IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁂ Peer-reviewed & Refereed journal ⁂ Vol. 12, Issue 5, May 2024

DOI: 10.17148/IJIREEICE.2024.12502

Machine Learning approach for the Evaluation of loan defaulters

Mr.A.B. Majgave

Asst.Prof, Computer Science and Engineering Department, D.K.T.E. Society's Textile and Engineering Institute, Ichalkaranji

Abstract: In today's world, obtaining loans from financial institutions has become a very common phenomenon. Every day many people apply for loans, for a variety of purposes. But not all the applicants are reliable, and not everyone can be approved. Every year, there are cases where people do not repay the bulk of the loan amount to the bank which results in huge financial loss. The risk associated with making a decision on a loan approval is immense. Hence, the idea of this project is to gather loan data from the Lending Club website and use machine learning techniques on this data to extract important information and predict if a customer would be able to repay the loan or not. In other words, the goal is to predict if the customer would be a defaulter or not.

Keywords: Lending Club, peer-to-peer Lending, EDA, Machine Learning.

I. INTRODUCTION

Peer to peer (P2P) lending is a way to borrow without using a traditional bank or credit union. For applicants with a good credit score (often a FICO credit score higher than 720), P2P loan rates can be surprisingly low. With less-than-perfect credit, an applicant still has a decent shot at being approved for an affordable loan with online lenders like Lending Club. P2P loans are loans made by individuals and investors – as opposed to loans that come from a bank. People with extra funds offer to lend that money to others (individuals and businesses) in need of cash. A P2P service (such as a website) matches lenders and borrowers so that the process is relatively easy for all involved. Loan default prediction is a common problem for such lending companies. This is the type of problem banks and credit card companies face whenever customers ask for a loan. This thesis focusses on using the Lending Club dataset which is freely available on their website. The objective is to make predictions about loan default and whether investors should lend to a customer or not. Data from 2007-2015 will be used because most of the loans from that period have already been repaid or defaulted on. Lending Club is the platform, or rather the marketplace, where investors and borrowers meet virtually. The Lending Club processes the application with their own data science methods. However, on the side of the investor, there is nothing to ensure the creditworthiness of the borrower and the level of risk involved in any given case. Applying machine learning to loan default predictions, showcases a useful application of this branch of artificial intelligence to solve real-world and business problems.

II. PROBLEM STATEMENT

If a model can identify credit-worthy customers that were not recognized by traditional credit scores, while minimizing their risk of default on the loans, this can be a lucrative niche market or micro-market, pushing higher the profit margin of the financial institution or investor. Although the prospect of more customers seems positive, it is important to be careful as to not lend to people that will default on the loan. Thus, a conservative approach and strict evaluation metrics were kept in mind throughout the project. The loan default prediction is a problem of binary classification (should the investor lend or not). Logistic Regression is a good model for this problem.

III. DATASET

The dataset was downloaded from a website called Kaggle. Kaggle has a collection of high quality public datasets. This dataset was verified with the dataset available on Lending Club's website. The Data Dictionary used for the project was downloaded from the Lending Club's website. The dataset consists of all accepted loan applications from 2007-2015. It has 74 features and 887379 applications. Such a huge dataset was helpful for my task. The following images are a part of the dataset.

Data Cleaning and Preprocessing

The dataset has 887379 rows and 74 columns. The columns represent different information gathered as part of the first inquiry by Lending Club. The data dictionary file provided with the dataset, indicates that columns are information about the borrower and the outcome of their loan repayment. Data from 2007-2015 was chosen because of the almost certainty that the loans have been repaid or defaulted on by now.

The first issue was to know if the columns were filled with useful information or were mostly empty. Data exploration uncovered many empty or almost empty columns which were removed from the dataset because it would prove a difficult task to go back and try to answer for each data point that did not seem necessary at the time of the loan application. Columns linking to the user's profile (with an URL) and a description (given by the customer) of the demand were removed because they were mostly filled with text data.

The columns that had more than 40% of missing values were also removed. This was done to free up space and make the processing faster. Fields including "recoveries" and "collection_recovery_fee" are data about the future about the loan. Fields including "last_pymnt_d" and "last_pmyny_amnt" describe the ending date of repayment, which are not possible to know in advance due to the fact that the customer may pay off the loan earlier than the original term.

The following five variables were all about the future of the loan, informing about how the repayment is proceeding: "out_prncp", "out_prncp_inv", "total_pymnt", "total_pymnt_inv" and "total_rec_prncp" . Hence, they were removed because such information would not be available to the investor.

The "total_rec_int" variable describes the interest received to date (meaning the loan has been approved) and "total_rec_late_fee" describes the late interest. These were not needed because such information would not be available to the investor. The variable "issue_d" is data about the month when the loan was funded. This means it reveals a future information. Hence, it was removed because such information would not be available to the investor.

The "zip_code" column did not add any value because that already existed in the state address contained in "addr_state". The variable "zip_code" could be used with other economic data to uncover a relationship with the environment in which a person lives and the risk of default. In addition, only the first 3 digits of the "zip_code" variable were present. The "id" and "member_id" features were removed because they did not provide any useful information about the customer. These were random features given by Lending Club.

The "funded_amnt" and "funded_amnt_inv" features were both concerns about the future, whether the loan has been approved at that point, and thus were not considered in the model. "Grade" and "sub_grade" were recurring data that were already included in the "int_rate" feature. Thus, they were removed as well.

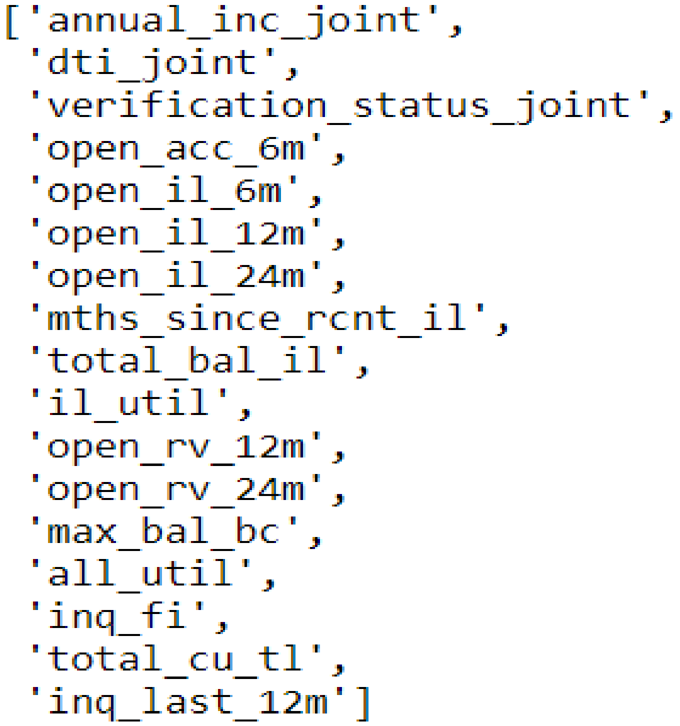
Although it could have been an area of improvement in the model, the "emp_title" feature would have been a hard feature to evaluate. Some form of sentiment analysis might be required, and certain metrics would need to provide a good estimate of a title's meaning and value in the lending context.

The process of data cleaning was executed in the following manner:

Step 1: Decided the target of the model
The target of the algorithm to be predicted was decided, namely the "loan_status" column. The loan status indicates whether the lender repays the loan in full or not.

Step 2: Dropped features that had only 1 distinct value
The features that have only one distinct value were dropped since they weren't useful for the task. Thus, the feature "policy_code" was dropped.

Step 3: Removed features that contained less than 5% of data
The features that had less than 5% of data were removed since they weren't helpful in creating a good model. Thus, the following features were dropped:



```
['annual_inc_joint',
 'dti_joint',
 'verification_status_joint',
 'open_acc_6m',
 'open_il_6m',
 'open_il_12m',
 'open_il_24m',
 'mths_since_rcnt_il',
 'total_bal_il',
 'il_util',
 'open_rv_12m',
 'open_rv_24m',
 'max_bal_bc',
 'all_util',
 'inq_fi',
 'total_cu_tl',
 'inq_last_12m']
```

This step left 887379 rows and 56 columns(features) remaining.

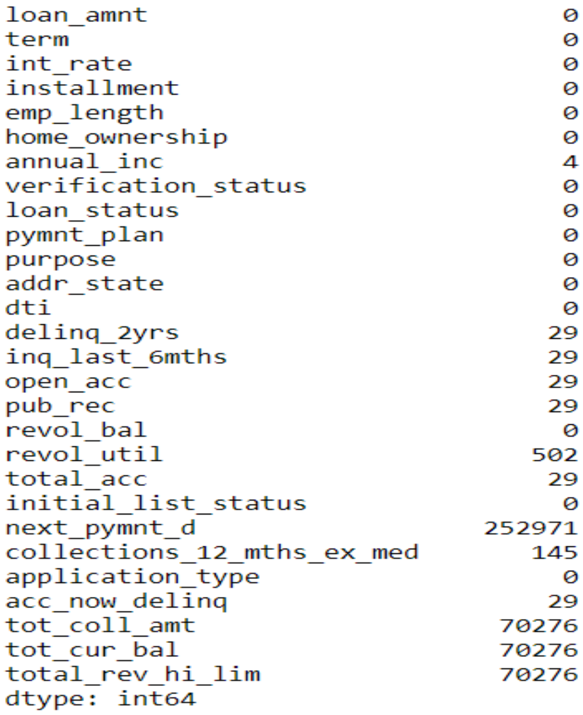
Step 4: Dropped features that were irrelevant for the goal.
The features that were irrelevant for the goal were dropped. The following features were dropped:
“id”, “url”, “member_id”, “zip_code”, “desc”, “emp_title”, “title”, “issue_d”, “last_credit_pull_d”, “earliest_cr_line”.

This left 46 remaining columns:
(['loan_amnt', 'funded_amnt', 'funded_amnt_inv', 'term', 'int_rate', 'installment', 'grade', 'sub_grade', 'emp_length', 'home_ownership', 'annual_inc', 'verification_status', 'loan_status', 'pymnt_plan','purpose', 'addr_state', 'dti', 'delinq_2yrs', 'inq_last_6mths', 'mths_since_last_delinq', 'mths_since_last_record', 'open_acc','pub_rec', 'revol_bal', 'revol_util', 'total_acc', 'initial_list_status', 'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'recoveries', 'collection_recovery_fee', 'last_pymnt_d', 'last_pymnt_amnt', 'next_pymnt_d','collections_12_mths_ex_med', 'mths_since_last_major_derog', 'application_type', 'acc_now_delinq', 'tot_coll_amt', 'tot_cur_bal', 'total_rev_hi_lim'].

Step 5: Removed features that could have caused data leakages.
The following features were removed the following features because they could have caused leakage of data.
'last_pymnt_d', 'last_pymnt_amnt','recoveries', 'collection_recovery_fee', 'out_prncp', 'out_prncp_inv', 'total_pymnt', 'total_pymnt_inv', 'total_rec_prncp', 'total_rec_int', 'total_rec_late_fee', 'funded_amnt', 'funded_amnt_inv'.
This step left 887379 rows and 33 columns(features) remaining.

Step 6: Grouped features that conveyed the same meaning.
The features “grade” and “sub_grade” were removed because they conveyed the same meaning as interest rate(“int_rate”).

Step 7: Removed columns that had more than 40% null values.
Three columns were removed.
The following features remained. The following image shows the count of null values in these features

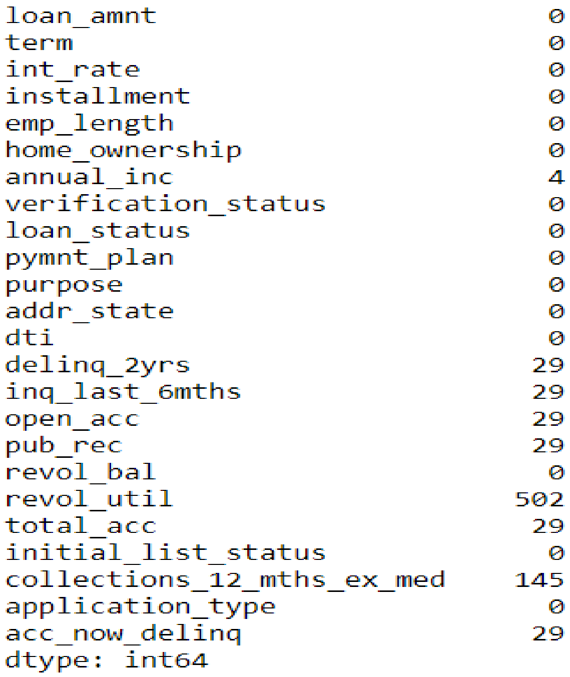


```
loan_amnt                        0
term                             0
int_rate                         0
installment                      0
emp_length                       0
home_ownership                   0
annual_inc                       4
verification_status              0
loan_status                      0
pymnt_plan                       0
purpose                          0
addr_state                       0
dti                              0
delinq_2yrs                     29
inq_last_6mths                  29
open_acc                        29
pub_rec                         29
revol_bal                        0
revol_util                     502
total_acc                       29
initial_list_status              0
next_pymnt_d                252971
collections_12_mths_ex_med     145
application_type                 0
acc_now_delinq                  29
tot_coll_amt                 70276
tot_cur_bal                  70276
total_rev_hi_lim             70276
dtype: int64
```

Fig. 3.2: List of count of null values in the features

Step 8: Removed features with most null values
The above image shows “next_pymnt_d”, “tot_coll_amt”, “tot_cur_bal” and ”total_rev_hi_lim” have numerous null values. Hence, these features were dropped.
Step 9: Removed all rows that had null values

The following image shows the count of null values in the features after executing step 8.



```
loan_amnt                      0
term                           0
int_rate                       0
installment                    0
emp_length                     0
home_ownership                 0
annual_inc                     4
verification_status            0
loan_status                    0
pymnt_plan                     0
purpose                        0
addr_state                     0
dti                            0
delinq_2yrs                   29
inq_last_6mths                29
open_acc                      29
pub_rec                       29
revol_bal                      0
revol_util                   502
total_acc                     29
initial_list_status            0
collections_12_mths_ex_med   145
application_type               0
acc_now_delinq                29
dtype: int64
```

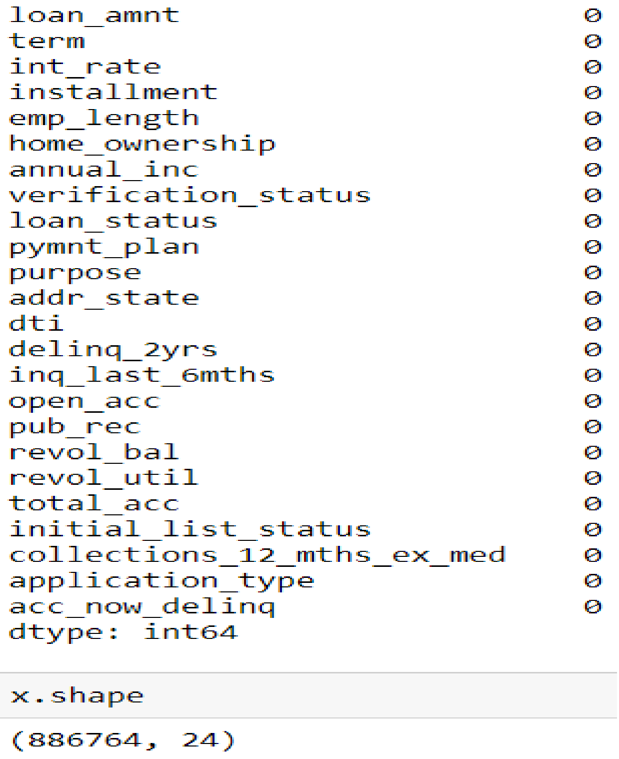
Fig. 3.3: List of count of null values after executing step 8

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.021 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 12, Issue 5, May 2024

DOI: 10.17148/IJIREEICE.2024.12502

All rows that had null values were dropped. The remaining 24 features all had zero null values. This is shown in the next figure



```
loan_amnt                      0
term                           0
int_rate                       0
installment                    0
emp_length                     0
home_ownership                 0
annual_inc                     0
verification_status            0
loan_status                    0
pymnt_plan                     0
purpose                        0
addr_state                     0
dti                            0
delinq_2yrs                    0
inq_last_6mths                 0
open_acc                       0
pub_rec                        0
revol_bal                      0
revol_util                     0
total_acc                      0
initial_list_status            0
collections_12_mths_ex_med     0
application_type               0
acc_now_delinq                 0
dtype: int64

x.shape

(886764, 24)
```

Fig. 3.4: List of features with no null values

Step 10: Rechecked the features
After reviewing all the features again, three of them were dropped namely “addr_state”, “initial_list_status” and “pymnt_plan” because they weren’t that useful for the model.

IV. EXPLORATORY DATA ANALYSIS

The first step was to analyze the total count of loan status types. The majority of loans were under the “Current” category. The “Fully Paid” and “Charged Off” categories were the target for prediction.

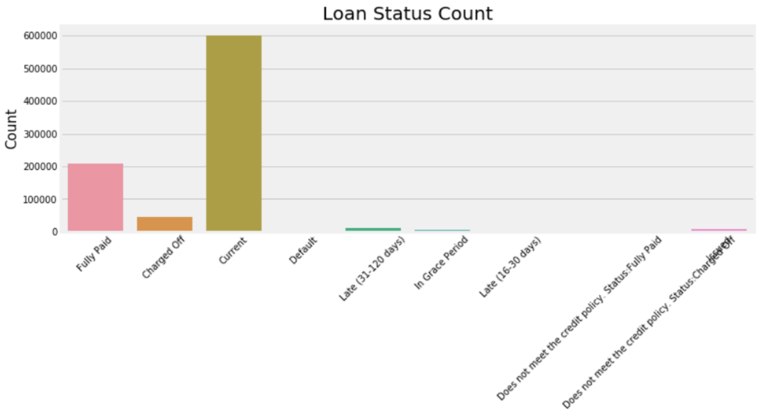


Fig. 4.1: Count of loan status by type

The purpose of the loans and the loan amount were then analyzed. It was observed that the loan amount for debt consolidation was the highest followed by credit card.

DOI: 10.17148/IJIREEICE.2024.12502

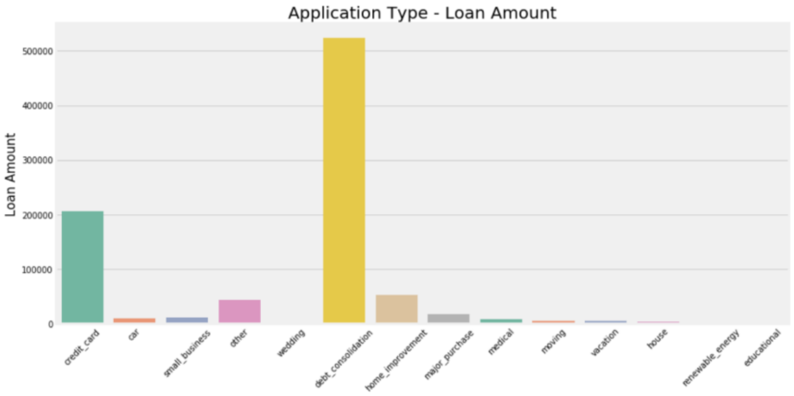


Fig. 4.2: Amount of loan by purpose

The loan status was analyzed by employment length. It was observed that people who were employed for more than 10 years had the highest percent of paying off loans in time. The highest count for defaulters (Charged Off) was observed with the same group.

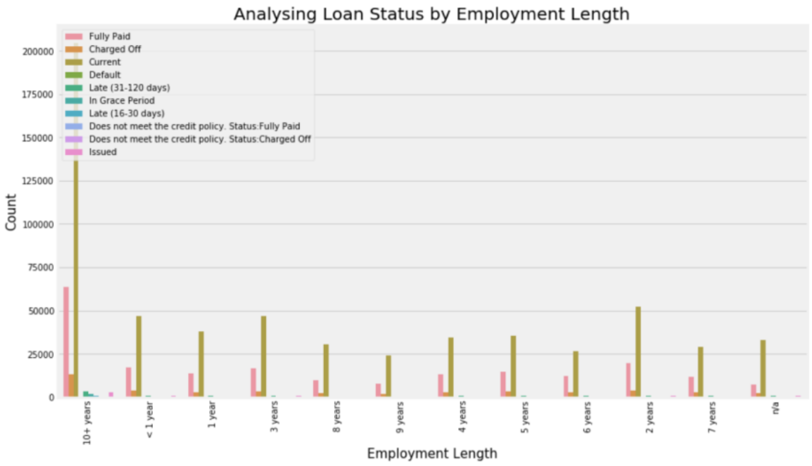
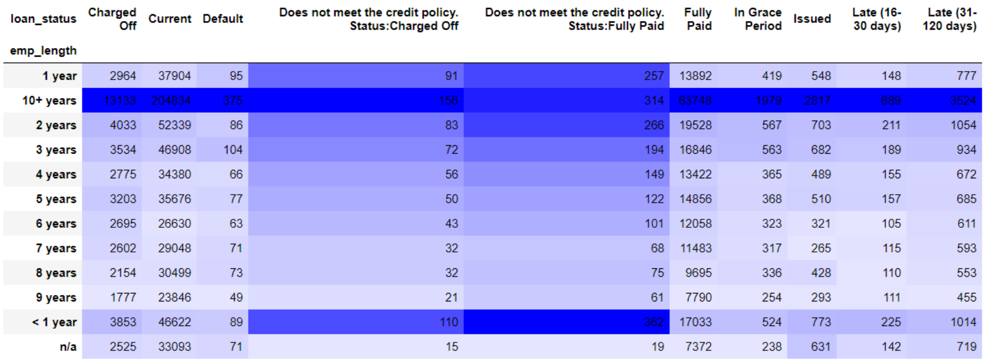


Fig. 4.3: Home ownership and loan amount distribution by application type



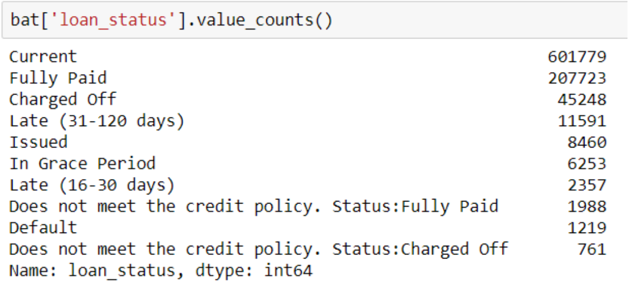
loan_status	Charged Off	Current	Default	Does not meet the credit policy. Status:Charged Off	Does not meet the credit policy. Status:Fully Paid	Fully Paid	In Grace Period	Issued	Late (16-30 days)	Late (31-120 days)
emp_length										
1 year	2964	37904	95	91	257	13892	419	548	148	777
10+ years	13133	204834	375	156	314	63748	1979	2817	689	3524
2 years	4033	52339	86	83	266	19528	567	703	211	1054
3 years	3534	46908	104	72	194	16846	563	682	189	934
4 years	2775	34380	66	56	149	13422	365	489	155	672
5 years	3203	35676	77	50	122	14856	368	510	157	685
6 years	2695	26630	63	43	101	12058	323	321	105	611
7 years	2602	29048	71	32	68	11483	317	265	115	593
8 years	2154	30499	73	32	75	9695	336	428	110	553
9 years	1777	23846	49	21	61	7790	254	293	111	455
< 1 year	3853	46622	89	110	362	17033	524	773	225	1014
n/a	2525	33093	71	15	19	7372	238	631	142	719

Fig. 4.4: Loan status by application type

The loan status was analyzed by home ownership. It was observed that most people who were charged off had either rented or mortgaged their home. Also, the majority of people who fully paid their loans either rented or mortgaged their home. There were less people who owned their house. Among them, the ratio for fully paid vs. charge off was close to approximately 4:1.

V. FEATURE ENGINEERING

These are the various types of loan status



```
bat['loan_status'].value_counts()

Current                                                601779
Fully Paid                                             207723
Charged Off                                             45248
Late (31-120 days)                                      11591
Issued                                                   8460
In Grace Period                                          6253
Late (16-30 days)                                        2357
Does not meet the credit policy. Status:Fully Paid       1988
Default                                                  1219
Does not meet the credit policy. Status:Charged Off       761
Name: loan_status, dtype: int64
```

Fig. 5.1: List of count of loan status

All the rows containing loan status as “Issued” were removed because they didn’t indicate whether the loan was repaid or not. This was not favorable for the model. The goal was to predict whether the applicant will pay off the loan or not. Therefore, all the loan status types except “Fully Paid” and “Charged Off” were discarded. The “Default” status loan type was not taken into consideration because there were very less default cases. “Charged Off” signifies the applicant would most likely not pay the loan/ default, and “Fully Paid” would mean the applicant would most likely pay the loan in time.

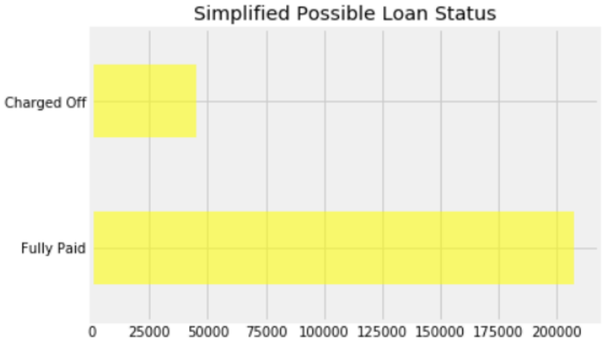


Fig. 5.2: Count of simplified loan status

The loan status was analyzed against different features.
The following images were the results:

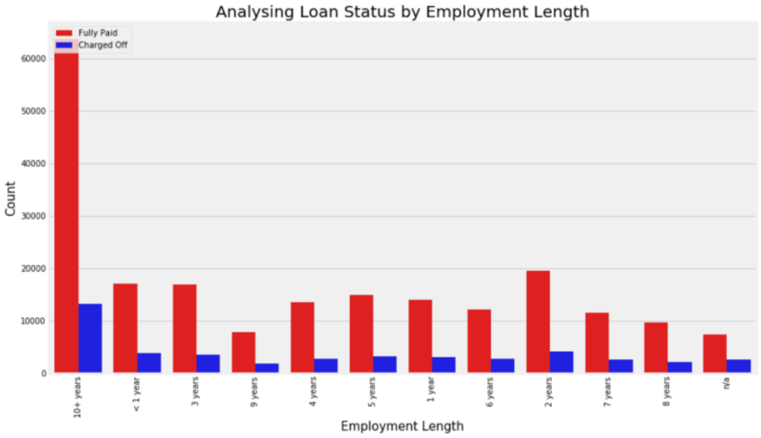
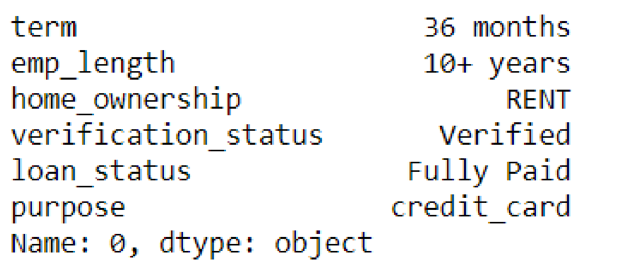


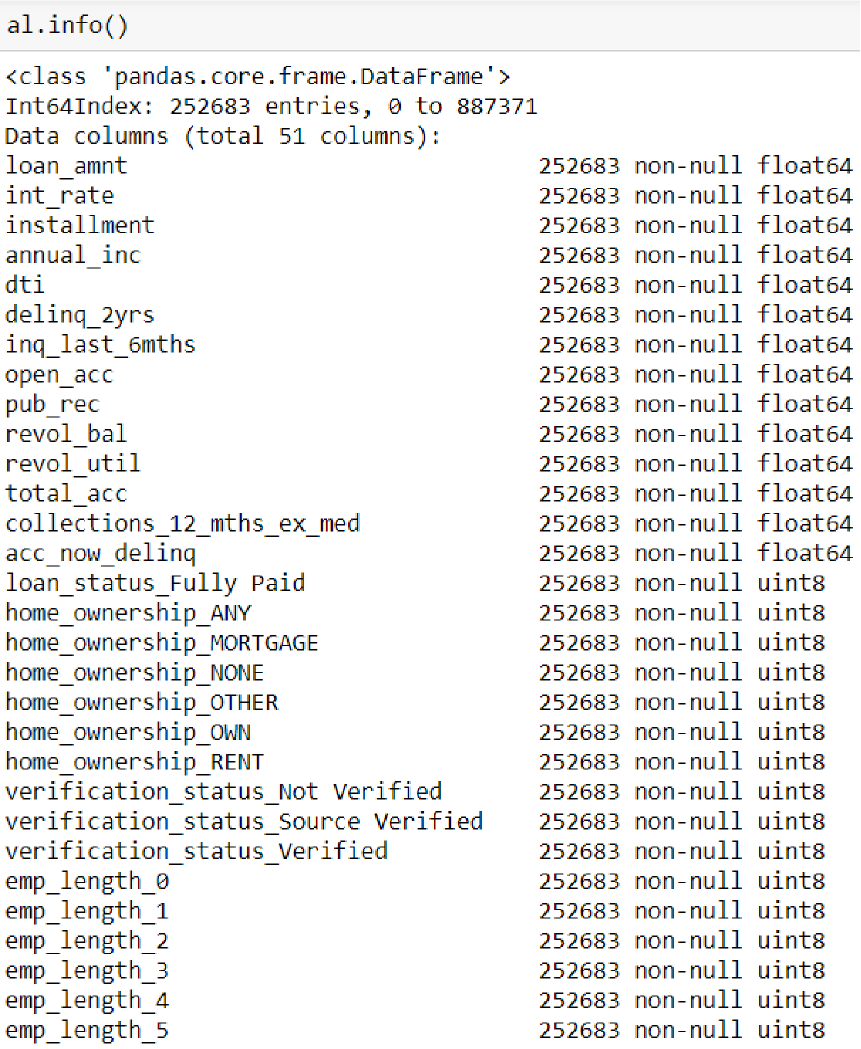
Fig. 5.3: Count of simplified loan status by employment length



```
term                         36 months
emp_length                   10+ years
home_ownership                    RENT
verification_status           Verified
loan_status                 Fully Paid
purpose                    credit_card
Name: 0, dtype: object
```

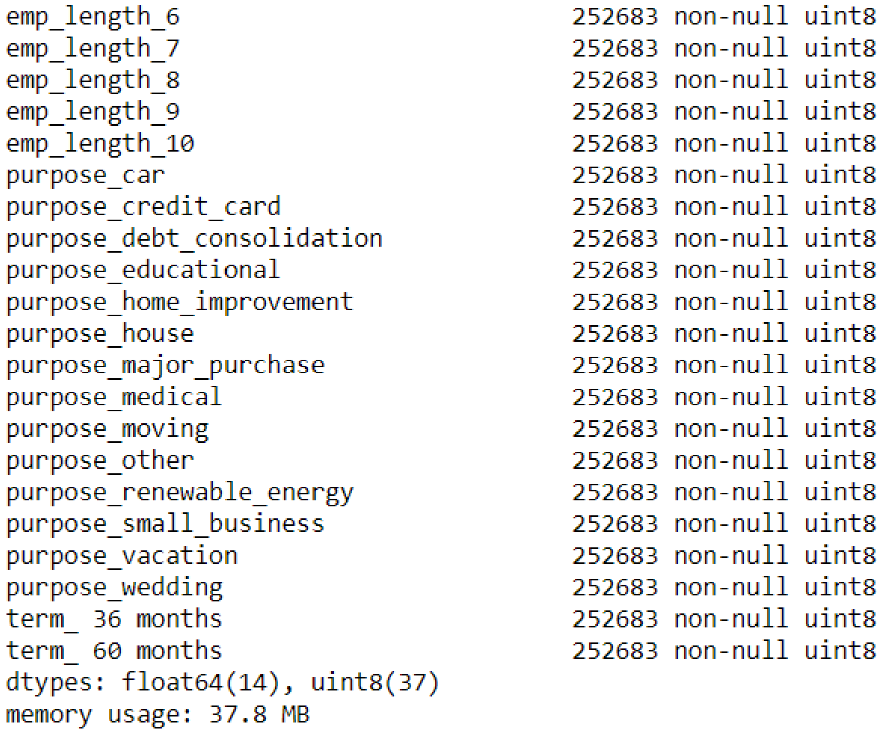
Fig. 5.4: List of categorical features

The above figure shows features that have non-numerical values. In order to use these features for the model, they were converted into numeric data types, using dummy variables. The “Loan status” variable was converted into “loan_status_Fully Paid” using dummy variables "1” and “0”. The value “1” indicates that the loan was fully paid and “0” indicates that the loan was charged off. By creating variables “1” and “0”, new columns were created which had the same meaning as the previous columns with the exception that their data type was changed. The following image shows the final set of features. There are 51 features and 252683 entries.



```
al.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 252683 entries, 0 to 887371
Data columns (total 51 columns):
loan_amnt                          252683 non-null float64
int_rate                           252683 non-null float64
installment                        252683 non-null float64
annual_inc                         252683 non-null float64
dti                                252683 non-null float64
delinq_2yrs                        252683 non-null float64
inq_last_6mths                     252683 non-null float64
open_acc                           252683 non-null float64
pub_rec                            252683 non-null float64
revol_bal                          252683 non-null float64
revol_util                         252683 non-null float64
total_acc                          252683 non-null float64
collections_12_mths_ex_med         252683 non-null float64
acc_now_delinq                     252683 non-null float64
loan_status_Fully Paid             252683 non-null uint8
home_ownership_ANY                 252683 non-null uint8
home_ownership_MORTGAGE            252683 non-null uint8
home_ownership_NONE                252683 non-null uint8
home_ownership_OTHER               252683 non-null uint8
home_ownership_OWN                 252683 non-null uint8
home_ownership_RENT                252683 non-null uint8
verification_status_Not Verified   252683 non-null uint8
verification_status_Source Verified 252683 non-null uint8
verification_status_Verified       252683 non-null uint8
emp_length_0                       252683 non-null uint8
emp_length_1                       252683 non-null uint8
emp_length_2                       252683 non-null uint8
emp_length_3                       252683 non-null uint8
emp_length_4                       252683 non-null uint8
emp_length_5                       252683 non-null uint8
```



```
emp_length_6                      252683 non-null uint8
emp_length_7                      252683 non-null uint8
emp_length_8                      252683 non-null uint8
emp_length_9                      252683 non-null uint8
emp_length_10                     252683 non-null uint8
purpose_car                       252683 non-null uint8
purpose_credit_card               252683 non-null uint8
purpose_debt_consolidation        252683 non-null uint8
purpose_educational               252683 non-null uint8
purpose_home_improvement          252683 non-null uint8
purpose_house                     252683 non-null uint8
purpose_major_purchase            252683 non-null uint8
purpose_medical                   252683 non-null uint8
purpose_moving                    252683 non-null uint8
purpose_other                     252683 non-null uint8
purpose_renewable_energy          252683 non-null uint8
purpose_small_business            252683 non-null uint8
purpose_vacation                  252683 non-null uint8
purpose_wedding                   252683 non-null uint8
term_ 36 months                   252683 non-null uint8
term_ 60 months                   252683 non-null uint8
dtypes: float64(14), uint8(37)
memory usage: 37.8 MB
```

Fig. 5.5: List of input features for the model

VI. SELECTING THE MODEL

The model was then selected for the prediction which was based on logistic regression. Logistic regression, sometimes called the logistic model or logit model, analyzes the relationship between multiple independent variables and a categorical dependent variable, and estimates the probability of occurrence of an event by fitting data to a logistic curve. There are two models of logistic regression: i) binary logistic regression and ii) multinomial logistic regression. Binary logistic regression is typically used when the dependent variable is dichotomous, and the independent variables are either continuous or categorical. When the dependent variable is not dichotomous and is comprised of more than two categories, a multinomial logistic regression can be employed.

When selecting the model for logistic regression analysis, another important consideration is the model fit. Adding independent variables to a logistic regression model will always increase the amount of variance. However, adding more and more variables to the model can result in overfitting, which reduces the generalizability of the model beyond the data on which the model is fit.

A classification task involves assigning which feature or label should be assigned to some data, according to some properties of the data. The target variable was “loan_status(Fully Paid)” and the rest of the variables were used for prediction. The dataset was divided into two parts. The entire dataset (252623 entries and 51 columns) was split into two parts: training dataset and testing dataset randomly. The dataset was split into 70% training and 30% testing dataset. The model was developed to fit on the training data and it was tested against the testing data.

Step 1: Checked the true positive rate and the false positive rate of the dataset.

Both, true positive rate and false positive rate were 1.00. The accuracy and precision were 0.82.

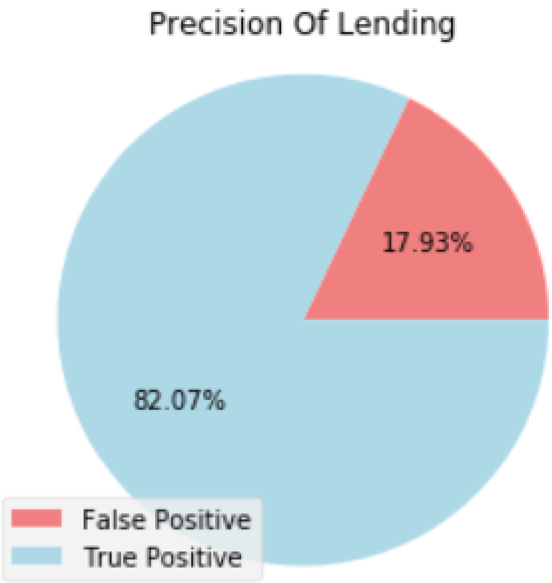
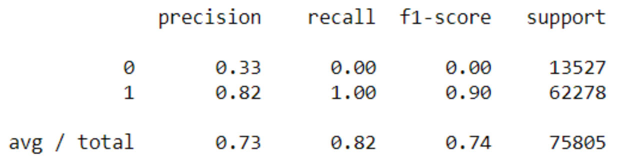


Fig. 6.1: Pie chart of True positive rate vs false positive rate

The imbalance in the target category of loan repayment in the dataset, was due to the fact that 82 out of 100 loans were repaid. This indicates money could be lent continuously (always predicting that the borrower would repay) and be correct about 82.07% of the time that the loan was repaid. However, that would mean that the model would not be profitable. For example, suppose the investor lends $1000 at 10% interest. Then the investor would expect a return of $100 on each loan. But after running the experiment 100 times, the investor would earn $8200 (82 x $100) and loose $18000 (due to a defaulter) i.e. with a great loss. The benchmark needs to encompass the weight of the defaulter and the optimization between the true positive rate (good borrowers) and the false positive rate (bad borrowers). This implies it is necessary to ensure a viable machine learning model and predict a higher percentage of potential defaulters to avoid lending to them. This results in 100% of true positive loans, but also 100% or the false positive because it was predicted that all the loans would be paid off. Hence, the dataset is imbalanced. The goal was to create a model which surpasses the 82.07% average loan repayment.

Step 2: a) The logistic regression model was used on the training set of 70% and testing set of 30% data from the filtered dataset with no weight changes i.e. with the imbalance.

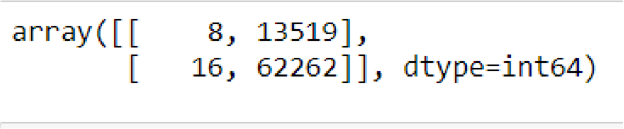
The following images show the classification report and confusion matrix of the result:



```
             precision    recall  f1-score   support

          0       0.33      0.00      0.00     13527
          1       0.82      1.00      0.90     62278

avg / total       0.73      0.82      0.74     75805
```



```
array([[    8, 13519],
       [   16, 62262]], dtype=int64)
```

Since an abnormally high number was obtained, the model was still predicting that all the loans will be paid off. Thus, weight was added in step 3.

b) Overfitting is the production of an analysis that corresponds too closely or exactly to a particular set of data and may therefore fail to fit additional data or predict future observations reliably. The essence of overfitting is to have unknowingly extracted some of the residual variation (i.e. the noise) as if that variation represented underlying model structure.

DOI: 10.17148/IJIREEICE.2024.12502

Two techniques were used to reduce overfitting namely regularization and cross validation. Regularization is a process of introducing additional information in order to solve an ill-posed problem or to prevent overfitting. Regularization basically adds the penalty as model complexity increases. L2 Regression (Ridge Regression) was used to reduce overfitting.

Cross-validation is a technique to evaluate predictive models by partitioning the original sample into a training set to train the model, and a test set to evaluate it. In K-fold cross-validation, the original sample is randomly partitioned into K equal size subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K-1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The k results from the folds can then be averaged (or otherwise combined) to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once.

VII. RESULTS

Through experiments, the model was found which best suits the dataset and serves the purpose of giving an investor a model which would increase their chances of a profit. It had an accuracy of 0.35 and a precision of 0.93. The investor might pass on a lot of loan opportunities, but there are very less chances of losing money.

5.2 Summary

At the start, the dataset was cleaned. Then exploratory data analysis and feature engineering were performed. Then a model was created which predicted whether the applicant would repay the loan or not.

5.3 Learning Experience

The project development provided me with a sense of new technologies that I was not familiar with at the beginning of this project. I learned to work with Jupyter notebooks and use different Python libraries. Plus, I understood the concepts of Machine Learning by building models.

5.4 Future Enhancements

Different machine learning techniques (Random Forests, Neural Networks etc.) can be implemented and compared to get better results.

REFERENCES

[1]. https://www.lendingclub.com/info/download-data.action
[2]. https://tryolabs.com/blog/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/
[3]. http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/
[4]. https://machinelearningmastery.com/logistic-regression-for-machine-learning/
[5]. http://scikitlearn.org/stable/modules/generated/sklearn.model_selection.KFold.html
[6]. http://scikitlearn.org/stable/modules/generated/sklearn.model_selection.cross_val_predict.html
[7]. http://scikitlearn.org/stable/modules/classes.html#module-sklearn.metrics
[8]. https://www.lendingclub.com/public/credit-score-
[9]. https://www.openml.org/a/estimation-procedures/1