COURSE-DATA SCIENCE(EDWISOR)

03/07/2020

Project Report

BANK LOAN DEFAULTER CASE

MUKESH KUMAR SABLANI

SUBMITTED BY-MUKESH KUMAR SABLANI

**Table of contents**

**1.Introduction**

1.1 Problem Statement

1.2 Type of Problem

1.3 Data

1.4 Software and Hardware requirement

**2 Methodology**

2.1 Exploratory Data Analysis…

2.2 Data Preprocessing

2.3 Outlier Analysis

2.4 Feature Selection

2.5 Feature Scaling

**3. Modelling**

3.1 Logistic Regression

3.2 Decision Tress

3.3 Random Forest

3.4 Naïve bayes

3.5 Prediction test on csv Format

**4. Model Selection**

**5. Model Fitting & Conclusion**

Introduction

At ​Bank loan defaulter ​, mission is to help people and businesses prosper. We are always looking for ways to help our customers understand their financial health and identify which products and services might help them achieve their monetary goals. Our data science team is continually challenging our machine learning algorithms , working with the global data science community to make sure we can more accurately identify new ways to solve our most common challenge, binary classification problems such as: is a customer satisfied? Will a customer buy this product? Can a customer pay this loan?



**1.1​ Problem Statement-**

There is a company named Mentor Housing finance Pvt limited that deals all home loans. They have presence across all urban and rural areas. Customer first apply for home loan after that company validates the customer eligibility for loan. However doing this manually takes a lot of time. Hence it wants to automate the loan eligibility process(real time) based on customer information.

So the final thing is to identify the factors/ customer segments that are eligible for taking loan. How will the company benefit if we give the customer segments is the immediate questions that arises. The solution is Banks would give loans to only those customer that are eligible so that they can be assured of getting money back.Hence the more accurate we are in predicting the eligible customer the more beneficial it would be for the Mentor Housing finance Pvt limited.

Loans default will cause huge loss for the banks, so they pay much attention on this issue and apply various method to detect and predict default behaviours of their customers.

The loan default dataset has 9 variables and 850 records, each record being loan default status for each customer. Each Applicant was rated as “Defaulted” or “Not-Defaulted”. New applicants for loan application can also be evaluated on these 8 predictor variables and classified as a default or non-default based on predictor variables.

**1.2 TYPES OF PROBLEM-**

The problem clearly see above statement is a classification problem as we need to classify whether the Loan \_status yes or no. So this can be solved any of the classification techniques like

1.Logistic Regression

2. Decision Trees

3. Random Forest

4. Naïve Bayes

**1.3 Data-**

Our task is to build classification model which will predict that the new applicant for loan application can be classified as default or non-default depending on these 8 predictor variables. There are 8 independent variable and one dependent variable which is my target variable out of 9 variable.

**In a loan risk prediction situation** of a loan financing company, the company would be interested in metrics such as how long it takes customers with certain attributes to pay back their loans and also, what is the possible risk of a default.

Generally, the company stands a higher risk of default from customers who have a bad credit rating or who have certain bad spending habits. In this situation, the company is very keen to find out if a customer will default or not. So, the past data observations gathered by the company are used to group customers into categories such as “Defaulter” or “Non-defaulter”.

Given below is a sample of the data set that we are using to predict the count

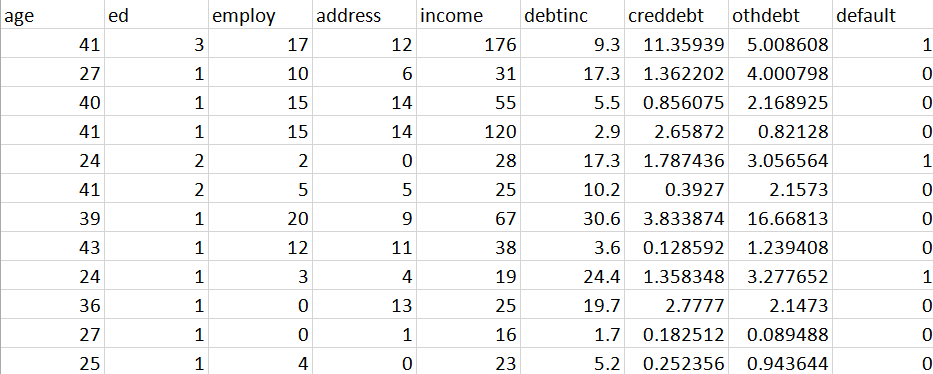


Table 1.1: Bank Loan Default Case. Sample data (Columns:1- 8)

Variables present in given dataset are 'age', 'ed', 'employ', 'address', 'income',

'debtinc', 'creddebt', 'othdebt' and 'default'.The details of variable present in the dataset are as follows :

|  |  |
| --- | --- |
| Sr.no | Column Name |
| 1 | Age |
| 2 | Education |
| 3 | Employment |
| 4 | Address |
| 5 | Income |
| 6 | Debitinc |
| 7 | creddebt |
| 8 | othdebt |
|  |  |
| 9 | default |

Table 1.2: Customer default status Prediction variables

Above the variables we used to predict the default status here 8 variables are independent variables and predictors and one variable ‘default’ is target variables

**1.4 Software and Hardware Requirement-**

1. R 3.6.1 for 64 bit
2. Anaconda 3 for 64 bit
3. R studio
4. 64 bit OS
5. Python 3
6. Jupyter Notebook
7. 4GB of RAM

**2.Methodology**

**2.1.1 Exploratory Data Analysis-**

It is a approach where we have analyze data sets to summarize and performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.It is a good practice to understand the data first and try to gather as many insights from it. EDA is all about making sense of data in hand, before getting them dirty with it. There are number of tools that are useful for EDA but EDA is characterized more by the attitude taken by particular techiniques which some of techiniques we describe further in our analysis step by step.

1.First of all we can derive our data index values in column where we used function

loan.column

Output =index(['age', 'ed', 'employ', 'address', 'income', 'debtinc', 'creddebt',

'othdebt', 'default'],

dtype='object')

2. we can input function first five rows of data

loan.head()

3. datatypes of the data

loan.dtypes

age int64

ed int64

employ int64

address int64

income int64

debtinc float64

creddebt float64

othdebt float64

default float64

dtype: object

4.Then we define data column value and non null count value in this function

loan.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 850 entries, 0 to 849

Data columns (total 9 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 age 850 non-null int64

1 ed 850 non-null int64

2 employ 850 non-null int64

3 address 850 non-null int64

4 income 850 non-null int64

5 debtinc 850 non-null float64

6 creddebt 850 non-null float64

7 othdebt 850 non-null float64

8 default 700 non-null float64

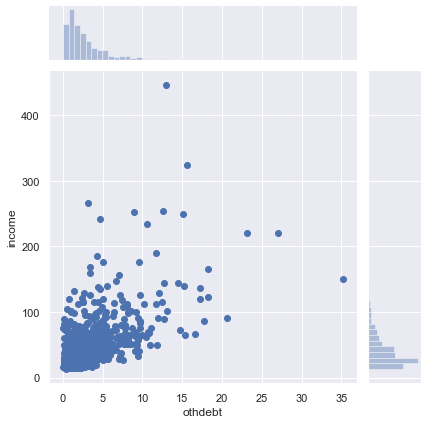
dtypes: float64(4), int64(5)

5. Descriptive statistics

loan.describe()

**6.Scatter plot-**

A Scatter Plot(Scatter Graph) uses dots to represent two different numeric variables. The Position of each dot on the horizontal and vertical axis indicated value of each



Figure(A): Scatter plot shows the linear relationship between x axis variable (othdebt) and y axis Variable (income)

data point. Scatter plots are used to observe relationships between variables

The above example of scatter plot shows the relationship **othdebt** variable lie on horizontal axis and **income** variable lie on vertical axis.Here we can see bottom of the right direction at x axis histogramme represent **othdebt** variables ,top of the left direction at y axis histogramme shows **income** variable an in between main plot we have drawn scatter plot **othdebt** veds **income.** One thing clearly seen this plot last two points are large distance othdebt variable which is lies range between at x axis 25 to 30 compare rest of the starting point at same variables which is lies on range between 0 to 10. So most of the data lies between 0 to 10 which shows less distance compare last two data or points so he we can easily conclude that positive correlation lies on the point.

**2.1.2 Data Preprocessing-**

**Missing values analysis-**

Missing values analysis is done to check is there any missing values present in given dataset. Missing data are a common occurrence and can have a significant effect on the statistical analysis. The concept of missing values is important to understand in order to successfully manage data Missing values can be easily treated using various methods like mean, median method, knn method to impute missing value.

In R function(x){sum(is.na(x))} is the function used to check the sum of missing values.

In python loan.isnull().sum() is used to detect any missing value.

Below table illustrate the missing value present in the data.

|  |  |
| --- | --- |
| Column Name | Missing Values |
| Age | 0 |
| Education | 0 |
| Employment | 0 |
| Address | 0 |
| Income | 0 |
| Debitinc | 0 |
| creddebt | 0 |
| othdebt | 0 |
| default | 150 |

Table 2.1 Missing Values in bank loan data

There are 150 missing values here and the percentage of missing values is 17.64%. So, we remove missing values using loan=loan.dropna(inplace=true) in python. Only 700 observations were left.

**2.1.3 Outlier Analysis-**

Outlier analysis is done to handle all inconsistent observations present in given dataset. As outlier analysis can only be done on continuous variable An outlier is an observation that is abnormal compared to other observations in that dataset. One of the most important tasks from large data sets is to find an outlier because outliers can significantly alter the results even though they are present in small proportions. First of all in this step we convert ed and default variable in a object data type

To find an outlier, inter quartile range (IQR) is found first. IQR represents the middle 50% of the data. The position of first quartile can be found using formula (N+1)/4 & third quartile can be found by 3\*(N+1)/4 where N is the total no. of observations. The difference of the values in the first & third quartiles is the IQR**. If any observation falls below 1.5 times IQR from the first quartile value, or if it falls above 1.5 times IQR from the third quartile value, then the value can be qualified as an outlier.**

Outliers can be found using box plot method which can be plotted in both R & Python. After finding the outliers, they can be removed from the dataset or they can be imputed by KNN method.

**Observations and results-**

Original loan data contained a total of 700 observations and After removing the outliers, **it reduced to 545**. So, **a sum of 155 outliers were detected from all variables and removed from the loan data.**

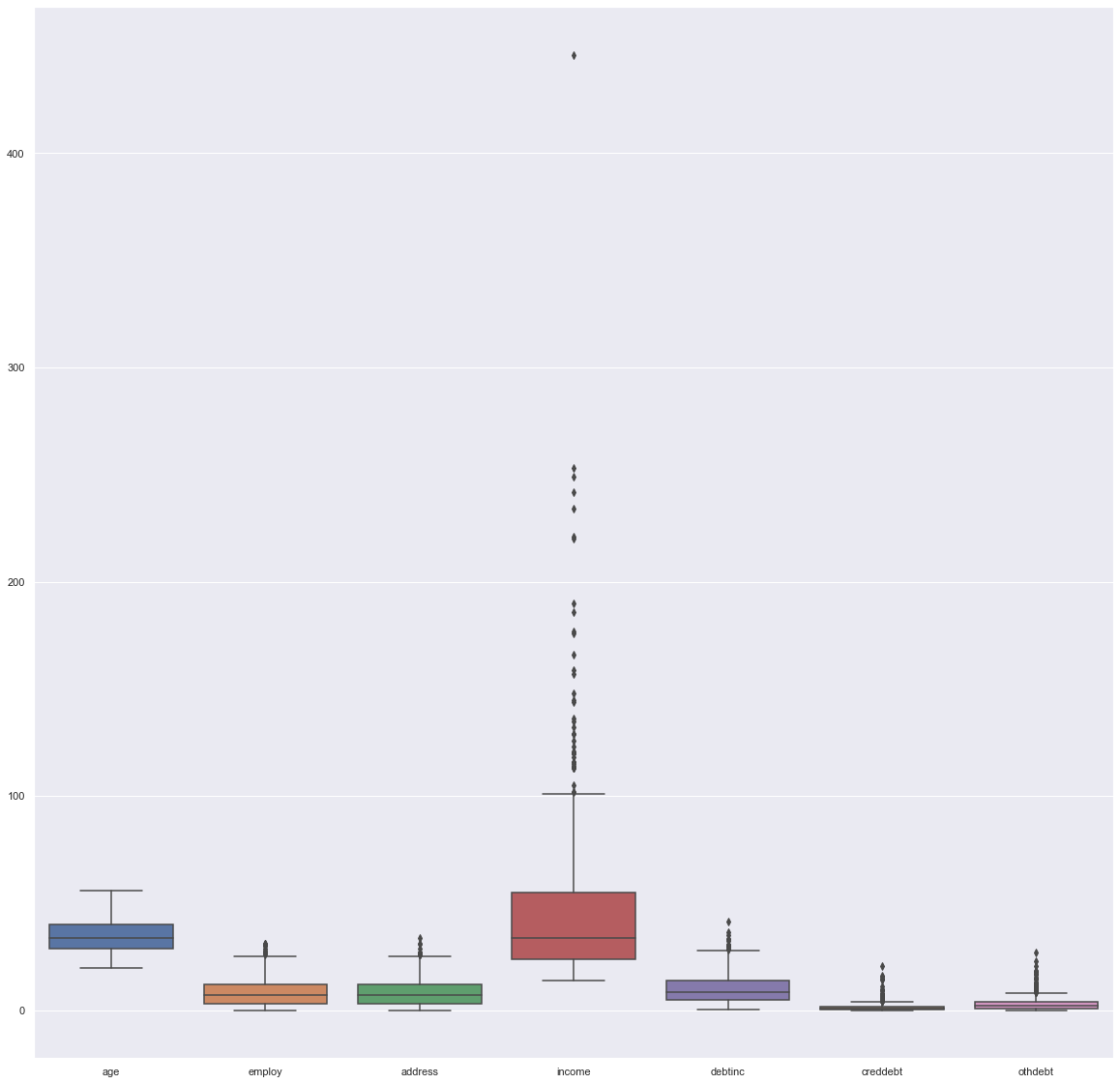
****

Figure 1.3 Boxplot graph of variables R and Python

**2.1.4 Feature Selections-**

Feature Selection is the process of selecting those features which contribute most to the prediction variable. Having irrelevant features in data can decrease the accuracy of the models and make the model learn based on irrelevant features. For this, relation between different variables are evaluated and if two variables are strongly correlated with each other, then one of them may be dropped. Correlation analysis is used in feature selection for numerical variables and chi square test is used for categorical variables.

Data and feature correlation is considered one important step in the feature selection phase of the data pre-processing especially if the data type for the features is continuous.

**Positive Correlation:**

means that if feature **A** increases then feature **B** also increases or if feature **A** decreases then feature **B** also decreases. Both features move in tandem and they have a linear relationship.

**Negative Correlation:** Means that if feature A increases then feature B decreases and vice versa.

**No Correlation:** No relationship between those two attributes.

**Correlation analysis with Heatmap-**

If correlation value is greater than 0.8 or less than -0.8, then it can be safely assumed that the two variables in consideration are highly correlated and one of them may be dropped. But, from correlation matrix in R, all the values were close to 0, which indicates that they are independent of each other.

Similar conclusion can be drawn from the heat map, in which no shade of red color which indicates dependency of the variables were found. Hence, none of the features were eliminated from the loan data.

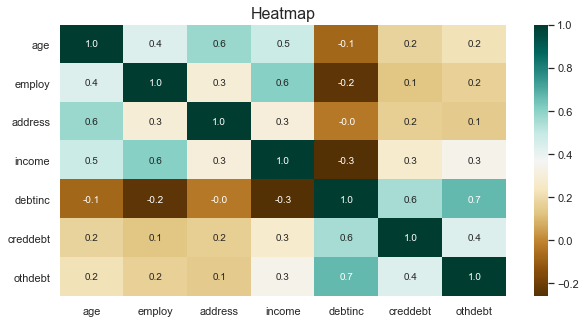


Figure 2.1.4 correlation plot

**Chi- Square test-**

It is a statistical hypothesis test that is valid to perform when the test is statistics under the null hypothesis check chi-square test for independence and print p value of ‘ed’ variable is equal to 0.055 that is very less than our significance leval so we can remove this feature.

It indicates there is sufficient evidence to conclude that the observed distribution is not the same as the expected distribution. You can conclude that a relationship exists between the

categorical variables. After that test we can drop variable ‘ed’ of a particular coloumn.

#select categorical data

c\_names = loan[['ed']]

#Chisquare test of independence

#loop for chi square values

for i in c\_names:

print(i)

chi2, p, dof, ex = chi2\_contingency(pd.crosstab(loan['default'], loan[i]))

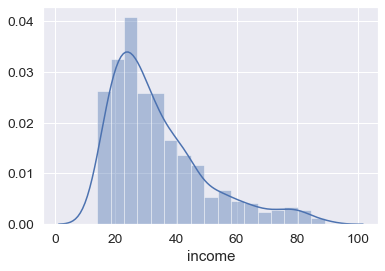
print(p)

loan = loan.drop('ed',axis = 1)

**2.1.5 Feature Scaling-**

**Feature scaling means adjusting data that has different scales into the same range.** Feature scaling is an important technique in Machine Learning and it is one of the most important steps during the preprocessing of data before creating a machine learning model. Most of the times, the dataset contains features highly varying in magnitudes, units and range. The two most important scaling techniques is Standardization and Normalization.

Normalization is the process of rescaling the features to the range of 0 to 1. Standardization is the process of rescaling data to have a mean of 0 and a standard deviation of 1. This is usually applied to the dataset which is normally distributed. Here we can see when we can draw distribution plot of ‘**income**’ variable is right skewed or positively skewed. So that’s why we don’t need follow standardization process because that process followed that data which is not normally distributed.I have done normalization both R and Python.



**Figure 2.1.4 Distribution of skewness**

**Observations & Results -**

Histograms were plotted to check the normality of the data and the plots of some of the variables are as follows which is not normally distributed after that take subset by removing ed variable both Python and R and check the range before normalization and after normalization , save the column names after that save the outpot in csv format converted to excel format in Both R and Python.

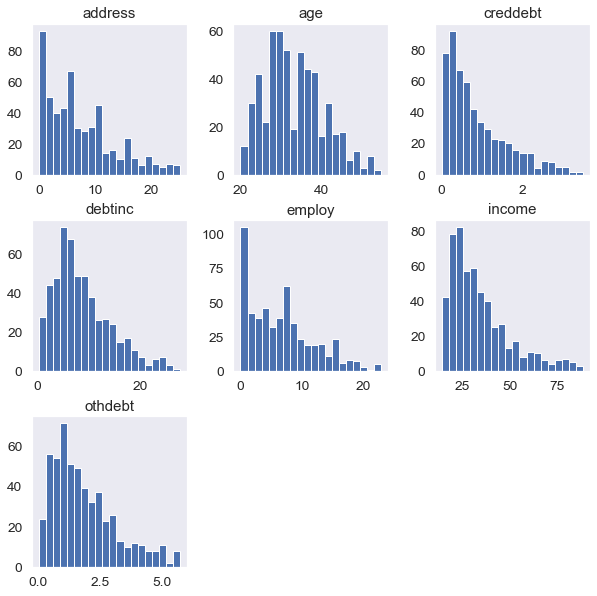
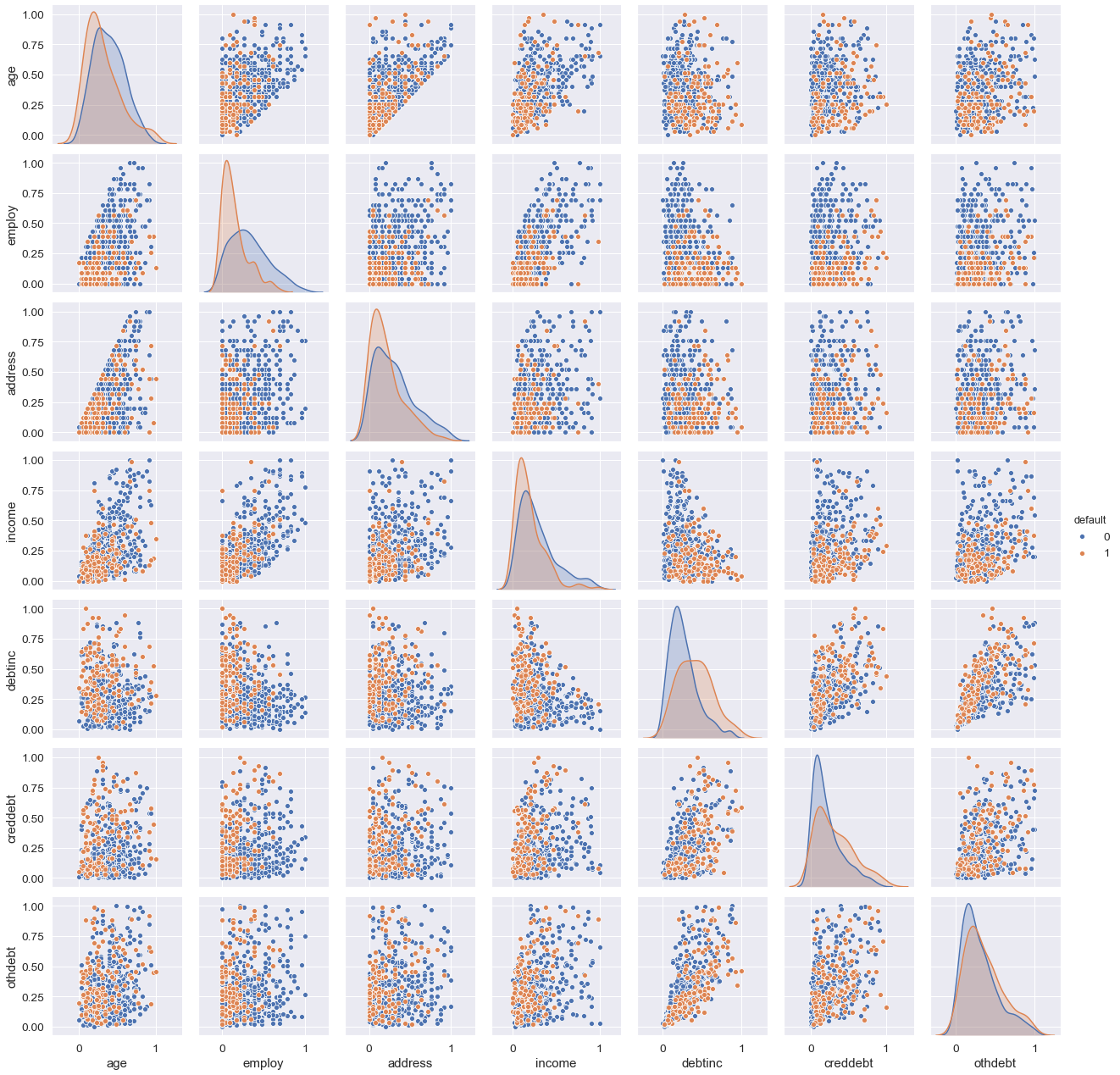
****

Figure 2.1.5 Plotting of Histogramm of some of the variables at a data points

**3.Model development**

A **pairplot** plot a pairwise relationships in a dataset. The **pairplot** function creates a grid of axes such that each variable in data will be shared in the y-axis across a single row and in the x-axis across a single column.

**Bivariate Analysis-**

****

Pair plot is equal to bivariate analysis and it is combination of scatter plot keeping two variables at a same time from pair plot we can indentified how variables are behaving with each other .

In this case we have to Each Applicant was rated as “Defaulted” or “Not-Defaulted”. New applicants for loan application can also be evaluated on the 8 predictor variables and classified as a default or non-default based on predictor variables. Model having less error rate and more accuracy will be our final model.In these we have divided the dataset into train and test part using random sampling. For this model we have divided the dataset into train and test part using random sampling Where train contains 80% data of data set and test contains 20% data and contains 8 variable where 8th variable is the target variable.

Here we can use this classification model some of machine learning algorithm.

**3.1Logistic Regression**-

Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable (0 or 1). In this, the independent variables should not be correlated with each other. That is, the model should have little or no multi-collinearity. Logistic Regression is one of the most popular ways to fit models for categorical data, especially for binary response data in Data Modeling.

Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.

Train-Test split data

**from sklearn.model\_selection import train\_test\_split**

**from sklearn import metrics**

**lr = LogisticRegression().fit(X\_train , y\_train)**

**lr\_pred = lr.predict(X\_test)**

**Confusion Matrix**

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are



Figure 7.1 Confusion Matrix

Known here,

**Class 1:** Positive

**Class 2:** Negative

**Definition of the Terms:**

• Positive (P): Observation is positive (for example: is an apple).

• Negative (N): Observation is not positive (for example: is not an apple).

• True Positive (TP): Observation is positive, and is predicted to be positive.

• False Negative (FN): Observation is positive, but is predicted negative.

• True Negative (TN): Observation is negative, and is predicted to be negative.

• False Positive (FP): Observation is negative, but is predicted positive.

**Classification Rate/Accuracy:**

**Precision**: Precision is fraction of items the classifier flags as being in the class actually are in the class.

**Precision = TP/(Predicted Yes)=TP/(TP+FP)**

**Recall**: What fraction of things that are in the class are detected by the classifier.

**Recall =TP/(Actual Yes)= TP/(TP + FN)**

**Accuracy**: Below is the actual over all Accuracy of the Model

**Accuracy = (TP+TN)/(TP+FP+TN+FN)**

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Model – R | Accuracy | FNR | TPR |
| Logistic  Regression | 80% | 45.23% | 54.76% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model-python** | Accuracy | FNR | TPR |
| Logistic Regression | 80.73% | 70.37% | 29.62% |

FPR=2.43%

TNR=97.56%

[80, 2],

[19, 8]

Defaulted 10

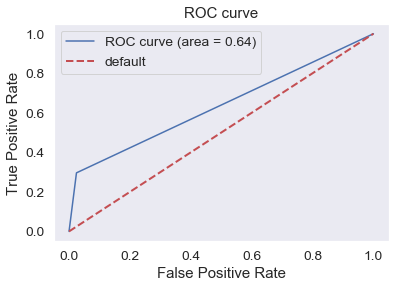
Non-defaulted 99

auc = metrics.auc(fpr, tpr)

print("AUC:", auc)

AUC : 0.6359530261969285

**Plot the ROC curve-**

****

**3.2Decision Trees-**

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

**Results-**

|  |  |  |  |
| --- | --- | --- | --- |
| Model – R | Accuracy | FNR | TPR |
| Decision Tree | 88.57% | 26.19% | 73.8% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model-python** | Accuracy | FNR | TPR |
| Decision Tree | 77.98% | 70.37% | 29.62% |

FPR=6.09%

TNR(Specifity)=93.90%

[77, 5],

[19, 8]

Defaulted 13

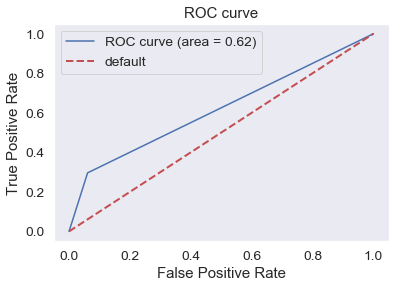
Non-defaulted 96

auc = metrics.auc(fpr, tpr)

print("AUC:", auc)

AUC: 0.6176603432700993

**Plot the ROC curve-**

****

**3.3RandomForest-**

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

**Results-**

|  |  |  |  |
| --- | --- | --- | --- |
| Model-R | Accuracy | FNR | TPR |
| Random Forest | 100% | 0.0% | 100% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model-Python** | Accuracy | FNR | TPR |
| Random Forest | 81.65% | 59.25% | 37.03% |

FPR=4.87%

TNR(Specifity)=95.12

[79, 3]

[17, 10]

Defaulted 13

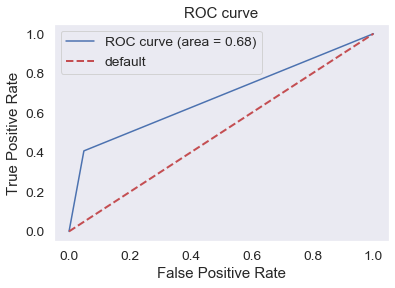
Non-defaulted 96

auc = metrics.auc(fpr, tpr)

print("AUC:", auc)

AUC: 0.6793134598012647

**Plot the ROC curve-**

****

**3.4 Naive Bayes-**

Naive Bayes is a probabilistic machine learning algorithm that can be used in a wide variety of classification tasks. The name naïve is used because it assumes the features that go into the model is independent of each other. That is changing the value of one feature, does not directly influence or change the value of any of the other features used in the algorithm.

The fundamental Naive Bayes assumption is that each feature makes an independent & equal contribution to the outcome. The Naïve Bayes classifier is based on the Bayes theorem which is given as

If, P (A | B) = P ( A & B ) / P(B)

& P(B | A) = P ( A & B ) / P(A)

Then, P (A | B) = [ P ( B | A) \* P(A) ] / P(B)

Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.

**Advantages:** This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

**Disadvantages:** Naive Bayes is is known to be a estimator bad

**Results-**

|  |  |  |  |
| --- | --- | --- | --- |
| Model – R | Accuracy | FNR | TPR |
| Naïve Bayes | 72.8% | 91.8% | 90.8% |

|  |  |  |  |
| --- | --- | --- | --- |
| **Model-python** | Accuracy | FNR | TPR |
| Naïve Bayes | 86.23% | 33.33% | 66.66% |

FPR=7.31%

TNR(Specifity)=92.68%

[76, 6],

[09, 18]

Defaulted 24

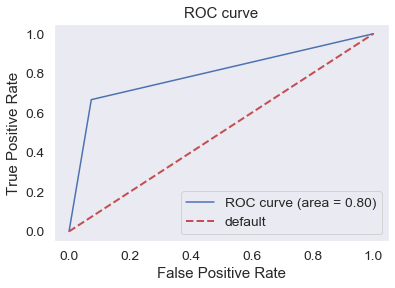
Non-defaulted 85

auc = metrics.auc(fpr, tpr)

print("AUC:", auc)

AUC: 0.7967479674796748

**Plot the ROC curve-**

****

**3.5 Prediction test on csv format-**

Here we can see all the model above in python Naïve bayes shows better accuracy and less false negative rate among all the machine learning model so we conclude this particular model find prediction test and save our file in csv format to excel format in our harddisk (C drive). So we follow some steps:

1. First of all naïve bayes predicted values save or store into dataframe.

**NB\_Predictions = pd.DataFrame(NB\_Predictions**)

1. Then we have load this file from our root directory.

**NB\_Predictions.to\_csv('bank -loan python.csv',header= True , index= False)**

1. #saving X\_test into directory

**X\_test.to\_csv("xtest\_final.csv", index = False , header= True)**

1. After that read that saving test in pd.read\_csv format and defined

**df=pd.read\_csv('xtest\_final.csv')**

5. After that we joining two dataframes in below format

**final\_result = pd.concat([df, NB\_Predictions], axis=1)**

1. Renaming column name

**final\_result.rename(columns={0: 'default'},inplace = True)**

**7.** #saving result in csv format

**final\_result.to\_csv('loan\_final\_result\_.csv',header= True , index= False)**

**4. Model Selection**

Here is the comparison of all the models we have developed.

|  |  |  |
| --- | --- | --- |
| **M.L. Model – R** | **Accuracy** | **FNR** |
| **Logistic regression** | **80%** | **45.23%** |
| **Decision Trees** | **88.57%** | **26.19%** |
| **Random Forest** | **100%** | **0.0%** |
| **Naïve Bayes** | **72.8%** | **91.8%** |

|  |  |  |
| --- | --- | --- |
| **M.L. Model - Python** | **Accuracy** | **FNR** |
| **Logistic regression** | **80.73%** | **70.37%** |
| **Decision Trees** | **77.98%** | **70.37%** |
| **Random Forest** | **81.65%** | **62.96%** |
| **Naïve Bayes** | **86.23%** | **33.33%** |

**5. Model Fitting and conclusion**

In R Random forest was performing well among all models and Python Naïve bayes was performing well among all the models so, we will freeze Random forest was chosee in R and Naïve Bayes was chosen in Python for the prediction of target variable in test data.

After the selection of the best possible model, it was fitted to the test dataset after selection the best possible model it was fitted to the test data for which the target variable was to be predicted.

Feature scaling was also done because the original train dataset was trained on the scaled data, thus the predicted results would be accurate only if the model fitting is done on the scaled test data.

.