MULTIVARIATE ANALYSIS OF BANK LOAN DATA

SAHIL SAINI (191118)
RACHNA CHAURASIA (191100)
ASHWINI LAKRA (191027)
RUPVANT DAYANAND WAGHMARE (191116)

M.Sc. Statistics, 2nd Year, Semester IV
Department of Mathematics and Statistics, Indian Institute of Technology, Kanpur

Date of Submission: May 10, 2021

Abstract

In this project we have analysed a bank loan data containing information of individuals applying for a loan. We have done data cleaning by label encoding, missing value imputations using KNN and Logistic regression, outlier detection, used quantile-based flooring and capping technique to get rid of outliers and have used and compared three classifiers- Logistic regression, SVM and Naïve Bayes used for classification problem. We have performed bivariate data analysis with the help of pie-charts and heat map. Through our analysis, by drawing insights from the data, we have drawn conclusions regarding defaulters i.e., when is a loan applicant most likely to be a defaulter. We have provided some recommendations to the bank regarding the changes to be made in their guidelines and whether the bank must change their approval standards or maintain the same approach regarding loans.

Contents

1. Introduction

Data cleaning is the key step before performing exploratory data analysis. It is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. Without cleaning the data none of the algorithms used for analyzing the data would give accurate results.

When an analyst plans to fit a model on the given data, it is not always possible to decide which classifier is good just by observing the dataset. So the basic practice is to compare different classifiers and choose one which is giving better results.

After completing the above two steps then we can perform the exploratory data analysis in order to draw insights from the data and thus obtain best results.

These are the three main objectives of our project. In section 2 we have given a description of the data. In section 3 we have done data cleaning by label encoding, missing value imputations using KNN and Logistic regression, outlier detection, used quantile-based flooring and capping technique to get rid of outliers. In section 4 we have applied and compared three classifiers- Logistic regression, SVM and Naïve Bayes used for classification problem. In Section 5 we performed bivariate data analysis using pie-charts and heat map. Lastly, in section 6 we have discussed about the insights drawn from the data and based on those have provided some suggestions.

2. Data Description

The dataset provided is information of 850 applicants over 11 variables for a bank loan. The description for the variables are as follows:

1. **age:** Age of the applicant (In Years)
2. **Gender:**
   0 Male
   1 Female
3. **Employment:** Field of Employment
   0 Employee
   1 Professional
   2 Business
4. **educ:** The education level of the applicant
   1 Did not complete high school
   2 High school degree
   3 Some college
   4 College degree
   5 Post-undergraduate degree
5. **employ:** Number of years with current employer
6. **Income:** Household income (In thousands)
7. **debtinc*:** Debt to income ratio
   *Debt-to-income ratio is one's all monthly debt payments divided by his/her gross monthly income. This number is one way, lenders measure someone's ability to manage the monthly payments to repay the money he/she plans to borrow.
8. **creddebt:** Credit card debts (In thousands)
9. **otherdebt:** Other types of debts (In thousands)
10. **default**:** Previously Defaulted Applicants
    0 NO
    1 YES
    **Default is the failure to repay a debt, including interest or principal
11. **address:** Years at current address

3. Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. So, in our raw dataset, there are many variables that require data cleaning. Here are some of the rows of our dataset:

	age	educ	employ	address	income	debtinc	creddebt	othdebt	default	Gender	Employment
0	41yrs	Post-undergraduate degree	17	12	176	9.3	11.36	5.01	yes	male	0.0
1	27yrs	Post-undergraduate degree	10	6	31	17.3	1.36	4.00	NO	male	2.0
2	40yrs	Did not complete high school	15	14	55	5.5	0.86	2.17	NO	F	1.0
3	41yrs	Post-undergraduate degree	15	14	120	2.9	2.66	0.82	NO	F	1.0
4	24yrs	High school degree	2	0	28	17.3	1.79	3.06	yes	male	2.0
...	...	...	...	...	...	...	...	...	...	...	...
845	34yrs	Did not complete high school	12	15	32	2.7	0.24	0.62	NaN	NaN	NaN
846	32yrs	Post-undergraduate degree	12	11	116	5.7	4.03	2.59	NaN	NaN	NaN
847	48yrs	Post-undergraduate degree	13	11	38	10.8	0.72	3.38	NaN	NaN	NaN
848	35yrs	College degree	1	11	24	7.8	0.42	1.45	NaN	NaN	NaN
849	37yrs	Post-undergraduate degree	20	13	41	12.9	0.90	4.39	NaN	NaN	NaN

850 rows × 11 columns

The variables '**age**', '**educ**', '**default**', '**Gender**' and '**Employment**' will be cleaned and the missing values will be imputed using model imputation. To the remaining 4 variables, i.e.,'**income**', '**debtinc**', '**creddebt**', '**othdebt**' we will give outlier correction treatment.

For the '**age**' variable, the rows should be numeric so we clean it by removing 'yrs' in it.

For the '**educ**' variable, we allocate labels to the 4 different classes and then we will do label encoding.
This is what we have,

```
Did not complete high school    240
High school degree              179
Some college                    154
Post-undergraduate degree       152
College degree                  125
Name: educ, dtype: int64
```

We give 'A' to 'Post-undergraduate degree', 'B' to both 'some college' and 'college degree' as we can consider them same, 'C' to 'High school degree' and 'D' to 'Did not complete high school'. So, we get the following counts:

```
B    279
D    240
C    179
A    152
Name: educ, dtype: int64
```

Now, we do **Label Encoding** and get '0' for 'A', '1' for 'B', '2' for 'C' and '3' for 'D'. For the '**Gender**' variable, there should have been just 2 classes i.e., male and female but we observed 8 classes as can be seen in the following:

```
FEMALE    172
M         106
F         106
Female     93
MALE       73
male       52
m          51
female     47
Name: Gender, dtype: int64
```

So, we take all FEMALE, F, Female, female classes and allocate them into 1 class and give label '0' while for the other class, give label '1'. Then we get as follows:

```
0    418
1    282
Name: Gender, dtype: int64
```

For the '**default**' variable, there should be just 2 classes i.e., yes and no so we similarly take all classes 'Y', 'Yes', 'yes' ,'YES' and allocate them to 1 class and give label '0' and for the other class, give label '1'. We get as follows:

```
0    517
1    183
Name: default, dtype: int64
```

For the '**Employment**' variable, we have the following where 0.0, 1.0, 2.0 correspond to different fields of employment as per in the description:

```
1.0    334
2.0    195
0.0    171
Name: Employment, dtype: int64
```

From above, we can see that for the '**Gender**', '**default**' and '**Employment**' variables, there are just 700 rows or observations and total is 850, so 150 rows are missing. Instead of just ignoring or dropping the missing values, we do **Missing Value Imputation**.

3.1 Missing value imputation in 'Gender' variable using KNN:

A popular approach to missing data imputation is to use a model to predict the missing values. Although any one among a range of different models can be used to predict the missing values, the k-nearest neighbor (KNN) algorithm has proven to be generally effective, often referred to as "nearest neighbor imputation". KNN replaces missing values using the mean squared difference of nearest non-missing feature values. So, here we use fancyimpute to replace missing values in huge data sets. Compared to commonly used imputing techniques like replacing with median and mean, this method yields better model accuracy.

```
0.0    512
1.0    338
Name: Gender, dtype: int64
```

3.2 Missing value imputation in 'Employment' variable by estimation using Logistic Regression:

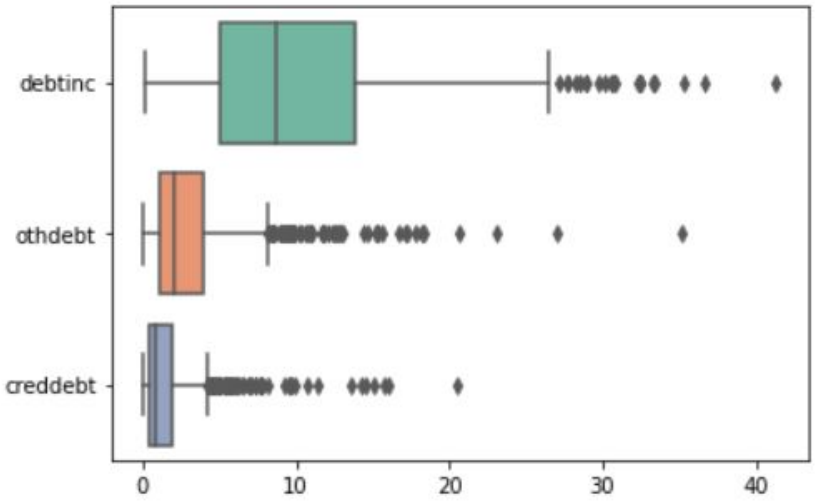
The technique that we are using here is called **Model imputation**. Here's a fun trick. To prepare a dataset for machine learning we need to fix missing values, and we can fix missing values by applying machine learning to that dataset! If we consider a column(here 'Employment' variable) with missing data as our target variable, and existing columns with complete data as our predictor variables, then we may construct a machine learning model using complete records as our train and test datasets and the records with incomplete data as our generalization target. This is a fully scoped-out machine learning problem. So, this is in general for any machine learning algorithm but here we are going to use Logistic Regression.
After applying logistic regression and thereafter predicting the missing values, we get:

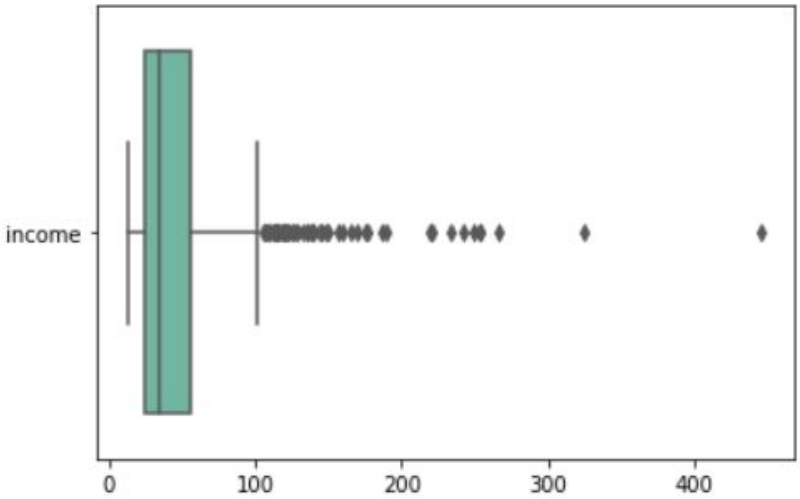
```
1.0    129
2.0     20
0.0      1
dtype: int64
```

This shows that the classification is not done properly as we can see that only 1 observation is classified to 0 class while 129 observation are classified to 1 class, so we are unable to model the '**Employment**' Variable using other predictor variables. We are not imputing the missing values in the 'Employment' variable because there are no necessary independent variables which can model the 'Employment' variable.

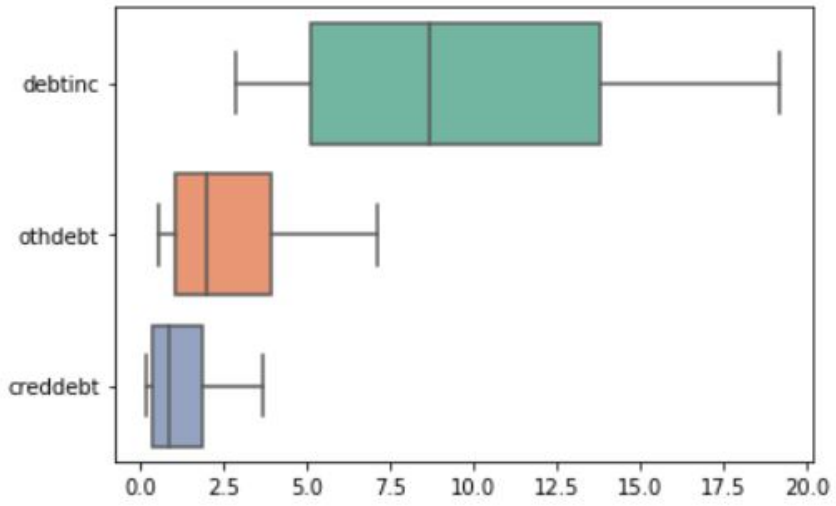
3.3 Outlier Detection

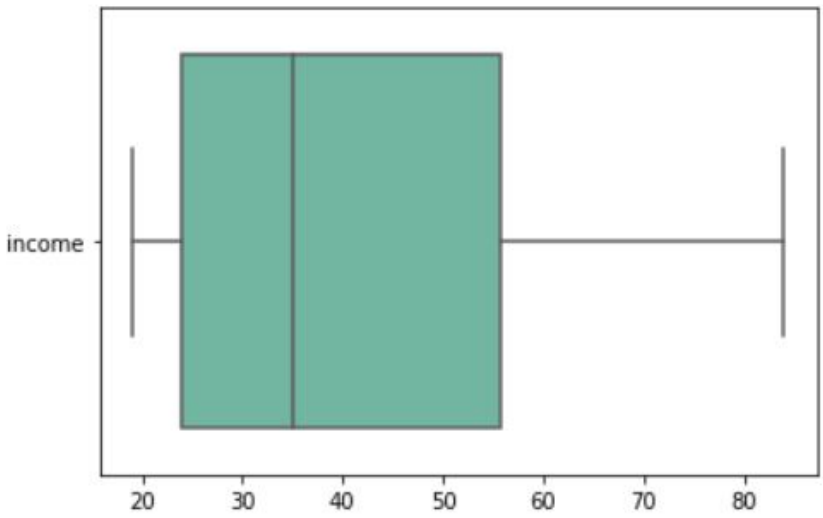
In descriptive statistics, a box plot is a method for graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (whiskers) indicating variability outside the upper and lower quartiles, hence the terms box-and-whisker plot and box-and-whisker diagram. Outliers may be plotted as individual points. Here, we show the box-and whisker plot for '**debtinc**', '**othdebt**', '**creddebt**' and '**income**'.





From the plots, we can observe that there are some outliers for each of the variables and we should get rid of them by using some correction technique. Here, we use **Quantile-based Flooring and Capping technique** to overcome the problem of outliers. In this technique, we will do the flooring (e.g., the 10th percentile) for the lower values and capping (e.g., the 90th percentile) for the higher values.





Now, we can see that there are no outliers in our data. Apart from this, our data is not symmetrical as the median is not in the center, rather it is positively skewed for these four variables.

4. Classification:

Proceeding further, the '**default**' variable that has 150 missing value is still remaining. Now, we will use Model imputation again but now with three different classifiers to predict the missing values of the '**default**' variable and compare which classifier gives the highest accuracy for our dataset. We consider the '**default**' variable as our response and remaining variables having no missing values as predictors.
Here we have imbalanced dataset as there are few number of defaulters and more number of non-defaulters, so we consider F1-score instead of accuracy.

4.1 Classification of 'default' variable using Logistic Regression:

Logistic regression is a supervised classification algorithm. In such problems, response variable can take only discrete values for a given set of predictors. Logistic prediction basically predicts whether something is true or false instead of predicting something continuous (like size). Also, instead of fitting a line to the data, logistic regression fits an 'S' shaped 'logistic function'.
Let y denotes the response variable and $x_1, x_2, \cdots, x_p$ denotes the predictors, the logistic function has the form

$$P_z = \frac{e^{\alpha+\beta_1 x_1+\beta_2 x_2+.....+\beta_p x_p}}{1+e^{\alpha+\beta_1 x_1+\beta_2 x_2+.....+\beta_p x_p}}$$

This equation is called the logistic regression equation. Here, Z is the linear function $\alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p$. The odds ratio is defined as:

$$odds = \frac{P_z}{1-P_z}$$

$$\Rightarrow P_z = \frac{odds}{1+odds}$$

By performing some algebraic manipulation and taking the natural logarithm of the odds (ln(odds) is also known as logit) we obtain:

$$ln(odds) = ln(\frac{P_z}{1-P_z}) = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p = Z$$

Thus, we obtain the multiple linear regression equation and as a result the logistic function is also known as the multiple logistic regression equation.
The fundamental assumption in logistic regression analysis is that ln(odds) is linearly related to the independent variables.Logistic regression can work with both continuous and discrete data which makes it a popular Machine Learning method.

Firstly by using Logistic Regression, we get an F1-score of 0.91.

	precision	recall	f1-score	support
0	0.88	0.94	0.91	77
1	0.72	0.57	0.63	23
accuracy			0.85	100
macro avg	0.80	0.75	0.77	100
weighted avg	0.84	0.85	0.84	100

4.2 Classification of 'default' variable using SVM Classifier:

Support Vector Machine (SVM) is a supervised Machine Learning algorithm mostly used for classification, outlier detection and sometimes for regression. SVM can only perform binary classification. In SVM each data item in a dataset is plotted in an N-dimensional space, where N is the number of features or attributes in the data. The objective of SVM is to find a hyperplane in a N-dimensional space that has the maximum margin i.e., the maximum distance between the data points of both classes and that distinctly classifies the data points. If the number of attributes is 2, then hyperplane is just a line and if the number of attributes is 3, then the hyperplane becomes a 2-d plane and so on.
Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, the margin of the classifier is maximized. When the support vectors are deleted the position of the hyperplane is changed. These are the points that help in building SVM.

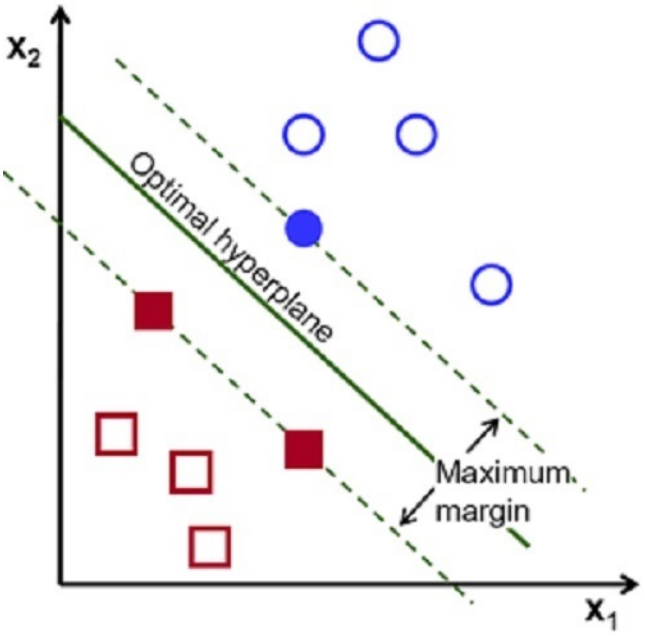


Figure 1: Pictorial Representation of SVM

It is effective in high dimensional spaces and also when the number of variables is greater than the number of samples.
Then, by using SVM classifier, we get an F1- score of 0.90.

	precision	recall	f1-score	support
0	0.87	0.94	0.90	77
1	0.71	0.52	0.60	23
accuracy			0.84	100
macro avg	0.79	0.73	0.75	100
weighted avg	0.83	0.84	0.83	100

4.3 Classification of 'default' variable using Naïve Bayes Classifier:

Naïve Bayes is a special form of discriminant analysis which makes predictions using Bayes theorem with the 'naïve' assumption of conditional independence between every pair of features given the value of the class variable. However, it works well even if independence assumption is clearly violated as there is no need for accurate probability estimates for classification so long as the greatest probability is assigned to the correct class. The Naïve Bayes classifier is very useful in high-dimensional problems. It is a stable algorithm and a small change in the training data will not make a big change in the model. It is also extremely fast compared to more sophisticated methods.
The methodology of the Naïve Bayes classifier is as follows:
Given, class variable y and dependent feature vector x_1 through x_n.
Bayes theorem states that:

$$P(y|x_1, \cdots\cdots, x_n) = \frac{P(y)P(x_1, \cdots\cdots, x_n|y)}{P(x_1, \cdots\cdots, x_n)}$$

Using the 'naïve' assumption of conditional independence that
$P(x_i|y, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n|y) = P(x_i|y)\ \forall$ i we get,

$$P(y|x_1, \cdots\cdots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \cdots\cdots, x_n)}$$

Since, the denominator is constant for a given output, we use the following classification rule:

$$P(y|x_1, \cdots\cdots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

$$\Rightarrow \hat{y} = arg\, max_y P(y)\prod_{i=1}^{n} P(x_i|y)$$

Thus finally, with Naïve Bayes Classifier, we get an F1- score of 0.86.

	precision	recall	f1-score	support
0	0.84	0.88	0.86	77
1	0.53	0.43	0.48	23
accuracy			0.78	100
macro avg	0.68	0.66	0.67	100
weighted avg	0.77	0.78	0.77	100

Among these classifiers, Logistic and SVM classifier have almost same F1- score, so we can predict the missing values using any of the two. Although, we use Logistic classifier and predict the missing values.
Now, we have finally fully cleaned our dataset and we have done anomaly detection. We don't have any missing values in our dataset except the '**Employment**' variable. So, here's what we have:

	age	educ	employ	address	income	debtinc	creddebt	othdebt	Gender	Employment	default
0	41.0	0.0	17.0	12.0	84.0	9.3	3.70	5.01	1.0	0.0	1
1	27.0	0.0	10.0	6.0	31.0	17.3	1.36	4.00	1.0	2.0	0
2	40.0	3.0	15.0	14.0	55.0	5.5	0.86	2.17	0.0	1.0	0
3	41.0	0.0	15.0	14.0	84.0	2.9	2.66	0.82	0.0	1.0	0
4	24.0	2.0	2.0	0.0	28.0	17.3	1.79	3.06	1.0	2.0	1
...	...	...	...	...	...	...	...	...	...	...	...
845	34.0	3.0	12.0	15.0	32.0	2.9	0.24	0.62	0.0	NaN	0
846	32.0	0.0	12.0	11.0	84.0	5.7	3.70	2.59	0.0	NaN	0
847	48.0	0.0	13.0	11.0	38.0	10.8	0.72	3.38	1.0	NaN	0
848	35.0	1.0	1.0	11.0	24.0	7.8	0.42	1.45	0.0	NaN	0
849	37.0	0.0	20.0	13.0	41.0	12.9	0.90	4.39	1.0	NaN	0

850 rows × 11 columns

Summary of the data:

	age	educ	employ	address	income	debtinc	creddebt	othdebt	Gender	Employment
count	850.000000	850.000000	850.000000	850.000000	850.000000	850.000000	850.000000	850.000000	850.000000	700.000000
mean	35.001176	1.596471	8.565882	8.371765	41.840000	9.737294	1.298065	2.702141	0.397647	1.034286
std	8.070873	1.079081	6.777884	6.895016	21.355377	5.332745	1.130080	2.083245	0.489700	0.722792
min	20.000000	0.000000	0.000000	0.000000	19.000000	2.900000	0.179000	0.550000	0.000000	0.000000
25%	29.000000	1.000000	3.000000	3.000000	24.000000	5.100000	0.380000	1.050000	0.000000	1.000000
50%	35.000000	1.000000	7.000000	7.000000	35.000000	8.700000	0.885000	2.005000	0.000000	1.000000
75%	40.000000	3.000000	13.000000	12.000000	55.750000	13.800000	1.900000	3.905000	1.000000	2.000000
max	56.000000	3.000000	33.000000	34.000000	84.000000	19.200000	3.700000	7.110000	1.000000	2.000000

5. Bivariate Data Analysis

Under this Data analysis part, we mainly study the variation in number of defaulters with respect to different features that we have and how they are impacting the count of defaulters. Also, we shall give some recommendations to the bank regarding the changes to be made in their guidelines in terms of the features that we have and whether the bank must consider changing their approval standards or maintain the same approach regarding loans.

To conduct the above analysis, we mainly plot pie-charts of different independent variables with respect to response (default) variable. Also, we plot a heat map to better understand our data and relation between the features.

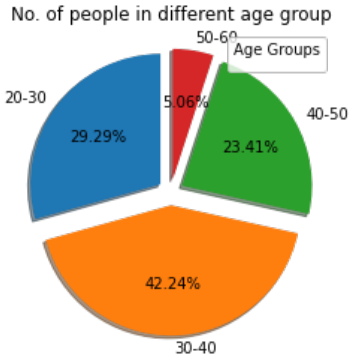
Our strategy to better analyze our data is that we shall firstly make some sub-groups of the feature on the basis of which we shall be comparing the number of defaulters in each sub-group of a feature with the corresponding number of people in that sub-group of that feature. From this, we can relate that out of total people falling in that sub-group, how many of them are defaulters and thereafter we can give some recommendations with respect to our observations from the pie-charts.

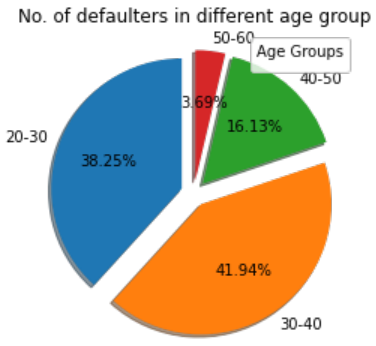
5.1 For 'age' variable:

We do grouping of the '**age**' variable in the following way-

Group 1	'20-30' years
Group 2	'30-40' years
Group 3	'40-50' years
Group 4	'50-60' years

Now, we plot the pie-charts for the number of people and number of defaulters in the above stated age groups-





Comparing the two pie-charts, we can clearly see that percentage of people in age group 20-30 is around 30% and percentage of defaulters in that age group is more than

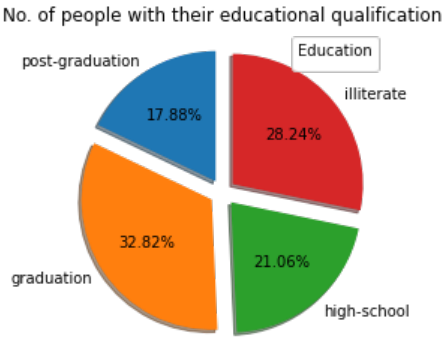
38%. So, we can say that a person falling in the age group '20-30' is more likely to be a defaulter.

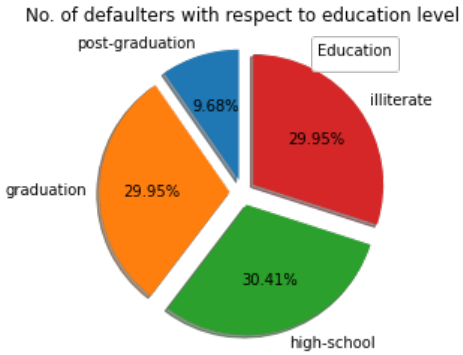
5.2 For 'educ' variable:

We do grouping of the '**educ**' variable in the following way-

Group 1	'post-graduation'
Group 2	'graduation'
Group 3	'high-school'
Group 4	'illiterate'

Now, we plot the pie-charts for the number of people and number of defaulters in the above stated education qualification groups-





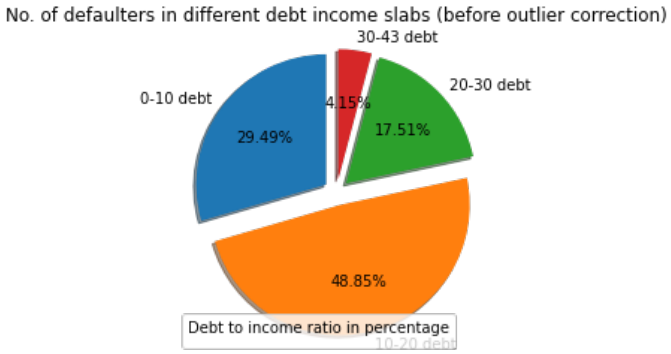
Comparing the above two pie-charts, we observe that percentage of people who have educational qualification as high school and illiterate is 21% and 28% respectively and the percentage of defaulters in the respective group is 31% and 30%. So, we can say that a person who is illiterate or high school pass out is more likely to be a defaulter.

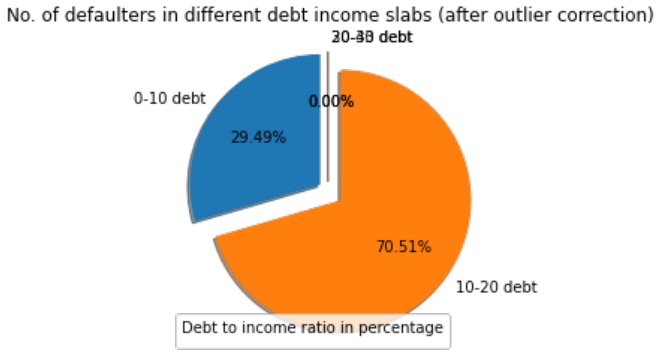
5.3 For 'debtinc' variable:

We do grouping of the '**debtinc**' variable in the following way-

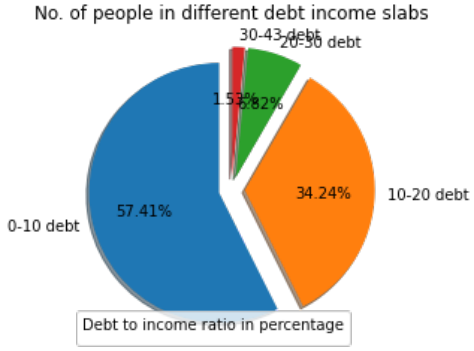
Group 1	'0-10 debt'
Group 2	'10-20 debt'
Group 3	'20-30 debt'
Group 4	'30-43 debt'

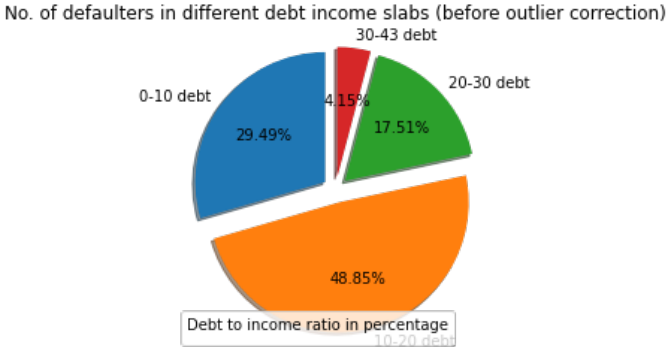
Now, we have two choices whether we should use the variable before outlier correction or after outlier correction. We will plot the pie chart using both and then we will see which pie-chart is giving more information.





It can be observed that the variable before outlier correction is giving more information so we will use it for bivariate analysis.





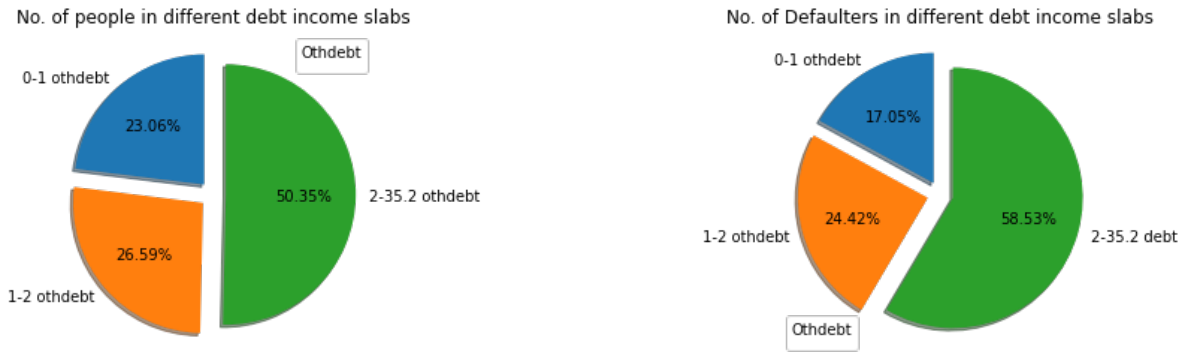
Comparing the above two pie-charts, we observe that percentage of people in group '30-43 debt' is around 1.5% and percentage of defaulters in that age group is 4.15% (which is roughly 3 times of the % of people). So, we can say that a person falling in the group '30-43 debt' is more likely to be a defaulter.

5.4 For 'othdebt' variable:

We do grouping of the '**othdebt**' variable in the following way-

Group 1	'0-1 othdebt'
Group 2	'1-2 othdebt'
Group 3	'2-35.2 othdebt'

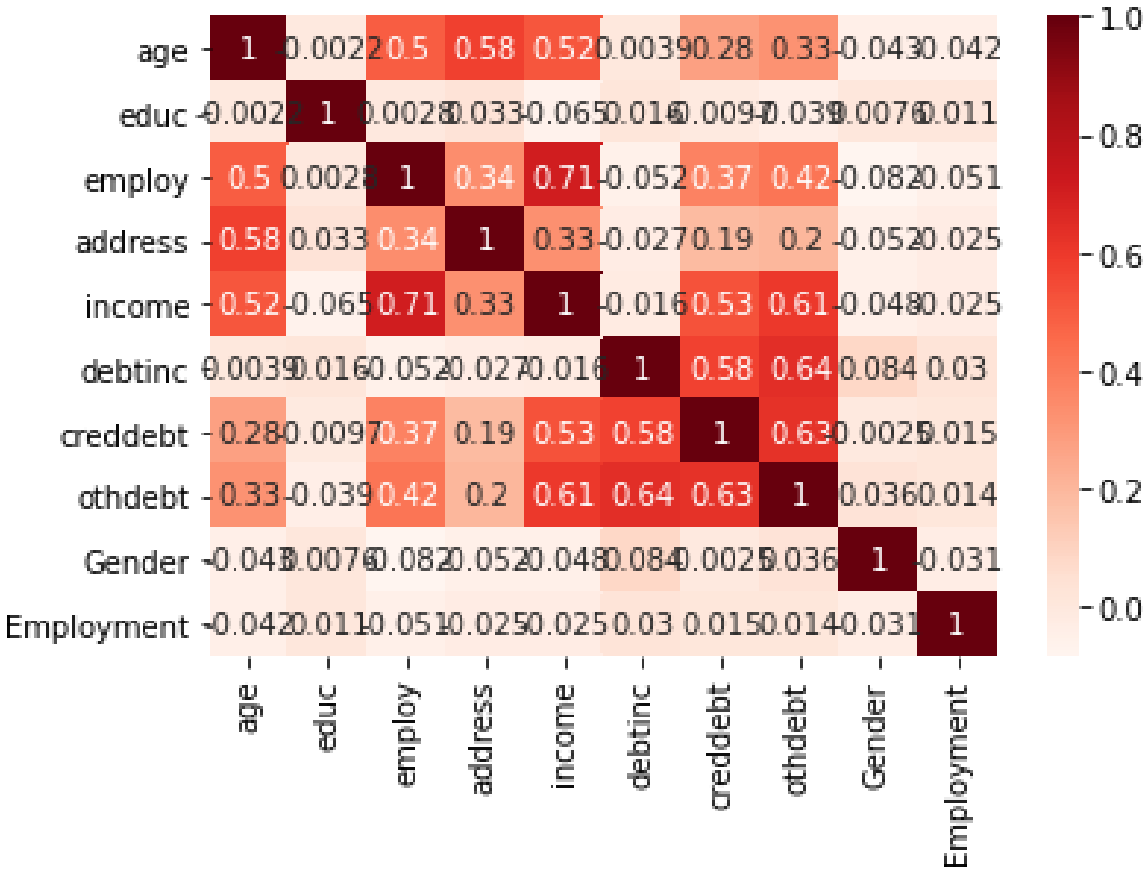
Now, we plot the pie-charts for the number of people and number of defaulters in the above stated number of other debt groups-



Comparing the above two pie-charts, we observe that percentage of people in group 2-35.2 other debt is around 50% and percentage of defaulters in that group is 59%. So, we can say that a person falling in the group 2-35.2 other debt is more likely to be a defaulter.

5.5 For other variables using Heat Map

This explains the correlation between different pairs of variables.



6. Conclusion

- Bank should make strict loan approving policy for the 'age' group 20-30 as there are high chances that a person in this age group may be a defaulter.
- It is recommended that for illiterates and high school pass outs, there should be strict scrutiny.
- If a person is having high debt income ratio, bank should give this case a special attention as there are high chances that the person is a defaulter.
- For 'othdebt', the above policy follows as 'debtinc'.
- 'othdebt' and 'income' has a correlation coefficient of 0.61 which implies a person having less income is more likely to be a defaulter. Likewise, a person having larger stay at a company is less likely to be a defaulter.
- It can be observed that 'income' and 'employ' variables are highly correlated with the correlation coefficient of 0.71 which implies that greater the number of years in a company, greater is the income.
- 'othdebt' and 'income' has a correlation coefficient of 0.61 which implies a person having less income is more likely to be a defaulter. Likewise, a person having larger stay at a company is less likely to be a defaulter.

7. References

1) https://datacadamia.com/data_mining/naive_bayes
2) https://scikit-learn.org/stable/modules/naive_bayes.html
3) https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/
4) https://scikit-learn.org/stable/modules/svm.html
5) Towards Data Science-Support Vector Machine (SVM)
6) https://www.geeksforgeeks.org/naive-bayes-classifiers/
7) https://www.geeksforgeeks.org/understanding-logistic-regression/
8) Afifi, A., May, S., Donatello, R., & Clark, V. A. (2019). Practical multivariate analysis. CRC Press.
9) https://www.quora.com/Why-do-we-call-an-SVM-a-large-margin-classifier

8. Acknowledgement

We would like to thank our professor Dr. Minerva Mukhopadhyay for giving us this golden opportunity and for encouraging us to do this project.