Czech University of Life Sciences, Prague

Faculty of Economics and Management

Department of Statistics

Diploma Thesis on:

The application of machine learning algorithms in credit scoring

Prepared by: Hailegnaw Niguss Solomon

Thesis Supervisor: Ing. Tomáš Hlavsa, Ph.D.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Ing. Niguss Solomon Hailegnaw

Informatics

Thesis title

The application of machine learning algorithms in credit scoring

---

Objectives of thesis

The goal of this thesis is to build a credit scoring model based using machine learning algorithms.
The author will focus on
- Appliying selected machine learning algorithm in credit scoring
- Identifying credit defaulter by efficient machine learning algorithm and implicate the accuracy potential of machine learning in the finacial sectors

Methodology

Big Data with observation greater than 20 000 rows will be obtained from legally available online sources or from other dates warehouse. The data will be preprocessed prior to our data mining technique, predictive analysis by Logistic regression. Under preprocessing step selected data will be cleaned to avoid any inconsistency, furthermore both missing data and outliers will be treated accordingly. Among the variables on the data set the most significant variables will be selected. After this all, the data will be ready for the subsequent data mining phase. In this phase predictive analysis specially, logistic regression will be done to predict credit scoring by using python.

The proposed extent of the thesis

60 – 80 pages

Keywords

Big Data, Sample size, Sampling Technique, Logistic Regression, Decision making

Recommended information sources

ABBOTT, D. Applied Predictive Analytics : Principles and Techniques for the Professional Data Analyst. Praha: John Wiley & Sons, Incorporated, 2014. ISBN 9781118727935.

AGRESTI, A. *Categorical data analysis.* Hoboken: John Wiley & Sons, 2013. ISBN 978-0-470-46363-5.

Dong, G., Lai, K.K., Yen, J., 2010. Credit scorecard based on logistic regression with random coefficients. Procedia Comput. Sci., ICCS 2010 1, 2463–2468. https://doi.org/10.1016/j.procs.2010.04.278

TUFFÉRY, S. Data Mining and Statistics for Decision Making. UK, West Sussex: Wiley, 2011. ISBN 978-0-470-68829-8.

VanderPlas, J., 2016. Python data science handbook: Essential tools for working with data. O'Reilly Media, Inc. USA

Yap, B.W., Ong, S.H., Husain, N.H.M., 2011. Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Syst. Appl. 38, 13274–13283. https://doi.org/10.1016/j.eswa.2011.04.147

Expected date of thesis defence

2019/20 SS – FEM

The Diploma Thesis Supervisor

Ing. Tomáš Hlavsa, Ph.D.

Supervising department

Department of Statistics

Electronic approval: 24. 1. 2019	Electronic approval: 5. 2. 2019
prof. Ing. Libuše Svatošová, CSc.	**Ing. Martin Pelikán, Ph.D.**
Head of department	Dean

Prague on 07. 12. 2019

Declaration

I declare that this diploma thesis work on Uptake of nutrients by wheat and maize from soil treated with ash from biomass incineration is my own work and all the sources I cited in it are listed in References.

Prague, 8th April 2015 ………………..

Niguss Solomon Hailegnaw

Acknowledgement

Above of all, I would like to express my thankfulness to almighty God, who gave me the strength, commitment and health to finish this master thesis.

Secondly, I would like to pass my appreciation to my supervisor Ing. Tomas Hlavsa Ph.D, for his support and advice while the process of writing this thesis work.

I would like also to pass my gratitude to my wife Feven Shewangizaw for her patience, love and constructive support, to my firstborn Iwnetim Niguss and my parents, who were always on the side of me to unleash all the barriers.

Finally, I would like to say thank all my friends, who have advised me, while writing this thesis.

Aplikace algoritmů strojového učení při bodování

Souhrn

V této práci byl použit princip strojového učení pomocí logistických regresí, rozhodovacích stromů a algoritmů neuronové sítě pro účely kreditního bodování. K dosažení tohoto cíle byla stažena velká datová sada vypůjčovatelů s více než 8 000 000 pozorováními od americké úvěrové společnosti LendingClub. Výběr důležitých znaků nebo proměnných byl proveden výběrem proměnných s chybějícími hodnotami s méně než 50% celkových pozorování. Chybějící hodnoty byly detekovány, vyplněny střední hodnotou, módem nebo mediánem proměnných nebo odpovídajícím způsobem odstraněny z datového souboru. Data byla také zkontrolována na případné duplicitní řádky, odlehlé hodnoty a kolinearitu. Po předzpracování dat byla provedena určitá vizualizace dat pro detekci možného vztahu mezi proměnnými a cílovou proměnnou zvanou půjčka_status. Poté, co byly všechny kredity modelovány pomocí logistické regrese, rozhodovacích stromů a neuronových sítí. Jejich predikční kvalita byla kontrolována pomocí kritérií zmatených metrik a oblastí charakteristik operátora příjemce (ROC) pod křivkou (AUC). Na základě výsledku našeho modelování Logistická regrese předčí rozhodovací strom i neuronovou síť. Na základě výsledků této práce se doporučuje použití logistické regrese pro konkrétní případ ratingu než rozhodovací strom a neuronová sí. Výše úvěru, doba výpůjčky, roční příjem, celková splátka dluhu dlužníka nad závazkem, dotazy za posledních 6 měsíců, revolvingové nástroje, celkový účet a poslední vysoká hodnota fico jsou nejdůležitějším plně prediktorem hodnoty defaulterů *p* nižší než 0,000.

Klíčová slova: Strojové učení; úvěrový rating; logistická regrese; rozhodovací stromy, neuronové sítě.

The application of machine learning algorithms in credit scoring

Summary

In this thesis the principle of machine learning by using logistic regression, decision tree and neural network algorithms have been used for the purpose of credit scoring. To achieve this, a big data set of borrowers with more than 8000, 000 observations from an American lending company called LendingClub have been downloaded. The selection of important features or variables was done by selecting variables having missing values with less than 50% of the total observations. Missing values was were detected, filled with mean, mode or median of the variables and or removed from the dataset accordingly. The data was also checked for any duplicated rows, outliers and collinearity. After the preprocessing for the data some data visualization has been done for the detection of some possible relation between the variables and the target variable called *loan_status*. After all the credits rating was modeled by using logistic regression, decision trees and neural networks. Their prediction quality checked by using confusion metrics criteria and Receiver Operator Characteristic (ROC) areas under curve (AUC). Based on the outcome of our modeling the Logistic regression overtakes both decision tree and neural network. Finally, based on the output of this work the use of logistic regression for the specific case of credit rating is recommended than decision tree and neural network. Loan amount, loan term, annual income, borrower's total debt payment over obligation, inquiries last 6 months, revolving utilities, total account and last fico range high are the most important power full predictor of defaulters *p* value of 0.000.

Key words: Machine learning; credit rating; logistic regression; decision trees, neural networks.

Contents

List of figures

List of tables

1. Introduction

Credit scoring refers the process of determining the risk of customers to default his financial obligation. Therefore, each customer will be assigned to two groups. The first groups are good customers, the customers are non-defaulter or customers who are most probably to repay their loan on time. The second groups are bad customers, who are considered to default the repayment of the loan. The identification of both good and bad type of customers is very important for the financial institution borrowing the money. Miss classification of good customers to the group of bad customers result in the loosing of income, which could possibly collect from these customers. On the other hand, miss classification of bad customers result in loose of money due to defaulters (Anderson, 2007). The need of powerful technique to identify bad and good customers and the production of big and complex data by the financial institution pushes toward the application of machine learning for credit scoring.

Machine learning is advantageous because of its powerful prediction capability with the large and complex data set. Non programable methods of credit scoring have the drawback of very slow performance with huge complex data set, in both speed and the accuracy of their predictions. On general way the algorithm used in machine learning could be supervised (learned from previous experience), unsupervised (no learning) or reinforcement where some rewarding system is applied.

Machine learning is a technique, where computers learns from a big data set and make a prediction of new data. Machine learning could be used for a wide range of application including the following but not limited to image and speech recognition, medical diagnosis, statistical arbitrage, learning associations, classification, prediction, extraction and regression.

Therefore, this thesis works mainly targeted on the application of selected machine learning algorithm for the credit scoring and the evaluation of their performance. It will present a step by step procedure of credit scoring, starting from data preprocessing up to the development of credit rating.

2. Objectives

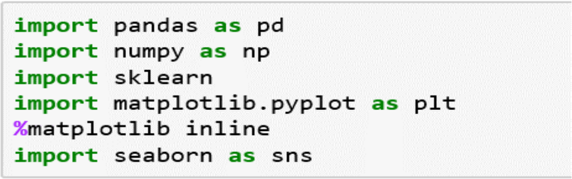
The main objective of this thesis work is to identify defaulters by efficient machine learning algorithm of logistic regression, decision tree and neural network and finally implicate the accuracy potential of machine learning in the financial sectors and compare different models.

3. Methodology

To achieve the objective stated above the data was downloaded ad 9-year separate CSV file was imported to python and merged. Especially due to the very large nature of the data used in this study, python will give us manifold advantage over statistical tools like SPSS, Statistica and even over R and SAS as long us speed is concerned. For this first we will import some python data manipulation module to Jupyter notebook, which is a web-based application allowing to write code, equations, visualization and any narrative text.

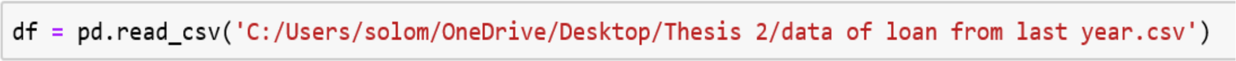
Getting important modules ready

First, important python libraries have been imported as follows.



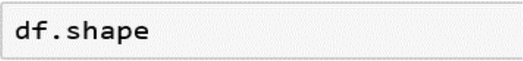
```
import pandas as pd
import numpy as np
import sklearn
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

After importin the libraries, then the downloaded data were imported to the Jupyter workbook as follows by statin the directory where the data is saved.



```
df = pd.read_csv('C:/Users/solom/OneDrive/Desktop/Thesis 2/data of loan from last year.csv')
```

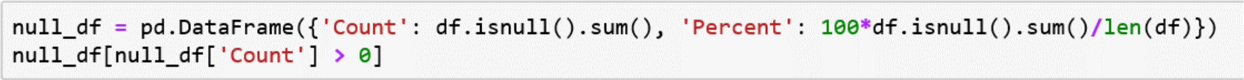
The next important step, which is checking the number of observation and number of features in the data set has been checked as follows.



```
df.shape
```

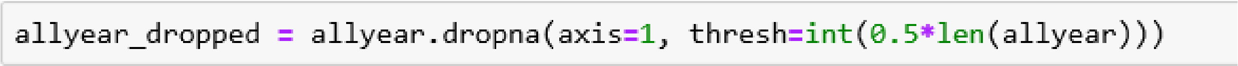
Missing values

As we have a bulk of observations with more than hundreds of features it is expected to have some sort of missing values, which have high potential of affecting our model. The missing values have been identified with the percentage of missing values for each feature as follows.



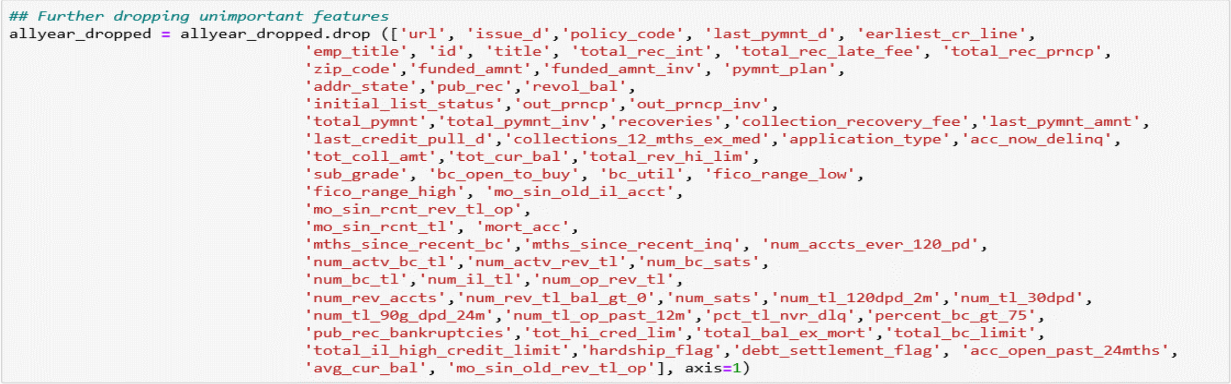
```
null_df = pd.DataFrame({'Count': df.isnull().sum(), 'Percent': 100*df.isnull().sum()/len(df)})
null_df[null_df['Count'] > 0]
```

Following the identification of missing values then the features with more than 50% of missing values have been removed as follows.



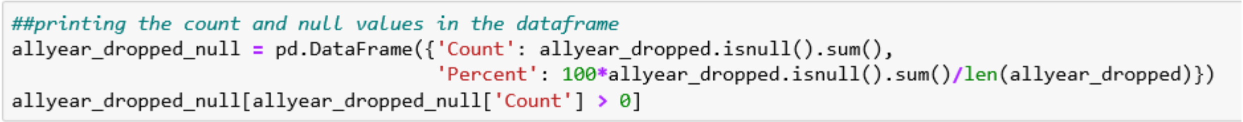
```
allyear_dropped = allyear.dropna(axis=1, thresh=int(0.5*len(allyear)))
```

Further, features with no meaning to our need of interest in regard to modeling the credit risks have been removed by the following code.



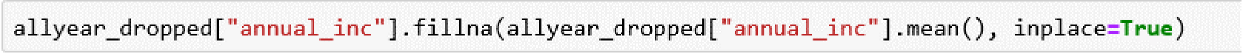
```
## Further dropping unimportant features
allyear_dropped = allyear_dropped.drop (['url', 'issue_d','policy_code', 'last_pymnt_d', 'earliest_cr_line',
                                         'emp_title', 'id', 'title', 'total_rec_int', 'total_rec_late_fee', 'total_rec_prncp',
                                         'zip_code','funded_amnt','funded_amnt_inv', 'pymnt_plan',
                                         'addr_state','pub_rec','revol_bal',
                                         'initial_list_status','out_prncp','out_prncp_inv',
                                         'total_pymnt','total_pymnt_inv','recoveries','collection_recovery_fee','last_pymnt_amnt',
                                         'last_credit_pull_d','collections_12_mths_ex_med','application_type','acc_now_delinq',
                                         'tot_coll_amt','tot_cur_bal','total_rev_hi_lim',
                                         'sub_grade', 'bc_open_to_buy', 'bc_util', 'fico_range_low',
                                         'fico_range_high', 'mo_sin_old_il_acct',
                                         'mo_sin_rcnt_rev_tl_op',
                                         'mo_sin_rcnt_tl', 'mort_acc',
                                         'mths_since_recent_bc','mths_since_recent_inq', 'num_accts_ever_120_pd',
                                         'num_actv_bc_tl','num_actv_rev_tl','num_bc_sats',
                                         'num_bc_tl','num_il_tl','num_op_rev_tl',
                                         'num_rev_accts','num_rev_tl_bal_gt_0','num_sats','num_tl_120dpd_2m','num_tl_30dpd',
                                         'num_tl_90g_dpd_24m','num_tl_op_past_12m','pct_tl_nvr_dlq','percent_bc_gt_75',
                                         'pub_rec_bankruptcies','tot_hi_cred_lim','total_bal_ex_mort','total_bc_limit',
                                         'total_il_high_credit_limit','hardship_flag','debt_settlement_flag', 'acc_open_past_24mths',
                                         'avg_cur_bal', 'mo_sin_old_rev_tl_op'], axis=1)
```

Now we have only 22 features including the target variable "*loan status*" remaining in our data set. This doesn't mean that all the remaining 22 features are without missing values, but at least they have missing value less than 50%. Therefore, for the 22 remaining variables the count of missing value has been displayed by the following code.



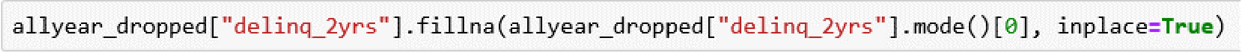
```
##printing the count and null values in the dataframe
allyear_dropped_null = pd.DataFrame({'Count': allyear_dropped.isnull().sum(),
                                     'Percent': 100*allyear_dropped.isnull().sum()/len(allyear_dropped)})
allyear_dropped_null[allyear_dropped_null['Count'] > 0]
```

For annual income 4 missing values are imputed by the mean of the column as follows.



```
allyear_dropped["annual_inc"].fillna(allyear_dropped["annual_inc"].mean(), inplace=True)
```

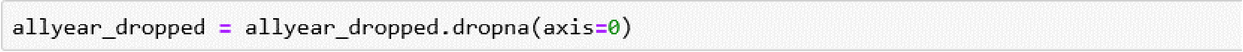
For *delinquency 2 years*, there are 29 missing values. And the values of *delinquency 2 years* are in the range between 0 and 39, with being 0 the most frequent one with total count of 717011. Therefore, missing value for this feature have been imputed with the mode of the feature, which is 0 as follows.



```
allyear_dropped["delinq_2yrs"].fillna(allyear_dropped["delinq_2yrs"].mode()[0], inplace=True)
```

Similar method has been implemented in the case of *employment length*, meaning that the missing values imputed by the most frequent value in the column, which is *10+ years*.

The imputation of missing values with the mode of the features have been done as well for *charged off within 12 months, tax liens* as that of *delinquency 2 years*. In the case of the remaining features with the missing values (*dti* (ratio of borrowers total debt payment over obligation), *revolving utility*, *total account* and *delinquency amount*), due to the unclear trend in the data and high number of observation it have been decided to drop rows with any possible missing values as follows.



```
allyear_dropped = allyear_dropped.dropna(axis=0)
```

And still we have remained with 886,936 observations**,** which indicates we do not need to worry much about the number of observations we have.

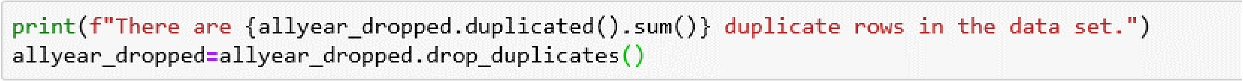
Outliers

The outliers have been checked by box plot and dealt accordingly. Basically, outliers are capped or transformed according to the specific category of the variables. For example, the outliers for the continuous numeric variables *annual income* are mainly the income above or below certain amount. For this the upper 95% and the lower 5% quantile have been used as a cutting-edge value. Similar technique has been applied for *total account, dti, and last fisco range high,*

In the case of categorical variables purpose, the category with less than 1% of distribution have been merged and assigned as new category in the variables.

Duplicated rows

All possible duplicity in the data set have been checked as follows.



```
print(f"There are {allyear_dropped.duplicated().sum()} duplicate rows in the data set.")
allyear_dropped=allyear_dropped.drop_duplicates()
```

Multicollinearity

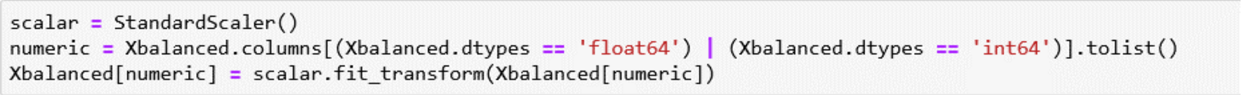
The high collinearity between the predictor variables is among the biggest challenge of the statistical data modeling. Multicollinearity could potentially affect the hypothesis testing and coefficient of the estimated parameters by creating inaccurate and unstable estimate of variance (Midi et al., 2010). A heatmap of Pearson correlation matrix for all predictor variable is presented below (Figure 16). Based on our heatmap *loan_amnt* and *installment*, *last_fico_range_high* and *last_fico_range_low* are highly correlated, meaning that we should avoid one of those variable from the combination to avoid the effect of multicollinearity from our model. Therefore, the *installment* and *last_fico_range_low* are dropped from being considered in the model.

Scaling the data to standard normal distribution

Due to very wide range of values within the variables a scaling of variables by standardization have been done. Any normal variable can be transformed to z-score as follows.

$$z = (X - \mu) / \sigma \qquad \text{eq 1.}$$

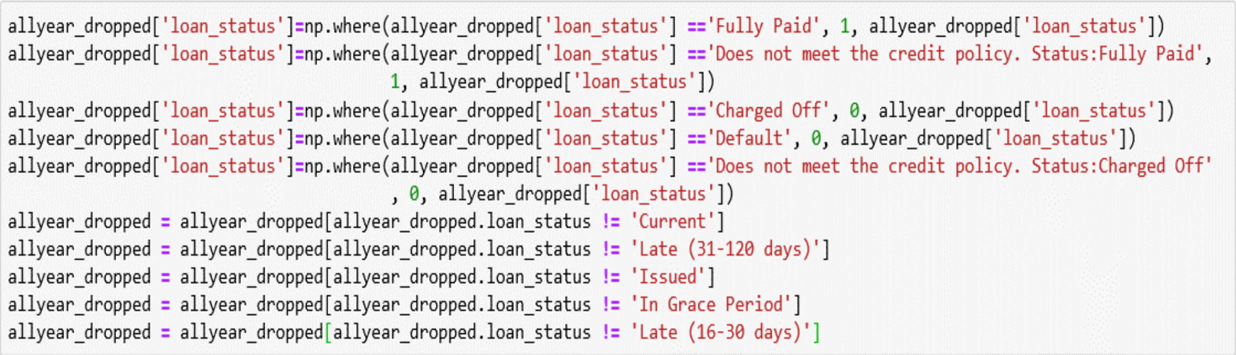
where X is variable of target μ is mean and σ is standard deviation. The scaling of all the variables can be done by python easily as follows.



```
scalar = StandardScaler()
numeric = Xbalanced.columns[(Xbalanced.dtypes == 'float64') | (Xbalanced.dtypes == 'int64')].tolist()
Xbalanced[numeric] = scalar.fit_transform(Xbalanced[numeric])
```

Feature engineering

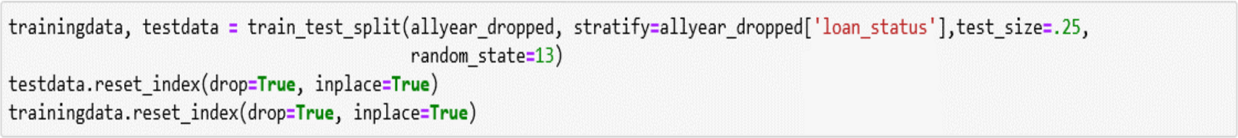
Afterward, to suits our model of credit scoring we need a binary target variable and here the categories *Fully paid and Does not meet the credit policy. Status:Fully Paid* considered as non-defaulter and assigned as 1, while *Charged Off, Default and Does not meet the credit policy, Status:Charged Off* are considered as defaulter and assigned as 0. The remaining categories are dropped as they are *current, late* and *ingrace period*, which possibly doesn't belong any of our two groups defaulter and non-defaulters. The merging is done as follows.



```
allyear_dropped['loan_status']=np.where(allyear_dropped['loan_status'] =='Fully Paid', 1, allyear_dropped['loan_status'])
allyear_dropped['loan_status']=np.where(allyear_dropped['loan_status'] =='Does not meet the credit policy. Status:Fully Paid',
                                        1, allyear_dropped['loan_status'])
allyear_dropped['loan_status']=np.where(allyear_dropped['loan_status'] =='Charged Off', 0, allyear_dropped['loan_status'])
allyear_dropped['loan_status']=np.where(allyear_dropped['loan_status'] =='Default', 0, allyear_dropped['loan_status'])
allyear_dropped['loan_status']=np.where(allyear_dropped['loan_status'] =='Does not meet the credit policy. Status:Charged Off'
                                        , 0, allyear_dropped['loan_status'])
allyear_dropped = allyear_dropped[allyear_dropped.loan_status != 'Current']
allyear_dropped = allyear_dropped[allyear_dropped.loan_status != 'Late (31-120 days)']
allyear_dropped = allyear_dropped[allyear_dropped.loan_status != 'Issued']
allyear_dropped = allyear_dropped[allyear_dropped.loan_status != 'In Grace Period']
allyear_dropped = allyear_dropped[allyear_dropped.loan_status != 'Late (16-30 days)']
```

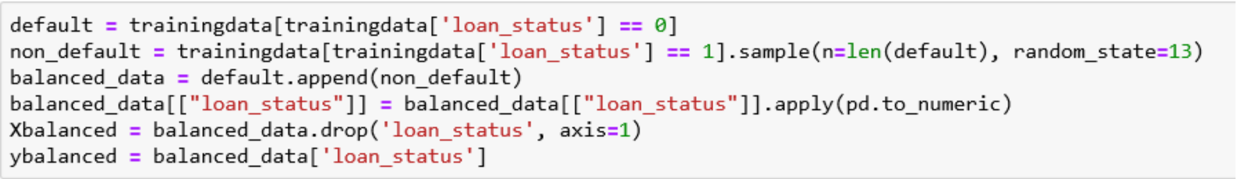
Modeling the credit score

Before building machine learning model, the whole data set have been divided in to training and test set as follows.



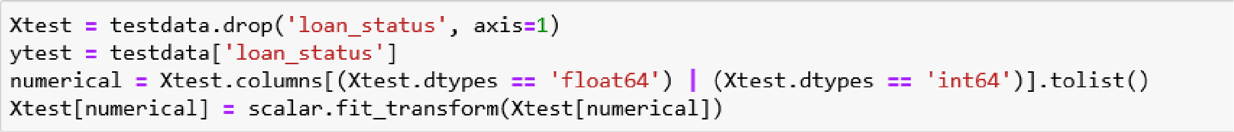
```
trainingdata, testdata = train_test_split(allyear_dropped, stratify=allyear_dropped['loan_status'],test_size=.25,
                                          random_state=13)
testdata.reset_index(drop=True, inplace=True)
trainingdata.reset_index(drop=True, inplace=True)
```

due the imbalance between defaulters and non-defaulters the balancing was needed. Balanced X (predictors) and balanced y (target variable loan status) is created from the training data set as follows.



```
default = trainingdata[trainingdata['loan_status'] == 0]
non_default = trainingdata[trainingdata['loan_status'] == 1].sample(n=len(default), random_state=13)
balanced_data = default.append(non_default)
balanced_data[["loan_status"]] = balanced_data[["loan_status"]].apply(pd.to_numeric)
Xbalanced = balanced_data.drop('loan_status', axis=1)
ybalanced = balanced_data['loan_status']
```

So, now we have balanced Xbalanced and ybalanced data set. As the test set data is not separated into predictors and target variable, the separation in to Xtest and ytest is done as follows.

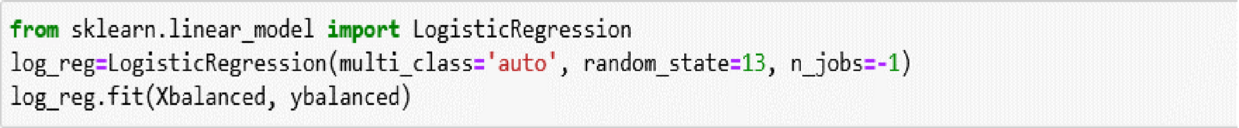


```
Xtest = testdata.drop('loan_status', axis=1)
ytest = testdata['loan_status']
numerical = Xtest.columns[(Xtest.dtypes == 'float64') | (Xtest.dtypes == 'int64')].tolist()
Xtest[numerical] = scalar.fit_transform(Xtest[numerical])
```

The test set is important for the evaluation of the model in the letter stage of the model.

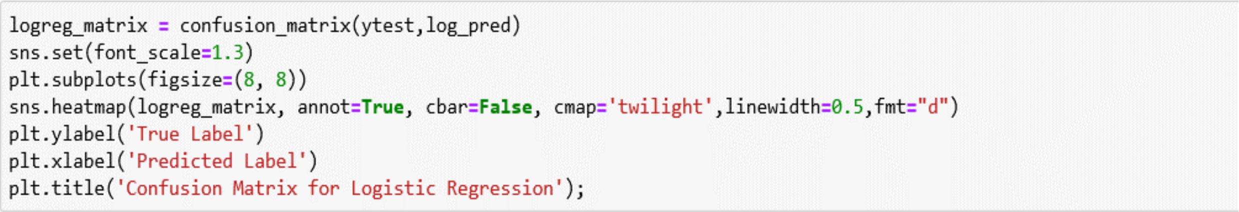
Logistic regression

The data is separated in to training and test set, then we identified the target and independent variable now it is the time to initiate python built in logistic regression called "*LogisticRegression*". Then we will use the "*fit*" function to train our training data (Xbalanced, ybalanced) as follows.



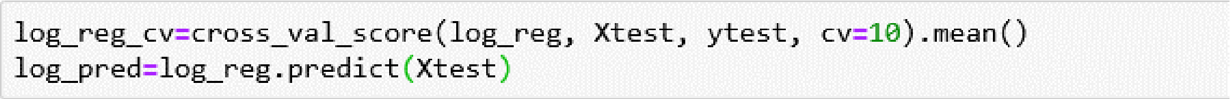
```
from sklearn.linear_model import LogisticRegression
log_reg=LogisticRegression(multi_class='auto', random_state=13, n_jobs=-1)
log_reg.fit(Xbalanced, ybalanced)
```

The performance of trained logistic regression model on the test set was checked by confusion matrix function as follows.



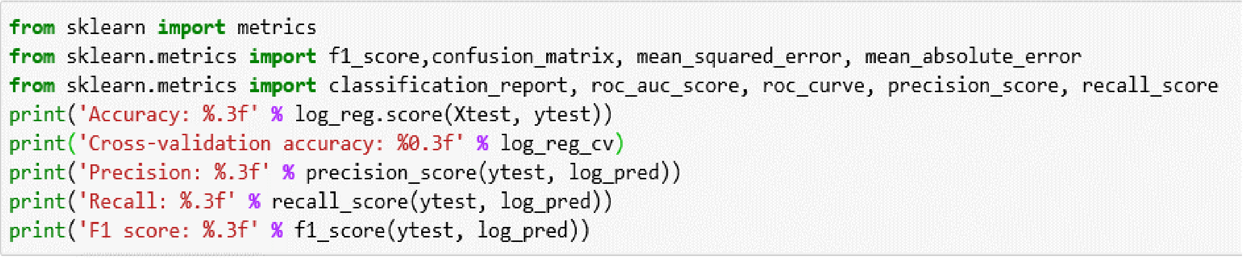
```
logreg_matrix = confusion_matrix(ytest,log_pred)
sns.set(font_scale=1.3)
plt.subplots(figsize=(8, 8))
sns.heatmap(logreg_matrix, annot=True, cbar=False, cmap='twilight',linewidth=0.5,fmt="d")
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.title('Confusion Matrix for Logistic Regression');
```

The accuracy of the model can be also cross checked by classifying the test set in to 10 different independent folds as follows. This can be done easily by importing python built in function called "*cross_val_score*" and fitting the X and ytest set into it. Therefore, the quality of the model will be checked by 10 separate test set and the average of cross validation accuracy will be displayed.



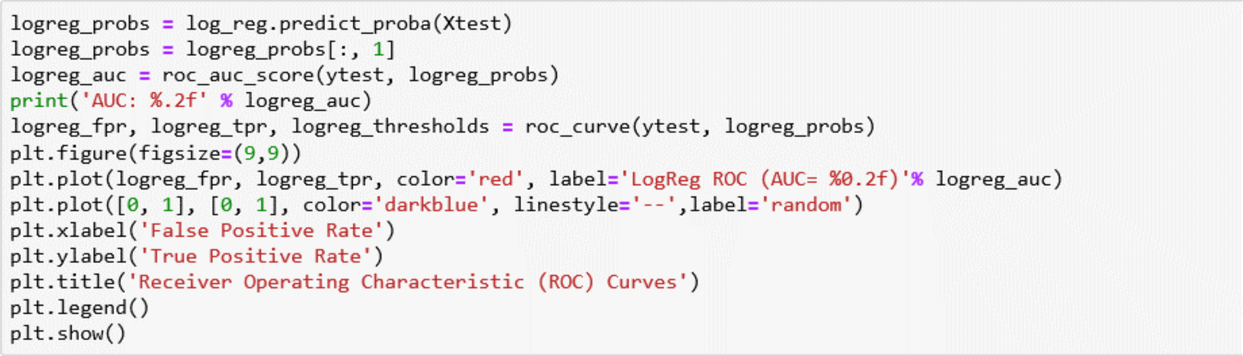
```
log_reg_cv=cross_val_score(log_reg, Xtest, ytest, cv=10).mean()
log_pred=log_reg.predict(Xtest)
```

Now let us print all accuracy checking criteria or metrics like cross validation accuracy, general accuracy, precision, recall and F1 score. The function is presented below.



```
from sklearn import metrics
from sklearn.metrics import f1_score,confusion_matrix, mean_squared_error, mean_absolute_error
from sklearn.metrics import classification_report, roc_auc_score, roc_curve, precision_score, recall_score
print('Accuracy: %.3f' % log_reg.score(Xtest, ytest))
print('Cross-validation accuracy: %0.3f' % log_reg_cv)
print('Precision: %.3f' % precision_score(ytest, log_pred))
print('Recall: %.3f' % recall_score(ytest, log_pred))
print('F1 score: %.3f' % f1_score(ytest, log_pred))
```

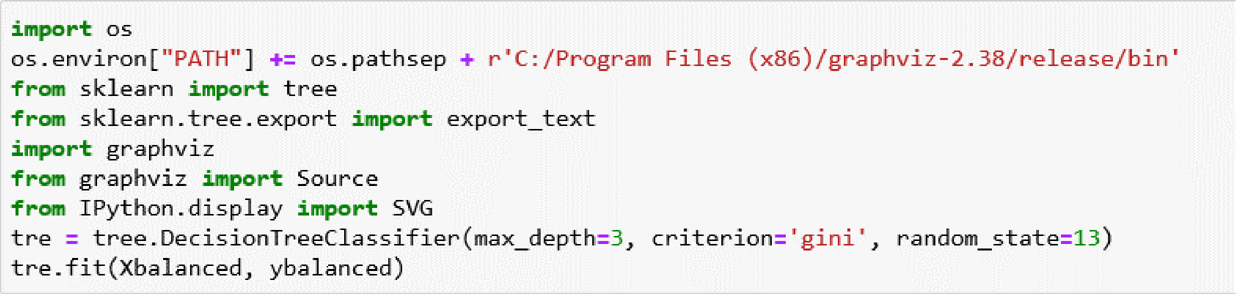
One classification model quality criterion the receiver Operator Characteristic (ROC) curve is built by using "*roc_curve()*" python built in scikit-learn function and the area under curve (AUC) is calculated by "*roc_auc_score()*" function as follows.



```
logreg_probs = log_reg.predict_proba(Xtest)
logreg_probs = logreg_probs[:, 1]
logreg_auc = roc_auc_score(ytest, logreg_probs)
print('AUC: %.2f' % logreg_auc)
logreg_fpr, logreg_tpr, logreg_thresholds = roc_curve(ytest, logreg_probs)
plt.figure(figsize=(9,9))
plt.plot(logreg_fpr, logreg_tpr, color='red', label='LogReg ROC (AUC= %0.2f)'% logreg_auc)
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--',label='random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curves')
plt.legend()
plt.show()
```

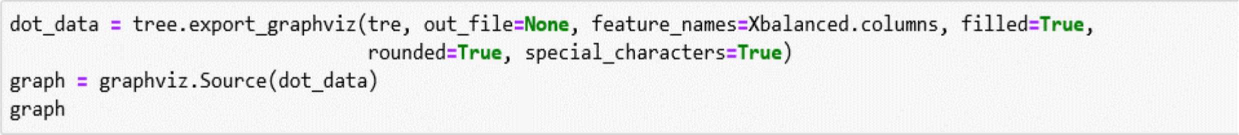
Decision tree

Most of the steps and procedure to be done in the case of decision tree are similar with that of logistic regression, so here the functions will be presented with some footnote only. The decision tree function is imported, initiated and the model is trained by the training data set as follows.



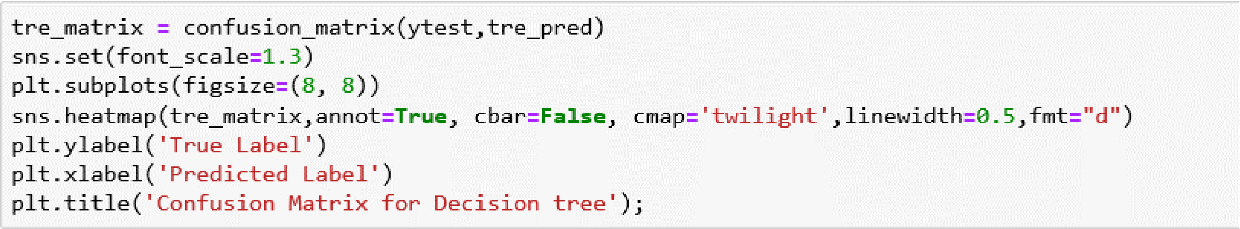
```
import os
os.environ["PATH"] += os.pathsep + r'C:/Program Files (x86)/graphviz-2.38/release/bin'
from sklearn import tree
from sklearn.tree.export import export_text
import graphviz
from graphviz import Source
from IPython.display import SVG
tre = tree.DecisionTreeClassifier(max_depth=3, criterion='gini', random_state=13)
tre.fit(Xbalanced, ybalanced)
```

The pruned version of the decision tree is plotted by the following function.



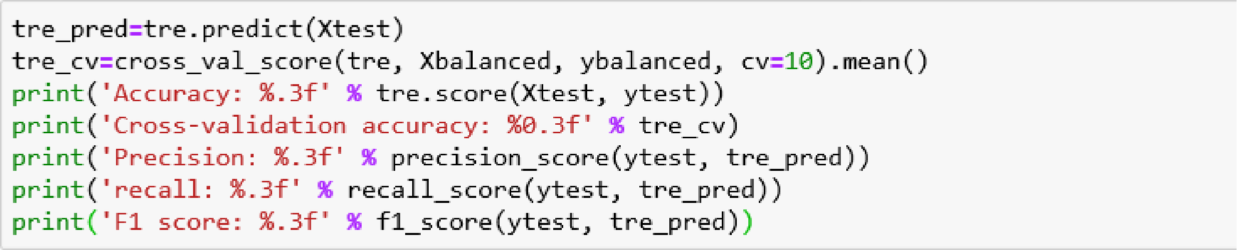
```
dot_data = tree.export_graphviz(tre, out_file=None, feature_names=Xbalanced.columns, filled=True,
                                rounded=True, special_characters=True)
graph = graphviz.Source(dot_data)
graph
```

Displaying the confusion matrix.



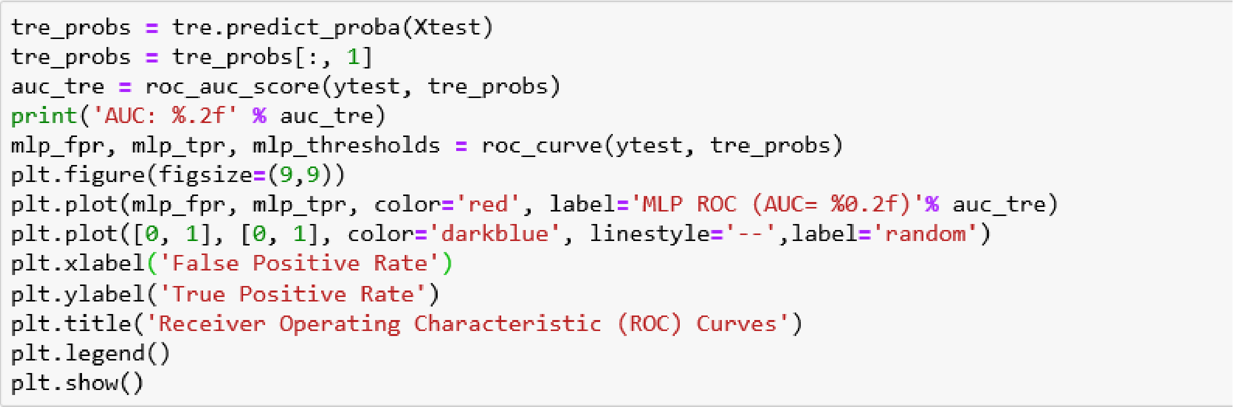
```
tre_matrix = confusion_matrix(ytest,tre_pred)
sns.set(font_scale=1.3)
plt.subplots(figsize=(8, 8))
sns.heatmap(tre_matrix,annot=True, cbar=False, cmap='twilight',linewidth=0.5,fmt="d")
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.title('Confusion Matrix for Decision tree');
```

Printing the model quality checking criterion or metrics (cross validation accuracy, general accuracy, precision, recall and F1 score).



```
tre_pred=tre.predict(Xtest)
tre_cv=cross_val_score(tre, Xbalanced, ybalanced, cv=10).mean()
print('Accuracy: %.3f' % tre.score(Xtest, ytest))
print('Cross-validation accuracy: %0.3f' % tre_cv)
print('Precision: %.3f' % precision_score(ytest, tre_pred))
print('recall: %.3f' % recall_score(ytest, tre_pred))
print('F1 score: %.3f' % f1_score(ytest, tre_pred))
```

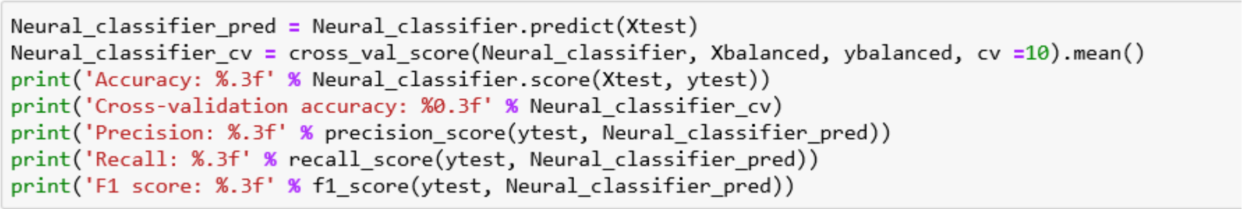
Plotting the ROC curve and AUC is done as follows.



```
tre_probs = tre.predict_proba(Xtest)
tre_probs = tre_probs[:, 1]
auc_tre = roc_auc_score(ytest, tre_probs)
print('AUC: %.2f' % auc_tre)
mlp_fpr, mlp_tpr, mlp_thresholds = roc_curve(ytest, tre_probs)
plt.figure(figsize=(9,9))
plt.plot(mlp_fpr, mlp_tpr, color='red', label='MLP ROC (AUC= %0.2f)'% auc_tre)
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--',label='random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curves')
plt.legend()
plt.show()
```

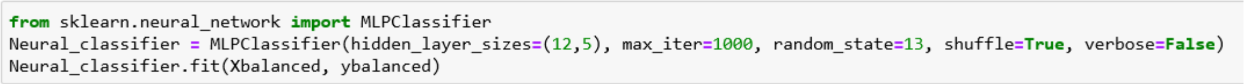
Neural network

Initiating neural network and determining size of the hidden layers.



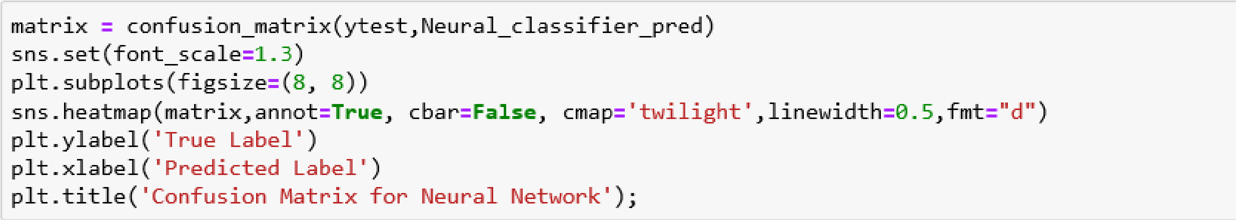
```
Neural_classifier_pred = Neural_classifier.predict(Xtest)
Neural_classifier_cv = cross_val_score(Neural_classifier, Xbalanced, ybalanced, cv =10).mean()
print('Accuracy: %.3f' % Neural_classifier.score(Xtest, ytest))
print('Cross-validation accuracy: %0.3f' % Neural_classifier_cv)
print('Precision: %.3f' % precision_score(ytest, Neural_classifier_pred))
print('Recall: %.3f' % recall_score(ytest, Neural_classifier_pred))
print('F1 score: %.3f' % f1_score(ytest, Neural_classifier_pred))
```

Printing model quality check criterion or metrics (cross validation accuracy, general accuracy, precision, recall and F1 score)



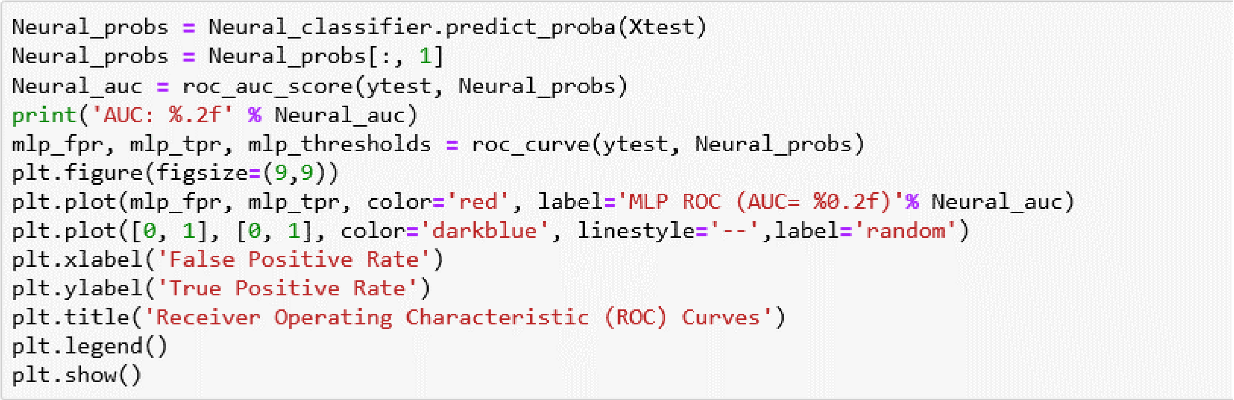
```
from sklearn.neural_network import MLPClassifier
Neural_classifier = MLPClassifier(hidden_layer_sizes=(12,5), max_iter=1000, random_state=13, shuffle=True, verbose=False)
Neural_classifier.fit(Xbalanced, ybalanced)
```

Plotting the confusion matrix



```
matrix = confusion_matrix(ytest,Neural_classifier_pred)
sns.set(font_scale=1.3)
plt.subplots(figsize=(8, 8))
sns.heatmap(matrix,annot=True, cbar=False, cmap='twilight',linewidth=0.5,fmt="d")
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.title('Confusion Matrix for Neural Network');
```

Plotting the ROC curve and calculating the AUC is done as follows.



```
Neural_probs = Neural_classifier.predict_proba(Xtest)
Neural_probs = Neural_probs[:, 1]
Neural_auc = roc_auc_score(ytest, Neural_probs)
print('AUC: %.2f' % Neural_auc)
mlp_fpr, mlp_tpr, mlp_thresholds = roc_curve(ytest, Neural_probs)
plt.figure(figsize=(9,9))
plt.plot(mlp_fpr, mlp_tpr, color='red', label='MLP ROC (AUC= %0.2f)'% Neural_auc)
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--',label='random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curves')
plt.legend()
plt.show()
```

Model performance measurements

To measure performance of the three models applied in this study the confusion matrix has been used. Below Table 1 detailed description of confusion matrix for our case is presented.

Labeled/Predicted	Defaulted (0) (Negative)	Non-defaulted (1) (Positive)
Defaulted (0) (Negative)	True Negative	False positive
Non-defaulted (1) (Positive)	False Negative	True Positive

Table 1. The detailed description of confusion matrix.

Then from the confusion matrix the following model performance measurement have been calculated. Those are the accuracy, Precision, Recall and the F1.

Accuracy: refers to the overall power of the model to predict accurately. Accuracy can be calculated by the following formula directly from the confusion matrix.

$$\frac{(\text{True Positives} + \text{True Negatives})}{(\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives})} \qquad \text{eq 2.}$$

Based on the result from python the Accuracy of our logistic regression model is 0.75. Meaning that this model is 75% accurate.

Cross validation accuracy is calculated after grouping the test set of data in to 10 separate folds and evaluating the accuracy of the built model then taking average score.

Precision: is the number of correctly predicted true positives out of all predicted positives. The formula is presented below.

$$\frac{(\text{True Positives})}{(\text{True Positives} + \text{False Positives})} \qquad \text{eq 3.}$$

Recall: is also termed as Sensitivity or true positive rate and it tell the rate of positive values correctly predicted. It can be calculated by the following formula.

$$\frac{(\text{True Positives})}{(\text{True Positives} + \text{False Negative})} \qquad \text{eq 4.}$$

F1 score: is generally the harmonic mean result of precision and recall. This measure is also always between 0 and 1. Signifying better model as the value approached to 1. The formula for the recall is presented below.

$$2 * \left(\frac{(\text{precission}*\text{recall})}{(\text{precision}+\text{recall})}\right) \qquad \text{eq 5.}$$

Receiver Operator Characteristic (ROC):

The ROC measures the accuracy of the model based on the area under curve (AUC). The curve for ROC is built between the true positive rate false positive rate of the model. As the curve hangs to 1 it tells the model is better in quality. The higher the area under the curve giving us high quality model.

4. Literature Review

4.1.Machine Learning

Machine learning is the mechanism by which we extract information from data. It could be also named as a predictive analytics and or statistical learning and it comprises statistics, artificial intelligence and computer science (Müller and Guido, 2017). The performance could be optimized in respect to the past data available. For the mathematical model building the theory of statistics will be used (Igual and Seguí, 2017). Generally, machine learning can be used for data mining, robotics, pattern recognition, vision processing, language processing, game development, expert system and forecasting. The algorithm for machine earning can be grouped mainly in to three groups; supervised machine learning, unsupervised machine learning and reinforcement machine learning. In all three cases the procedure incorporates problem defining, data preparation, choosing a model, evaluation of algorithms and prediction.

4.1.1. Supervised Machine Learning

Supervised machine learning is the prediction of new data set based on the learned algorithm, which is gained from the large data set with answers (Rebala et al., 2019). Supervised machine learning algorithms build a model based on a data set having the output. The data with the answers is called training data having a set of data with both inputs and outputs. Therefore, the model built can be used to predict new datasets without any output. Main principles are presented (Figure 1). For supervised machine learning the datasets must contain descriptions, class label (labelled), target (the desired output). Therefore the algorithm learns the relationship, which exist between the input and output of the training dataset and uses its experience or knowledge to predict output of new dataset (Sarkar et al., 2017). In broad sense the problems of supervised machine learning can fall in to two real-world problem-solving categories: classification and regression problems. Classification aims on sorting each data sets to a distinct class it belongs. For example, predicting male/female, positive/negative, benign/malignant and bad/good credit scoring and classifying an object as table, chair, car or dog can be considered as classification problems. Classification is used when there is a restricted and very limited set of outputs rather than a continuous output (Rebala et al., 2019). Regression type of problems are problems with a continuous numeric output within certain range. Regression is in one way or another the ability of predicting values of continuous variable. Examples can be predicting price of houses with defined specification in ten years, predicting price of specific commodity in 10 years etc. (Rebala et al., 2019). Examples of supervised machine learning algorithm includes: Logistic Regression, Neural Networks, Support Vector Machines (SVM) and Naïve Bayes classifiers.

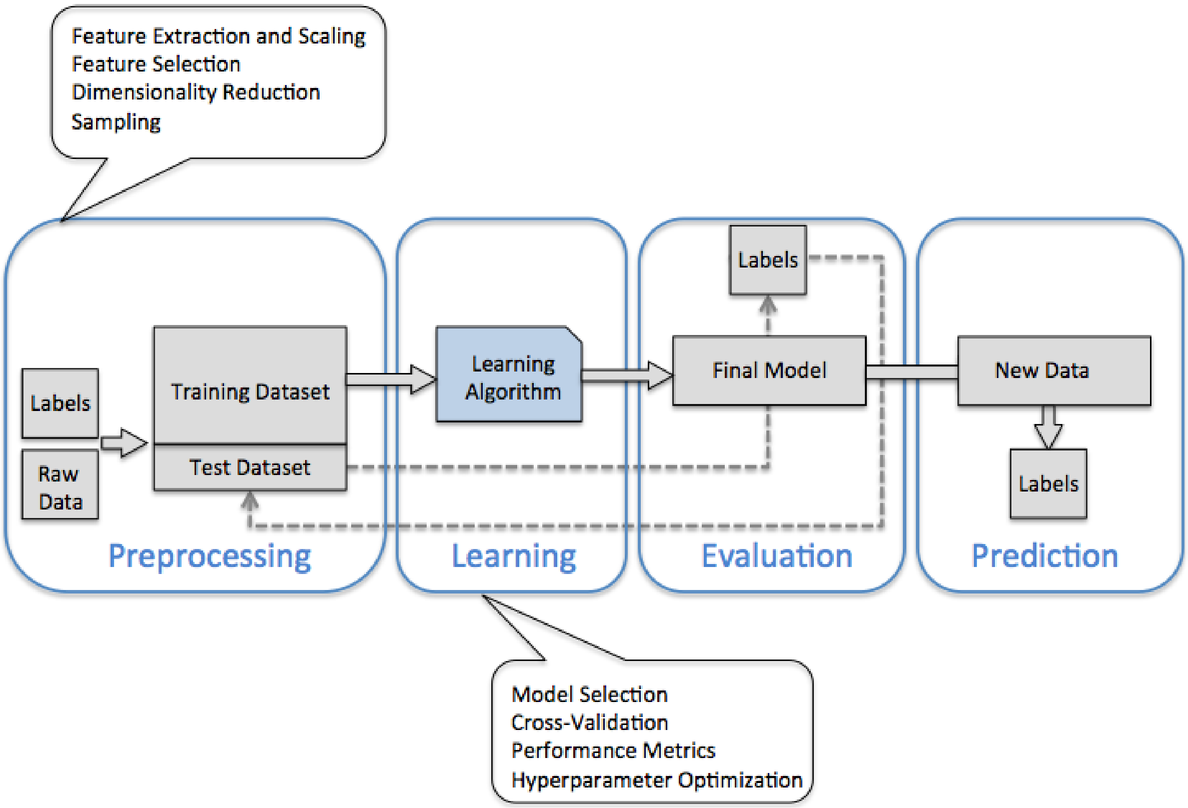


Figure 1. The main working principle of supervised machine learning (Dangeti, 2017)

4.1.2. Unsupervised Machine Learning

In the case of unsupervised machine learning there is no target variable. Therefore, the machine will be provided with a set of data without any outputs and the machine learns to find some similarity between the data points or finds any trends and then allocate them to certain group or cluster. After grouping, individual groups or clusters could be labelled according to specific group they belongs (Abbott, 2014). The major problems to solve with unsupervised machine learning are the dimension reduction problem and also clustering problems (Dangeti, 2017). The following machine learning tasks are mainly categorized as unsupervised machine learning.

- Clustering
- Dimensionality reduction
- Anomaly detection
- Association rule mining (Sarkar et al., 2017)

4.1.3. Reinforcement Learning

As we can understand from the name reinforcement learning enquires some source of reinforcement or reward to achieve the classification. The machine (agent) will be given some events as a sort of data sets without any supervision and rewarded as +1 or -1. The overall process is presented below (Figure 2). Therefore based on the final reward as +1 or -1 the agent or algorithm determine the paths (Dangeti, 2017).

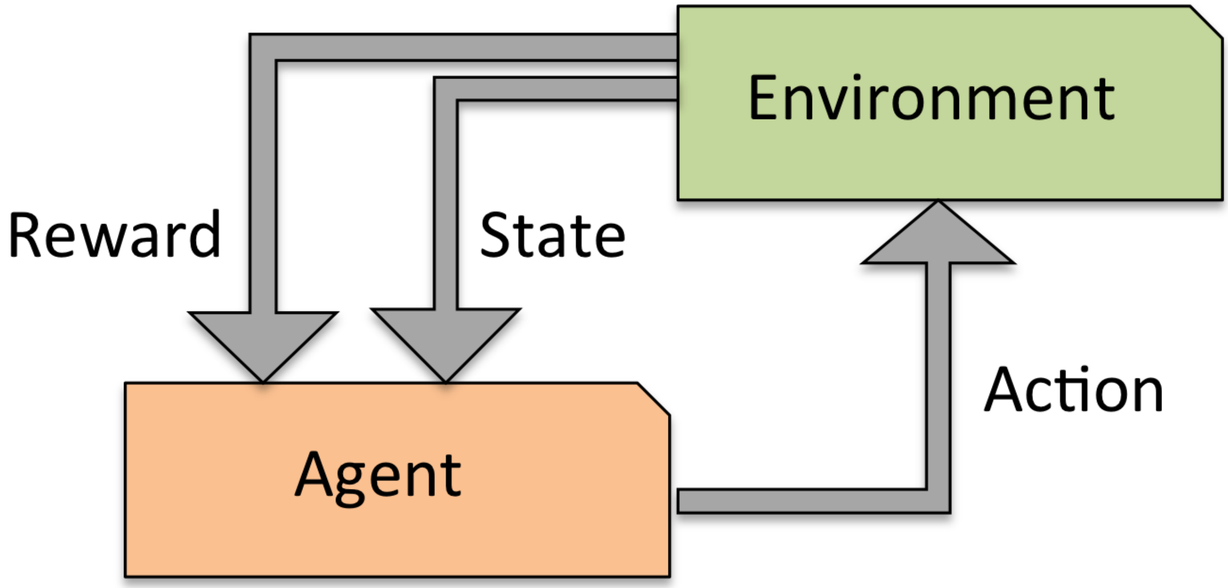


Figure 2. The typical process implemented in reinforcement learning.

More specifically reinforcement learning comprises the following steps.

- Preparing the agent with a set of initial strategies and policies
- Observing the environment and current status
- Selecting optimal policy and perform action
- Get reward or penalty per specific action
- Update policy based on the reward and penalty if needed
- Repeating the above steps until the agent learn the most optimal policies or strategies (Sarkar et al., 2017).

4.2.Credit Scoring

4.2.1. Introduction to credit scoring

Credit scoring is the term, which uses to describe the credit worthiness of a person or the term telling the ability of a person to repay his credit. It is a very important practice in managing risk of any financial institution (Maldonado et al., 2020). The decision of credit defaulters can be made by the system called credit scoring models or simple judgmental techniques. The judgmental technique is mainly based on the 3C's, 4C's or 5C's, which are character, capital, collateral, capacity and condition (Yap et al., 2011). Credit scoring help the lenders to decide for who should they give credit, amount of credit to be given and generally provide information, which could possibly boost their profitability (Thomas et al., 2017). Know a day, credit scoring models almost totally replaced judgmental methods in the assessment of applicants. This is mainly due to the increased number of customers looking for credit and the advancement of computer technology, which allowed the implementation of sophisticated credit scoring models (Henley, 1994). In U. S. Fair Isaac Corporation have developed formula to estimate credit score of a person. Principally peoples argue that this secret algorithm could be generally the ratio of debt and available credit, which could be then adjusted with the payment history, reputation of credit application and negative events like bankruptcy of the specific person (Arya et al., 2013). The FICO model, on the contrary side of its easiness and simplicity to apply by lending institution, it could lead to the decision that is under inclusive and disadvantage borrowers that do not have awareness about the credit system before. Lenders my make two types of decision to lend for specific person based on specific type of borrowers; weather to give credit to new customer or applicant, which we call this technique as credit scoring and to increase credit of an existing customer, this technique being called behavioral scoring (Thomas et al., 2017). Model of credit scoring uses the relationship of person being bad or good to be given credit with some predictor variables which are in one way or another related to features of sample applicants, whose credit worthiness is already identified (Henley, 1994). As the main objective of a lender is to get benefit from borrowers, the high rate of avoiding person with bad credit scoring may reduce benefit. This is because the person who is regarded with bad credit scoring may be potentially profitable to the lenders (Henley, 1994). There may be a plenty of character for single credit score card/credit scoring. However, the characters must be selected according to its logical sense, power of prediction, low correlation with other selected characters, stable and available for use, relate to customers and its ignorance may result in an acceptable information loss (Anderson, 2007).

4.2.2. Scorecard development process

Preliminaries and planning

Development of good scorecard needs proper planning and analytical work before entering to the main credit scoring task. This step of credit scorecard development includes the identification and prioritization of organizational objective to the specific scorecard development project. By doing this the organization could be able to focus on increasing revenue or decreasing loos, which could possibly arise after giving loan to the client. Other main organizational objective may include the following:

- Reducing bad debt
- Increasing approval rate or market share, where loan is secured
- Increasing profitability
- Increasing operation efficiency
- Cost saving and creating efficient predictive power

Identifying project risk

For effective credit card development, the following possible risks should be identified:

- No availability of data or insufficient data

- Poor quality of data (dirty or unreliable)
- Delays/difficulties in accessing data
- Nonpredictive or weak data
- Scorecard characteristics or derivations that cannot be handled by operational systems
- Changes in organizational direction/priorities
- Possible implementation delays
- Other legal or operational issues

The above possible risks should be identified, and possible backup should be formulated in advance.

Data review and project parameters

This stage is important to determine whether the scorecard development is feasible and to set high level of performance. The availability of reliable and clean data with the minimum acceptable number of "good" and "bad" is needed for the development of scorecard. Even if, there is no exact number needed for the development but usually, as a rule of thumb there should be approximately 2,000 "good" and 200 "bad" customer accounts that can be selected.

4.3.Possible Methods of Credit scoring

Scoring models could be built by using a variety of statistical methods based on statistical pattern recognition. Those techniques may be in the range between very advanced statistical methods and traditional statistical methods. The use of sophisticated statistical techniques like neural network is very advantageous in that they can model complex functions.

4.3.1. Fisher's Linear Discriminant Analysis (LDA)

Discriminant analysis was one of the major methods for classification before Fisher's basic work in 1936 for credit scoring. The main principle in discriminant function analysis is determining the variables, which have good discrimination power of two or more naturally occurring groups (Tufféry, 2011).

LDA in machine learning uses a classification method try to point out a linearly combined features, which can classify two or more groups. Then the identified features will be used for dimension reduction before applying them using them as a linear classifier (Bhatia et al., 2017).

After one or more linear functions will be constructed by including explanatory variables. The general model can be expressed as follows,

$$Z = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_P X_p, \qquad \text{eq 6.}$$

Where Z = discrimination score, α = intercept, β_i = coefficient responsible for the linear contribution of i^{th} explanatory variable Xi, where I = 1, 2, …, p.

Discriminant analysis have assumptions which are almost similar to analysis of variance (ANOVA). It mainly assumes the normal distribution of the data, homogeneity of variance or covariances of variables, no correlation between means and variances and the variables used to do discriminate the group should not be repeated.

4.3.2. k-nearest Neighbor (k-NN)

The principle for k-nearest Neighbor Classifier is simply by assuming; the object with similarity are situated in proximity each other than unrelated one. Therefore it classify the new object (with input vector y) by examining the k nearest training data set pints to the entry data point y and assign y to the class where these majority of k belongs (Hand et al., 2001). In credit scoring k-NN estimate the probability of good and bad among the k most similar points in the training samples. Principally the algorithm of k-NN first determine the k and distance metrics, then it will find the k nearest neighbors of the sample of interest for classification and the label will be assigned by the rule of majority vote. For example, figure 3 clearly demonstrate how our sample data point “ ? ” is assigned to the triangle class. It is assigned to the triangle because most of its neighbor 3 of them are from the triangle groups, while it have only one neighbor from “ – “ and also one from “ + ” group (Raschka, 2015).

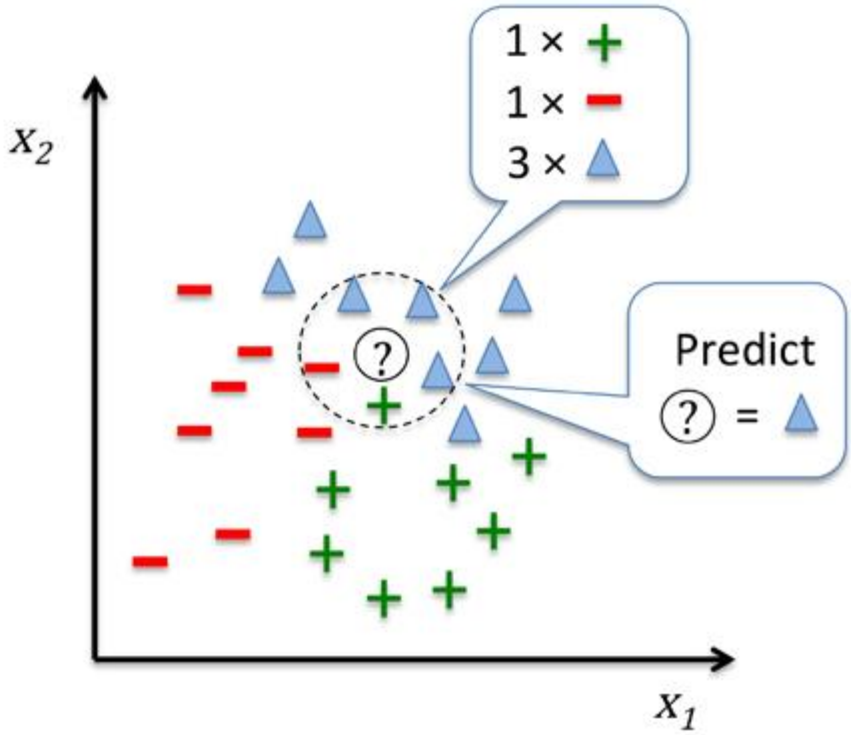


Figure 3. The principle of k-NN (Raschka, 2015).

4.3.3. Decision trees

Refers a set of decision with a tree shaped structure (Figure 4). Then each decision generates rule for classification. The most common methods of decision tree include classification and regression trees (CART) producing binary split and the chi square automatic interaction detection (CHAID), which able to produce a multiple branches of a single root (Berry and Linoff, 2004). The use of decision tree algorithm has the following main advantages:

- The models developed are easy to understand and by non-experts also can be easily visualize.

- No need of preprocessing like normalization and standardization of feature is needed as each feature is processed separately (Müller and Guido, 2017).
- Any extreme values will be separated in small nodes and will not affect the classification problem (Tufféry, 2011).

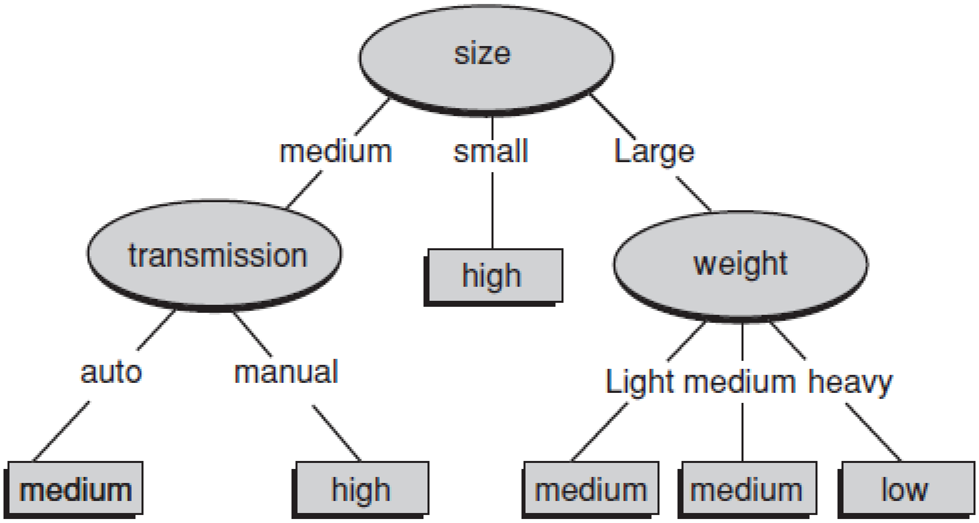


Figure 4. Example of simple decision tree (Sumathi and Sivanandam, 2006).

The primary techniques used by machine learning for decision tree is the robotic process automation (RPA) and it includes the following procedures (Anderson, 2007).

- Binning, where the methods for predictors binning will be determined,
- Splitting, where the character to be used will be selected,
- Stopping, where the stopping of new sub nodes will be stopped,
- Pruning, where the nodes with possible overfitting will be dropped,
- Assignment, where each node will be classified as bad and good.

Therefore, the algorithms decision making, which variable is the most important and automatically sort out the target variable in a category (Yap et al., 2011).

4.3.4. Support vector machine

Support vector machine (SVM) is one of well-known and best machine learning approach for classification problems, where it is applied in a wide area including handwritten digit recognition, face detection, text categorization and credit rating. The application of SVM machine in a dependent variable with two possible output 0 or 1, the classification will be performed by surface or plane in the space separating the attribute 0 from attribute 1 (Figure 5).

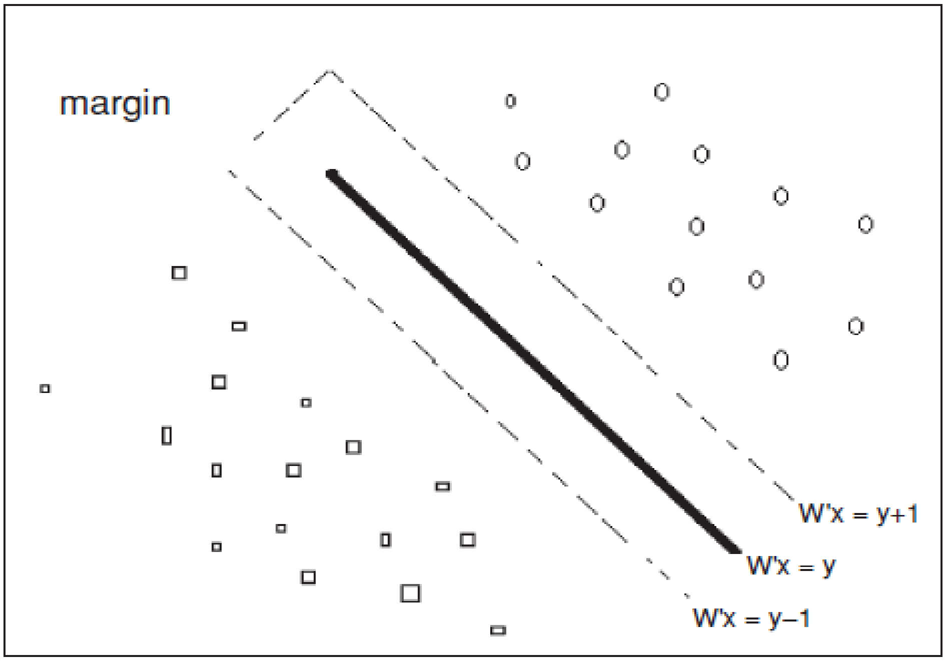


Figure 5. Example of SVM having dependent attribute 0 represented by ○'s and 1 represented by □'s. (Sumathi and Sivanandam, 2006).

SVM uses three main principles for classification (Dangeti, 2017). These are

- Maximum margin classifiers,
- Support vector classifiers and
- Support vector machines.

Maximum margin classifier enables to select the ideal hyperplane with the maximum margin of classification width among the set of infinite hyperplanes. Support vector classifiers are just the extension of maximum margin classifiers in which some violation will be considered in the case of non-separable data set to find the best fit even by tolerating some errors within the threshold. This type of cases is most likely to happen in real life as purely separable classes are rare. Support vector machines are used for data set, where Maximum margin classifier and Support vector classifiers are not applicable, meaning that in the case of linearly inseparable cases (Dangeti, 2017).

4.3.5. Neural Networks (NN)

In neural network the replica of human brain is used to determine the relationship between the input and the output signals. This method uses the artificial neurons, which are interconnected each other to solve machine learning problems like credit. The neuron computes a function by imputing each neuron having specific input with a given amount of weight. Then an activation functions like sigmoid, tanh, restified linear unit (ReLU) will be applied on the linear combination weighted inputs on the aggregated sum as shown on Figure 6 (Dangeti, 2017).

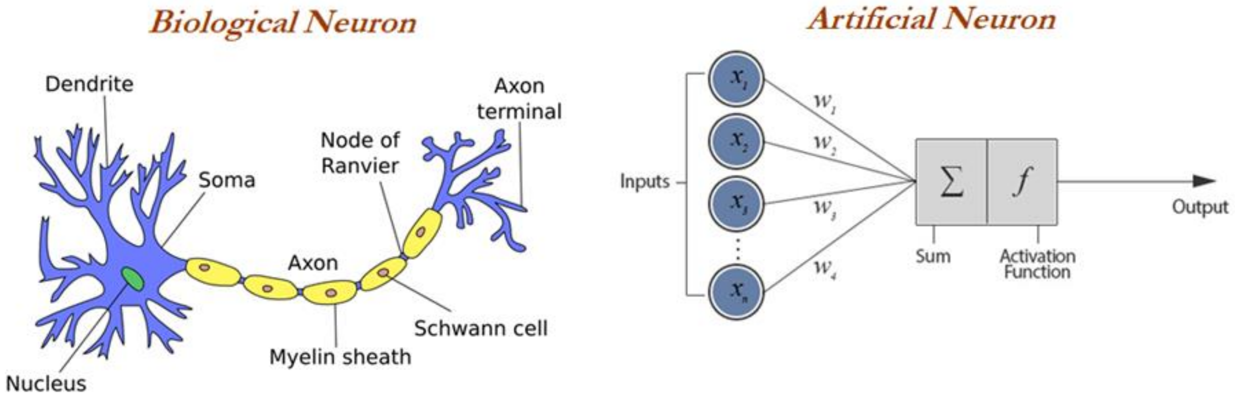


Figure 6. Biological Neuron vs Artificial Neuron (Dangeti, 2017).

The activation functions are simply the principle by which the neurons process the inputs and transform them. Sigmoid function takes any real number and change to value between 0 and 1 by using $\sigma(x) = 1 / (1+e^{-x})$, while Tanh function change the input real numbers to the range between -1 and +1 by using $\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$.

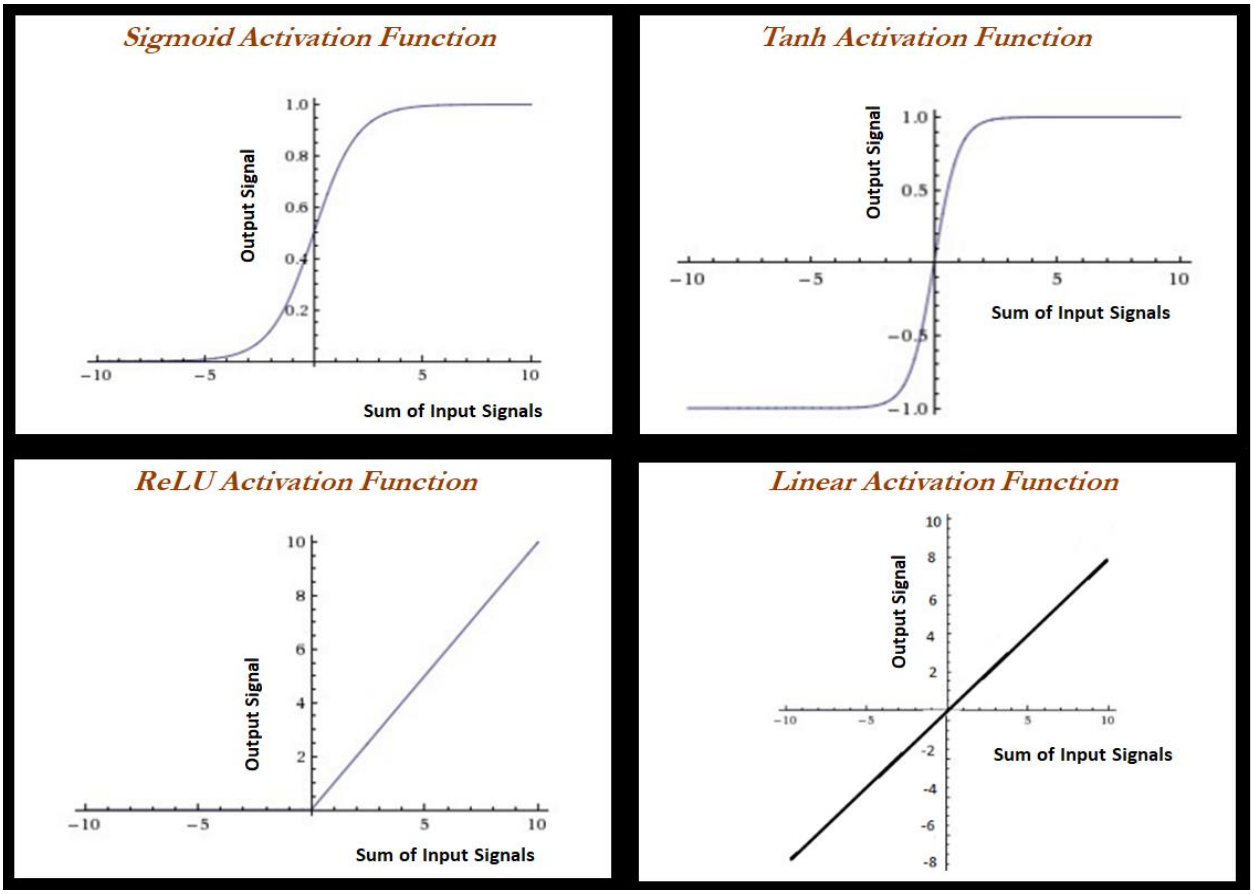


Figure 7. The diagrammatic illustration of activation functions used in ANN.

In the cause of ReLU uses the function f(x) = max (0, x) and simply use the threshold at zero. Additionally, linear function f(x) = x could be also used as an activation function. Generally, the activation functions used in ANN is presented (Figure 7). The prediction or classification using ANN could be implemented first by identification of the input/output data, normalizing data, establishing the network with the suitable structure, learning the machine, testing the algorithm and applying the model generated by the machine (Tufféry, 2011). The use of neural network is advantageous for the following cases:

- It can process a large sate of data, capture information and make a complex model.

- Very efficient than other machine learning algorithms for tasks requiring classification and regression (Müller and Guido, 2017).

4.3.6. **Logistic regression**

Generally logistic regression can be assumed unique due to its ability to perform the following three different purposes (Hilbe, 2016):

1. For predicting the probability of response variable equals 1
2. For categorization of outcomes or predictions
3. For accessing the add or risk, which is associated to the models of predictors

The ability of logistic regression to perform these three different tasks and its applicability in the binary variables make it from other types of regression. The model assumes:

- No correlation of predictor variables
- Significant correlation of predictors to the response variable
- No correlation of the observation or data elements of the model

Logistic regression is a non-parametric technique, which classify the probability of attribute distribution. In the application of logistic regression the log of probability of odds will be matched with linear combination of selected characteristic variables (Thomas et al., 2017).

Logistic regression applies a logistic transformation to the input and restrict the output ranging from - ∞ to + ∞ to the probability between 0 and 1. Therefore as there are only two outputs in the case of credit scoring (good and bad), logistic regression will be greatly beneficial. A binary regression model is presented.

$$g(x) = \ln\left(\frac{p_g}{1-p_g}\right) = b_0 + b_1x_1 + b_2x_2 + \ldots + b_nx_n \qquad \text{eq 7.}$$

Where:

$g(x)$ = The logit transforms of p_g

p_g = The probability of being in good class

$\frac{p_g}{1-p_g}$ = The odds ratio

Being different from linear regression here we cannot use ordinary least square to estimate our coefficients, rather here we use the coefficient of regression by estimating with maximum likelihood estimate method (MLE). Then the probability to be in class membership of good (p_g) will be estimated as follows by using our input features.

$$p_g = \frac{\exp(b_0+b_1x_1+\cdots+b_nx_n)}{1+\exp(b_0+ b_1x_1+\cdots b_nx_n)} \qquad \text{eq 8.}$$

Then the probability to be in class membership of bad (p_g) will be estimated as follows

$$p_b = \frac{1}{1+\exp(b_0+ b_1x_1+\cdots b_nx_n)} \qquad \text{eq 9.}$$

In the case, where p_g= p_b, then the specific instance could have equal probability of both classes.

Generally, the instance will be defined to be member of $\mathrm{p_g}$, when it satisfies the following condition

$$\mathrm{b_0} + \mathrm{b_1 x_1} + \cdots \mathrm{b_n x_n} > 0 \qquad \text{eq 10.}$$

or it will be defined to be member of $\mathrm{p_b}$, when it satisfies

$$\mathrm{b_0} + \mathrm{b_1 x_1} + \cdots \mathrm{b_n x_n} < 0 \qquad \text{eq 11.}$$

Thus, in credit scoring the major aim of logistic regression is determining the probability of specific applicant or customer to be defaulter or non-defaulter (Yap et al., 2011). Due to its nature of robustness and transparency, the logistic regression is the most commonly used technique used for credit scoring in the banking industry (Dong et al., 2010). Assessment of the statistical significance for all coefficients is done by Wald test. The calculation is based on z statistic. The value of z will be squared to give a Wald statistic with chi-square distribution at p +1 of degree of freedom, where p is parameters.

$$z = \frac{\mathrm{b}_n}{\mathrm{SE}\ (\mathrm{b}_n)} \qquad \text{eq 12.}$$

Where SE is standard error.

4.4.Data preparation and preprocessing

The preparation of data is the crucial stage in both descriptive statistics and in predictive data modeling. Specially in the predictive modeling, the data preparation stage is believed to be the most time taking step taking from 60 to 90 % of the predictive modeling. The preprocessing stage of data mining plays the greatest role in ensuring better analytical output (Lup Low et al., 2001). This is mainly due to the principle of " garbage in, garbage out" (Lee et al., 1999). The Cross Industry Standard Process for Data Mining (CRISP-DM) is a critical way of data mining methodology. Therefore, based on CRISP-DM the business requirement should be understood, then the data should be understood, data prepared, possible method of modelling implemented, the model developed should be evaluated and the model will be used for application (Chapman et al., 2000).

4.4.1. Understanding the data

The understanding of the data should be the first step in the predictive modeling, so that the analyst identifies imperfections, errors and problems related to the data that could affect the model. At this stage the analyst will have the first glance to know type of the data, number of records, number of variables, number of target variables, presence of missing values, presence of outliers, identify unexpected distributions and characteristics of the data (Abbott, 2014). Our data can be quantitative (numerical), which may be again continuous or discrete. The data can be also qualitative (categorical). Therefore the analyst have to identify with which type of data he is facing because all statistical and data mining methods my not accept all data types for further analysis and the transformation of data from one type to another may be needed (Tufféry, 2011). For the understanding of single variable, we could implement simple summary statistics by incorporating mean, standard deviation, skewness and kurtosis. The mean (μ) helps to understand the middle of the distribution or the most typical value in the distribution for the data with normal or uniform distribution (Abbott, 2014). This stage also should answer if the data is distributed as we expected and if the data is useful for the building of predictive model (Abbott, 2014). The normal distribution, which is mainly performed for a Fisher discriminant analysis or linear regression can be checked by test of normality to deal with extreme values. Normally distributed data should have a belly curve with Gaussian distribution. On the other side, the uniform distribution of data tell is the presence of a variable value within a fixed finite range incorporating the mean value in the center of the range (Tufféry, 2011).

4.4.2. Data cleaning

The cleaning of variables includes handling of incorrect of irrelevant values, finding duplicated and miscoded values. Therefore such variables, which need fixation should be precisely identified and action have to be taken accordingly (Abbott, 2014). External files, abbreviation standards and automated spell reader could be used for the identification and fixation of irrelevant values (Lup Low et al., 2001). The simplest and standard way of determining duplicate in database or any data source is sorting the data and checking identity of neighbors (Bitton and DeWitt, 1983). A method called sorted neighborhood method (SNM) is proposed for the identification of duplicate, which require the creating of keys, sorting the data and the merging steps (Hernández and Stolfo, 1995). This method is further improved as duplicate elimination SNM (DE - SNM), consisting both identification and elimination (Hernández, 1996).

4.4.3. Missing values

Missing values in data set includes data coded as null value or any empty cells. This type of data problem is the most common problem in data analytics. There are some issues, which make handling missing data more difficult. For example, missing value as empty cell could be just part of the data history and should be treated as a legitimate data point, some may be simply due to data entry error, others may be due the overwriting or corruption of database tables. Handling missing value is the most important and time consuming step in data analytics and can be treated accordingly as follows (Abbott, 2014). Missing values can be grouped in to three categories. The first group of missing values are Missing Completely at Random (MCAR), which the missed attribute values are independent of the parent attribute. The second group are Missing at Random (MAR), here the missing values are not depending on the source attribute itself but may depend on other attributes. The third one is Missing Not at Random (MNAT), this groups of missing values are values, which are dependent on the parent attribute itself (Yadav and Roychoudhury, 2018).

- Assigned by a constant number.
- Replacing by mean and median
- Replacing by random number from the own data distribution
- Or could be deleted according to the relevance (Abbott, 2014)

- Different statistical imputation mechanisms could be also used (Yadav and Roychoudhury, 2018)

4.4.4. Incorrect values or outliers

Incorrect values represent the value, which is coded incorrectly. This value includes unusual values occurring very infrequently in categorical data. In the case of continuous data type such incorrect values are most often regarded as outlier or spikes in the distribution, where a single value appears rarely (Abbott, 2014). Outliers are simply unusual extreme data values that may occur very far below or above the point of data we are expecting. The presence of outlier in the data set may affect the development of models and the equations to be developed (Berman, 2016). Outliers in the data set can be identified using data visualization tools by box plot and scatter plot or by Z-score or interquartile range (IQR) scores. Once they are identified by the above methods then we can handle the outliers as follows (Abbott, 2014).

- Removing outliers from the modeling data set. This approach can be used, when it is believed that the presence of the outlier causes a distortion of the model. This approach is very common in the case of linear regression, k-nearest neighbor, K-Means clustering, and Principal Component Analysis.
- Develop a separate model just for the outliers. It can be done by separating outliers and developing separate model. This approach is useful to overcome the problem of the first approach, which removes the outliers.
- Transforming the outliers. For this a simple min max transformation of the outliers could be applied, therefore, they will lie in the range of most data set.

- Binning the data. This approach could be applied for the outliers, which are very extreme for transformation. Binning could be done by transforming numeric outliers to categorical data type.
- Leaving outliers as they are without modification. This approach could be used only for algorithms, which can not be affected by outliers like decision trees.

4.4.5. Consistency on the data format

All the data in the given variable must be consistent across the rows in a single column. This type of error is most common, when a single data is sourced from multiple tables or data sources. As example date writing format difference and ZIP code assigning as string or integer from data source to data source can be mentioned as the most common source of data inconsistency (Abbott, 2014).

5. Practical part

5.1.Data source and description

The data, which is used in this diploma thesis was downloaded from LendingClub (https://www.lendingclub.com/). LendingClub is an American based company with a headquarter in San Francisco, California. Up to date the company have customers over 3, 000, 000 peoples. This make it a company borrowing over 50 billion dollars to date. The data set used here is a loan, which is borrowed by this company in the year from 2007 – 2015. Our full data totally contains around 887, 440 observations and 150 features. This data is available on LendingClub website on separate file. The number of loans given to customers each year is presented on figure 8. In this thesis work all separate files were concatenated together for the analysis by using python.

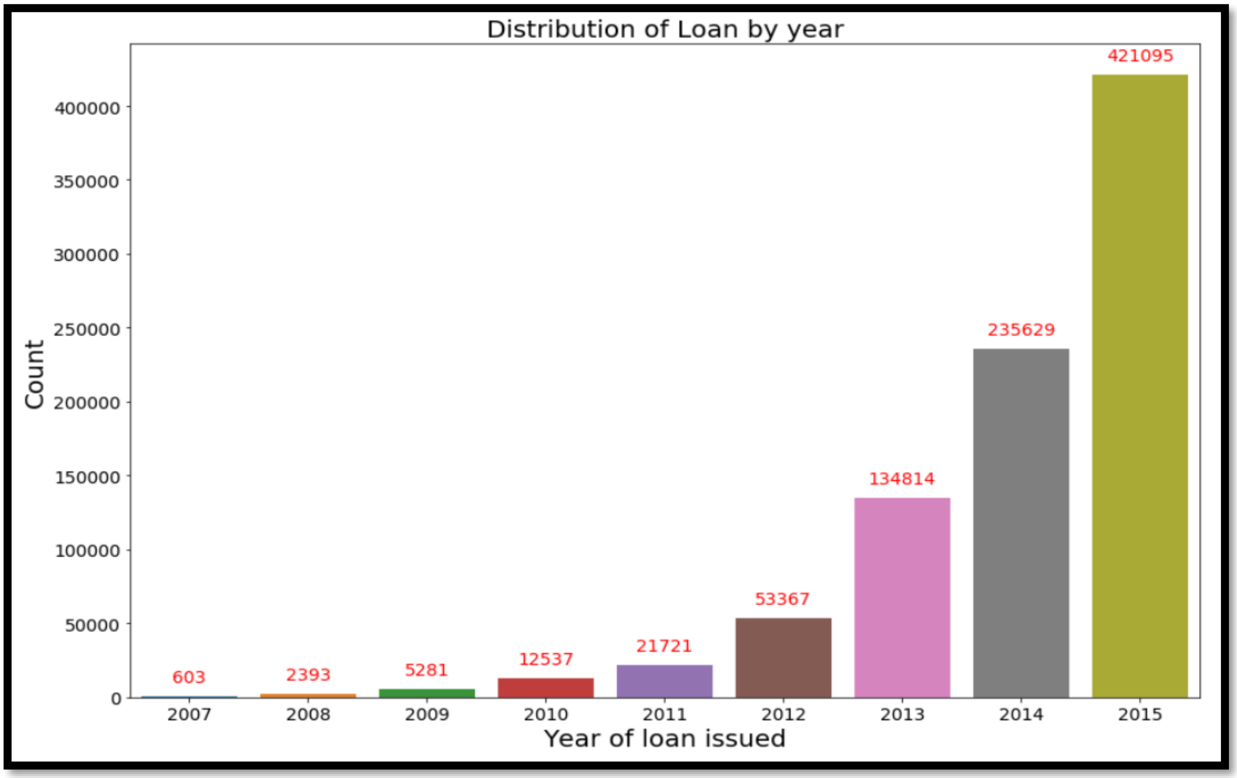


Figure 8. Number loans issued by years (data source: https://www.lendingclub.com/) own visualization.

As we can clearly see from figure 8the number of loan takers grown from 603 peoples in 2007 to 421, 095 peoples. This indicates the lending or mortgage industry is in a need of powerful machine learning technique for segregating good loan takers from bad one before granting the loan. The main features including the target variable, which are used for the modeling in this thesis work is presented with their description (Table 3).

Table 2. List of selected features used with their description. data source: https://www.lendingclub.com/

Features	**Description**
Loan_status	Current status of the loan
Annual_Inc	The self-reported annual income provided by the borrower during registration.
chargeoff_within_12_mths	Number of charge-offs within 12 months
Delinq_2_Yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
Delinq_Amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
dti	A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income.

Emp_Length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
grade	LC assigned loan grade
Home_Ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
Inq_Last_6_Mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
Int_Rate	Interest Rate on the loan
Loan_Amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
openAcc	The number of open credit lines in the borrower's credit file.
verification_status	
purpose	A category provided by the borrower for the loan request.
Revol_Util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
tax_liens	Number of tax liens
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
totalAcc	The total number of credit lines currently in the borrower's credit file

The target variable "loan status" have 9 categories in total, namely: Fully paid, charged off, doesn't meet the credit policy (status: fully paid), doesn't meet the credit policy (status charged off), current, late (31-120 days), late (16-30 days), in grace period and default (Figure 9).

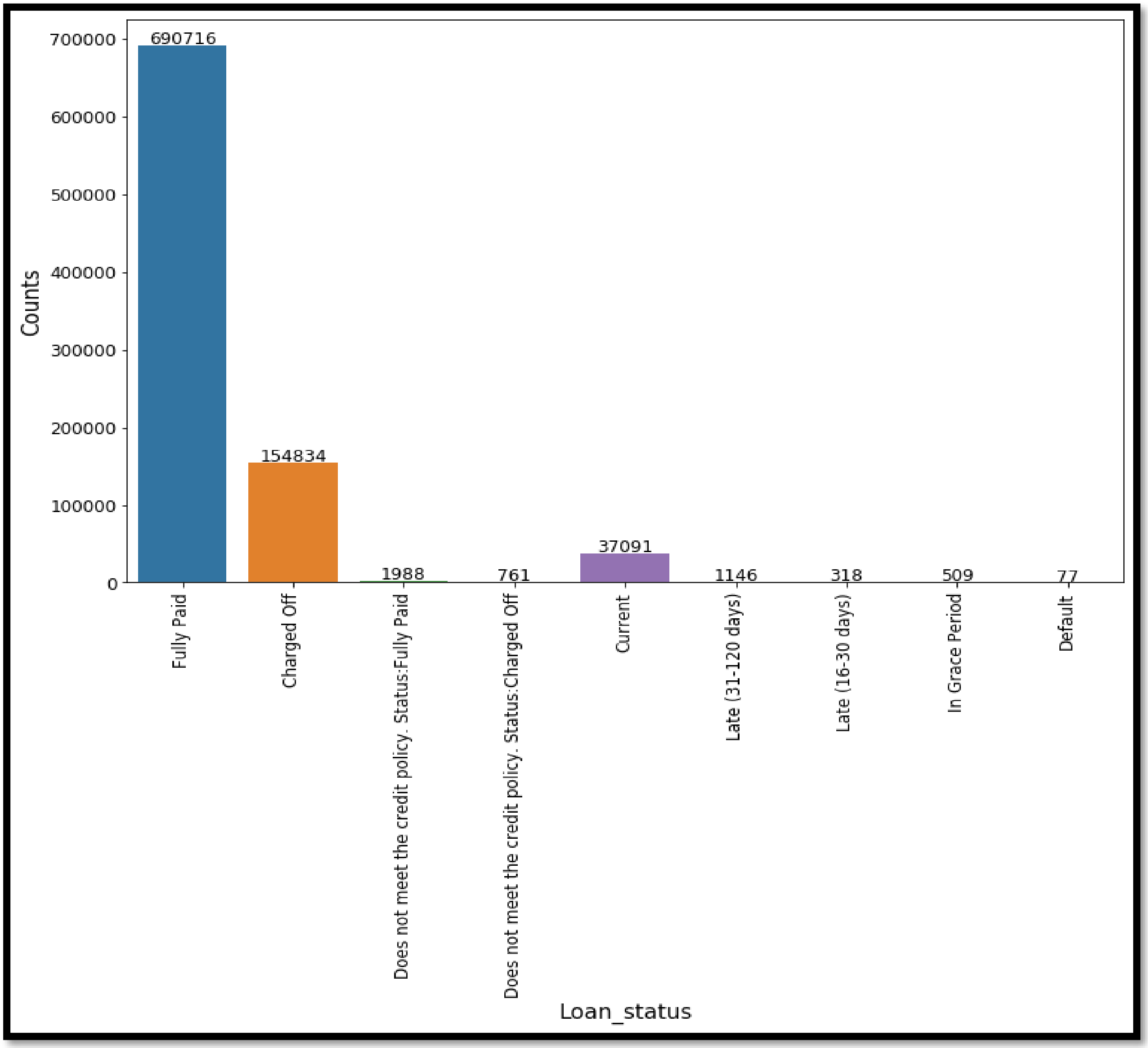


Figure 9. The target variable Loan status. (data source: https://www.lendingclub.com/). Own visualization.

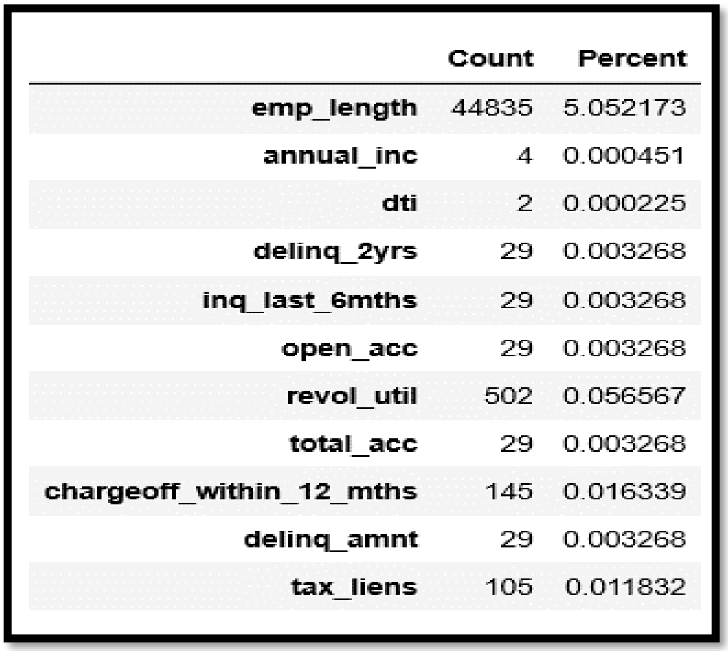
5.2.Data cleaning

In the analysis of any data, the preparation, understanding the nature of data and cleaning the data is very import. Especially in our data due to its bulkiness a huge number of missing data, outliers, duplicated values and other data error could be expected.Therefore, we will start by cleaning our data set. The data is cleaned was done in regard of both target and explanatory variables.

5.2.1. Missing values

As we have a bulk of observations with more than hundreds of features it is expected to have some sort of missing values, which have high potential of affecting our model. Therefore, those missing values must be identified, and different techniques of missing value handling must be applied starting from dropping observations with missing values to imputation. In this thesis work all the features with the missing value of more than 50% have been removed due to their possible effect on the model predicting capacity. In addition to this other feature, which have no meaning for the inclusion in the model have been removed. Those features are features like *'pymnt_plan'*, *'addr_state'*, *'inq_last_6mths'*, *'mths_since_last_delinq'*, *'open_acc'*, *'pub_rec'*, *'revol_bal'*, *'revol_util'*, *'total_acc'*, *'initial_list_status'*, *'out_prncp'*, *'out_prncp_inv'*, *'total_pymnt'*, *'total_pymnt_inv'*, *'recoveries'*, *'collection_recovery_fee'*, *'last_pymnt_amnt'*, *'last_credit_pull_d'*, *'collections_12_mths_ex_med'*, *'application_type'*, *'acc_now_delinq'*, *'tot_coll_amt'*, *'tot_cur_bal'*, *'total_rev_hi_lim'*, *'issue_d'*, *'member_id'*, *'member_id'*, *'url'*, *'policy_code'*, *'last_pymnt_d'*, *'next_pymnt_d', 'earliest_cr_line', 'emp_title', 'id', 'title', 'total_rec_int', 'total_rec_late_fee', 'total_rec_prncp', 'zip_code'.*

After removing the features with more than 50% missing value and feature, which have no any significance to the analysis we remained with only 22 features including the target variable "*loan_status*". This doesn't mean that all the remaining 22 features are without missing values, but at least they have missing value less than 50%. So, let us keep dealing with missing values.



	Count	Percent
emp_length	44835	5.052173
annual_inc	4	0.000451
dti	2	0.000225
delinq_2yrs	29	0.003268
inq_last_6mths	29	0.003268
open_acc	29	0.003268
revol_util	502	0.056567
total_acc	29	0.003268
chargeoff_within_12_mths	145	0.016339
delinq_amnt	29	0.003268
tax_liens	105	0.011832

Table 3. Features with the count and percentage of missing values. Data source: https://www.lendingclub.com/. Own calculation.

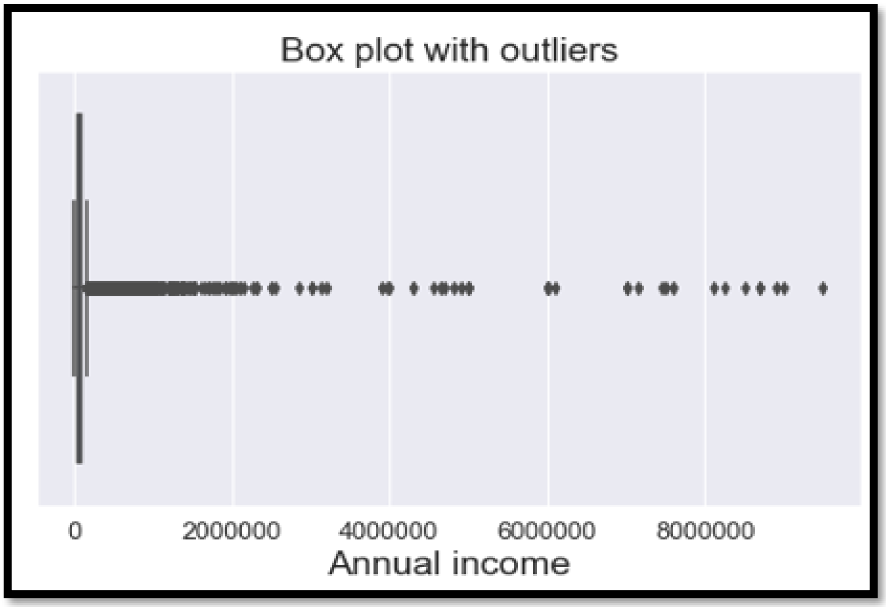
As we can see from table 1, the feature called *annual_inc* have 4 missing values and are imputed by the mean of the column. In the case of *delinq_2yrs,* there are 29 missing values. Values of *delinq_2yrs* are in the range between 0 and 39, with being 0 the most frequent one with total count of 717011. Therefore, missing value for this feature have been imputed with the mode of the feature, which is 0. Similar method have been implemented in the case of *emp_length*, meaning that the missing values imputed by the most frequent value in the column, which is *10+ years*. The imputation of missing values with the mode of the features have been done as well for *chargeoff_within_12_mths, tax_liens, delinq_2yrs* and *delinq_2yrs.* In the case of the remaining features with the missing values (*dti*, *revol_util*, *total_acc and delinq_amnt*), due to the unclear trend in the data and high number of observation it have been decided to drop rows with any possible missing values. And still we have remained with 886,936 observations**,** which indicates we do not need to worry much about the number of observations we have.

5.2.2. Duplicated rows

Other possible factor, which affect quality of model is rows with similar values through the column in other words they are called duplicated rows. Duplicity have been checked easily according the code described in the materials and method of this thesis work. Luckily our data set have no duplicated values.

5.2.3. Outliers

An outliers are any data observation, which are significantly different from others. The outliers have been identified using box plot as it described in the materials and methods. The box plot for *annual_inc* before and after the capping is presented (Figure 10).



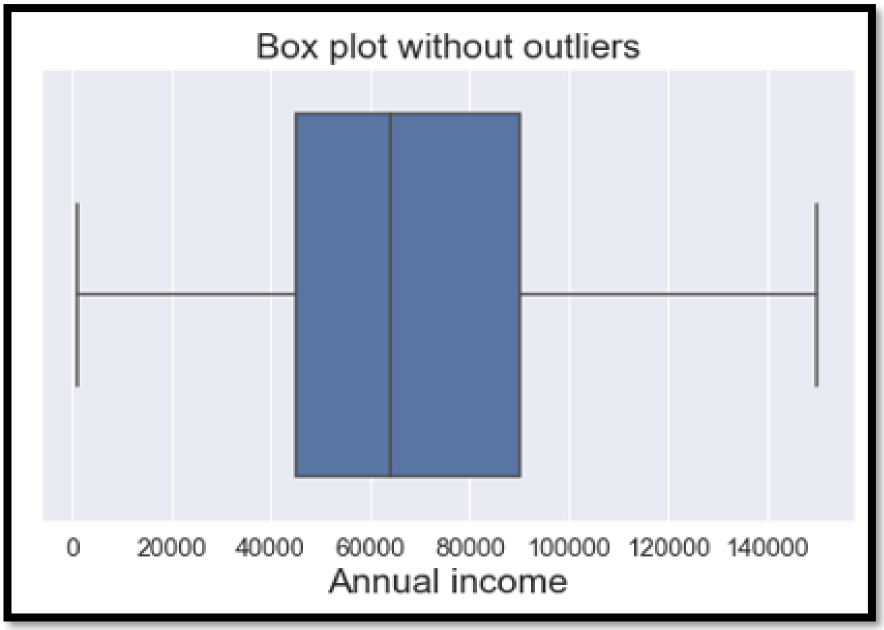


Figure 10. Annual income box plot (with and without outliers). Data source:

https://www.lendingclub.com/. Own visualization.

The same principle have been applied for *dti* (ratio of debt payments on the total debt obligations) (Figure 11). Dti is a variable,w hich is created by the Lending club as a ratio of borrower’s total monthly debt payments on the total debt obligations and the borrower’s self-reported monthly income.

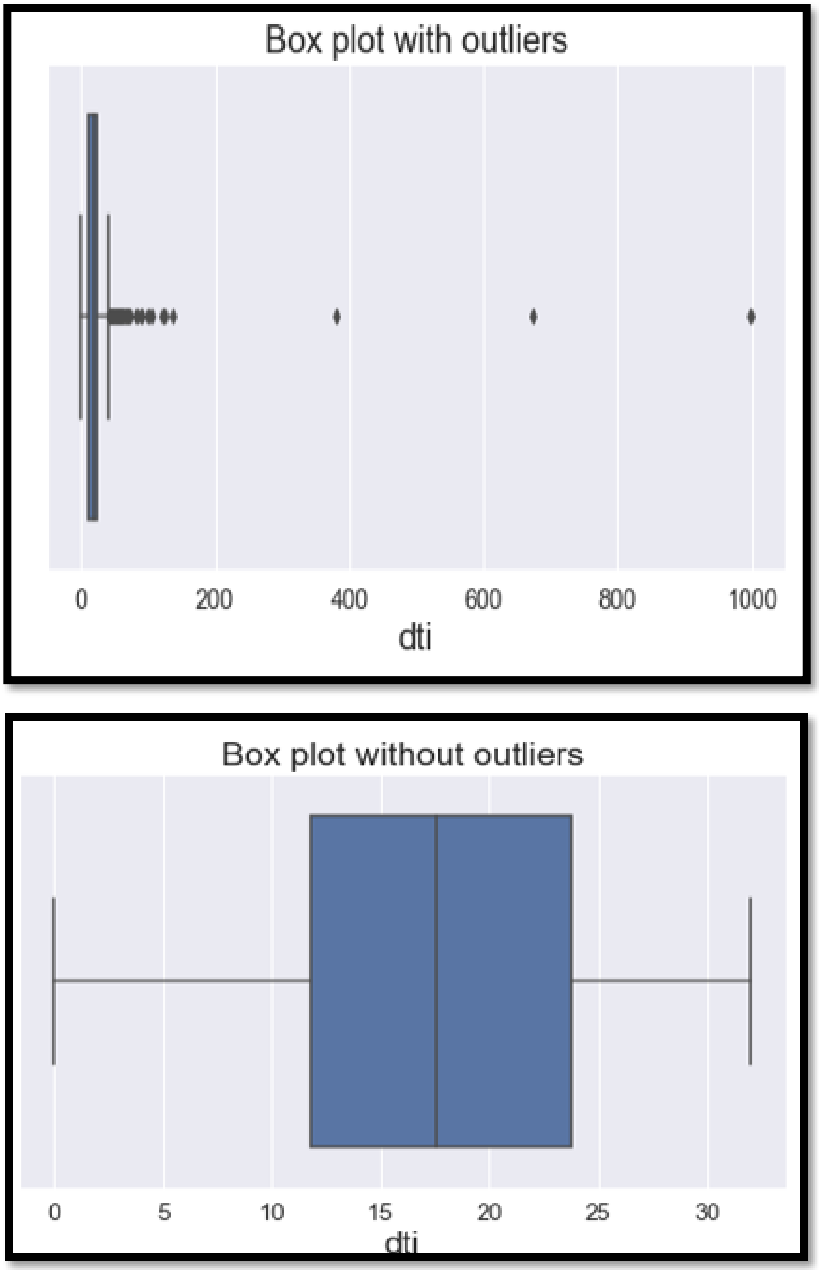


Figure 11. dti box plot (with and without outliers). Data source: https://www.lendingclub.com/. Own visualization.

Similar technique has been applied for ‘*total account’ and ‘last_fisco_range_high’*. Box plot for *‘total account’* and *‘last_fisco_range_high’* before and after the handling of missing value is displayed on figure 12 and 13 respectively.

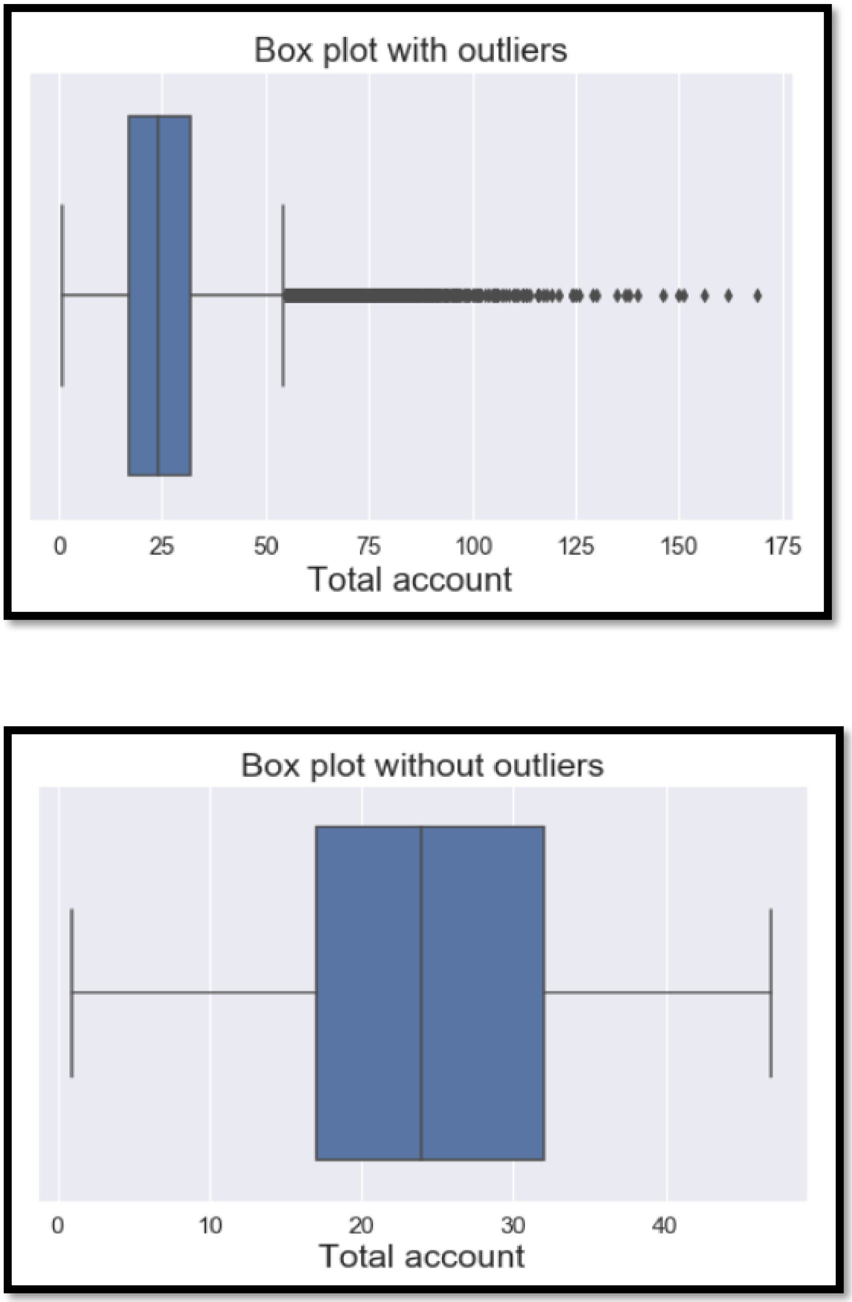


Figure 12. total account box plot (with and without outliers). Data source: https://www.lendingclub.com/. Own visualization.

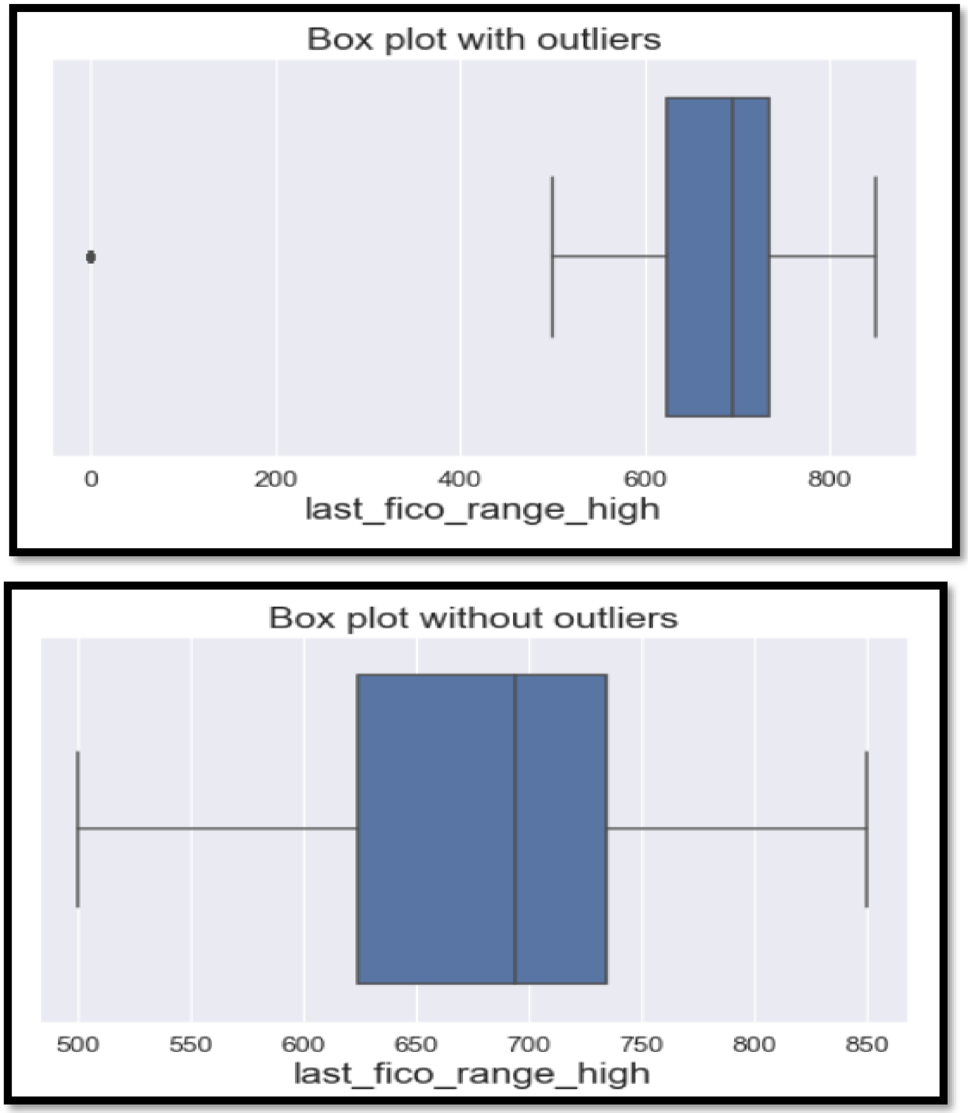
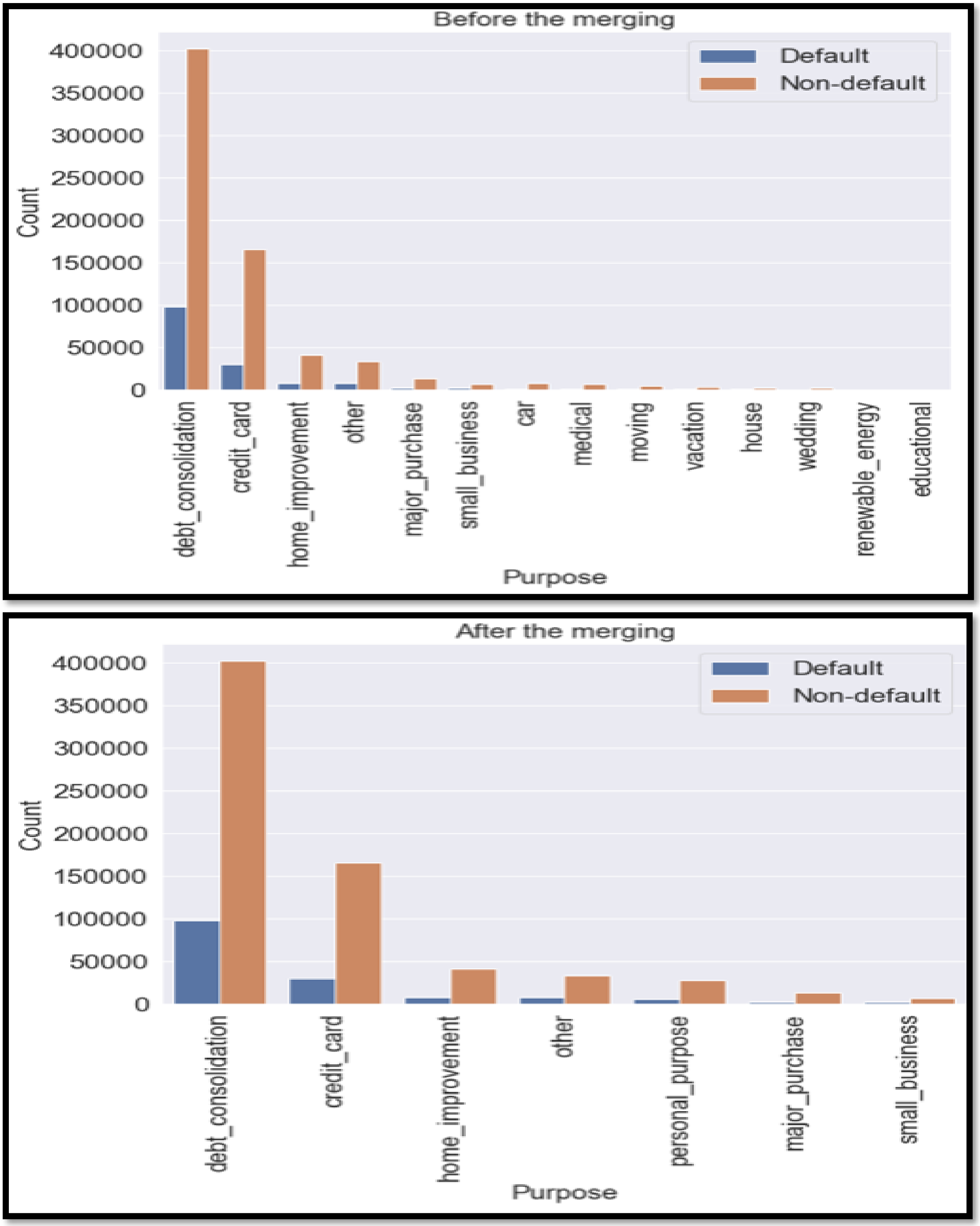


Figure 13. last_fico_range_high box plot (with and without outliers). Data source: https://www.lendingclub.com/. Own visualization.

In the case of categorical variables purpose, the category with less than 1% counts have been merged and assigned as new category in the variables. The original purpose variable have total 14 variable from which 8 of them have observation with the count of less than 1%. Therefore those variables with the count of less than 1% namely: medical, car, weeding, moving, vacation, house, education and renewal energy are assigned to the new created variable called personal purpose.

Therefore, modified variable purpose have only 7 categories.



Before the merging

After the merging

Figure 14. Bar chart for the variable purpose before and after the merging of categories. Data source: https://www.lendingclub.com/. Own visualization.

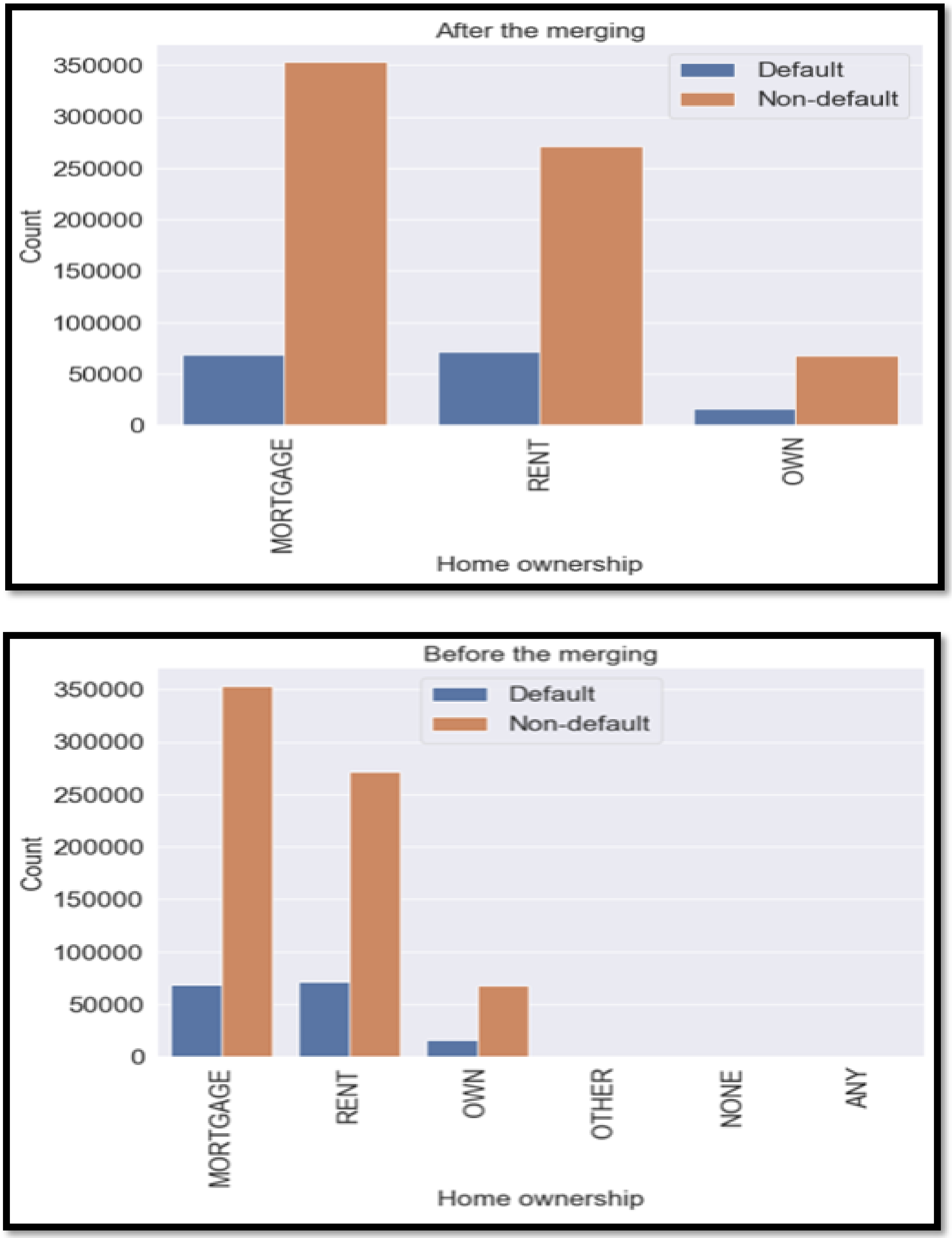


Figure 15. Count plot for the variable home_ownership before and after the merging of categories.

Data source: https://www.lendingclub.com/. Own visualization.

On the other side for the variable *home_ownership* with total category of 7 have been transformed to 3 categories by dropping three categories *OTHER*, *NONE* and *ANY*, which all together accounts very less than 1% of the total observations in the data set. Dropping is preferred rather than merging them to other category due to the wide variation in context than others (Figure 15). The outliers from the remaining numerical and categorical variables have been also dropped, caped or combined accordingly.

5.2.4. Multicollinearity

The high collinearity between the predictor variables is among the biggest challenge of the statistical data modeling. Multicollinearity could potentially affect the hypothesis testing and the estimate of the parameters by creating inaccurate and unstable estimate of variance (Midi et al., 2010). A heatmap of Pearson correlation matrix for all predictor variable is presented below (Figure 16). Based on our heatmap *loan_amnt* and *installment, last_fico_range_high* and *last_fico_range_low* are highly correlated, meaning that we should avoid one of those variable from the combination to avoid the effect of multicollinearity from our model. Therefore, the *installment* and *last_fico_range_low* are dropped from being considered in the model.

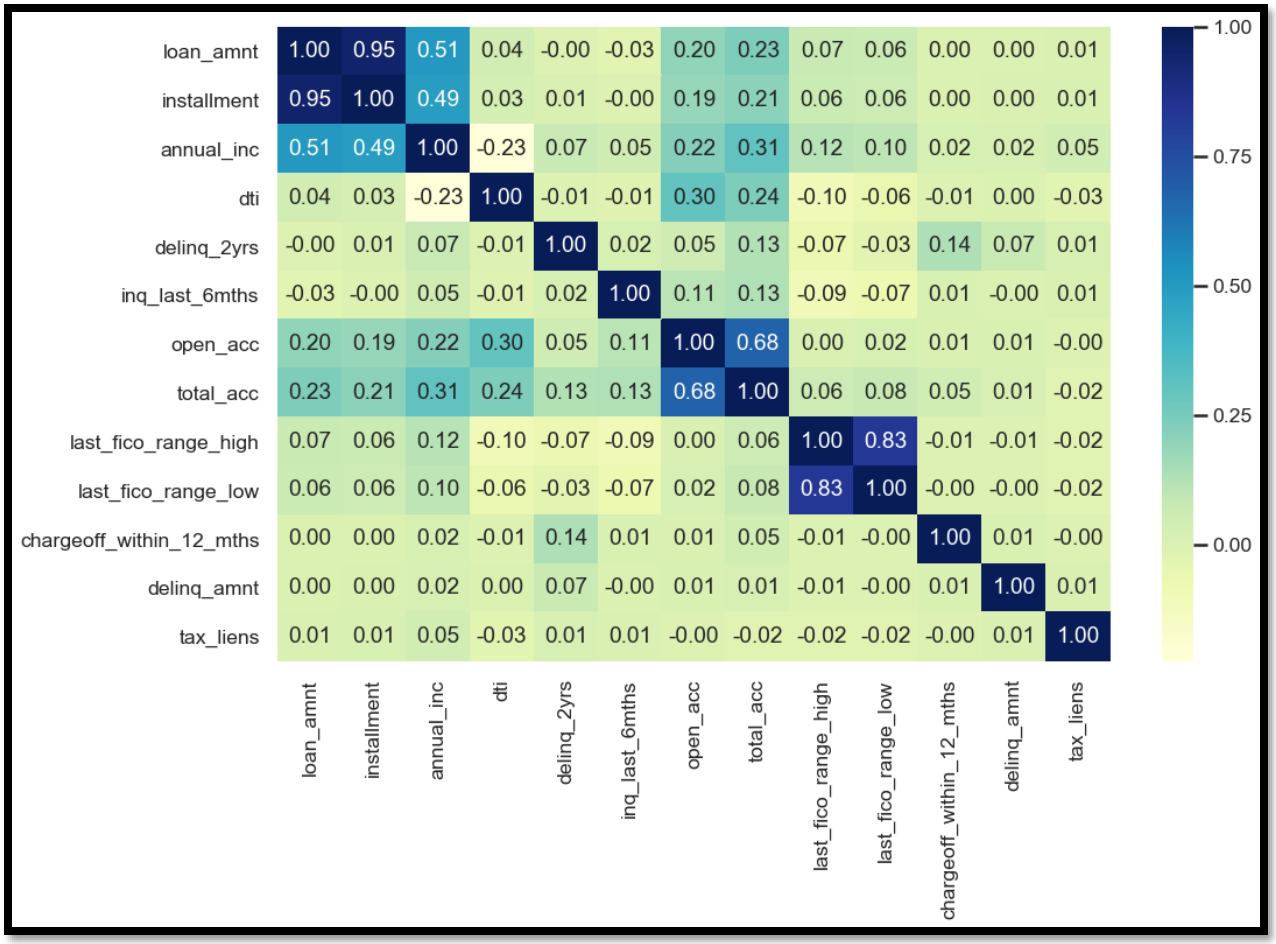


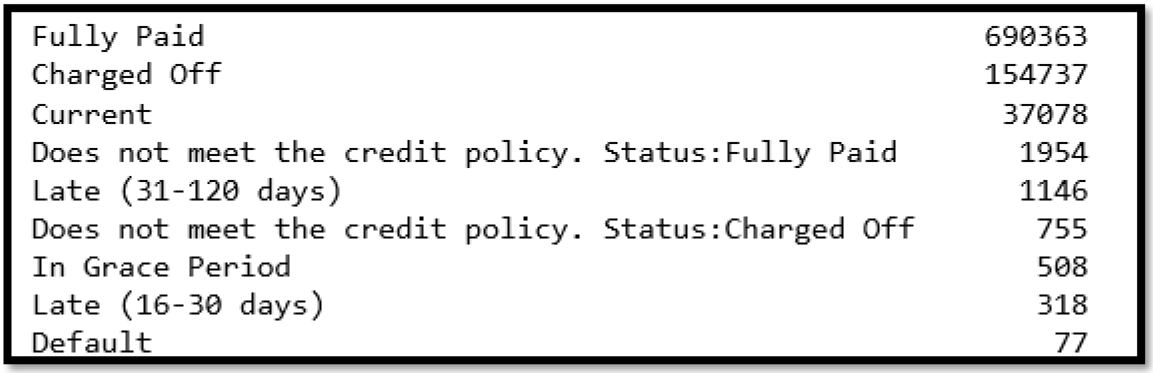
Figure 16. Heatmap of the correlation matrix. Data source: https://www.lendingclub.com/. Own visualization.

5.2.5. Standard normal distribution

Standard normal distribution is very essential in the any statistical modeling as the variables going to be used may have differ in measurements and some of the statistical models may also perform very poor with this type of data, which is doesn't have standard normal distribution. It also greatly helps in speeding up the algorithms we are going to use. Therefore due to this reason the standardizing of the data have been implemented according to the method described in the materials and method section of this thesis.

5.2.6. Visualization and data exploration

Now before further doing let us deal with the target variable *loan status.* In the loan status, which is our target variable have 7 categories as we can see below.



```
Fully Paid                                            690363
Charged Off                                           154737
Current                                                37078
Does not meet the credit policy. Status:Fully Paid      1954
Late (31-120 days)                                      1146
Does not meet the credit policy. Status:Charged Off      755
In Grace Period                                          508
Late (16-30 days)                                        318
Default                                                   77
```

To suits our model of credit scoring we need a binary target variable and here the categories *Fully paid and Does not meet the credit policy (Status:Fully Paid)* considered as non-defaulter and assigned as 1, while *Charged Off, Default and Does not meet the credit policy (Status:Charged Off)* are considered as defaulter and assigned as 0. The remaining categories are dropped as they are *current, late and in grace period*, which possibly doesn't belong any of our two groups defaulter and non-defaulters.

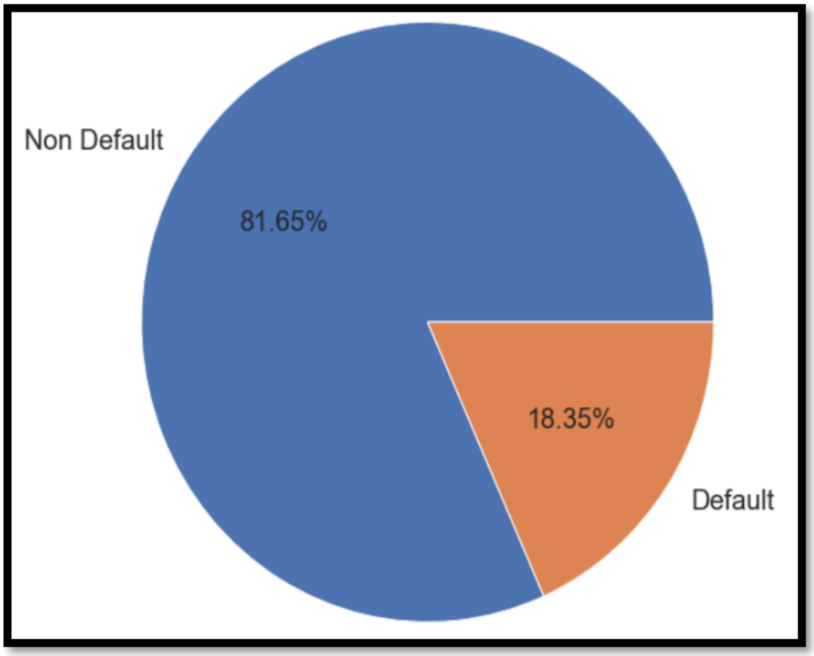


Figure 17. The pie plot for the distribution of defaulters and non-defaulters. Data source: https://www.lendingclub.com/. Own visualization.

Based on the newly reformed target variable (figure 17), the 81.65% of loan status belongs to non-defaulter and 18.35% to the defaulter. We can clearly see that we there are much more customers paying their loan on time than non-paying. The imbalance in number between the defaulter and non-defaulter will greatly affects the model we are going to build. While the modeling we are going to balance the number of defaulters and non-defaulters and this is dealt in the coming chapters.

Now let us do some visualization to see how categories of the categorical variables are distributed among defaulters and non-defaulters and discuss if they could be potential predictors of defaulters.

Variable ***Grade:*** this variable is a characteristic of the borrowers, which is assigned by the lending club from A - G depending on the potential of the borrowers to qualify for the loan. From the figure 18 we can see that most of borrowers are from the grade B. when we see numbers of defaulters the highest count is from grade C, but when we with less grade like D, E, F and G the ratio of defaulters become almost 50:50. Meaning that the lower the grade the borrowers gave it is most probable they will be defaulters. therefore, the use of grade as predictor could be very useful for the model of credit scoring

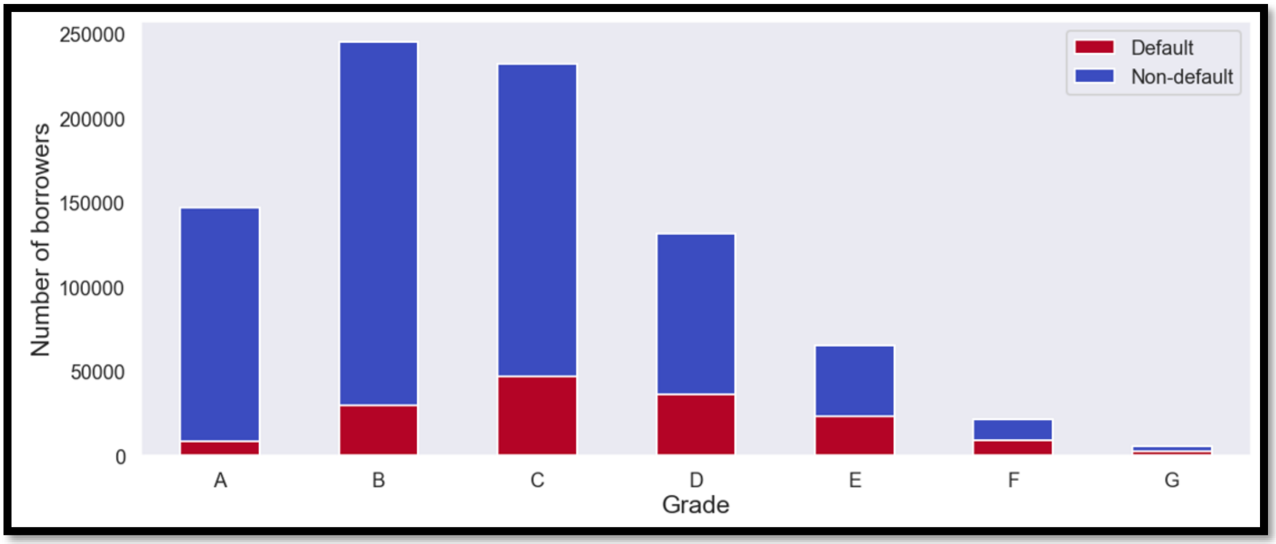


Figure 18. The distribution of category of variable grade among defaulters and non-defaulters. Data source: https://www.lendingclub.com/. Own visualization.

Variable ***home_ownership*****:** it refers the home ownership status of the borrowers based on the information provided by the borrower during the application of the loan. As we have discussed in the earlier chapter this variable has been merged to three categories from 6. As the dropped categories have count of observation less than 1% and now, we have three categories rent, own and mortgage (figure 19). As we can see majority of loan takers are peoples with some mortgage and the list are have their own home. When we see defaulters 8.1, 8.4 and 1.9% are defaulters from mortgage, rent and homeowners respectively. This could potentially shows that the use of the variable ownership to be potential predictors.

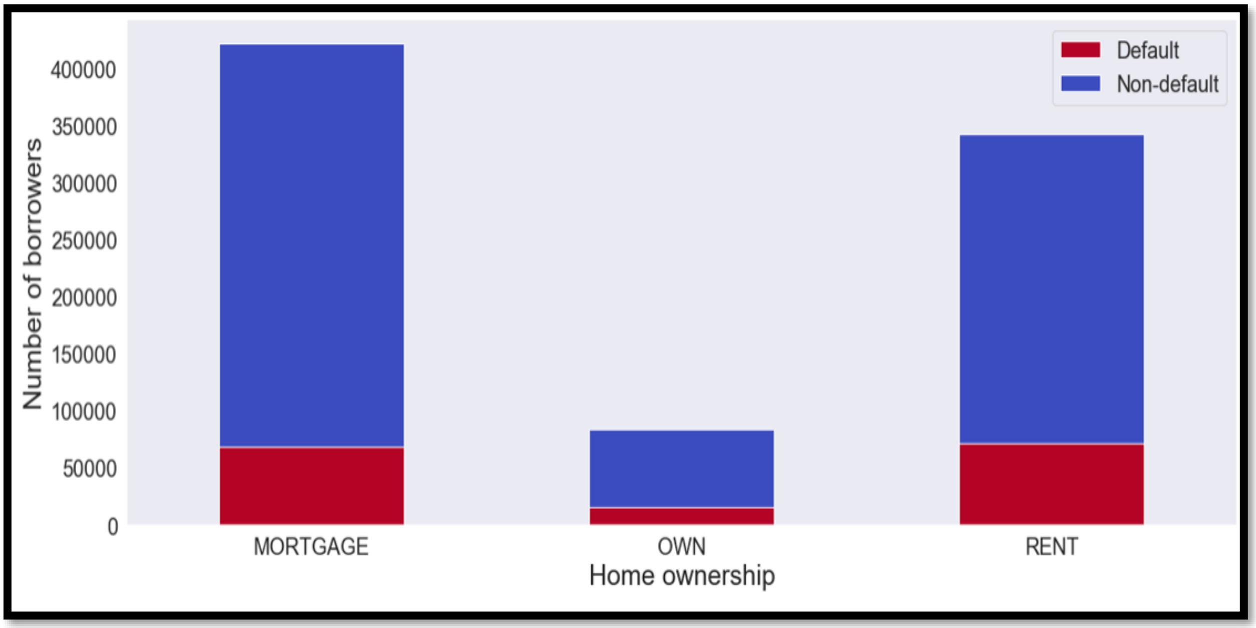


Figure 19. Distribution of home ownership category as defaulter and non-defaulter. Data source: https://www.lendingclub.com/. Own visualization.

Variable ***term***: refers the number of payments on the loan and its values are in months and can be either 36 or 60. As we can see figure 20 most of loans are for 36 months and from this 10.2% are defaulters. when we come to the 60 months loan takers, they account 26.7% of the total data set but from this 8.1% of them are defaulters meaning that almost 30% of the 60 months loan takers are defaulters, while only 14% of the 36 months loan takers are defaulters. Variable ***verification_status***: this variable refers the income status of the borrower if it is verified or not verified. Most of the borrowers have a verified income source (figure 21). Among the borrowers with verified income 20% of them are defaulters and from the borrowers with unverified income source 14% of them are defaulters.

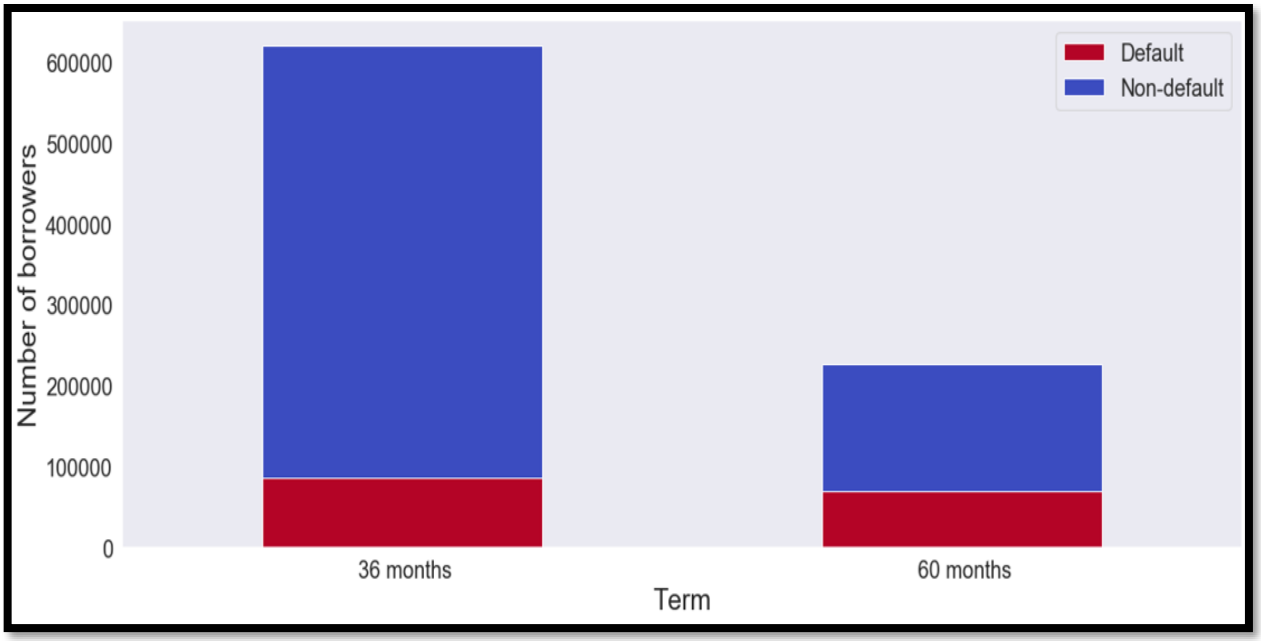


Figure 20. The distribution of defaulters and non-defaulters among the categories of loan term. Data source: https://www.lendingclub.com/. Own visualization.

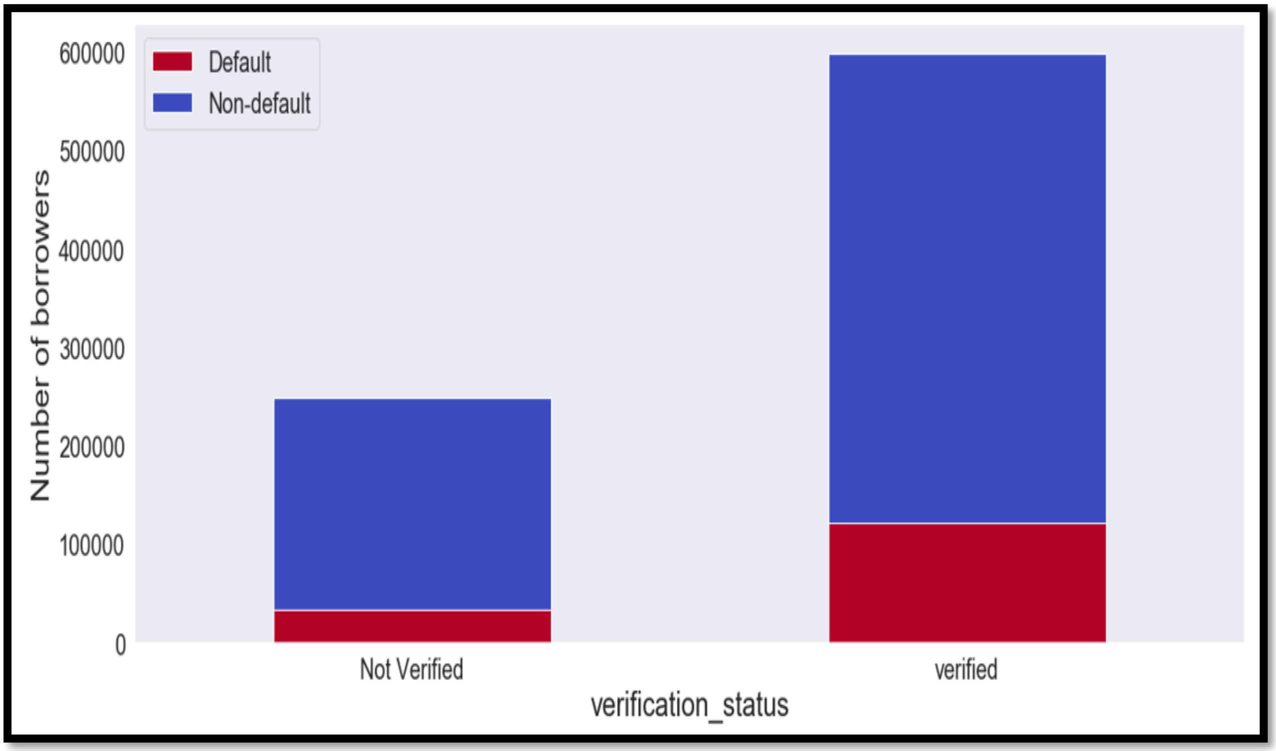


Figure 21. Distribution of defaulters and non-defaulters among borrowers with verified income source and not verified. Data source: https://www.lendingclub.com/. Own visualization.

Variable ***purpose***: is the category of purpose, which the borrower requested the loan for. Most of the loan takers borrowed the loan for debt consolidation and among the defaulters 63% of them are from category of *debt_consolidation* and the list 0.2% are from *personal_purpose* and *small_business* category (Figure 22).

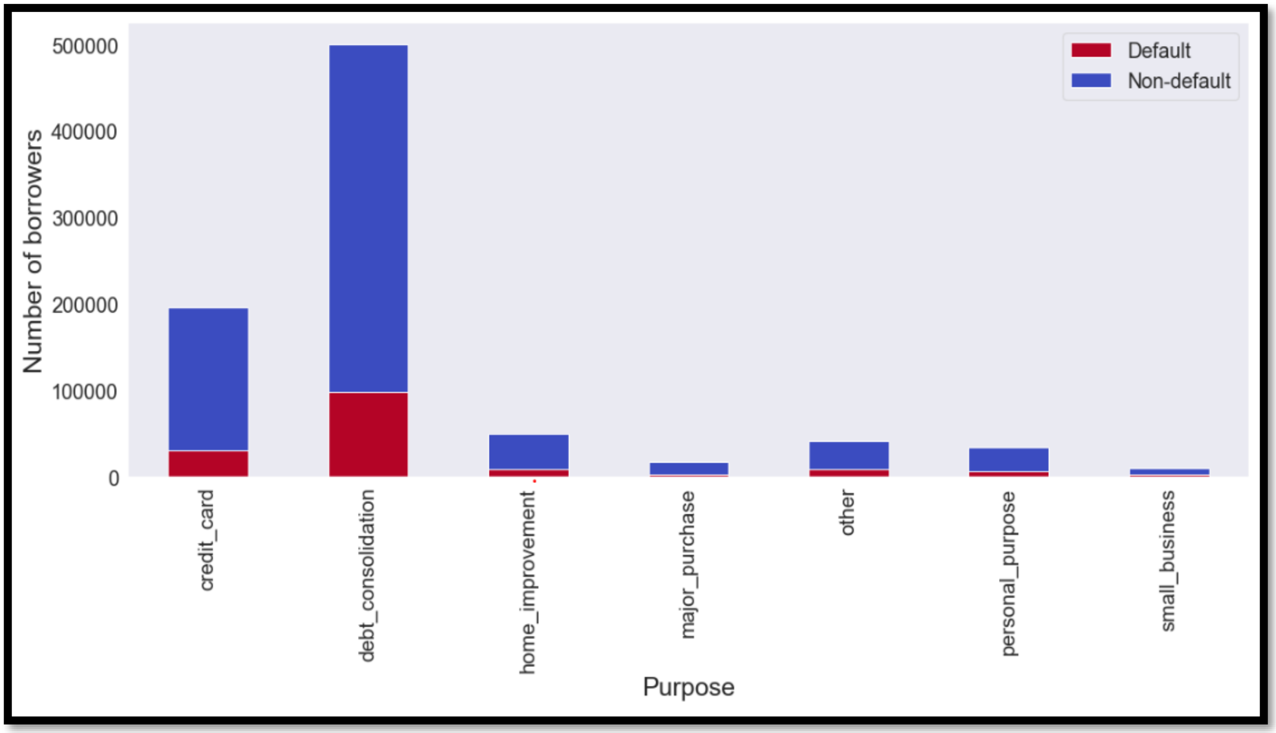


Figure 22. Distribution of defaulters and non-defaulters among categories of purpose. Data source: https://www.lendingclub.com/. Own visualization.

Variable ***emp_length***: refers the length of years since the borrowers become employed and it ranges from 10+ more to < 1 year. Most of the borrowers have employment length of 10 and more years figure 23. Among the defaulters 38% of them are from category 10+ years.

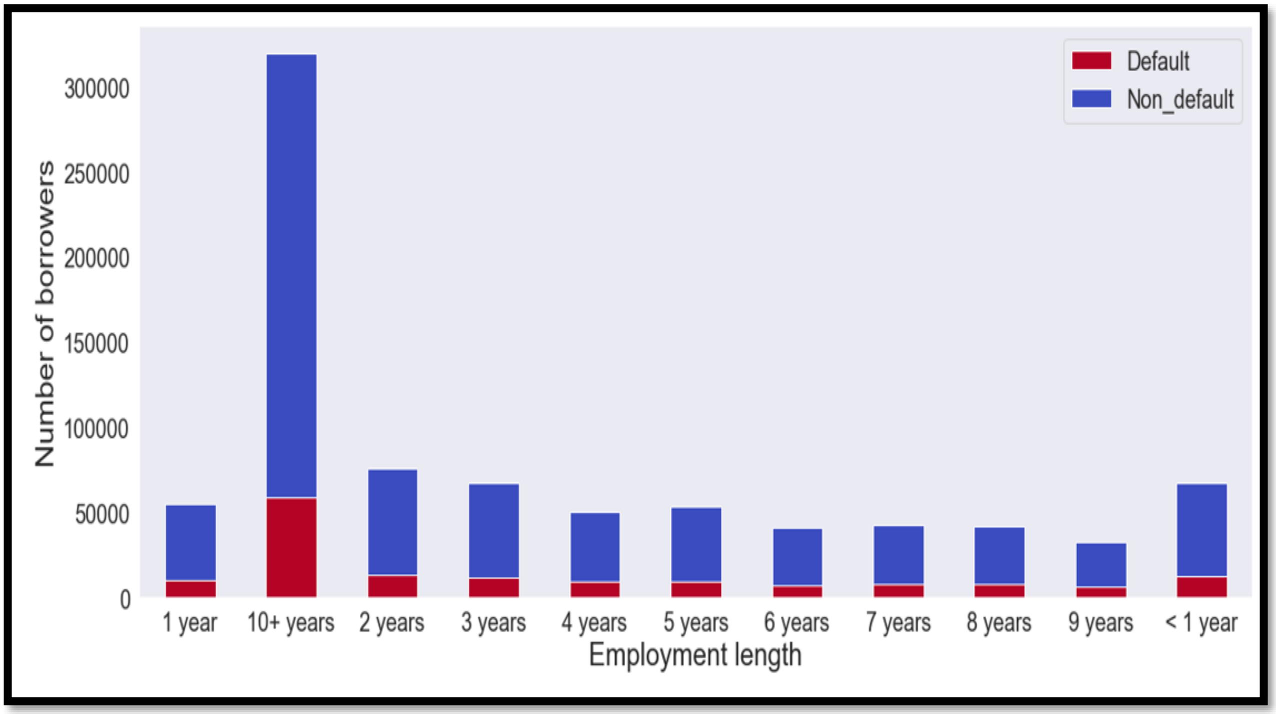


Figure 23. Distribution of defaulters and non-defaulters among categories employment length. Data source: https://www.lendingclub.com/. Own visualization.

6. Result and discussion

In this chapter of this thesis three type of predictive modeling (logistic regression, decision tree and Neural networks) will be implemented and their power to predict credit defaulters will be evaluated. For each model's a confusion matrix and receiver operator characteristics (ROC) curve is presented diagrammatically. The accuracy, precision, Recall, F1 score, ROC will be presented in table for comparison and their predicting power will be ranked based on accuracy, precision, Recall, F1 score, ROC.

6.1.Modeling the credit score

The first step of any modeling by the techniques of machine learning using python is to split our data in to two part, the training set and the test set. This section of the modeling is very crucial because after modeling our algorithm we need to check for the quality of the model with separate data set, which the algorithm have no knowledge about. Therefore, we can test the quality on the test set and no biases will happen. For this purpose, 25% of the whole dataset is divided to be test set of the model. The next step is balancing the number of defaulters and non-defaulters. The effect of imbalanced data set have been widely discussed in literature (Brown and Mues, 2012). In the data set there are much higher number of non-defaulters than defaulters and if we use this un-proportional then the model could prone to be biased in differentiating between defaulters and non-defaulters. Balanced X (predictors) and balanced y (target variable loan status) is created from the training data set. So, now we have balanced Xbalanced and ybalanced data set. As the sest set data is not separated into predictors and target variable, the separation in to Xtest and ytest. The test set is important for the evaluation of the model in the letter stage of the model. The whole procedure for this is clearly described in the materials and method chapter of this thesis.

6.1.1. Logistic regression

Logistic regression is the most widely used method of credit scoring. Its simplicity and easiness to understand make logistic regression the most widely used method of credit scoring. Based on the result of confusion matrix (Figure 24) from our logistic regression, the model was able to predict 37,295 of defaulters correctly out of 38, 892. This shows that the model can predict 95.9 % defaulters correctly. On the other side the model predicted 121,475 of non-defaulters correctly out of 173,080, this tells the model can correctly predict 70.2% of non-defaulter correctly. From this, we can tell that the model is better in predicting defaulter than non-defaulter. Source: own

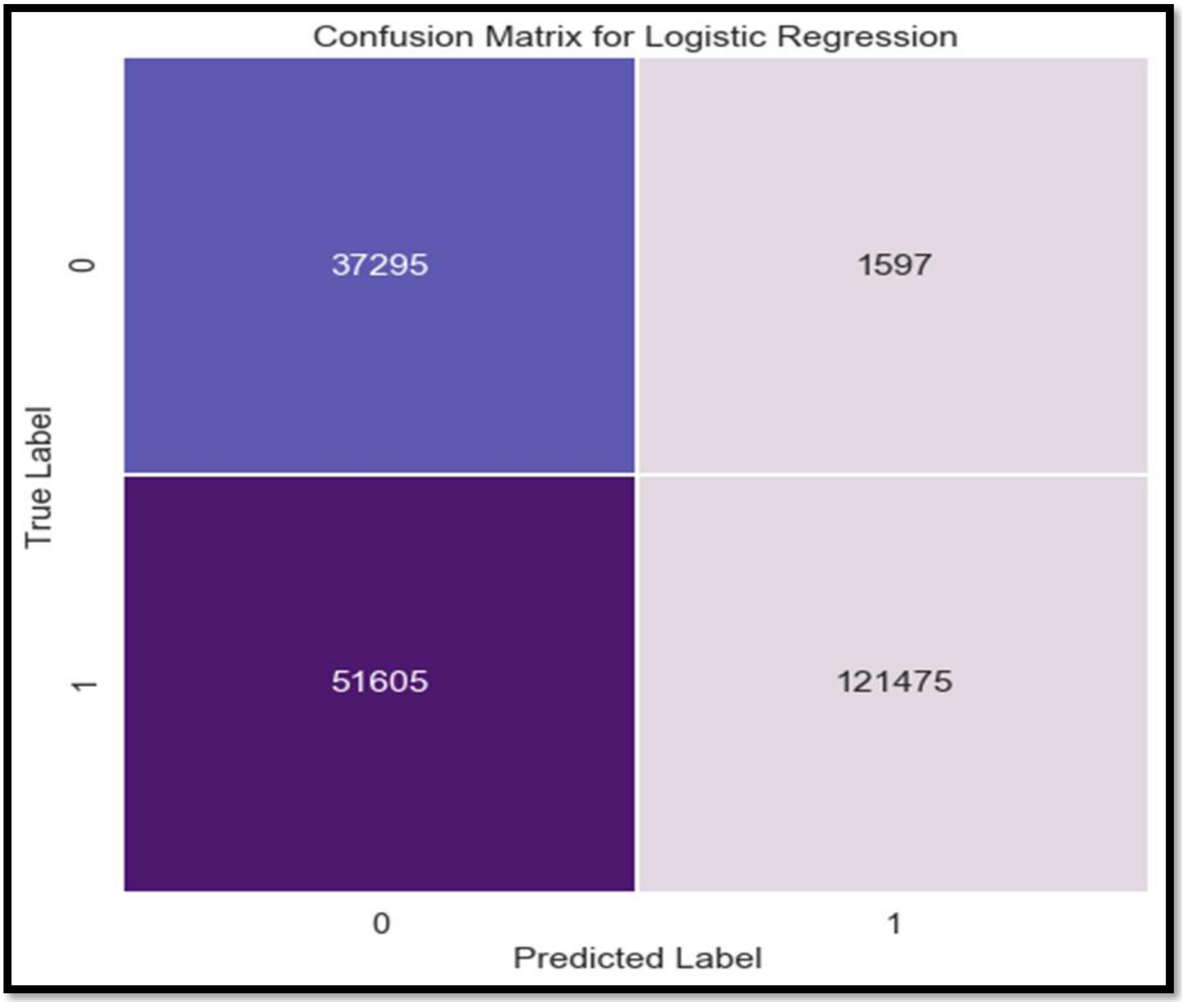


Figure 24. Confusion matrix from logistic regression model. Data source: https://www.lendingclub.com/. Own visualization.

This can be seen from two business point, first our model is better in avoiding risk by lending for potential defaulters but the secondly our model is categorizing almost 30% of non-defaulter as defaulter, this tells potential of pushing peoples, who are non-defaulters from lending them and this potentially reduce income of the lendingclub.

From the confusion matrix we got the most common machine learning model quality evaluation criteria called the accuracy, Precision, Recall and the F1. The over whole meaning and calculation is clearly described in the materials and method chapter of this thesis.

Accuracy:

Based on the result from python the Accuracy of our logistic regression model is 0.75. Meaning that this model is 75% accurate.

Cross validation accuracy:

Based on the cross validation our logistic regression is 88% accurate.

Precision:

The precision of our model, as it is calculated by python is 0.987. Which means, among the non-defaulters predicted by the model, the 98.7% of them are truly Non-defaulters.

Recall:

From our logistic regression model, the Recall was 0.702. meaning that, out of the actual Non-defaulters our model was able to predict 70% of them correctly.

F1 score:

Our regression model F score value as computed by python is 0.82, meaning that we have somehow better model according to F score.

Receiver Operator Characteristic (ROC):

ROC curve build from logistic regression is presented (Figure 25). From our ROC curve we can see that the curve is hang up quite better to the top left corner of the plot. This signify our model have good quality. In addition to this the AUC value is presented on the plot us 0.92. This tells that by using our model there is probability of 92% to distinguish between defaulters and non-defaulters. The AUC value approaching to 1 refers again better quality (James et al., 2013).

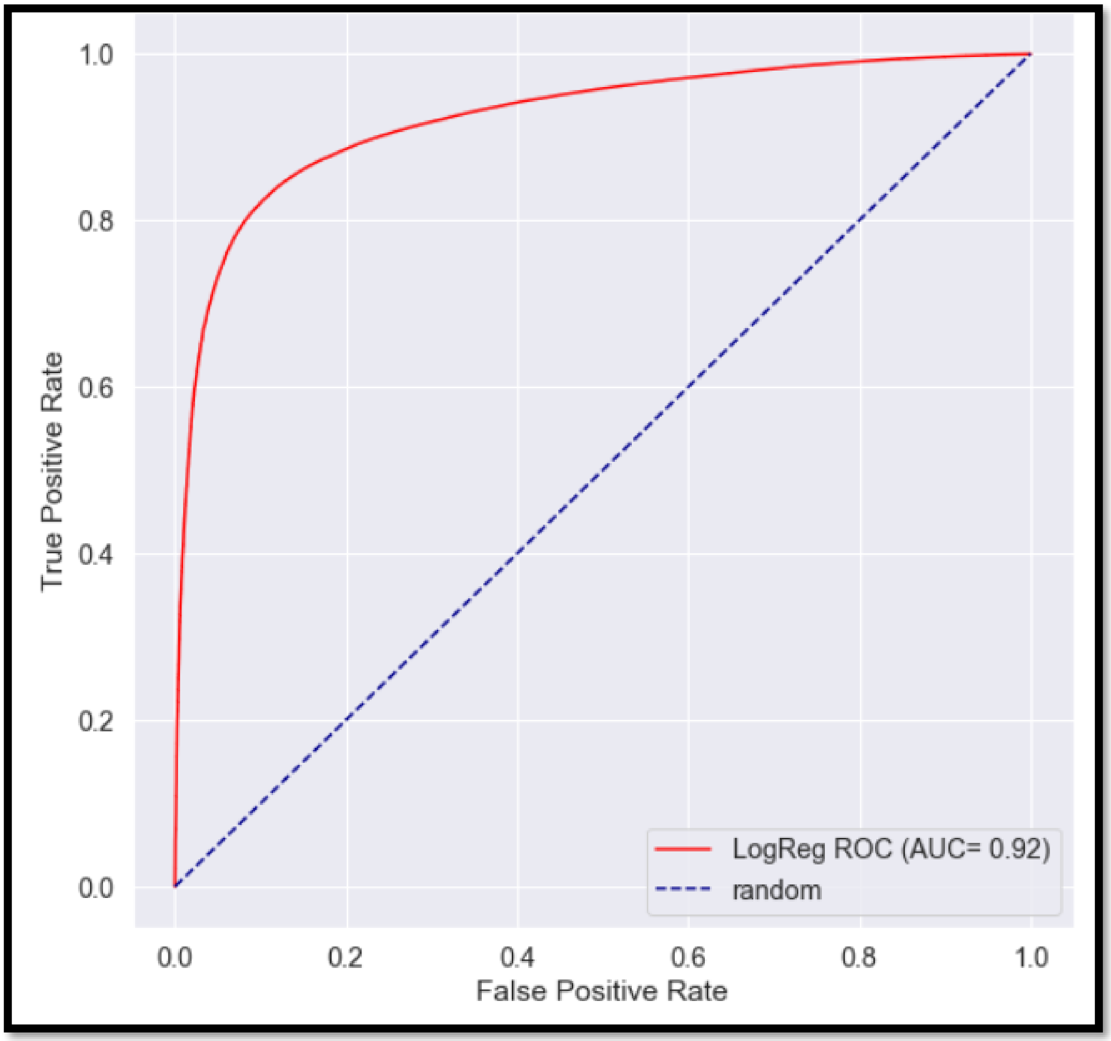
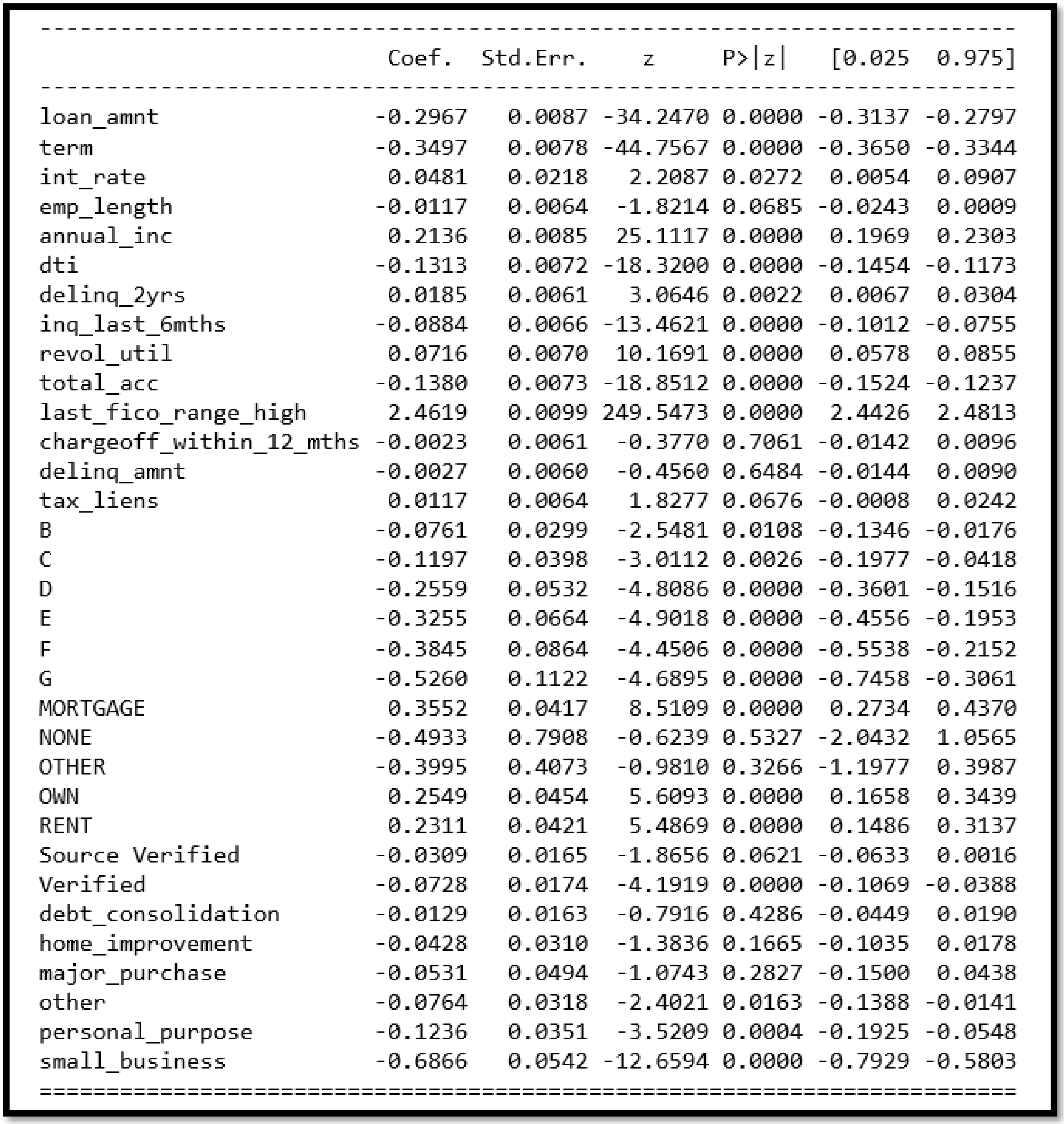


Figure 25. ROC curve of logistic regression. Data source: https://www.lendingclub.com/. Own visualization.

Based on table 4 output of logistic regression summary statistics the loan amount, loan term, annual income, borrowers total debt payment over obligation, inquiries last 6 months, revolving utilities, total account and last fico range high are the most important predictor of defaulters with p = 0.000, delinquency of 2 years with p = 0.01, interest rate with p = 0.05.



```
--------------------------------------------------------------------------
                           Coef.  Std.Err.    z     P>|z|   [0.025  0.975]
--------------------------------------------------------------------------
loan_amnt                 -0.2967   0.0087 -34.2470 0.0000 -0.3137 -0.2797
term                      -0.3497   0.0078 -44.7567 0.0000 -0.3650 -0.3344
int_rate                   0.0481   0.0218   2.2087 0.0272  0.0054  0.0907
emp_length                -0.0117   0.0064  -1.8214 0.0685 -0.0243  0.0009
annual_inc                 0.2136   0.0085  25.1117 0.0000  0.1969  0.2303
dti                       -0.1313   0.0072 -18.3200 0.0000 -0.1454 -0.1173
delinq_2yrs                0.0185   0.0061   3.0646 0.0022  0.0067  0.0304
inq_last_6mths            -0.0884   0.0066 -13.4621 0.0000 -0.1012 -0.0755
revol_util                 0.0716   0.0070  10.1691 0.0000  0.0578  0.0855
total_acc                 -0.1380   0.0073 -18.8512 0.0000 -0.1524 -0.1237
last_fico_range_high       2.4619   0.0099 249.5473 0.0000  2.4426  2.4813
chargeoff_within_12_mths  -0.0023   0.0061  -0.3770 0.7061 -0.0142  0.0096
delinq_amnt               -0.0027   0.0060  -0.4560 0.6484 -0.0144  0.0090
tax_liens                  0.0117   0.0064   1.8277 0.0676 -0.0008  0.0242
B                         -0.0761   0.0299  -2.5481 0.0108 -0.1346 -0.0176
C                         -0.1197   0.0398  -3.0112 0.0026 -0.1977 -0.0418
D                         -0.2559   0.0532  -4.8086 0.0000 -0.3601 -0.1516
E                         -0.3255   0.0664  -4.9018 0.0000 -0.4556 -0.1953
F                         -0.3845   0.0864  -4.4506 0.0000 -0.5538 -0.2152
G                         -0.5260   0.1122  -4.6895 0.0000 -0.7458 -0.3061
MORTGAGE                   0.3552   0.0417   8.5109 0.0000  0.2734  0.4370
NONE                      -0.4933   0.7908  -0.6239 0.5327 -2.0432  1.0565
OTHER                     -0.3995   0.4073  -0.9810 0.3266 -1.1977  0.3987
OWN                        0.2549   0.0454   5.6093 0.0000  0.1658  0.3439
RENT                       0.2311   0.0421   5.4869 0.0000  0.1486  0.3137
Source Verified           -0.0309   0.0165  -1.8656 0.0621 -0.0633  0.0016
Verified                  -0.0728   0.0174  -4.1919 0.0000 -0.1069 -0.0388
debt_consolidation        -0.0129   0.0163  -0.7916 0.4286 -0.0449  0.0190
home_improvement          -0.0428   0.0310  -1.3836 0.1665 -0.1035  0.0178
major_purchase            -0.0531   0.0494  -1.0743 0.2827 -0.1500  0.0438
other                     -0.0764   0.0318  -2.4021 0.0163 -0.1388 -0.0141
personal_purpose          -0.1236   0.0351  -3.5209 0.0004 -0.1925 -0.0548
small_business            -0.6866   0.0542 -12.6594 0.0000 -0.7929 -0.5803
==========================================================================
```

Table 4. Estimators statistics and model evaluation. Data source: https://www.lendingclub.com/. Own visualization.

6.1.2. Decision tree

Most of the steps and procedure to be done in the case of decision tree are similar with that of logistic regression, so here the functions will be presented with some footnote only. The decision tree function is imported, initiated and the model is trained by the training data set as follows. The pruned version of the decision tree is plotted by the following function. The decision tree is pruned for the better fit (Figure 26). In this thesis the classification and regression tree (CART) specifically classification tree has been used. To evaluate the quality of decision tree model developed by python, the confusion matrix (Figure 27) and ROC curve (Figure 28) is presented. The model was able to predict 38,043 of defaulters correctly out of 38, 892. This shows that the model can predict 99.2 % defaulters correctly. On the other side the model predicted 99,224 of non-defaulters correctly out of 173,080, this tells the model can correctly predict 34% of non-defaulter correctly.

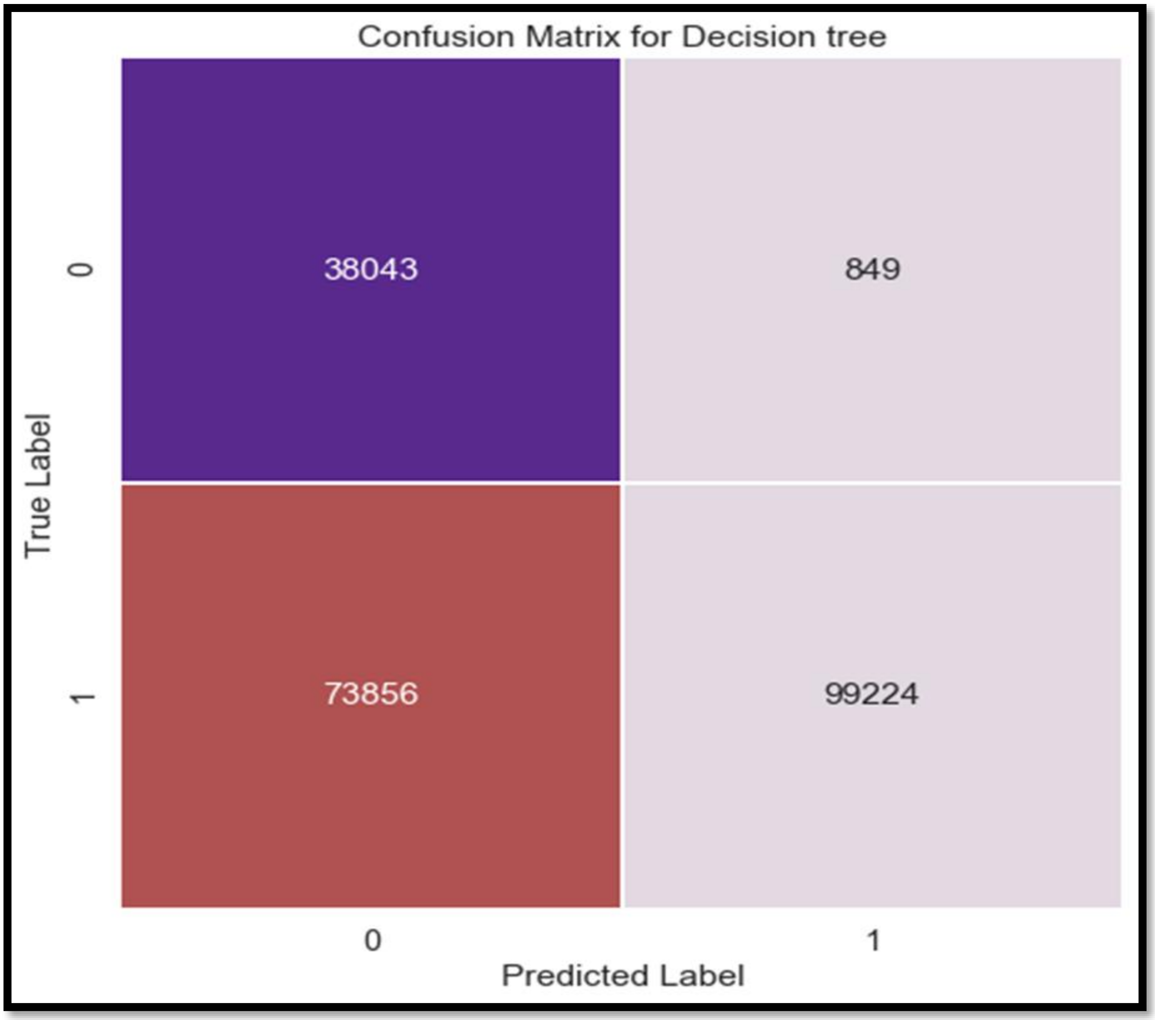


Figure 26. Confusion matrix of decision trees. Data source: https://www.lendingclub.com/. Own visualization.

Based on the confusion matrix let us describe the decision tree model by presenting the accuracy, precision, recall and the F1 score computed by python.

Accuracy: The accuracy of decision tree model is 0.65. Meaning that the model is 65% accurate overall.

Cross validation accuracy: Is 0.85.

Precision: The precision of decision tree developed 0.992. Which means, from the non-defaulters predicted by the model, 99.1% of them are truly Non-defaulters.

Recall: The computed recall was 0.573. This tells that the decision tree developed predicted only 57.4% of non-defaulters correctly.

F1 score: The harmonic mean of precision and recall for decision tree is 0.727, signifying moderate quality of the model.

ROC curve: The decision tree model developed also have quiet good ROC curve (Figure 27), which hanged to the left top of the plot. And the calculated AUC for the decision tree is 0.89. This indicates that the model has 89% probability to distinguish between defaulters and non-defaulters.

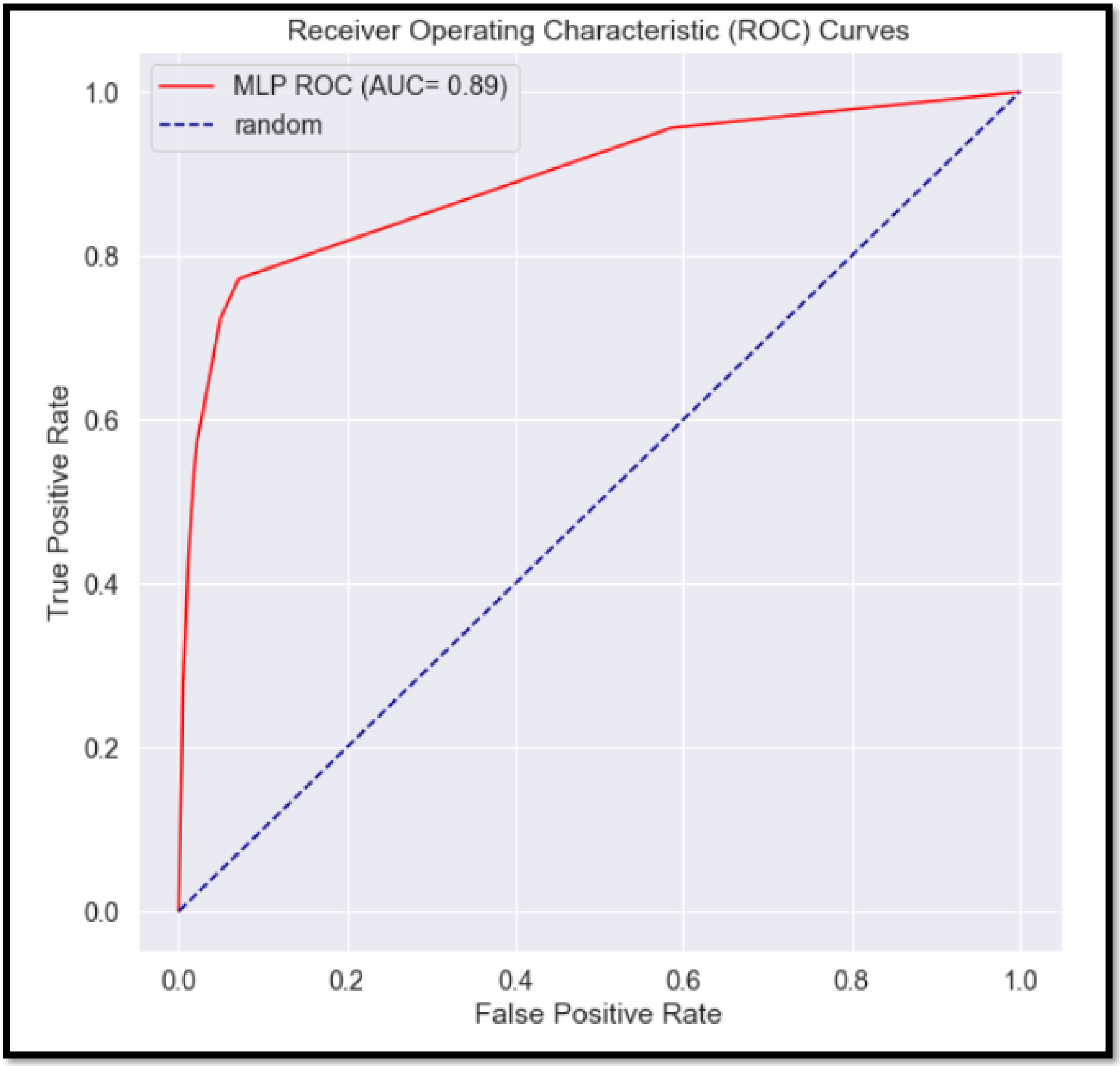


Figure 27. ROC curve of Decision tree. Data source: https://www.lendingclub.com/. Own visualization.

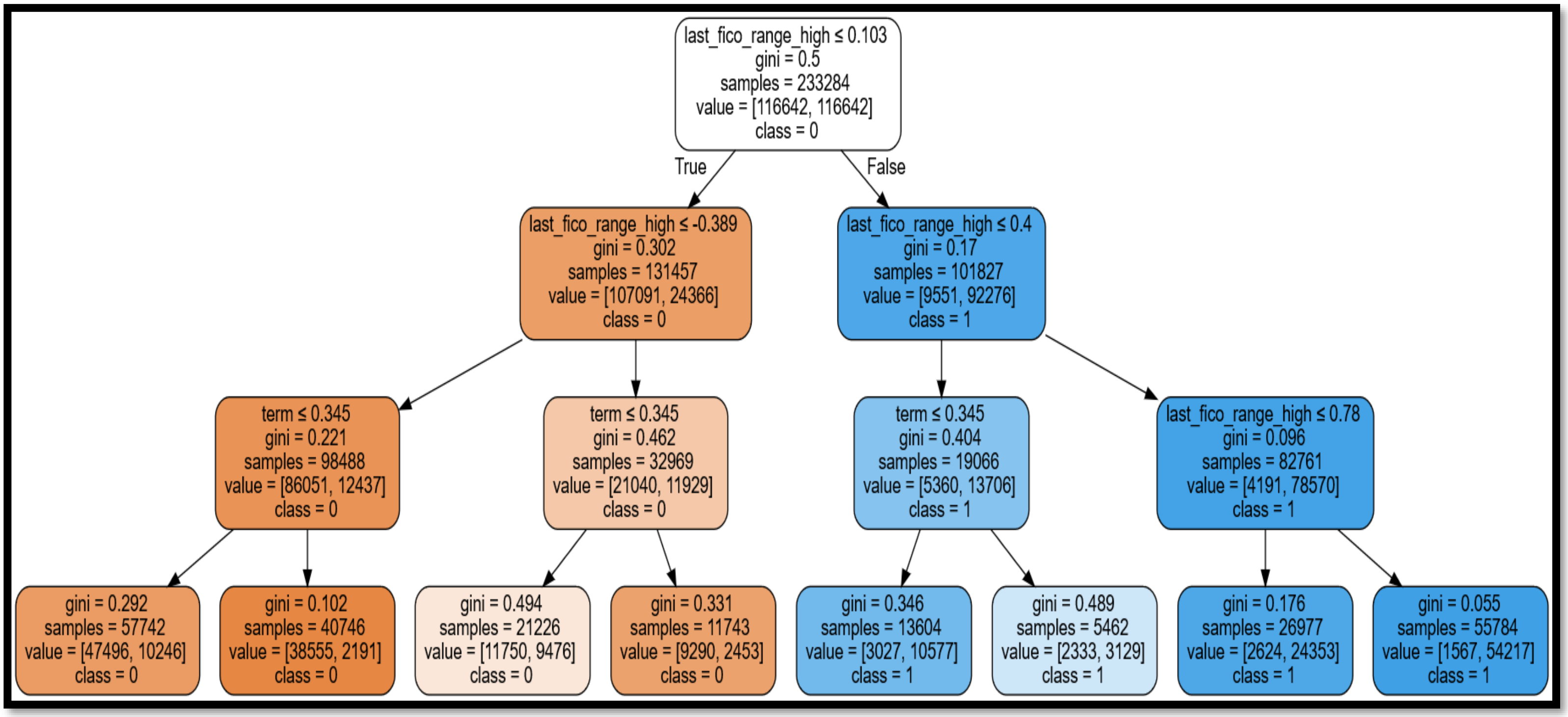


Figure 28. Pruned decision tree. Data source: https://www.lendingclub.com/. Own visualization.

6.1.3. Neural network

Based on the confusion matrix (Figure 29) the neural network model was able to predict 37,853 of defaulters correctly out of 38, 892. This shows that the model can predict 99% defaulters correctly. On the other side the model predicted 110,144 of non-defaulters correctly out of 173,080, this tells the model can correctly predict 38.2% of non-defaulter correctly.

Based on the confusion matrix the accuracy, precision, recall and the F1 score the neural network model is presented discussed.

Accuracy: The accuracy of the neural network model is 0.698. Meaning that the model is 69% accurate overall.

Cross validation accuracy: Is 0.86.

Precision: The precision of neural network model is 0.991. Which means, from the non-defaulters predicted by the model, 99.1% of them are truly non-defaulters.

Recall: The computed recall was 0.636. This tells that the neural network model developed predicted only 62% of non-defaulters correctly.

F1 score: The harmonic mean of precision and recall for neural network is 0.775, signifying moderate quality of the model

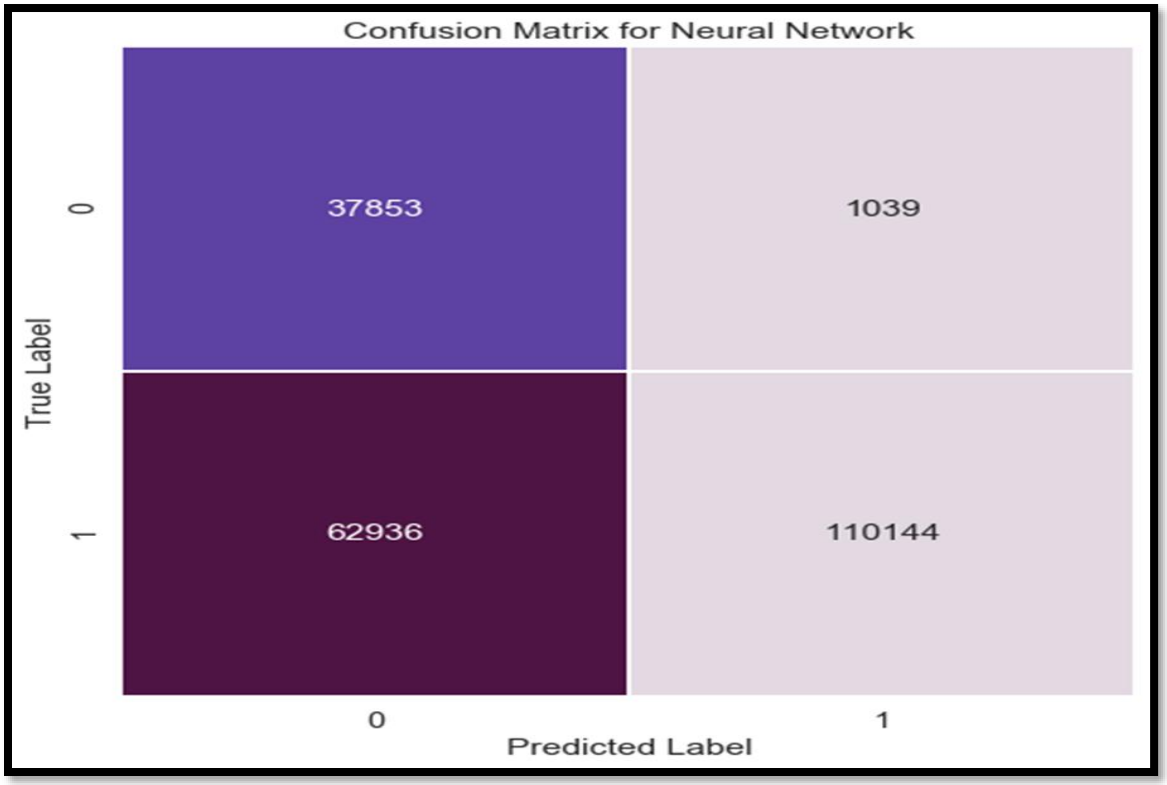


Figure 29. Confusion matrix of Neural network. Data source: https://www.lendingclub.com/. Own visualization.

ROC curve: The decision tree model developed also have quiet good ROC curve (Figure 30), which hanged to the left top of the plot. And the calculated AUC for the decision tree is 0.92. This indicates that the model has 87% probability to distinguish between defaulters and non-defaulters.

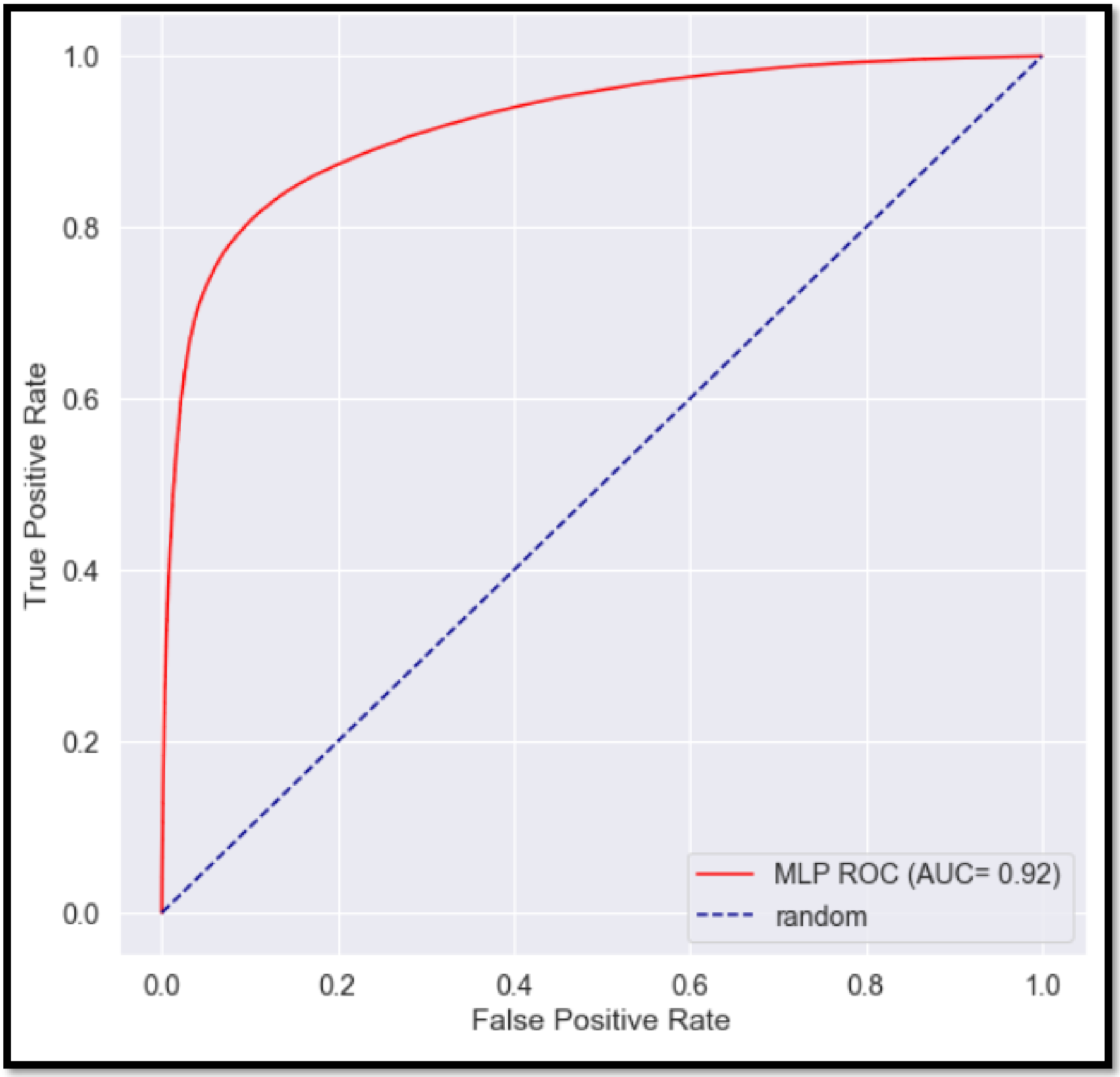


Figure 30. ROC curve of Neural network. Data source: https://www.lendingclub.com/. Own visualization.

6.2.Comparison of models built

Based on table 3 logistic regression had the highest value of most common quality measures in machine learning (accuracy, CV accuracy, recall and F1 score), while decision tree have the highest the precision and neural network have the highest ROC AUC.

	Accuracy	CV accuracy	Precision	Recall	F1 score	ROC AUC
Logistic regression	0.749	0.881	0.987	0.702	0.820	0.924
Decision tree	0.648	0.855	0.992	0.573	0.727	0.892
Neural network	0.698	0.862	0.991	0.636	0.775	0.925

Table 5. Summary of the quality measure for all three models developed. Data source: https://www.lendingclub.com/. Own visualization.

When we come to the average rank (table 4) logistic regression is ranked first with the average rank of 1.5, neural network ranked second with the average rank of 1.84 and decision tree the list with the average rank of 2.67. However, when we look to the specific case of precision of predicting defaulters logistic regression is the worst with the prediction capability of defaulter only 34% while decision tree 37.6% and neural network 38.2%.

	Accuracy	**CV accuracy**	**Precision**	**Recall**	**F1 score**	**ROC AUC**	**Average score**
Logistic regression	1	1	3	1	1	2	1.5
Decision tree	3	3	1	3	3	3	2.67
Neural network	2	2	2	2	2	1	1.84

Table 6. Rank of models based on each quality measures. Data source: https://www.lendingclub.com/. Own visualization.

7. Conclusion

Now a day's python is becoming very powerful and taking power over both the traditional way of data analytics and R programming. Python gives a relief for handling big data like we used in this study. The size of the data used in this study ranges over 800, 000 with more than 100 features. Handling this all data with tools like SPSS, statistica or excel for simple data visualization or modeling is unthinkable and python did this all without any trouble.

The main aim was this study was to show the implementation of machine learning by using python in the credit scoring. In this thesis data of borrowers from Lending club in the year 2007 to 2015 was downloaded from Lendingclub data repository. Data was imported to Jupiter notebook, where all codes was written. The imported data was checked for missing values, outliers and for any possible duplicate in the observation.

Missing values were dropped or filled with mean and median per respective features accordingly. In the case of outliers each outlier were capped in the case of continuous features like annual income and some possible combining or merging of features have been implemented in the case of categorical feature. Features with some possible multicollinearity were dropped from the model and features, with possible relevance selected. To avoid variation in the unit of measurement used for the features and to have normally distributed data standard normalization have been implemented. After all the preprocessing step the data was divided in to training and test set to avoid any bias in the testing of developed model. As there were big difference in the number of defaulters and non-defaulters the balancing of data set was done.

Model was developed for logistic regression, decision tree and neural network. The quality of each model has been checked by the use of confusion metrics and ROC AUC. Based on the output of our quality measure logistic regression outperforms both decision tree and neural network. Therefore, we generally recommend the use of machine learning algorithm for the credit scoring and logistic regression is recommended over decision tree and neural network based our finding. And based on the output of logistic regression the loan amount, loan term, annual income, borrower's total debt payment over obligation, inquiries last 6 months, revolving utilities, total account and last fico range high are the most important power full predictor of defaulters.

8. References

Abbott, D. Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst. Praha: John Wiley & Sons, Incorporated, 2014. ISBN 9781118727935.

Anderson, R. The credit scoring toolkit: theory and practice for retail credit risk management and decision automation. Oxford University Press. 2007. ISBN: 9780199226405.

Arya, S., Eckel, C., Wichman, C., 2013. Anatomy of the credit score. J. Econ. Behav. Organ. 95, 175–185. https://doi.org/10.1016/j.jebo.2011.05.005.

Berman, J.J. Data simplification: taming information with open source tools. Morgan Kaufmann. 2016. ISBN 978-0128037812.

Berry, M.J., Linoff, G.S. Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons. 2004. ISBN 0-471-47064-3.

Bhatia, S., Sharma, P., Burman, R., Hazari, S., Hande, R., 2017. Credit scoring using machine learning techniques. Int. J. Computer. Appl. 161, 1–4.

Bitton, D., DeWitt, D.J., 1983. Duplicate Record Elimination in Large Data Files. ACM Trans Database System 8, 255–265. https://doi.org/10.1145/319983.319987.

Pete, C., Julian, C., Randy, K., Thomas, K., Thomas, R., Colin, S., Rudiger, W., 2000. CRISP-DM 1.0. The Modelling Agency. [Online] 2000. https://the-modeling-agency.com/crisp-dm.pdf

Dangeti, P. Statistics for machine learning. Packt Publishing Ltd. 2017. ISBN: 9781788295758.

Dong, G., Lai, K.K., Yen, J., 2010. Credit scorecard based on logistic regression with random coefficients. Procedia Computer Science, ICCS 2010 1, 2463–2468. https://doi.org/10.1016/j.procs.2010.04.278.

Hand, D.J., Hand, P. in the D. of S.D.J., Mannila, H., Smyth, P. Principles of Data Mining. MIT Press. 2001. ISBN 0-262-08290-X.

Henley, W.E., 1994. Statistical aspects of credit scoring. (PhD Thesis). Open University. http://oro.open.ac.uk/id/eprint/57441

Hernández, M., 1996. A Generalization of Band Joints and the Merge/-Purge Problem. Band Ph. D (PhD Thesis). thesis. Columbia University. https://academiccommons.columbia.edu/doi/10.7916/D8FB5B3W.

Hernández, M.A., Stolfo, S.J., 1995. The merge/purge problem for large databases, in: ACM Sigmod Record. ACM, pp. 127–138.

Hilbe, J.M. Practical guide to logistic regression. CRC Press. 2016. ISBN 9781498709576.

Igual, L., Seguí, S., 2017. Introduction to Data Science: A Python Approach to Concepts, Techniques and Applications, Undergraduate Topics in Computer Science. Springer International Publishing.

Lee, M.L., Lu, H., Ling, T.W., Ko, Y.T., 1999. Cleansing data for mining and warehousing, in: International Conference on Database and Expert Systems Applications. Springer, pp. 751–760.

Louzada, F., Ara, A., Fernandes, G.B., 2016. Classification methods applied to credit scoring: Systematic review and overall comparison. Surv. Oper. Res. Manag. Sci. 21, 117–134. https://doi.org/10.1016/j.sorms.2016.10.001

Lup Low, W., Li Lee, M., Wang Ling, T., 2001. A knowledge-based approach for duplicate elimination in data cleaning. Inf. Syst., Data Extraction,Cleaning and Reconciliation 26, 585–606. https://doi.org/10.1016/S0306-4379(01)00041-2

Maldonado, S., Peters, G., Weber, R., 2020. Credit scoring using three-way decisions with probabilistic rough sets. Inf. Sci. 507, 700–714. https://doi.org/10.1016/j.ins.2018.08.001

Müller, A.C., Guido, S. Introduction to machine learning with Python: a guide for data scientists. O'Reilly Media, Sebastopol, CA. 2017. ISBN 978-1-449-36941-5.

Raschka, S. Python machine learning. Packt Publishing Ltd. 2015. ISBN 978-1787125933

Rebala, G., Ravi, A., Churiwala, S. An Introduction to Machine Learning. Springer. 2019. 978-3-030-15729-6.

Sarkar, D., Bali, R., Sharma, T. Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems. Apress. 2017. ISBN 978-1-4842-3207-1.

Sumathi, S., Sivanandam, S.N. Introduction to data mining and its applications. Springer. 2006. ISBN 978-3-540-34350-9.

Thomas, L., Crook, J., Edelman, D. Credit scoring and its applications. Siam. 2017. ISBN 9781611974560 1611974569.

Tufféry, S. Data mining and statistics for decision making. John Wiley & Sons. 2011. ISBN: 978-0-470-68829-8.

Yadav, M.L., Roychoudhury, B., 2018. Handling missing values: A study of popular imputation packages in R. Knowl.-Based Syst. 160, 104–118. https://doi.org/10.1016/j.knosys.2018.06.012

Yap, B.W., Ong, S.H., Husain, N.H.M., 2011. Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Syst. Appl. 38, 13274–13283. https://doi.org/10.1016/j.eswa.2011.04.147

Appendix 1. The overall features in the dataset with the description

BrowseNotesFile	**Description**
acceptD	The date which the borrower accepted the offer
accNowDelinq	The number of accounts on which the borrower is now delinquent.
accOpenPast24Mths	Number of trades opened in past 24 months.
addrState	The state provided by the borrower in the loan application
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
annualInc	The self-reported annual income provided by the borrower during registration.
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
avg_cur_bal	Average current balance of all accounts
bcOpenToBuy	Total open to buy on revolving bankcards.
bcUtil	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
chargeoff_within_12_mths	Number of charge-offs within 12 months
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
creditPullD	The date LC pulled credit for this loan
delinq2Yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years

delinqAmnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
desc	Loan description provided by the borrower
dti	A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income
earliestCrLine	The date the borrower's earliest reported credit line was opened
effective_int_rate	The effective interest rate is equal to the interest rate on a Note reduced by Lending Club's estimate of the impact of uncollected interest prior to charge off.
emp_title	The job title supplied by the Borrower when applying for the loan.*
empLength	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
expD	The date the listing will expire
expDefaultRate	The expected default rate of the loan.
ficoRangeHigh	The upper boundary range the borrower’s FICO at loan origination belongs to.
ficoRangeLow	The lower boundary range the borrower’s FICO at loan origination belongs to.

fundedAmnt	The total amount committed to that loan at that point in time.
grade	LC assigned loan grade
homeOwnership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
id	A unique LC assigned ID for the loan listing.
ils_exp_d	wholeloan platform expiration date
initialListStatus	The initial listing status of the loan. Possible values are – W, F
inqLast6Mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
intRate	Interest Rate on the loan
isIncV	Indicates if income was verified by LC, not verified, or if the income source was verified
listD	The date which the borrower's application was listed on the platform.
loanAmnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
memberId	A unique LC assigned Id for the borrower member.
mo_sin_old_rev_tl_op	Months since oldest revolving account opened
mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened
mo_sin_rcnt_tl	Months since most recent account opened
mortAcc	Number of mortgage accounts.

msa	Metropolitan Statistical Area of the borrower.
mths_since_last_major_derog	Months since most recent 90-day or worse rating
mths_since_oldest_il_open	Months since oldest bank installment account opened
mthsSinceLastDelinq	The number of months since the borrower's last delinquency.
mthsSinceLastRecord	The number of months since the last public record.
mthsSinceMostRecentInq	Months since most recent inquiry.
mthsSinceRecentBc	Months since most recent bankcard account opened.
mthsSinceRecentLoanDelinq	Months since most recent personal finance delinquency.
mthsSinceRecentRevolDelinq	Months since most recent revolving delinquency.
num_accts_ever_120_pd	Number of accounts ever 120 or more days past due
num_actv_bc_tl	Number of currently active bankcard accounts
num_actv_rev_tl	Number of currently active revolving trades
num_bc_sats	Number of satisfactory bankcard accounts
num_bc_tl	Number of bankcard accounts
num_il_tl	Number of installment accounts
num_op_rev_tl	Number of open revolving accounts
num_rev_accts	Number of revolving accounts
num_rev_tl_bal_gt_0	Number of revolving trades with balance >0
num_sats	Number of satisfactory accounts

num_tl_120dpd_2m	Number of accounts currently 120 days past due (updated in past 2 months)
num_tl_30dpd	Number of accounts currently 30 days past due (updated in past 2 months)
num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in last 24 months
num_tl_op_past_12m	Number of accounts opened in past 12 months
openAcc	The number of open credit lines in the borrower's credit file.
pct_tl_nvr_dlq	Percent of trades never delinquent
percentBcGt75	Percentage of all bankcard accounts > 75% of limit.
pub_rec_bankruptcies	Number of public record bankruptcies
pubRec	Number of derogatory public records
purpose	A category provided by the borrower for the loan request.
reviewStatus	The status of the loan during the listing period. Values: APPROVED, NOT_APPROVED.
reviewStatusD	The date the loan application was reviewed by LC
revolBal	Total credit revolving balance
revolUtil	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
serviceFeeRate	Service fee rate paid by the investor for this loan.
subGrade	LC assigned loan subgrade
tax_liens	Number of tax liens
term	The number of payments on the loan. Values are in months and can be either 36 or 60.

title	The loan title provided by the borrower
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
tot_hi_cred_lim	Total high credit/credit limit
total_il_high_credit_limit	Total installment high credit/credit limit
total_rev_hi_lim	Total revolving high credit/credit limit
totalAcc	The total number of credit lines currently in the borrower's credit file
totalBalExMort	Total credit balance excluding mortgage
totalBcLimit	Total bankcard high credit/credit limit
url	URL for the LC page with listing data.
verified_status_joint	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
open_acc_6m	Number of open trades in last 6 months
open_il_6m	Number of currently active installment trades
open_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
mths_since_rcnt_il	Months since most recent installment accounts opened
total_bal_il	Total current balance of all installment accounts
il_util	Ratio of total current balance to high credit/credit limit on all install acct

open_rv_12m	Number of revolving trades opened in past 12 months
open_rv_24m	Number of revolving trades opened in past 24 months
max_bal_bc	Maximum current balance owed on all revolving accounts
all_util	Balance to credit limit on all trades
inq_fi	Number of personal finance inquiries
total_cu_tl	Number of finance trades
inq_last_12m	Number of credit inquiries in past 12 months