Bank Loan Default

Prajakta Deshmukh

(6/9/2020)

Content

Chapter 1 : Introduction

1. Problem statement

Given problem statement is about bank default loan case.

The loan default dataset has 8 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.

Aim of this is to identify/rate applicant is as 'default' or 'non-default'.

So, here classification is done on basis of data variables. '*default*' is target or depend variable. After classifying applications, bank can easily perform further process to give loan to applicant.

2. Data

There are total 8 variables in data are as follows:

- age : age of each customer (Numerical type)
- ed : Education categories (Categorical type)
- employ : Employment status – Numerical Corresponds to job status and being converted to numeric format (Numerical type)
- address : Address Geographic area numerical value (Numerical type)
- income : Gross Income of each applicant (Numerical type)
- debtinc : Individual's debt from gross income(Numerical type)
- creddebt : debt-to-credit ratio is a Numerical measurement of how much you owe your creditors as a percentage of your available credit (credit limits)
- othdebt : Any other debts (Numerical type)

Predictor variables or independent variables are as follow:

Predictor variables
Age
Education
Employ
Address
Income
Debitinc
Creddebt
Othdebt

Given below is the dataset which we will be using to classify applicant as ‘default’ or ‘non-default’ :

1	age	ed	employ	address	income	debtinc	creddebt	othdebt
2	41	3	17	12	176	9.3	11.3594	5.00861
3	27	1	10	6	31	17.3	1.3622	4.0008
4	40	1	15	14	55	5.5	0.85608	2.16893
5	41	1	15	14	120	2.9	2.65872	0.82128
6	24	2	2	0	28	17.3	1.78744	3.05656
7	41	2	5	5	25	10.2	0.3927	2.1573
8	39	1	20	9	67	30.6	3.83387	16.6681
9	43	1	12	11	38	3.6	0.12859	1.23941
10	24	1	3	4	19	24.4	1.35835	3.27765

Chapter 2 : Methodology

Data pre-processing

Data pre-processing first step is to Analysis of Data means transforming raw data in proper format required for modeling. Explore data and transform it, means the analysis of data.

There are total 850 record and 8 variables. Target Value 'defaulted' contains 85% of data contains 'not-defaulted' Customers and 14.5 % of data contains defaulted' , it may be chance that class imbalance problem may occurs because of less proportion of data contains defaulted customers .

we should be very careful on during evaluation of Model instead of concentration on only Accuracy we should also concentrate on Precision and Recall also and we should make sure that Precision and Recall should also be high.

Missing Value Analysis

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models. Missing value analysis is done to check is there any missing value present in given dataset. Missing values can be easily treated using various methods like mean, median, mode method, KNN method to impute 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

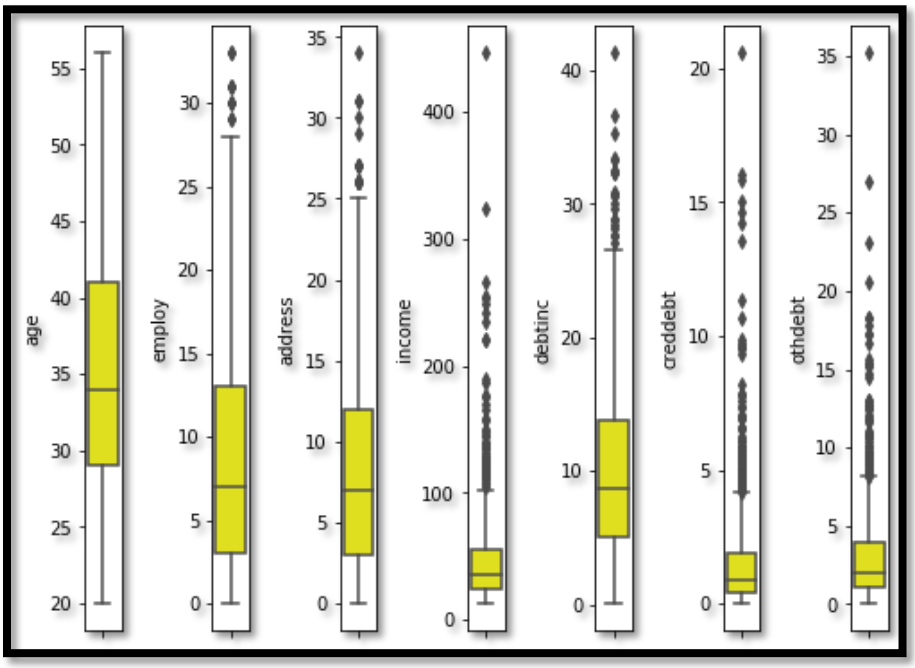
There are 150 missing values here and the percentage of missing values is 17.64%.

Imputed missing value in data set by taking mode as mode is 0.

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.

Visualization of numeric variable present in our dataset to detect outliers using boxplot and distribution plots. Outliers will be detected with black color. According to above visualizations there all features columns shows outliers. All independent variables are right skewed/positively skewed.



Outliers in data set are removed for better performance of model, after data contain total 702 records in dataset.

Feature Selection

Feature selection is extracting meaningful features from dataset. It is also called as dimension reduction process.

For our model we have selected correlation analysis method. It gives association between data variables in range of -1 to 1.

Feature Selection is the process of selecting the attributes that can make the predicted variable more accurate or eliminating those attributes that are irrelevant and can decrease the model accuracy and quality.

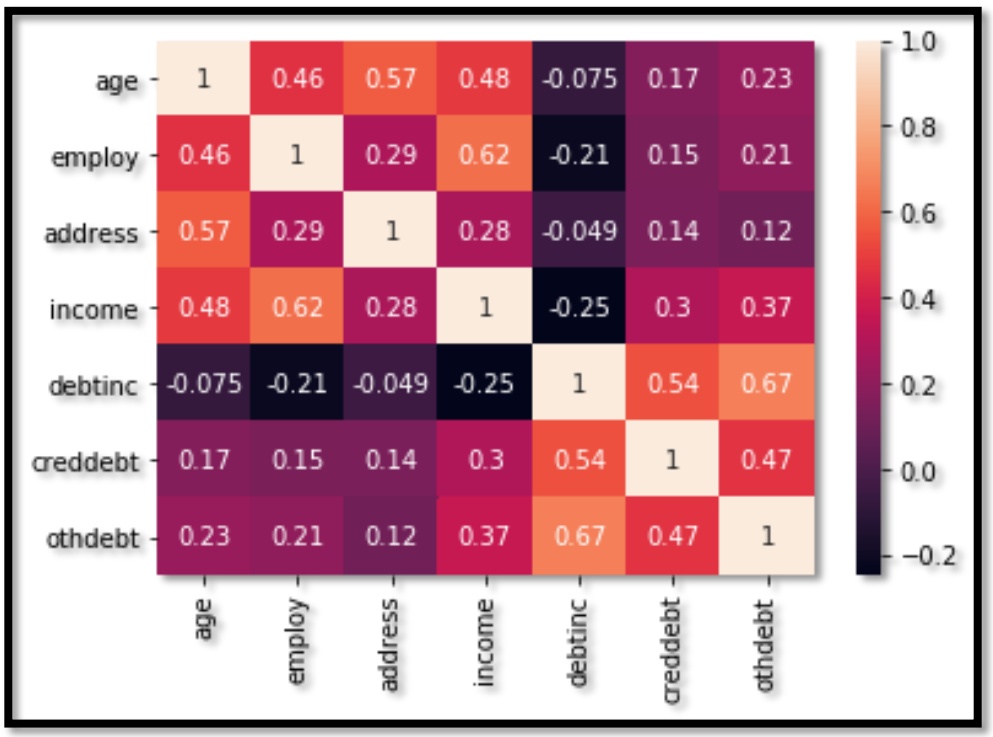
Positive Correlation: Means that if feature increases then feature B also increases or if feature decreases then feature B also decreases. Both features move in tandem and they have a linear relationship.

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

No Correlation: No relationship between those two attributes.

If there is a strong and perfect positive correlation, then the result is represented by a correlation score value of 0.9 or 1.If there is a strong negative correlation, it will be represented by a value of -1.

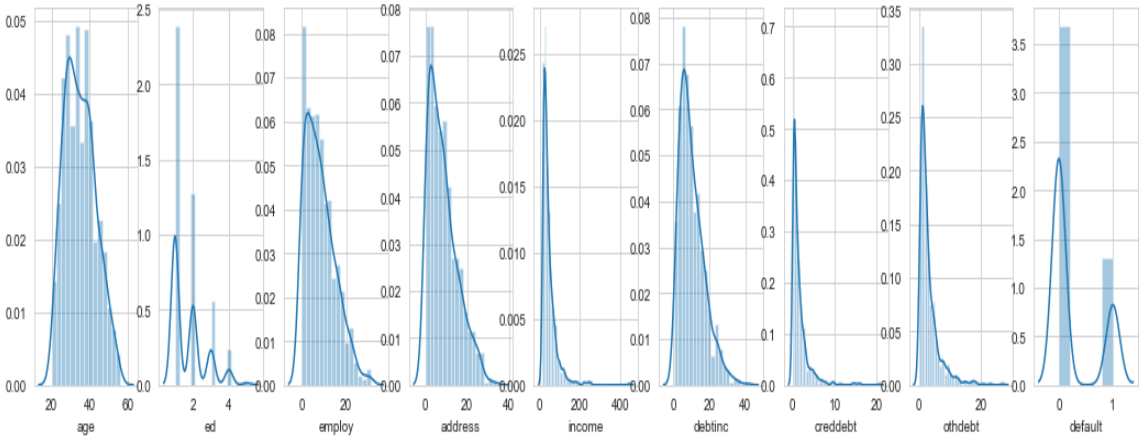
Below fig illustrates that relationship between all numeric variables using heat map.



Feature Scaling

To check feature scaling in give data set we need to perform some test as follow:

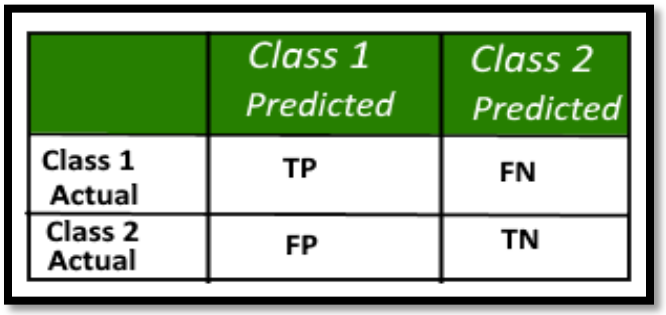
Feature scaling includes two functions normalization and standardization. It is reduce unwanted variation either within or between variables and to bring all of the variables into proportion with one another.



Chapter 3 : Modeling

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



	Class 1 Predicted	Class 2 Predicted
Class 1 Actual	TP	FN
Class 2 Actual	FP	TN

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/TP+FP

Recall: What fraction of thingsthat is in the class are detected by the classifier.

Recall = TP/TP + FN

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

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

F1 Score: It is the combination of the Precision and recall

F1 Score = 2*(Precision*Recall)/(Precision + Recall)

However, there are problems with accuracy. It assumes equal costs for both kinds of errors. 99% accuracy can be excellent, good, mediocre, poor or terrible depending upon the problem.

ROC CURVES and AUC

Receiver operating characteristic (ROC) curve: plots the true positive rate (TPR) versus the false positive rate (FPR) as a function of the model's threshold for classifying a positive.

Area under the curve (AUC): metric to calculate the overall performance of a classification model based on area under the ROC curve.

Problem type:

In Data Modeling at first, we need to identify the type of Problem statement.

In general, there are 4 kinds of Problem statement—

1. Predictive/Forecasting: The dependent variable has to be of type continuous.

2. Classification: The dependent variable has to be of type categorical.

3. Optimization

4. Unsupervised Learning

So, we conclude that give problem statement is of classification type

Classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

We will be using the following Machine Learning Algorithms to analyze and build our model are as follow:

- Logistic regression
- Random Forest
- Decision Tree
- XG Boost

Based on accuracy in R we have selected Random forest and in python we have selected XG Boost

R_ code

```
#remove objects from RAM

rm(list=ls())


#set working directory

setwd("C:/Users/HP/Desktop/Bank Project")


x = c("ggplot2", "corrgram", "DMwR", "caret", "randomForest", "unbalanced", "C50",
"dummies", "e1071", "Information",

    "MASS", "rpart", "gbm", "ROSE", 'sampling', 'DataCombine', 'inTrees')

lapply(x, require, character.only = TRUE)

rm(x)


########### load dataset ###############


bankdata = read.csv("bank-loan.csv")


############## Understand Data ################


class(bankdata)


str(bankdata)


dim(bankdata)
```

```
#change datatype as per data

bankdata$age = as.numeric(bankdata$age)
bankdata$ed = as.factor(bankdata$ed)
bankdata$employ = as.numeric(bankdata$employ)
bankdata$address=as.numeric(bankdata$address)
bankdata$income = as.numeric(bankdata$income)

str(bankdata)

###################### understand summary of all variables ###############

summary(bankdata)

###################### missing value anlysis and imputaion ####################

sum(is.na(bankdata))
summary(is.na(bankdata))

#calculated and imputed NA by taking mode of default column

bankdata[is.na(bankdata)] <- 0

sum(is.na(bankdata))
```

```
############################## outlier analysis ############################

boxplot(bankdata[,c('age','employ','address','income','debtinc','creddebt','othdebt')])

########## outlier detection and removal  ##########

for (x in c('age','employ','address','income','debtinc','creddebt','othdebt'))
{
  value = bankdata[,x][bankdata[,x] %in% boxplot.stats(bankdata[,x])$out]
  bankdata[,x][bankdata[,x] %in% value] = NA
}

library(tidyr)
bankdata = drop_na(bankdata)
as.data.frame(colSums(is.na(bankdata)))

dim(bankdata)

######################### feature selection ####################

numeric_col=c('age','employ','address','income','debtinc','creddebt','othdebt')

library(corrgram)
```

```
corrgram(bankdata[,numeric_col],order=FALSE,upper.panel = panel.pie,
         text.panel = panel.txt,
         main= "Correlation Analysis Plot")



########################## Feature scaling ###################################


##### histogram


hist(bankdata$age)
hist(bankdata$employ)
hist(bankdata$address)
hist(bankdata$income)
hist(bankdata$debtinc)
hist(bankdata$creddebt)
hist(bankdata$othdebt)


##### skeweness test


library(e1071)


for(x in numeric_col)
{
  print(x)
  skewtest = skewness(bankdata[,x])
```

```
  print(skewtest)
}



#################### sampling of data as training and testing ####################

set.seed(1234)

train.index = createDataPartition(bankdata$default, p = .80, list = FALSE)
train = bankdata[ train.index,]
test  = bankdata[-train.index,]

train$default=as.factor(train$default)
str(train$default)

################ Logistic Regression ################

library(caret)
logit_model = glm(default ~ ., data = train, family = "binomial")
summary(logit_model)

#prob_pred = predict(logit_model, test)
#logit_predict = predict(logit_model , test[] ,type = 'response' )
#logit_predict <- ifelse(logit_predict > 0.5,1,0) # Probability check
```

```
#CM= table(test , logit_predict)



##Decision tree for classification
#Develop Model on training data
library(C50)
str(bankdata$default)
bankdata$default[bankdata$default %in% "1"] = "yes"
bankdata$default[bankdata$default %in% "0"] = "no"

C50_model = C5.0(default ~., train, trials = 100, rules = TRUE)

#Summary of DT model
summary(C50_model)



#Lets predict for test cases
C50_Predictions = predict(C50_model, test, type = "class")

##Evaluate the performance of classification model
ConfMatrix_C50 = table(test$default, C50_Predictions)
confusionMatrix(ConfMatrix_C50)

#False Negative rate
```

```
#FNR = FN/FN+TP



################## Random Forest ##################


RF_model = randomForest(default ~ ., train, importance = TRUE, ntree = 500)
#Presdict test data using random forest model
RF_Predictions = predict(RF_model, test[])
##Evaluate the performance of classification model
ConfMatrix_RF = table(test$default, RF_Predictions)
confusionMatrix(ConfMatrix_RF)
#False Negative rate
#FNR = FN/FN+TP



########################## Here We got more accuracy with Random Forest ########
Result = data.frame('Actual_target' = test$default, 'Predicted_target' =RF_Predictions  )
write.csv(Result,"PREDICTION_R.csv",row.names=FALSE)
```

REFERENCES

- StackOverflow
- www.towardsdatascience.com
- www.medium.com