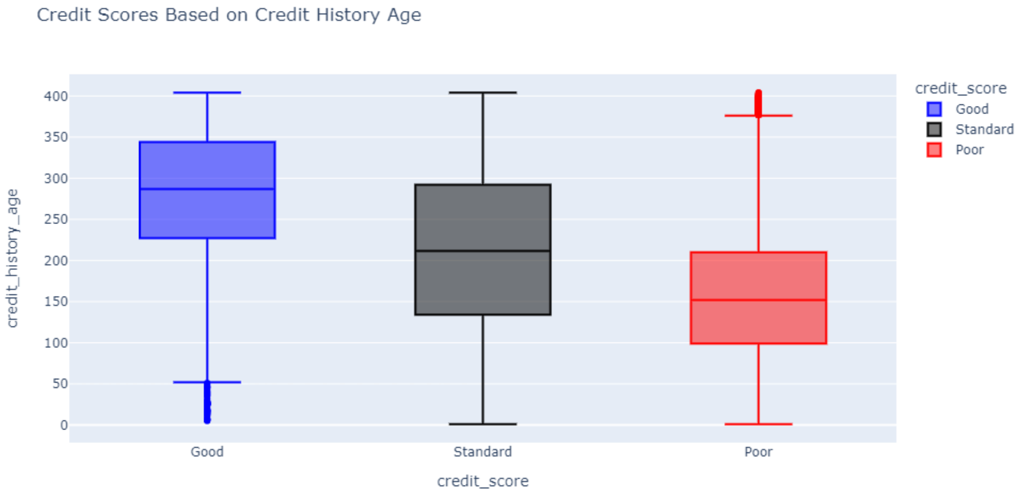
H_0:There is no association between the credit history age and credit score.

H_1:There is association between the credit history age and credit score.

The value obtained as follows: P-value: 0.00
Test Statistic:-94.862

Here we can observe that, obtained p-value is lesser than 0.05. Hence we reject the null hypothesis. There is association between credit history age and credit score.

By Ordered Model analysis, we examined the relationship between the credit score and three predictor variables. Longer credit history is associated with higher credit scores and smaller credit history is associated with lower credit score.



By looking the graph, who are have long credit history age their credit score is good. And who are have the small credit history age their credit score is poor.

3.2.20 To determine whether the factor credit mix effects the credit score.

Chi Square Test

The following hypothesis are:

H_0:There is no association between credit mix and credit score.
H_1:There is association between credit mix and credit score.

Table 3: The following table shows the group of credit score on credit mix.

credit score\credit mix	**Good**	**Standard**	**Bad**
Good	6355	1026	187
Standard	9680	16893	5224
Poor	5518	5906	8239

The value obtained as follows:
P-value: 0.00
Statistics:13534.4605

Here we can observe that the obtained p-value is lesser than 0.05. Hence we reject the null hypothesis and accept $H1$. There is association between credit mix and credit score. Hence factor credit mix effect the credit score.

3.3 Data Splitting

The data is splitted into train and test in the ratio 80:20. Only effecting factors are considered. The shape of the data for train and test after splitting is as shown below.

x train shape:(47222, 23)

y train shape:(47222,)

x test shape:(11806, 23)

y test shape:(11806,)

Following featurization are only applied to the train data, since test data is considered as unseen data. Test data will be transformed using the featurization objects of train data.

3.4 Predictive Models

Label ('Good','standard','Poor') is the target variable, that has to be predicted. The below feature transformation are done before fitting the model.

- Categorical variable are encoded using label encoder and one hot encoder.
- Standardizing numerical variables.

3.5 Model Building

3.5.1 Logistic Regression

Hyper-parameter tuning is done using Randomised search cross validation.

Result:
Test accuracy:0.66
Train accuracy:0.67
Test F1-score:0.60
Train F1-score:0.62

Below confusion matrix of logistic regression represents test data:

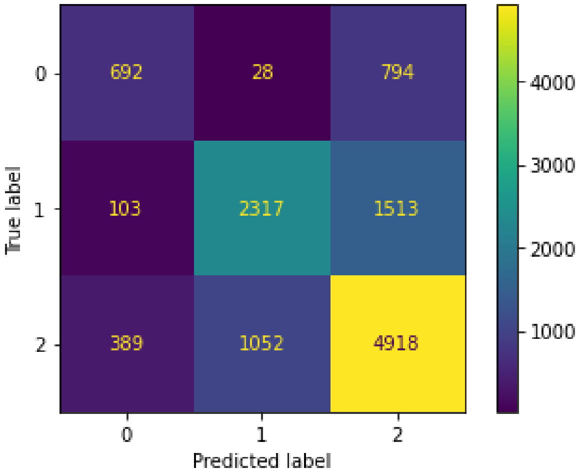


Figure 14: Confusion matrix of Logistic Regression

- 692 data points which belongs to good credit score are classified correctly.

- 28 data points which is belongs to standard are misclassified as good credit score.
- 794 data points which is belongs to poor credit score are misclassified as good credit score.
- 103 data points which is belongs to standard credit score are misclassified as good credit score.
- 2317 data points which belongs to standard credit score are classified correctly.
- 1513 data points are belongs to standard credit score are misclassified as as poor credit score.
- 389 data points which belongs to poor credit score are misclassified as good credit score.
- 1052 data points are belongs to poor credit score are misclassified as standard credit score.
- 4918 data points are belongs to poor credit score are classified correctly.

3.5.2 K-Nearest Neighbour

In KNN, used the elbow method to obtain parameter and to avoid the overfitting validation curve is used to measure the influence of a single hyperparameter.

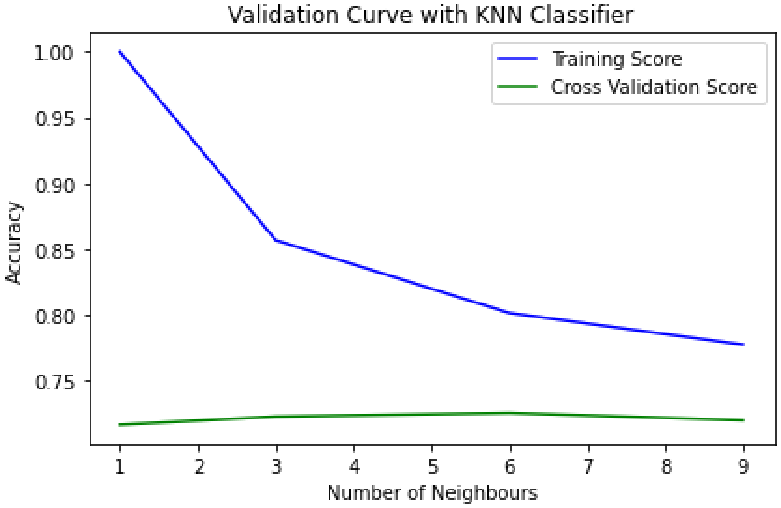


Figure 15: Validation curve of KKN

From this above graph we observed that training score and cross validation score having right accuracy score when n-neighbour= 9.

Result:
Test accuracy:0.74
Train accuracy:0.79
Test F1-score:0.71
Train F1-score:0.76

Below confusion matrix of KNN represents test data:

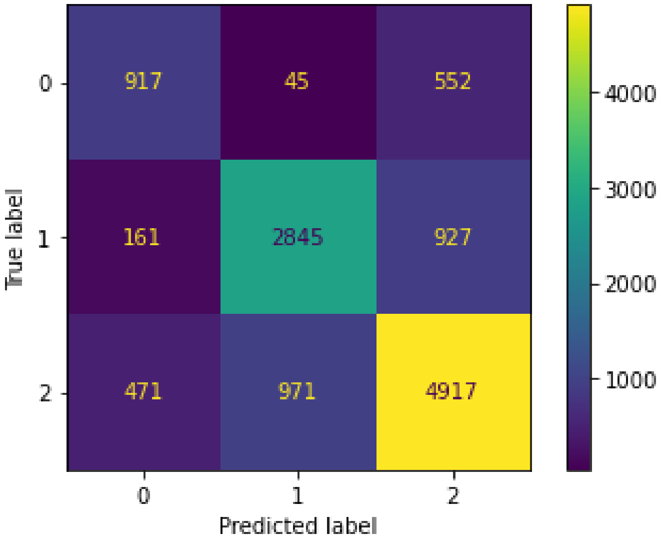


Figure 16: Confusion matrix of KNN

- 917 data points which belongs to good credit score are classified correctly.
- 45 data points which is belongs to standard are misclassified as good credit score.
- 552 data points which is belongs to poor credit score are misclassified as good credit score.
- 161 data points which is belongs to standard credit score are misclassified as good credit score.
- 2845 data points which belongs to standard credit score are classified correctly.
- 927 data points are belongs to standard credit score are misclassified as as poor credit score.
- 471 data points which belongs to poor credit score are misclassified as good credit score.
- 971 data points are belongs to poor credit score are misclassified as standard credit score.

- 4917 data points are belongs to poor credit score are classified correctly.

3.5.3 Random Forest Classifier

Hyper-parameter tuning is done using Randomised search cross validation.

Result:
Test accuracy:0.81
Train accuracy:0.80
Test F1-score:0.78
Train F1-score:0.80

Below confusion matrix of Random forest classifier represents test data:

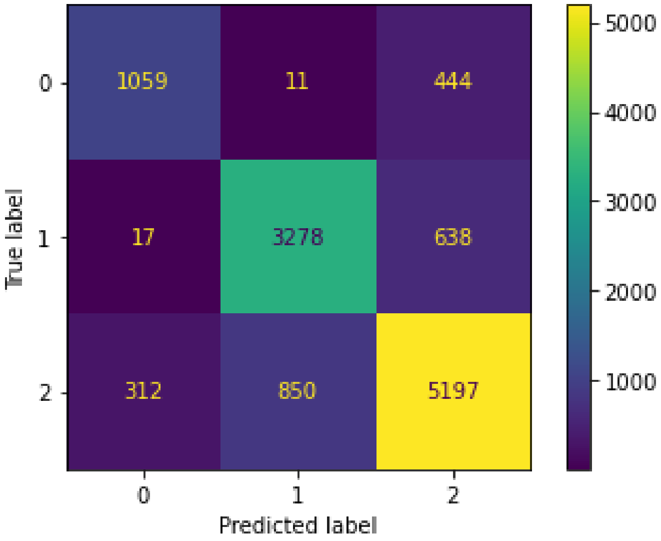


Figure 17: Confusion matrix of Random Forest Classifier

- 1059 data points which belongs to good credit score are classified correctly.
- 11 data points which is belongs to standard are misclassified as good credit score.

- 444 data points which is belongs to poor credit score are misclassified as good credit score.
- 17 data points which is belongs to standard credit score are misclassified as good credit score.
- 3278 data points which belongs to standard credit score are classified correctly.
- 638 data points are belongs to standard credit score are misclassified as as poor credit score.
- 312 data points which belongs to poor credit score are misclassified as good credit score.
- 850 data points are belongs to poor credit score are misclassified as standard credit score.
- 5197 data points are belongs to poor credit score are classified correctly.

3.5.4 Backpropagation Algorithm

Learning rate =0.5
hidden layers=2

The below graph represent the mean square error:

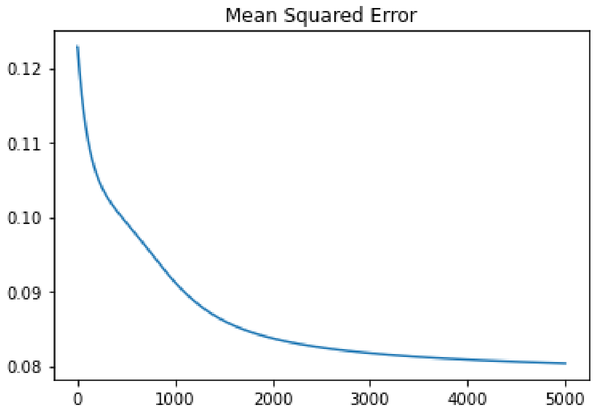


Figure 18: Graph represent the mean square error

From this above graph we observed that as number of iteration increases mean square error decreased.

The below graph represent the accuracy :

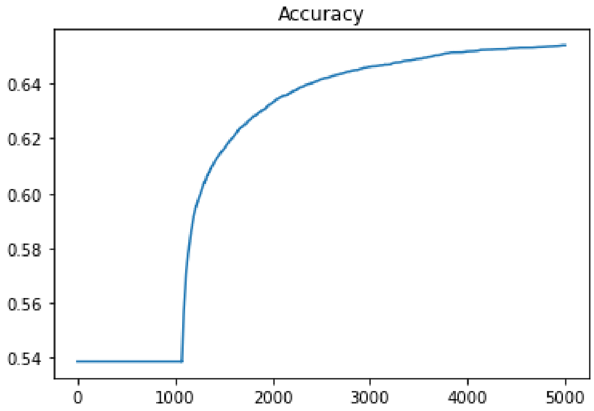


Figure 19: Graph represent the accuracy

From the above graph we observed that number of iteration is 5000 got good accuracy i.e 0.65

3.6 Model Comparison

Totally 3 predictive models and 1 neural net are trained and the comparison summary of the metrics of those models is given below table.

Table 4: The following the table represent the model comparison.

Model name	**Accuracy**	**f1 score**	**Precision**	**Recall**
KNN	73.81	69.8	70.1	70.2
Logistic Regression	67.2	63.2	65.3	61.01
Random Forest	80.8	78.7	79.2	77.6
Backpropagation	65.04	-	-	-

3.7 Conclusion

From the above model comparison, accuracy of Logistic regression and BacKpropagation is almost same. Random forest classifier have more accuracy compared to other model. Even though the performance of K-nearest neighbours is better compared to logistic regression and Backpropogation. But KNN is simplest model.

By comparing the performance of the model, we can say that Random forest classifier is the best model.

4 Chapter 4

Conclusion

4.1 Conclusion

1. Based on the analysis, it can be concluded that the factors "amount invested by monthly" and "EMI per month" do not have a significant effect on the credit score. However, it is important to note that the other factors under consideration do impact the credit score.

2. Results indicate that the age of credit history, Good/Standard credit status, and Standard/Poor credit status all play important roles in determining credit scores. Longer credit history is associated with higher credit scores, indicating that individuals with a more established credit history tend to have better credit scores.

3. Result indicate that there is a relationship between the minimum payment of amount and outstanding debt, which impact the credit score. That individuals who make minimum payments on their outstanding debts tend to have poorer credit scores, while those who do not make minimum payments have credit scores falling into the categories of good and standard.

4. Impact of credit mix on credit scores is essential for individuals and financial institutions.Individuals with a "Bad" credit mix tend to have credit scores falling into the "Poor" category.Individuals with a "Good" credit mix tend to have credit scores categorized as "Good."

5. Based on the model built for credit score classification, the Random Forest classifier achieved the highest accuracy among the models evaluated. This indicates that the Random Forest algorithm performed the best in predicting credit scores compared to other models considered.

4.2 Summary

Credit score classification data is a secondary data collected from Kaggle.com. This data contains 100000 rows and 28 columns. Credit score classification is the process of predicting or categorizing individuals into different credit score groups based on various factors. Main aims to help lenders and financial institutions make informed decisions about granting loans, credit cards, mortgages, or other forms of credit.

In this Credit score classification:Unveiling patterns through machine learning project, main goal was to classifies credit scores based on various factors. To achieve this, undertook a exploratory data analysis (EDA) to gain insights into the dataset. The EDA phase allowed us to identify missing values and potential outliers, enabling us to prepare a clean and reliable dataset for analysis. Various statistical methods used to determine the key factors influencing credit scores. From all this get to know that the main facotr like credit history age credit mix influencing the credit score. Along with other factor collectively effects the credit score. Further for the credit score classification, explored different machine learning models, including logistic regression, K nearest neighbours, Backpropagation, and Random Forest. Random Forest model emerged as the top-performing algorithm compared to other model.

5 Bibliography

1. www.investopedia.com/terms/c/credit_score.asp

2. www.onlinelibrary.wiley.com/

3. www.consumerfinance.gov

4. www.sciencedirect.com/science/article/abs/pii/S1876735416300101

5. www.analyticsvidhya.com

6. www.wallstreetmojo.com/kruskal-wallis-test/

7. www.g2.com/articles/statistical-analysis-methods

8. www.neptune.ai/blog/backpropagation-algorithm-in-neural-networks-guide

9. www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/

10. www.geeksforgeeks.org

11. www.askpython.com

12. Bornvalue Chitambira (2022): *Credit scoring using machine learning method.*-Vasteras, Sweden, SE-721 23

13. Martin VOJTEK (2006): *Prediction of credit score in the retail segment.*, 23(05)

14. Rashmi Malhotra , D. K. Malhotra (2003): *Evaluating consumer loans.* Omega, 31(2):83–96.

15. Hussein A Abdou (2011): *Credit Scoring, Statistical Techniques and Evaluation Criteria*, 34(3), 333 375

6 Appendix

```
import pandas as pd
import numpy as np
import math
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from scipy import stats as st

import warnings
warnings.filterwarnings("ignore")
from sklearn.preprocessing import StandardScaler,LabelEncoder,OneHotEncoder
from sklearn.model_selection
import train_test_split,cross_val_score,
RandomizedSearchCV,
StratifiedKFold
from sklearn.pipeline import Pipeline
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import plot_tree
from sklearn.metrics import precision_recall_curve
from optuna.visualization import plot_optimization_history
from sklearn.metrics import classification_report,
precision_score, recall_score,accuracy_score, f1_score,ConfusionMatrixDisplay

* cleaning the data
# to check the balance of the data
sns.histplot(df["Credit_Score"],shrink=0.75)
df[df["Monthly_Balance"]=="__-333333333333333333333333333__"].head(2)
df["Occupation"]=df["Occupation"].replace("_______",np.nan)
df["Amount_invested_monthly"]=df["Amount_invested_monthly"]
.replace("__10000__",10000)
df["Changed_Credit_Limit"]=df["Changed_Credit_Limit"]
```

```
.replace("_",np.nan)
df["Credit_Mix"]=df["Credit_Mix"].replace("_",np.nan)
df.columns = df.columns.str.lower()
df=df.drop(columns=["id","ssn","name","monthly_inhand_salary"])
df.head(2)

# listing the categorical column and numerical column
cat_col=["month",
        "occupation",
        "type_of_loan",
        "credit_mix",
        "payment_of_min_amount",
         "credit_history_age",
         "payment_behaviour",
        "credit_score"]

num_col=["age","annual_income","monthly_inhand_salary","num_bank_accounts",
        "num_credit_card","interest_rate","num_of_loan",
        "delay_from_due_date",
        "num_of_delayed_payment","changed_credit_limit",
        "num_credit_inquiries","outstanding_debt",
        "total_emi_per_month","amount_invested_monthly",
        "monthly_balance"]
object_but_num_col=[col for col in df.columns
if df[col].dtypes=="object" and col in num_col]
def remove_underscore(val):
    if "_" in str(val):
        return val.strip("_")
    return val
for col in object_but_num_col:
    df[col]=df[col].apply(remove_underscore)
    #df[col]=df[col].apply(float)
    df[col]=df[col].astype(float)

*filling the missing value
index=df[df["payment_behaviour"]=="!@9#%8"].index
df.loc[index,"payment_behaviour"]=np.nan
data=df[-df["payment_behaviour"].isna()]
```

```
ser=df.groupby("customer_id")["payment_behaviour"]
.agg(st.mode).apply(lambda x:x[0][0])
df.drop("payment_behaviour",axis=1,inplace=True)
df=pd.merge(left=df,right=ser,how="left",
left_on="customer_id",right_index=True)

# credit card histroy age a=year and b=months -> a*12+b
def convert_credit_history(history_age):
    if str(history_age)=="nan":
        return np.nan
    else:
        year=int(history_age.split(" ")[0])
        month=int(history_age.split(" ")[3])
        return (12*year)+month
df["credit_history_age"]=df["credit_history_age"]
.apply(convert_credit_history)

df["occupation"]=df["occupation"].fillna(method="ffill")
df[(df["type_of_loan"]=="No loan") & (df["num_of_loan"]==0)]
df["type_of_loan"]=df["type_of_loan"].replace(np.nan,"No loan")

* Checking the outliers
plt.figure(figsize=(20,20))
for i in range(len(num_col)):
    plt.subplot(7,3,i+1)
    sns.boxplot(y=num_col[i])
plt.title("Box plot")

col=["age","annual_income","num_bank_accounts","num_credit_card",
     "interest_rate","num_of_loan","num_of_delayed_payment",
     "changed_credit_limit",
     "num_credit_inquiries","outstanding_debt",
     "total_emi_per_month","amount_invested_monthly","monthly_balance"]

for col in col:
    index=df[df[col]<=0].index
    df.loc[index,col]=np.nan
    df[col]=df[col].fillna(df.groupby("customer_id")[col].transform("mean"))
```

```
col=["num_bank_accounts","num_of_loan","num_of_delayed_payment",
"num_credit_inquiries",
     "total_emi_per_month"]

for col in col:
    df[col]=df[col].fillna(df[col].mean())

percentile25 = df['num_bank_accounts'].quantile(0.25)
percentile75 = df['num_bank_accounts'].quantile(0.75)
iqr=percentile75-percentile25
print("upper bound",percentile75 + (1.5 * iqr))
print("lower bound",percentile25 - (1.5 * iqr))
df=df[(df["num_bank_accounts"]>0) &
(df["num_bank_accounts"]<15)]

plt.figure(figsize=(8,5))
sns.distplot(df["annual_income"])
plt.title("histogram represent the annual income")
plt.show()

percentile25 = df['annual_income'].quantile(0.25)
percentile75 = df['annual_income'].quantile(0.75)
iqr=percentile75-percentile25
print("annual income")
print("upper bound",percentile75 + (1.5 * iqr))
print("lower bound",percentile25 - (1.5 * iqr))
df=df[(df["annual_income"]>0) &
(df["annual_income"]<=150228)]

* bivariate Analysis
plt.figure(figsize=(16,8))
sns.countplot(x=df["occupation"],hue=df["credit_score"])
plt.title("Credit Score Based On Occupation")
plt.xticks(rotation = 45)

from scipy.stats import shapiro
statistic,p_value=shapiro(df["annual_income"])
```

```
print("p_value:",p_value)
print("statistic:",statistic)
if(p_value<0.05):
    print("reject the null hypothesis")
else:
    print("accept the null hypothesis")

# checking the normalities
import statsmodels.api as sm
sm.qqplot(df["annual_income"],line="s")
plt.title("Q-Q plot represent the
distrubution of annual income ")

The Kruskal-Wallis H test¶
from scipy.stats import kruskal
standard,good,bad=df.groupby("credit_score")["annual_income"]
standard=pd.DataFrame({"standard":standard[1]})
good=pd.DataFrame({"good":good[1]})
bad=pd.DataFrame({"bad":bad[1]})

statistic,p_value=kruskal(standard["standard"]
,good["good"],bad["bad"])
print("Statistic:",statistic)
print("p_value:",p_value)

* Box pot
fig = px.box(df,
             x="credit_score",
             y="annual_income",
             color="credit_score",
             title="Credit Scores Based on Annual Income",
             color_discrete_map={'Poor':'red',
                                 'Standard':'yellow',
                                 'Good':'green'})
fig.update_traces(quartilemethod="exclusive")
fig.show()

# outstanding debt
```

```
sns.violinplot(x=df["credit_score"],y=df["outstanding_debt"])
plt.title("Credit Score Based On Outstanding debt")
plt.xticks(rotation = 45)

ctab=pd.crosstab(df["credit_score"],df["payment_of_min_amount"])
ctab
ctab=ctab.to_numpy()
from scipy.stats import chi2_contingency
stat,p_value,dof,expected=chi2_contingency(ctab)
print(p_value)
print("Expected value:\n")
d=pd.DataFrame(expected,index=["Good","Poor","Standard"]
,columns=["NM","No","Yes"])
print(d)
alpha=0.05
print(stat)
if(p_value<0.05):
    print("reject the null hypothesis")
else:
    print("accept the null hypothesis()")

plt.figure(figsize=(14,8))
sns.violinplot(x=df["payment_of_min_amount"],
y=df["outstanding_debt"],hue=df["credit_score"])

from pandas.api.types import CategoricalDtype
cat_type = CategoricalDtype(categories=["Good","Standard","Poor"], ordered=True)
df1["credit_score"] = df1["credit_score"].astype(cat_type)
plt.hist(df1[’credit_history_age’],bins=20,color=’b’)
plt.xlabel(’credit history age’)
plt.title(’Distribution ’)

from statsmodels.miscmodels.ordinal_model import OrderedModel
mod_prob = OrderedModel(df1[’credit_score’],
                        df1["credit_history_age"],
                        distr=’probit’)
res_prob = mod_prob.fit(method=’bfgs’)
res_prob.summary()
```

```
from scipy.stats import f_oneway
standard,good,bad=df.groupby("credit_score")
["credit_utilization_ratio"]
standard=pd.DataFrame({"standard":standard[1]})
good=pd.DataFrame({"good":good[1]})
bad=pd.DataFrame({"bad":bad[1]})
statistic,p_value=f_oneway(standard["standard"]
,good["good"],bad["bad"])
print(p_value)
print(statistic)

from scipy.stats import chi2_contingency
stat,p_value,dof,expected=chi2_contingency(ctab)
print(p_value)
print("Expected value:\n")
d=pd.DataFrame(expected,index=["Good","Poor","Standard"])
print(d)
alpha=0.05

if(p_value<0.05):
    print("reject the null hypothesis")
else:
    print("accept the null hypothesis()")

* model buliding
le=LabelEncoder()
ohe=OneHotEncoder()
df["credit_mix"]=le.fit_transform(df["credit_mix"])
s=pd.DataFrame(ohe.fit_transform(df[["payment_behaviour",
"payment_of_min_amount"]]).toarray())
df=df.drop(columns=["payment_behaviour","payment_of_min_amount"])
df=df.join(s)
y=df["credit_score"]
X=df.drop(columns=["credit_score"])
y=le.fit_transform(df["credit_score"])
se=StandardScaler()
X=se.fit_transform(X)
```

```
train_X,test_X,train_y,test_y=train_test_split
(X,y,test_size=0.2,stratify=y)
print(train_X.shape,test_X.shape,train_y.shape,test_y.shape)


def train_and_evaluate_model(model):
        model.fit(train_X,train_y)
        y_pred=model.predict(train_X)
        print(classification_report(train_y,y_pred))
        acc=accuracy_score(train_y,y_pred)
        precision=precision_score(train_y,y_pred,average="micro")
        recall=recall_score(train_y,y_pred,average="micro")
        f1=f1_score(train_y,y_pred,average="micro")
        ConfusionMatrixDisplay.from_predictions(train_y,y_pred)
        plt.show()
model_names1 = []
accuracy_scores1 = []
precision_scores1 = []
recall_scores1 = []
f1_scores1 = []

def test_and_evaluate_model(model):
    model.fit(train_X,train_y)
    y_pred=model.predict(test_X)
    print(classification_report(test_y,y_pred))
    acc=accuracy_score(test_y,y_pred)
    precision=precision_score(test_y,y_pred,average="micro")
    recall=recall_score(test_y,y_pred,average="micro")
    f1=f1_score(test_y,y_pred,average="micro")
    ConfusionMatrixDisplay.from_predictions(test_y,y_pred)
    plt.show()
    model_names1.append(model)
    accuracy_scores1.append(acc)
    precision_scores1.append(precision)
    recall_scores1.append(recall)
    f1_scores1.append(f1)

train_and_evaluate_model(RandomForestClassifier())
```

```
test_and_evaluate_model(RandomForestClassifier())

param_grid = {’criterion’: [’gini’,’entropy’],
              ’max_features’: [’sqrt’,’log2’],
              ’max_depth’: [2,12,38,68,98,128,201]
             }
grid_dt = RandomizedSearchCV(RandomForestClassifier(),
param_grid,n_jobs=1,cv=5)
train_and_evaluate_model(grid_dt)

grid_dt = RandomizedSearchCV(RandomForestClassifier(),
param_grid,n_jobs=1,cv=5)
test_and_evaluate_model(grid_dt)

train_and_evaluate_model(LogisticRegression(
penalty="l2",random_state=1,multi_class="multinomial"))

test_and_evaluate_model(LogisticRegression(
penalty="l2",random_state=1,multi_class="multinomial"))

test_and_evaluate_model(LogisticRegression(
penalty="l2",random_state=1,multi_class="multinomial"))

param_grid = {’penalty’: [’l1’,’l2’,’elasticnet’],
              ’C’: [0.001,0.01,0.1,1,10],
              ’solver’: [’newton-cg’, ’lbfgs’,
              ’liblinear’, ’sag’, ’saga’],
              ’multi_class’: [’ovr’, ’multinomial’],
              ’l1_ratio’: [0.2,0.5,0.8]
             }

grid_lr = RandomizedSearchCV(LogisticRegression(),param_grid,verbose=2)

test_and_evaluate_model(grid_lr)

def elbow(k):
    test_error=[]
    for i in k:
        clf=KNeighborsClassifier(n_neighbors=i)
        clf.fit(train_X,train_y)
        tmp=clf.predict(test_X)
        tmp=f1_score(test_y,tmp,average="micro")
        error=1-tmp
        test_error.append(error)
    return test_error

k=range(6,20)
test=elbow(k)

plt.plot(k,test)
plt.xlabel("k Neighbors")
plt.ylabel("test_error")
plt.title("elbow curve for test")
train_and_evaluate_model(KNeighborsClassifier(n_neighbors=6))
test_and_evaluate_model(KNeighborsClassifier(n_neighbors=6))

from sklearn.model_selection import validation_curve
param_range =[1,3,6,9]
train_score, test_score = validation_curve
            (KNeighborsClassifier(), train_X, train_y,
            param_name="n_neighbors",
             param_range=param_range,
            cv=5, scoring="accuracy")

mean_train_score = np.mean(train_score, axis=1)
std_train_score = np.std(train_score, axis=1)

 plt.plot(param_range, mean_train_score,
         label="Training Score", color=’b’)
plt.plot(param_range, mean_test_score,
         label="Cross Validation Score", color=’g’)
plt.title("Validation Curve with KNN Classifier")
plt.xlabel("Number of Neighbours")
plt.ylabel("Accuracy")
plt.tight_layout()
plt.legend(loc=’best’)
plt.show()
train_and_evaluate_model(KNeighborsClassifier(n_neighbors=9))
test_and_evaluate_model(KNeighborsClassifier(n_neighbors=9))

model_performmance=pd.DataFrame({"model":model_names1,
         "accuracy":accuracy_scores1,
        "precision":precision_scores1,"recall":recall_scores1,
        "f1_score":f1_scores1}).sort_values("accuracy",ascending=False)
model_performmance

learning_rate = 0.1
iterations = 5000
N = train_y.size
# number of input features
input_size = 23
# number of hidden layers neurons
hidden_size = 2
# number of neurons at the output layer
output_size = 3
results = pd.DataFrame(columns=["mse", "accuracy"])

np.random.seed(10)
# initializing weight for the hidden layer
W1 = np.random.normal(scale=0.5, size=(input_size, hidden_size))
print("Weight for the hidden layer \n", W1, "\n")
# initializing weight for the output layer
W2 = np.random.normal(scale=0.5, size=(hidden_size , output_size))
print("Weight for the output layer \n", W2)

def sigmoid(x):
    return 1 / (1 + np.exp(-x))
def mean_squared_error(y_pred, y_true):
    y_true_one_hot = np.eye(output_size)[y_true]
    y_true_reshaped = y_true_one_hot.reshape(y_pred.shape)
    return ((y_pred - y_true_reshaped)**2).sum() / (2*y_pred.size)
def accuracy(y_pred, y_true):
    acc = y_pred.argmax(axis=1) == y_true.argmax(axis=1)
    return acc.mean()
```

```
print(W2.shape)
print(A2.shape)
print(train_y.shape)

import warnings

warnings.filterwarnings('ignore')
for itr in range(iterations):
    Z1 = np.dot(train_X, W1)
    A1 = sigmoid(Z1)
    Z2 = np.dot(A1, W2)
    A2 = sigmoid(Z2)
    mse = mean_squared_error(A2,train_y)
    acc = accuracy(np.eye(output_size)[train_y], A2)
    results=results.append({"mse":mse, "accuracy":acc},
    ignore_index=True )
    # backpropagation
    E1 = A2 - np.eye(output_size)[train_y]
    dW1 = E1 * A2 * (1 - A2)
    E2 = np.dot(dW1, W2.T)
    dW2 = E2 * A1 * (1 - A1)

    # weight updates
    W2_update = np.dot(A1.T, dW1) / N
    W1_update = np.dot(train_X.T, dW2) / N
    W2 = W2 - learning_rate * W2_update
    W1 = W1 - learning_rate * W1_update
results.mse.plot(title="Mean Squared Error")

Z1 = np.dot(test_X, W1)
A1 = sigmoid(Z1)
Z2 = np.dot(A1, W2)
A2 = sigmoid(Z2)
acc = accuracy(np.eye(output_size)[test_y], A2)

print("Accuracy: {}".format(acc))
model=RandomForestClassifier()
```

```
from sklearn.model_selection import StratifiedKFold
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=1)
scores = []
folds=list(kfold.split(train_X,train_y))
for k,(train,test) in enumerate(folds):
    model.fit(train_X[train], train_y[train])
    score = model.score(train_X[test], train_y[test])
    scores.append(score)
    print('Fold: %s, Class dist.: %s, Acc: %.3f' %
    (k+1, np.bincount(train_y[train]),score))
print('CV accuracy: %.3f +/- %.3f' % (np.mean(scores), np.std(scores)))
```