CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Predicting Loan Defaults using Machine Learning Techniques

A thesis submitted in partial fulfillment of the requirements

For the degree of Master of Science in Computer Science

by

Abhishek Bhagat

MAY 2018

The thesis of Abhishek Bhagat is approved by:

_____	_____
Dr. Kyle Dewey	Date
_____	_____
Dr. Taehyung Wang	Date
_____	_____
Dr. Robert McIlhenny, Chair	Date

California State University, Northridge

Acknowledgements

I would like to express my heartfelt gratitude to my advisor Dr. Robert McIlhenny for the continuous support of my Master's Study and Research, for his patience, motivation, enthusiasm and immense knowledge.

I would also like to thank my thesis committee members Dr. George Wang and Dr. Kyle Dewey for their support and guidance.

Table of Contents

List of Figures

Abstract

Predicting Loan Defaults using Machine Learning Techniques

By

Abhishek Bhagat

Master of Science in Computer Science

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.

Chapter 1: Introduction

1.1 Background

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.

1.2 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.

Chapter 2: 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.

```
In [4]: bat.head()
```

Out[4]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...	total_bal_il	il_util	open_rv_12m	open_
0	1077501	1296599	5000.0	5000.0	4975.0	36 months	10.65	162.87	B	B2	...	NaN	NaN	NaN	
1	1077430	1314167	2500.0	2500.0	2500.0	60 months	15.27	59.83	C	C4	...	NaN	NaN	NaN	
2	1077175	1313524	2400.0	2400.0	2400.0	36 months	15.96	84.33	C	C5	...	NaN	NaN	NaN	
3	1076863	1277178	10000.0	10000.0	10000.0	36 months	13.49	339.31	C	C1	...	NaN	NaN	NaN	
4	1075358	1311748	3000.0	3000.0	3000.0	60 months	12.69	67.79	B	B5	...	NaN	NaN	NaN	

5 rows × 74 columns

Fig. 1: ***First 5 entries in the dataset***

2.1 Features

The following table consists the features of the dataset with their description:

No.	Variable Name	Description
1	id	A unique LC assigned ID for the loan listing.
2	member_id	A unique LC assigned Id for the borrower member.
3	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.
4	funded_amnt	The total amount committed to that loan at that point in time.
5	funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
6	term	The number of payments on the loan. Values are in months and can be either 36 or 60.
7	int_rate	Interest Rate on the loan
8	installment	The monthly payment owed by the borrower if the loan originates.
9	grade	LC assigned loan grade
10	sub_grade	LC assigned loan subgrade
11	emp_title	The job title supplied by the Borrower when applying for the loan.
12	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.

13	home_ownership	The home ownership status provided by the borrower during registration. Our values are: RENT, OWN, MORTGAGE, OTHER.
14	annual_inc	The self-reported annual income provided by the borrower during registration.
15	verification_status	Indicates if the borrowers income was verified by LC, not verified, or if the income source was verified
16	issue_d	The month which the loan was funded
17	loan_status	Current status of the loan
18	pymnt_plan	Indicates if a payment plan has been put in place for the loan
19	url	URL for the LC page with listing data.
20	desc	Loan description provided by the borrower
21	purpose	A category provided by the borrower for the loan request.
22	title	The loan title provided by the borrower
23	zip_code	The first 3 numbers of the zip code provided by the borrower in the loan application.
24	addr_state	The state provided by the borrower in the loan application
25	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.
26	delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years

27	earliest_cr_line	The month the borrower's earliest reported credit line was opened
28	inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
29	mths_since_last_delinq	The number of months since the borrower's last delinquency.
30	mths_since_last_record	The number of months since the last public record.
31	open_acc	The number of open credit lines in the borrower's credit file.
32	pub_rec	Number of derogatory public records
33	revol_bal	Total credit revolving balance
34	revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
35	total_acc	The total number of credit lines currently in the borrower's credit file
36	initial_list_status	The initial listing status of the loan. Possible values are – W, F
37	out_prncp	Remaining outstanding principal for total amount funded
38	out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
39	total_pymnt	Payments received to date for total amount funded
40	total_pymnt_inv	Payments received to date for portion of total amount funded by investors
41	total_rec_prncp	Principal received to date
42	total_rec_int	Interest received to date
43	total_rec_late_fee	Late fees received to date
44	recoveries	post charge off gross recovery

45	collection_recovery_fee	post charge off collection fee
46	last_pymnt_d	Last month payment was received
47	last_pymnt_amnt	Last total payment amount received
48	inq_last_12m	Number of credit inquiries in past 12 months
49	next_pymnt_d	Next scheduled payment date
50	last_credit_pull_d	The most recent month LC pulled credit for this loan
51	collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
52	mths_since_last_major_derog	Months since most recent 90-day or worse rating
53	policy_code	publicly available policy_code=1, new products not publicly available policy_code=2
54	application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
55	annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
56	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
57	verification_status_joint	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
58	acc_now_delinq	The number of accounts on which the borrower is now delinquent.
59	tot_coll_amt	Total collection amounts ever owed
60	tot_cur_bal	Total current balance of all accounts
61	open_acc_6m	Number of open trades in last 6 months

62	open_il_6m	Number of currently active installment trades
63	pen_il_12m	Number of installment accounts opened in past 12 months
64	open_il_24m	Number of installment accounts opened in past 24 months
65	mths_since_rcnt_il	Months since most recent installment accounts opened
66	total_bal_il	Total current balance of all installment accounts
67	il_util	Ratio of total current balance to high credit/credit limit on all install acct
68	open_rv_12m	Number of revolving trades opened in past 12 months
69	open_rv_24m	Number of revolving trades opened in past 24 months
70	max_bal_bc	Maximum current balance owed on all revolving accounts
71	all_util	Balance to credit limit on all trades
72	total_rev_hi_lim	Total revolving high credit/credit limit
73	inq_fi	Number of personal finance inquiries
74	total_cu_tl	Number of finance trades

Chapter 3: Performance Metrics

To measure the success rate of the model, the best metric was the precise prediction of a loan default. The profitability of the investor or the financial institution depends on the decision of the model. These are the error types (false positive, false negative, true positive and true negative) which were used for determining a conservative evaluation of the loan default rate.

True Positive(TP): These are cases where the model predicts that the loan will be repaid and the original outcome in the dataset is the same.

True Negative(TN): These are cases where the model predicts that the loan will default and the original outcome in the dataset is the same.

False Positive(FP): These are cases where the model predicts that the loan will be repaid, but the original outcome in the dataset is that it will default.

False Negative(FN): These are cases where the model predicts that the loan will default, but the original outcome in the dataset is that it will not default.

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Fig. 2: ***Example of a confusion matrix***

The above figure is a confusion matrix which displays the error types. Using this, other performance metrics like precision, accuracy, true positive rate, false positive rate are calculated. In the above figure, 'n' denotes the total number of cases. There are two possible Predicted and Actual classes: 'YES' and 'NO'. Actual 'Yes' means that the loan was originally paid off and Actual 'NO' means the loan wasn't paid off. Predicted 'YES' means that the model classifier predicted that the loan would be paid off and Predicted 'NO' means that the model classifier predicted that the loan would not be paid off. Thus, the classifier made a total of 165 predictions that were equal to the number of actual outcomes. 'TN', 'FP', 'FN' and 'TP' denote True Negatives, False Positives, False Negatives and True Positives respectively.

The best metric to evaluate the model is the precision of the algorithm to predict whether a customer is going to repay the loan. This is achieved by training the model on the training dataset and then predicting (based on the features) the faithfully paying customers from those that default. The training results in being able to measure the precision, practicality and realism of the model. The precision, accuracy, true positive rate and false positive rate of the model are measured as follows:

Precision= (True Positive/ (True Positive + False Positive))

Accuracy= (True Positive/ (True Positive+ False Positive+ True Negative + False Negative)

True Positive Rate: (True Positive/ (True Positive + False Negative))

False Positive Rate: ((False Positive/ (False Positive + True Positive)))

So, for the confusion matrix in Fig. 2, the precision, accuracy, true positive rate and false positive rate are calculated as follows:

Accuracy: (TP + TN) / Total = (100 + 50)/ 165 = 0.91

Precision: TP/ Predicted YES = 100/110 = 0.91

True Positive Rate: TP/ Actual YES = 100/ 105 = 0.95

False Positive Rate: FP/ Actual NO = 10/ 60 = 0.17

Chapter 4: Methodology

4.1 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']
```

Fig. 3: ***List of dropped features***

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. 4: ***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. 5: ***List of count of null values after executing step 8***

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. 6: ***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.

4.2 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.

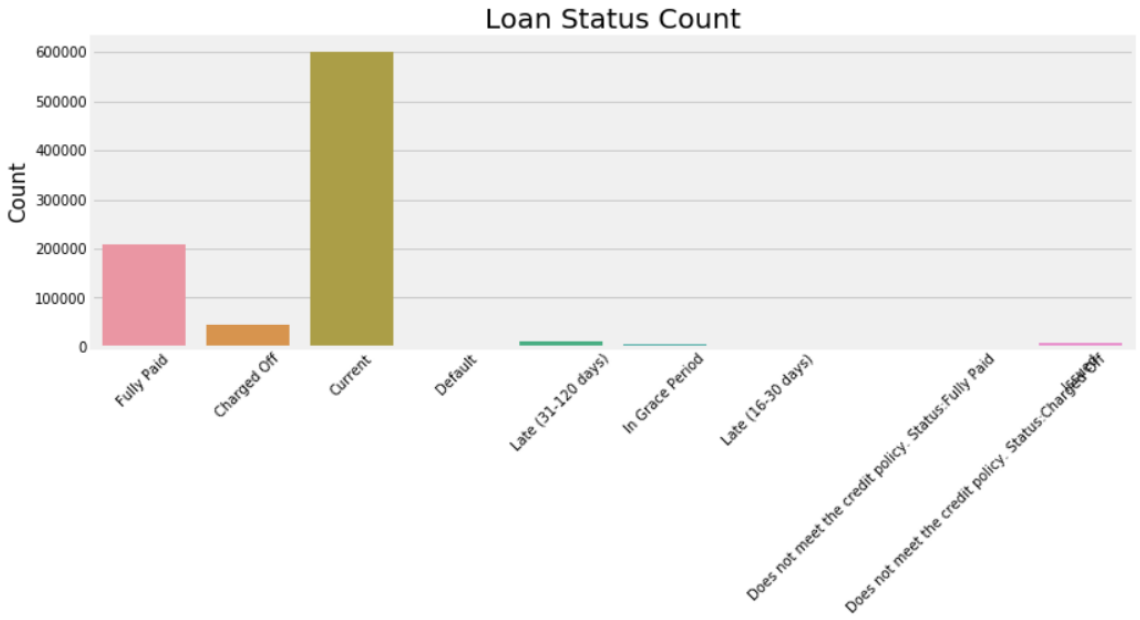

Fig. 7: ***Count of loan status by type***

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.

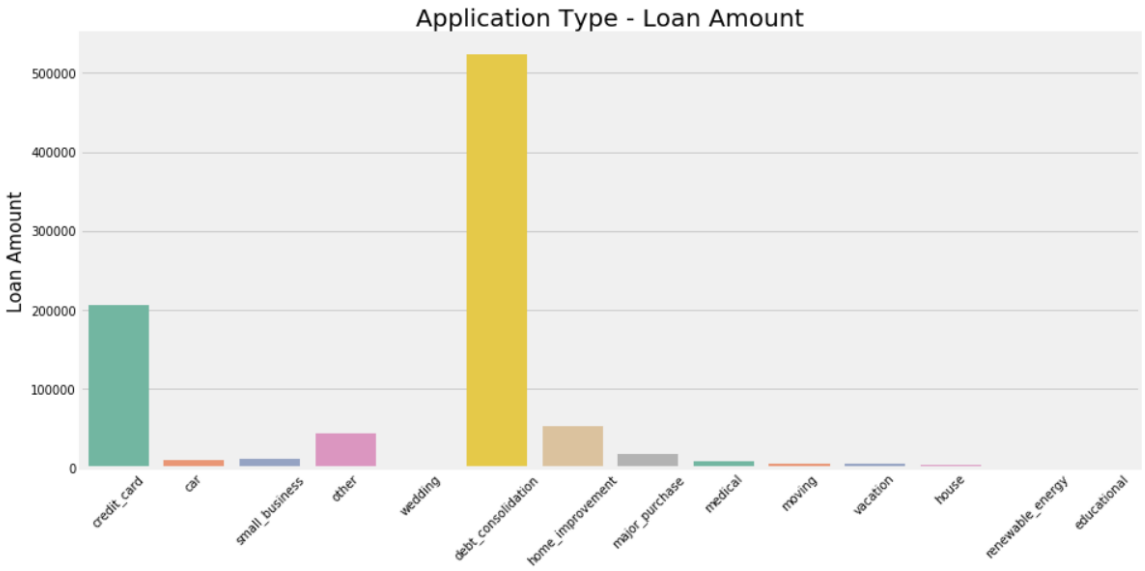

Fig. 8: ***Amount of loan by purpose***

The frequency and normal distribution of the loan amount were then analyzed.

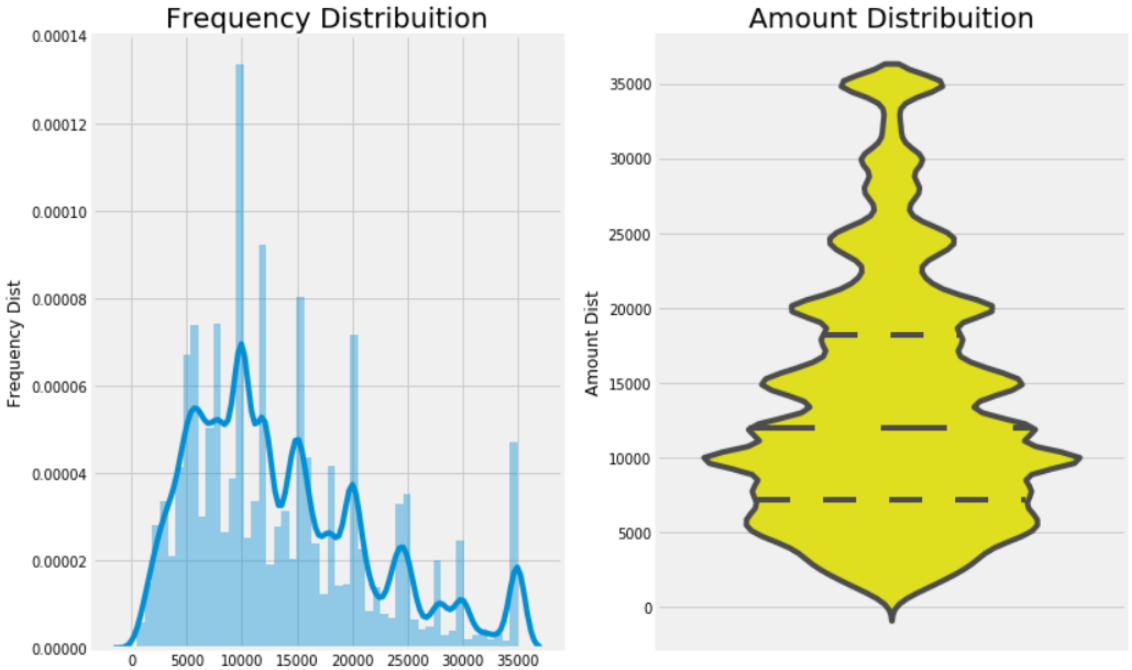

Fig. 9: ***Frequency and normal distribution of loan amount***

The interest rate normal and frequency distribution were then observed.

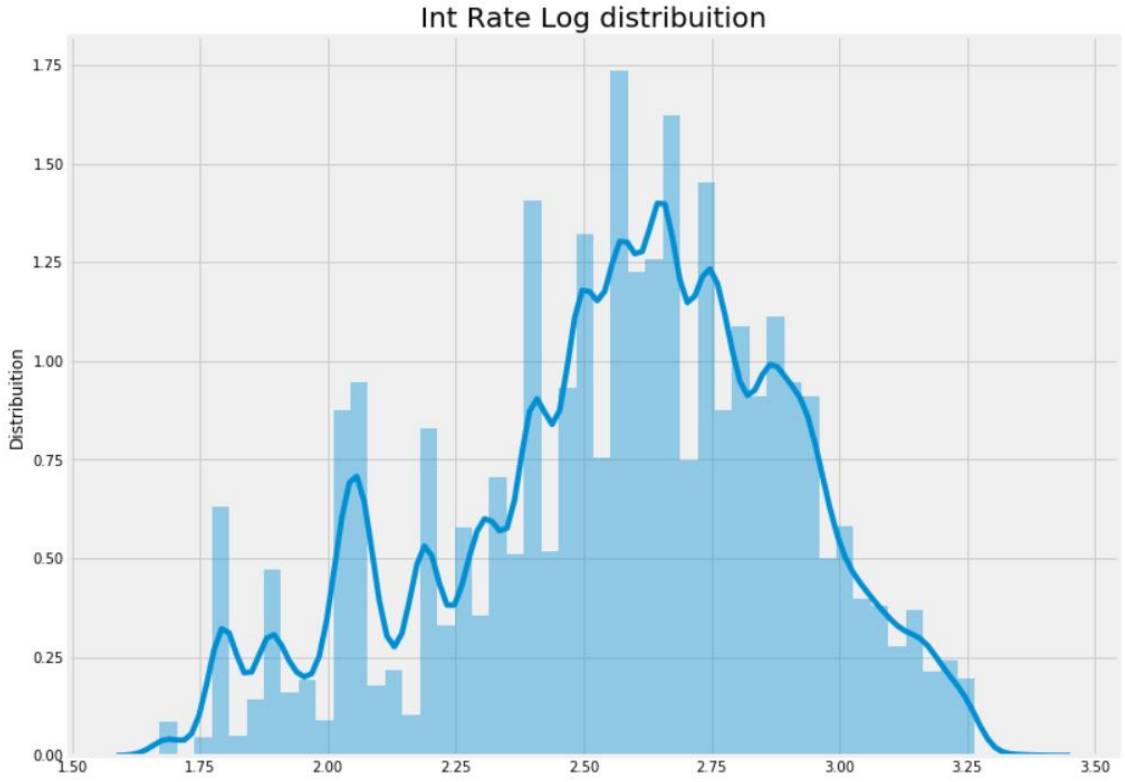

Fig. 10: ***Interest rate log Distribution***

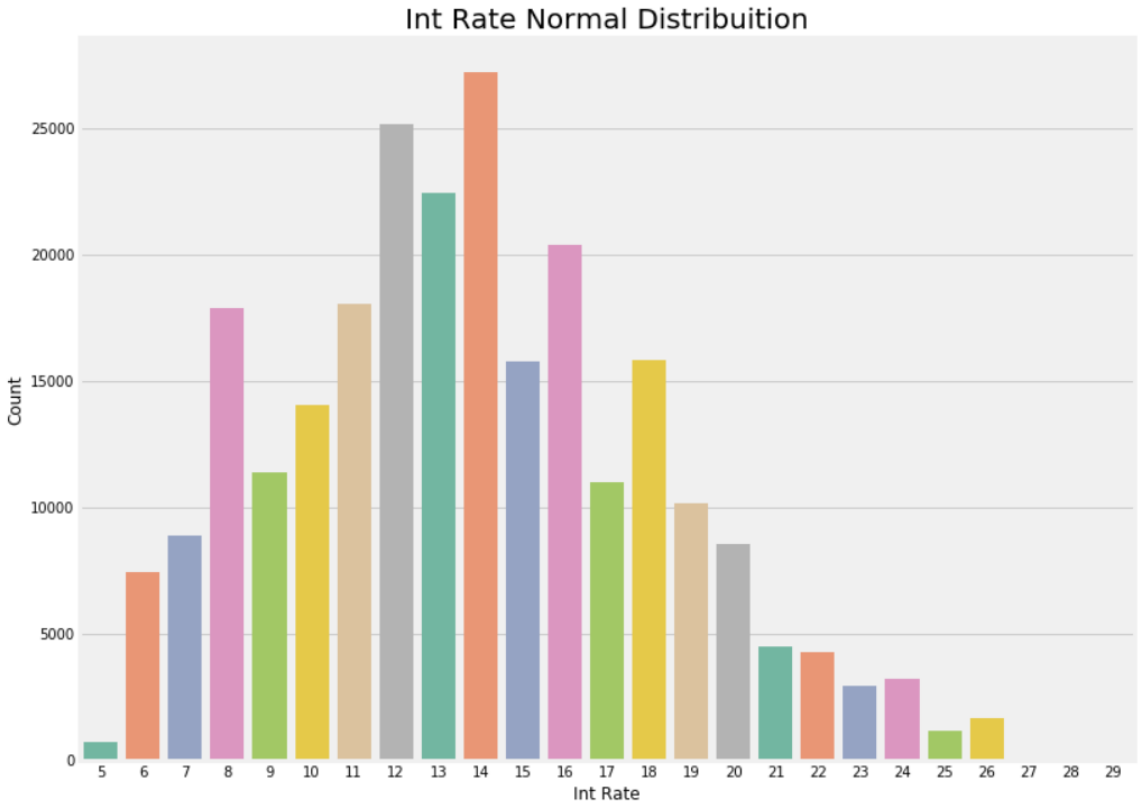

Fig. 11: ***Interest rate normal distribution***

The distribution of the application type through the loan amount and interest rate were analyzed.

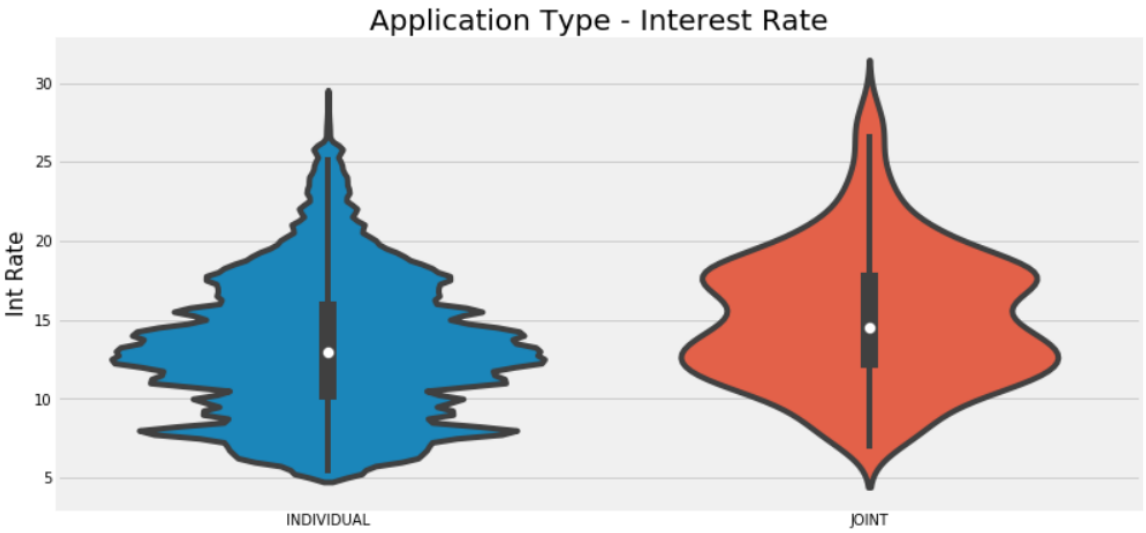

Fig. 12: ***Distribution of application type through interest rate***

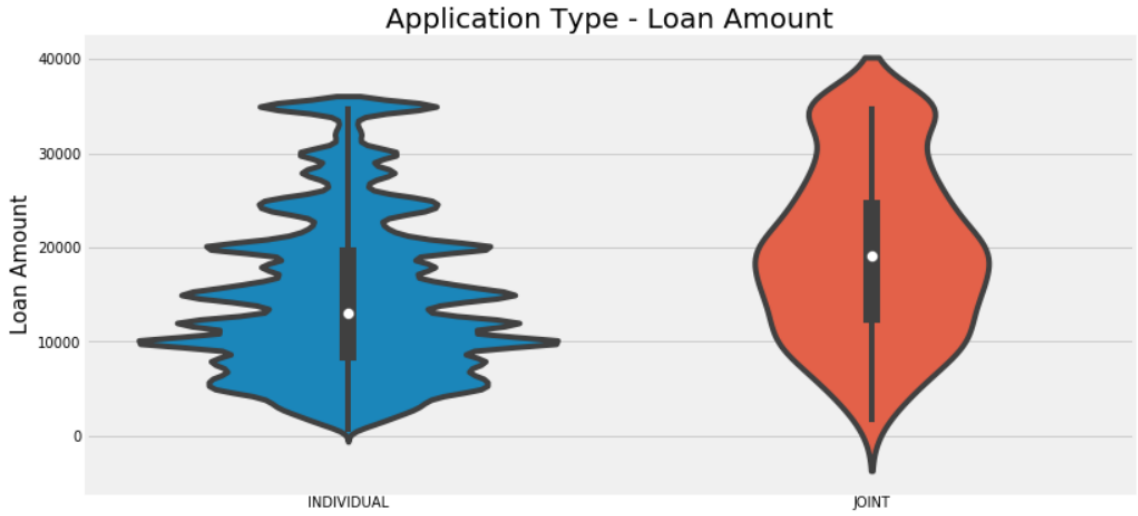

Fig. 13: ***Distribution of application type through loan amount***

The home ownership and loan amount distribution with the application type were observed.

It was observed that joint application type applicants had either rented, owned or had their homes mortgaged. The highest number of joint application type had their homes mortgaged.

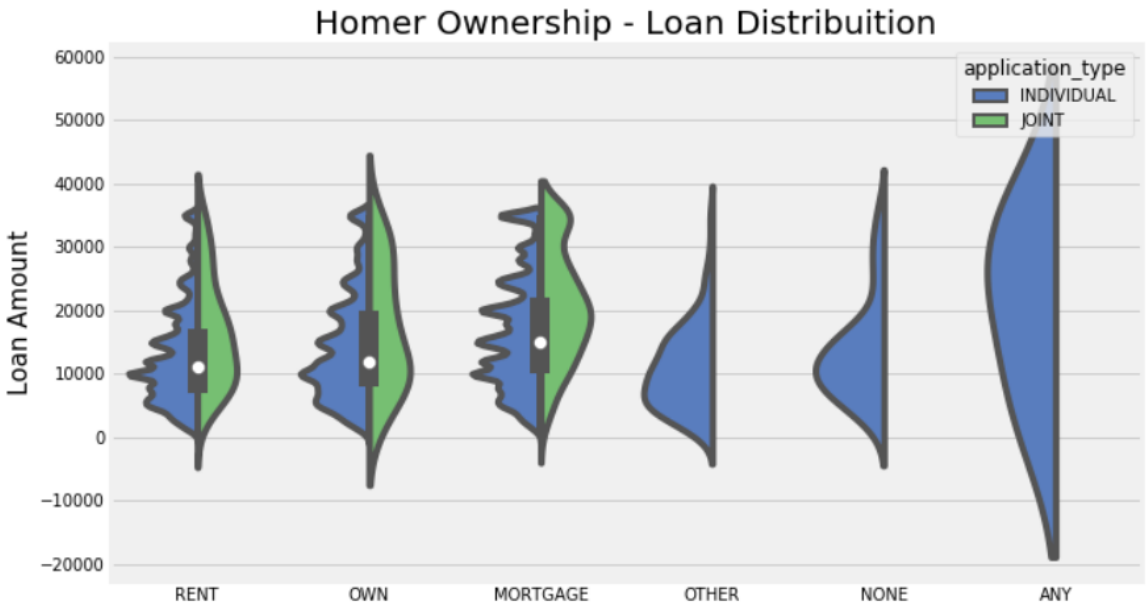

Fig. 14: ***Home ownership and loan amount distribution by application type***

application_type	INDIVIDUAL	JOINT
loan_status		
Charged Off	45248	0
Current	601338	441
Default	1219	0
Does not meet the credit policy. Status:Charged Off	761	0
Does not meet the credit policy. Status:Fully Paid	1988	0
Fully Paid	207722	1
In Grace Period	6250	3
Issued	8396	64
Late (16-30 days)	2357	0
Late (31-120 days)	11589	2

Fig. 15: *Loan status by application type*

The purpose distribution was observed against the loan amount with the application type. There were many observations. The selection of the small business as the purpose of the loan was usually seen for joint application types over individual. Joint application types rarely had purpose as “moving”, “vacation”, “house”, “educational” or “renewable energy”.

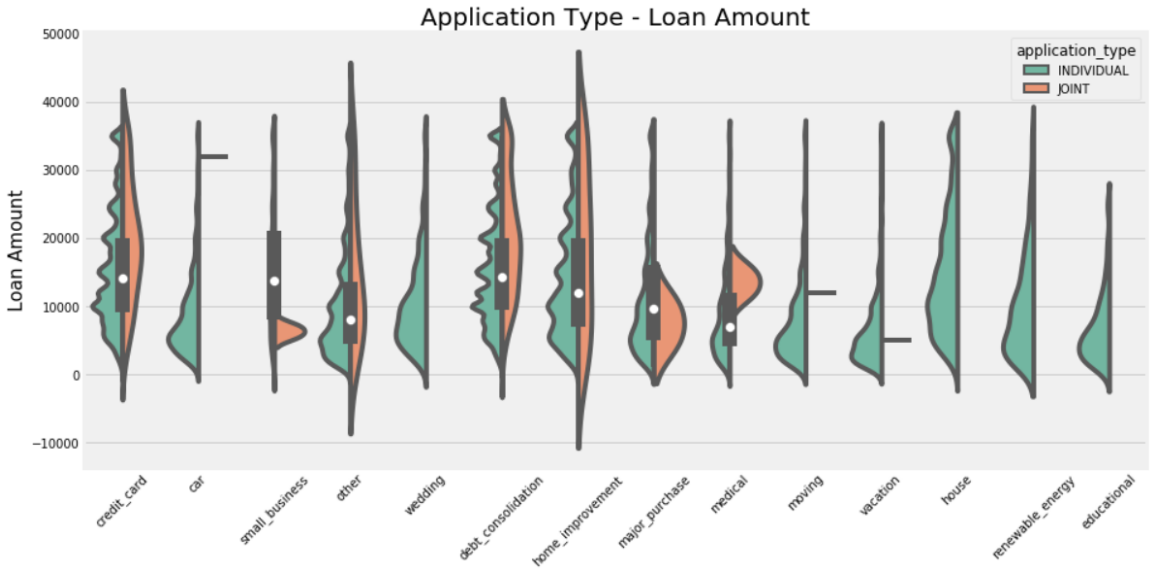

Fig. 16: ***Purpose distribution against loan amount by application type***

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.

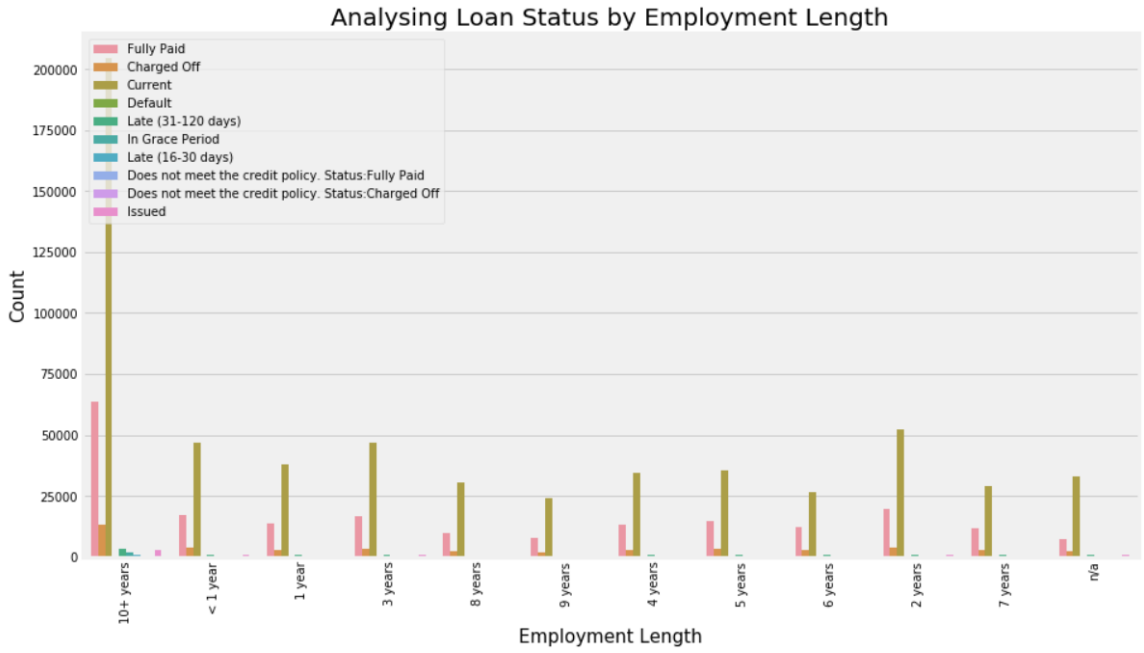

Fig. 17: ***Home ownership and loan amount distribution by application type***

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. 18: *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.

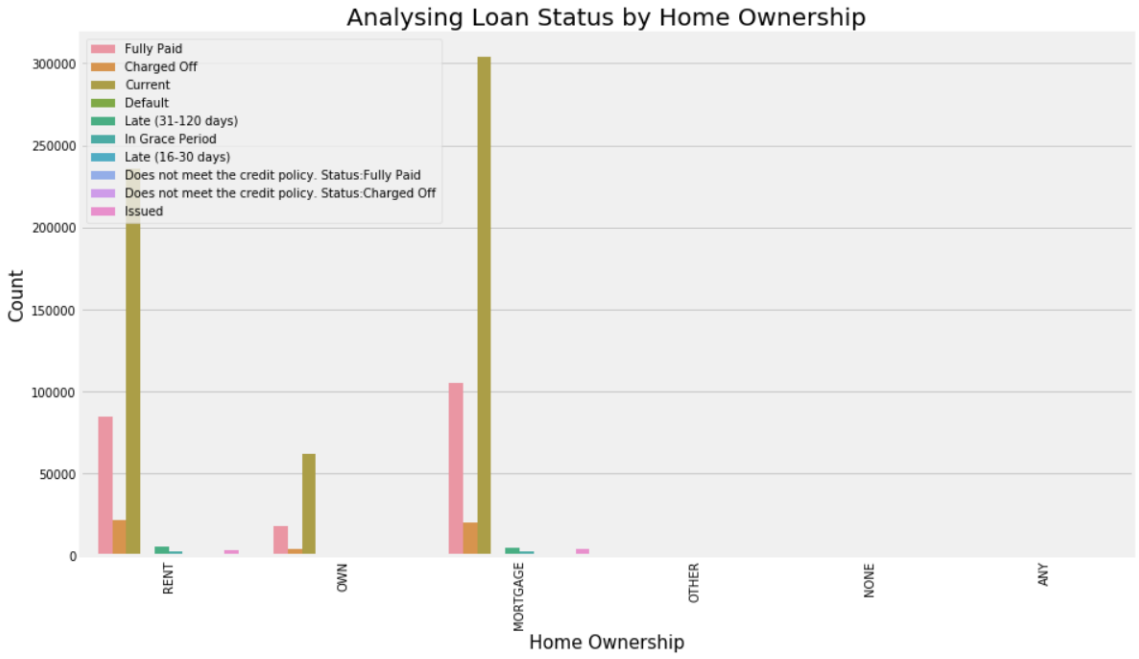

Fig. 19: ***Count of loan status types by home ownership***

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)
home_ownership										
ANY	0	2	0	0	0	1	0	0	0	0
MORTGAGE	19878	303764	498	348	908	104966	2855	4220	1101	5019
NONE	7	2	0	1	4	36	0	0	0	0
OTHER	27	3	0	11	27	114	0	0	0	0
OWN	4025	62041	110	49	138	17960	637	1038	260	1212
RENT	21311	235967	611	352	911	84646	2761	3202	996	5360

Fig. 20: ***Home ownership by loan status***

The loan status was observed by purpose. The highest no. of people who fully paid/charged off wanted the loan for debt consolidation. The second highest were for credit cards.

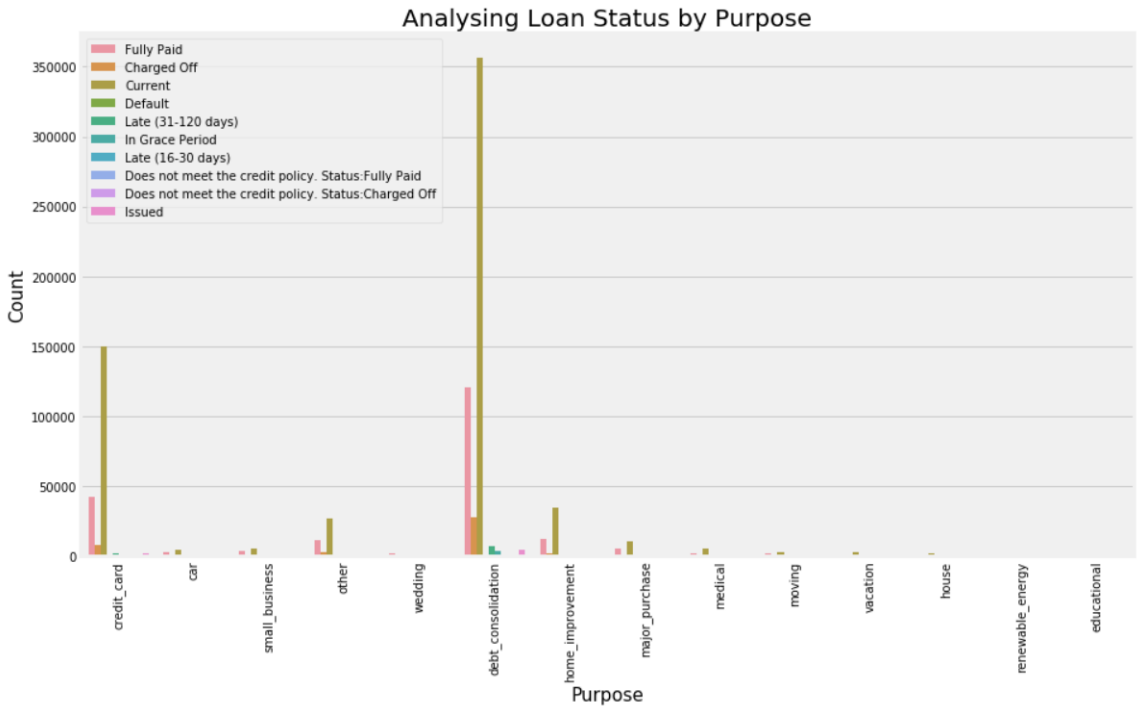

Fig. 21: Count of *Loan status types by purpose*

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)
purpose										
car	448	4937	10	13	51	3198	40	81	15	70
credit_card	7826	149835	233	69	271	42250	1150	2071	381	2096
debt_consolidation	27599	356239	790	292	808	120764	3998	4796	1510	7419
educational	56	1	0	32	65	269	0	0	0	0
home_improvement	2269	34980	47	71	143	12660	367	493	137	662
house	286	1854	7	11	33	1366	37	37	15	61
major_purchase	874	10308	14	23	100	5391	125	184	51	207
medical	569	5324	15	22	36	2285	56	91	17	125
moving	425	3121	11	15	31	1603	43	52	23	90
other	2936	26607	65	121	303	11341	310	480	136	595
renewable_energy	54	282	0	1	2	213	8	6	0	9
small_business	1371	5020	19	72	89	3375	79	112	50	190
vacation	270	2946	8	6	13	1318	37	57	22	59
wedding	265	325	0	13	43	1690	3	0	0	8

Fig. 22: *Loan status type by purpose*

The loan status was observed by verification status. The number of charge off's was similar between those who had their profiles source verified or not verified.

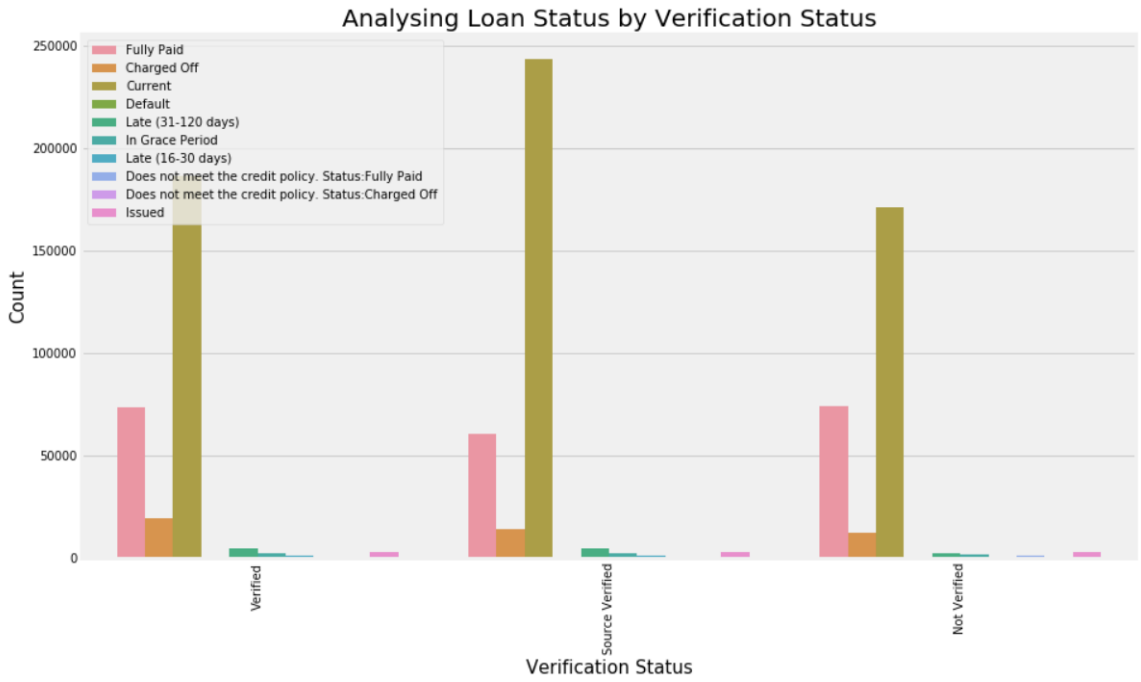

Fig. 23: ***Count of loan status types by verification status***

verification_status	Not Verified	Source Verified	Verified
loan_status			
Charged Off	12208	13740	19300
Current	171400	243705	186674
Default	278	462	479
Does not meet the credit policy. Status:Charged Off	511	82	168
Does not meet the credit policy. Status:Fully Paid	1321	208	459
Fully Paid	73856	60271	73596
In Grace Period	1403	2523	2327
Issued	2779	2836	2845
Late (16-30 days)	473	1006	878
Late (31-120 days)	2521	4725	4345

Fig. 24: ***Loan status by verification status***

The loan status was observed by “grade”. Grade C had the highest number of people who charged off and grade G had the lowest.

grade	A	B	C	D	E	F	G
loan_status							
Charged Off	2617	9519	12642	10486	6258	2934	792
Current	103322	171735	171175	91984	47061	13589	2913
Default	47	198	360	312	201	79	22
Does not meet the credit policy. Status:Charged Off	8	85	148	197	158	93	72
Does not meet the credit policy. Status:Fully Paid	90	269	481	494	378	154	122
Fully Paid	39679	66546	52678	30020	12928	4726	1146
In Grace Period	365	1240	1887	1405	908	354	94
Issued	1448	2529	2472	1185	593	194	39
Late (16-30 days)	134	410	678	569	368	155	43
Late (31-120 days)	492	2004	3339	2890	1852	768	246

Fig. 25: ***Loan status by grade***

The loan status was analyzed with respect to term.

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)
term										
36 months	29083	402848	715	649	1789	167575	3975	5982	1483	7026
60 months	16165	198931	504	112	199	40148	2278	2478	874	4565

Fig. 26: ***Loan status by term***

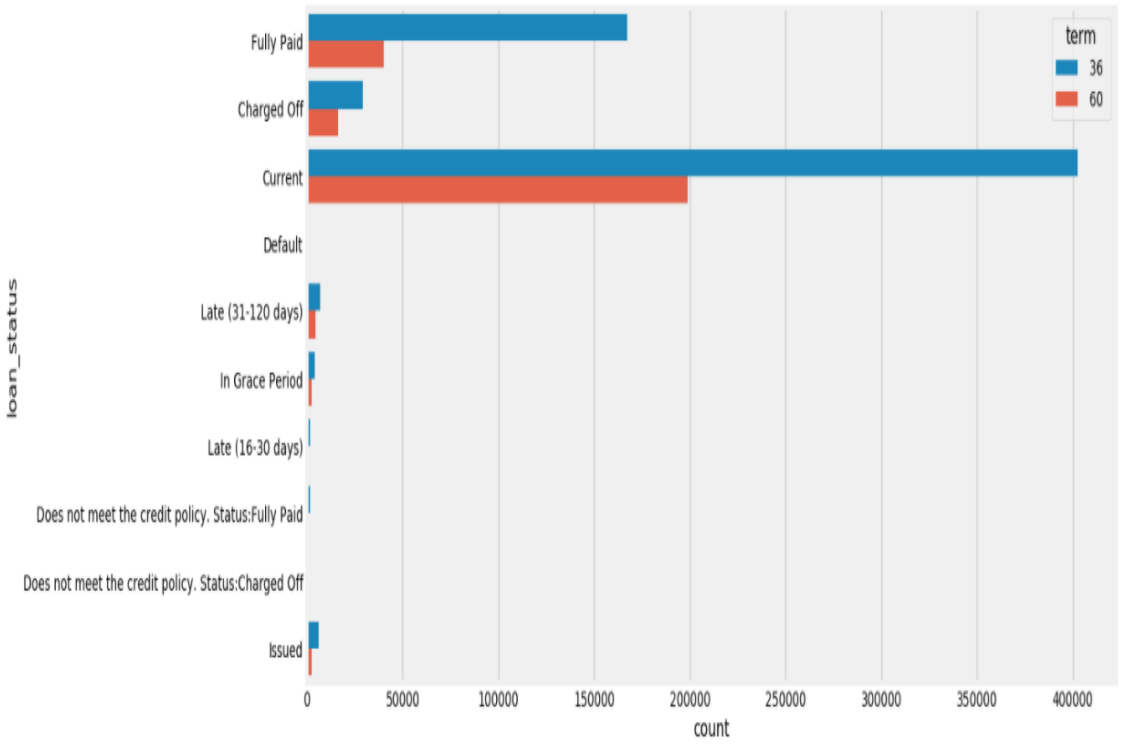

Fig. 27: ***Count of Loan status by term***

4.3 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. 28: ***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.

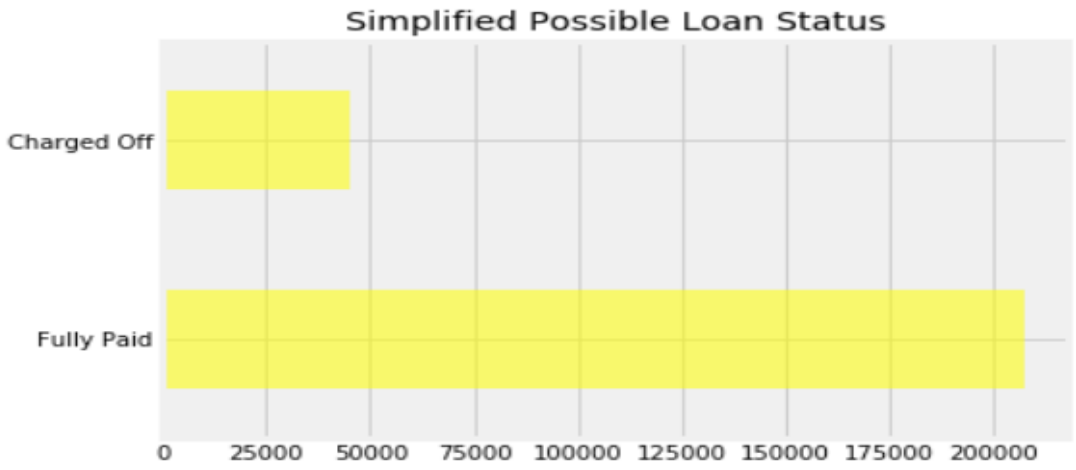

Fig. 29: ***Count of simplified loan status***

The loan status was analyzed against different features.

The following images were the results:

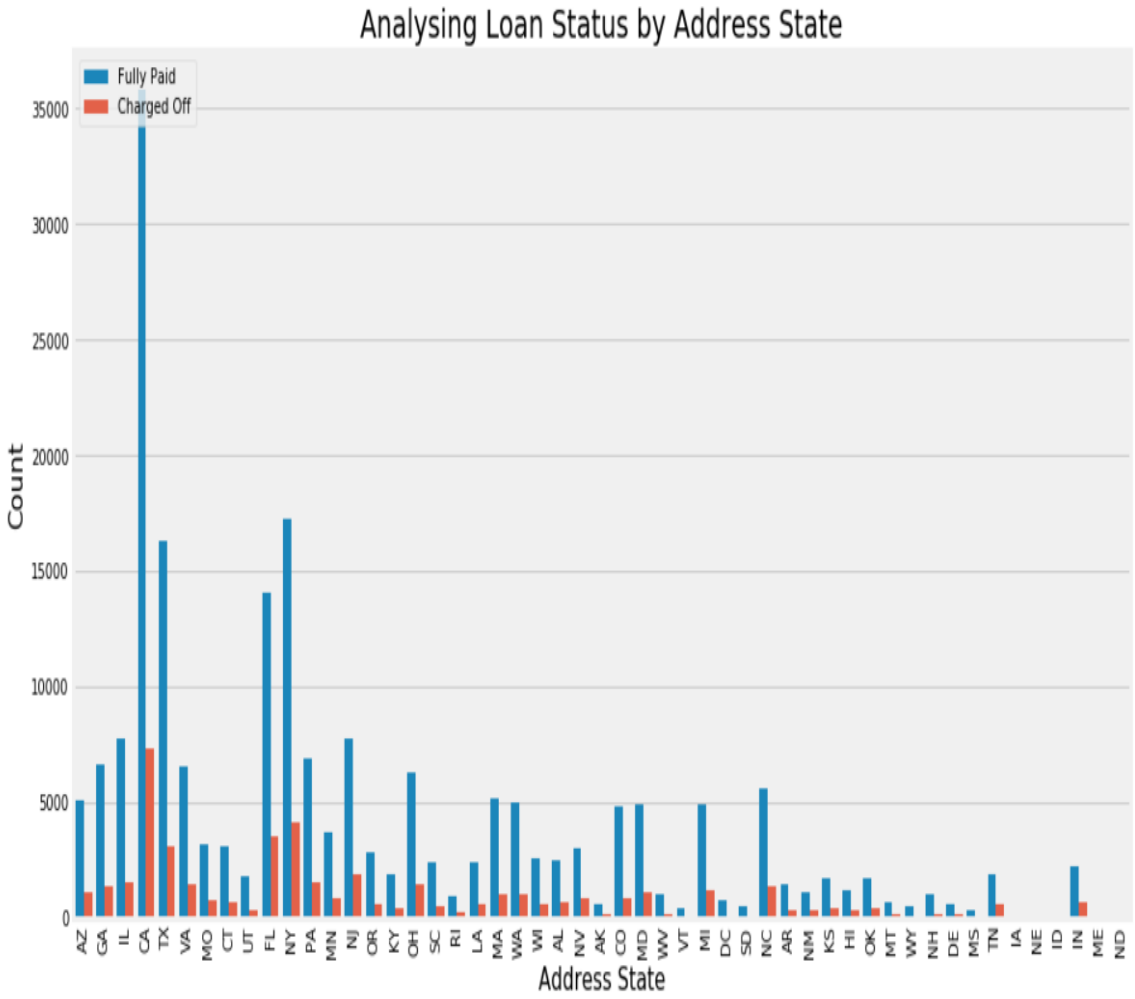

Fig. 30: ***Count of Simplified loan status by address state***

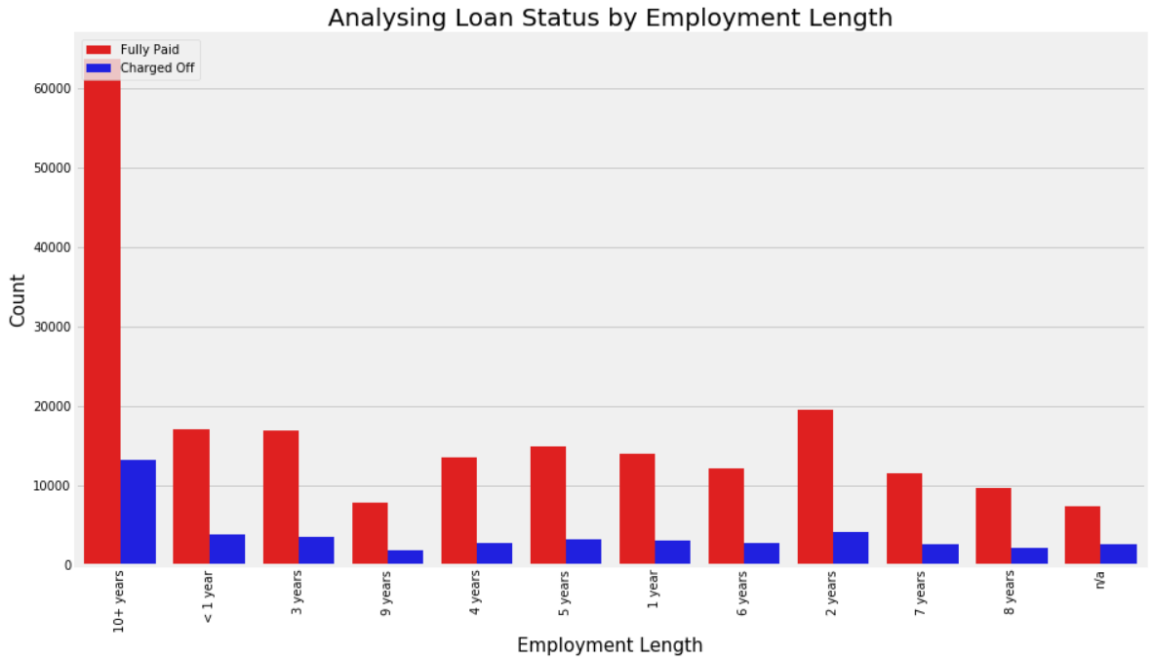

Fig. 31: ***Count of simplified loan status by employment length***

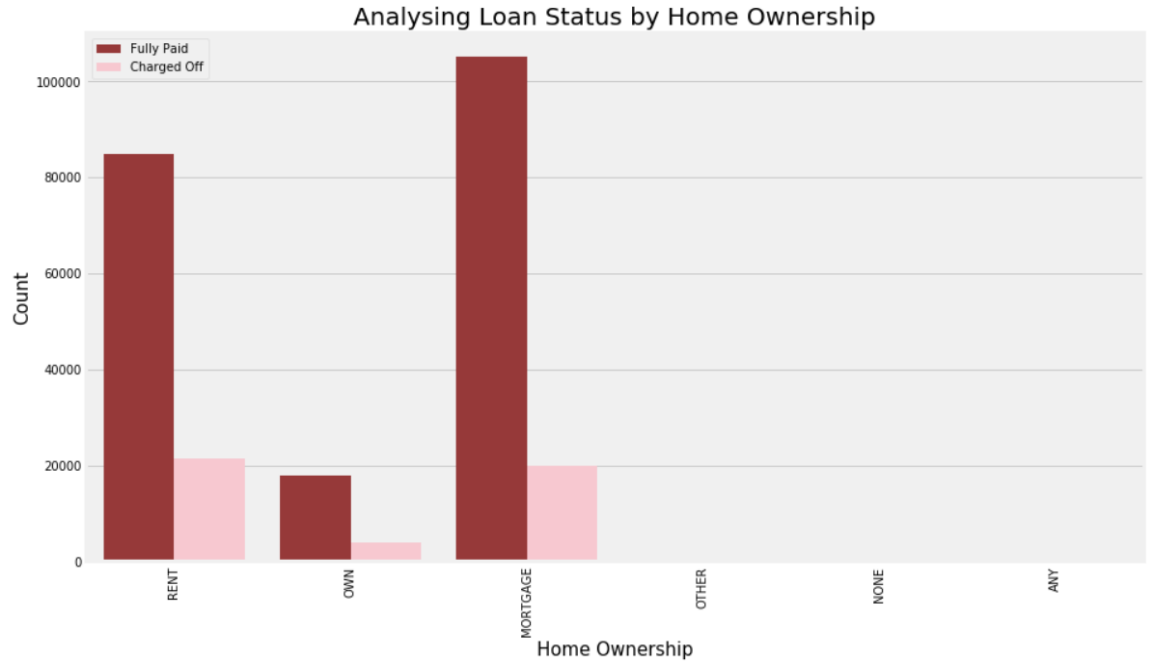

Fig. 32: ***Count of simplified loan status by home ownership***

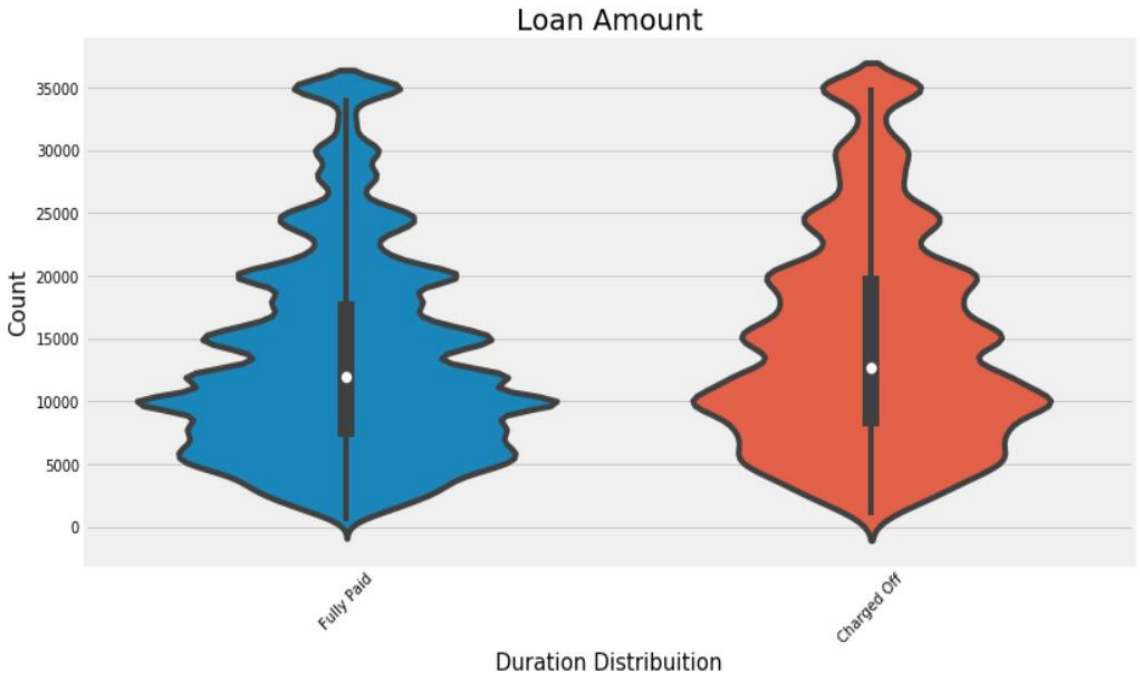

Fig. 33: ***Simplified loan amount duration distribution***

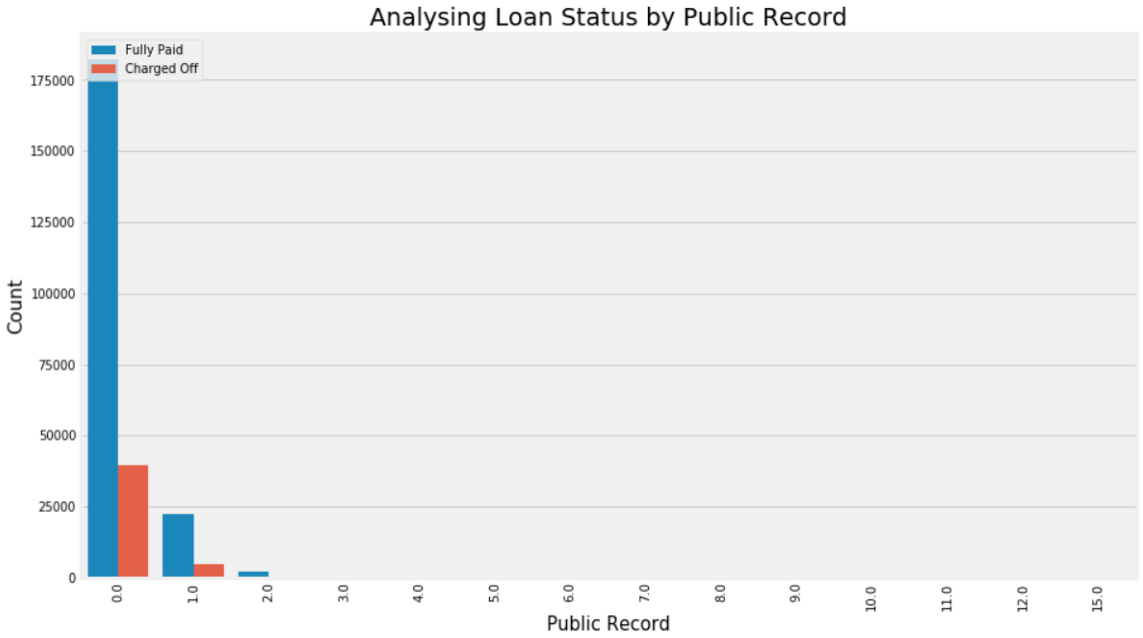

Fig. 34: ***Count of simplified loan status by public record***

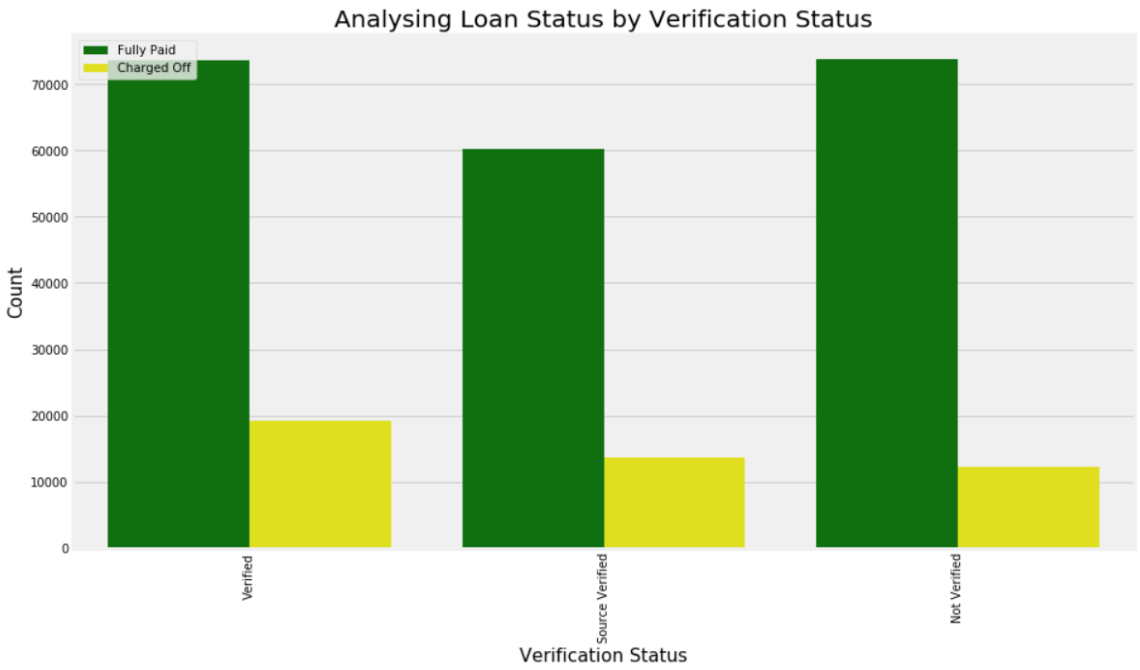

Fig. 35: ***Count of simplified loan status by verification status***

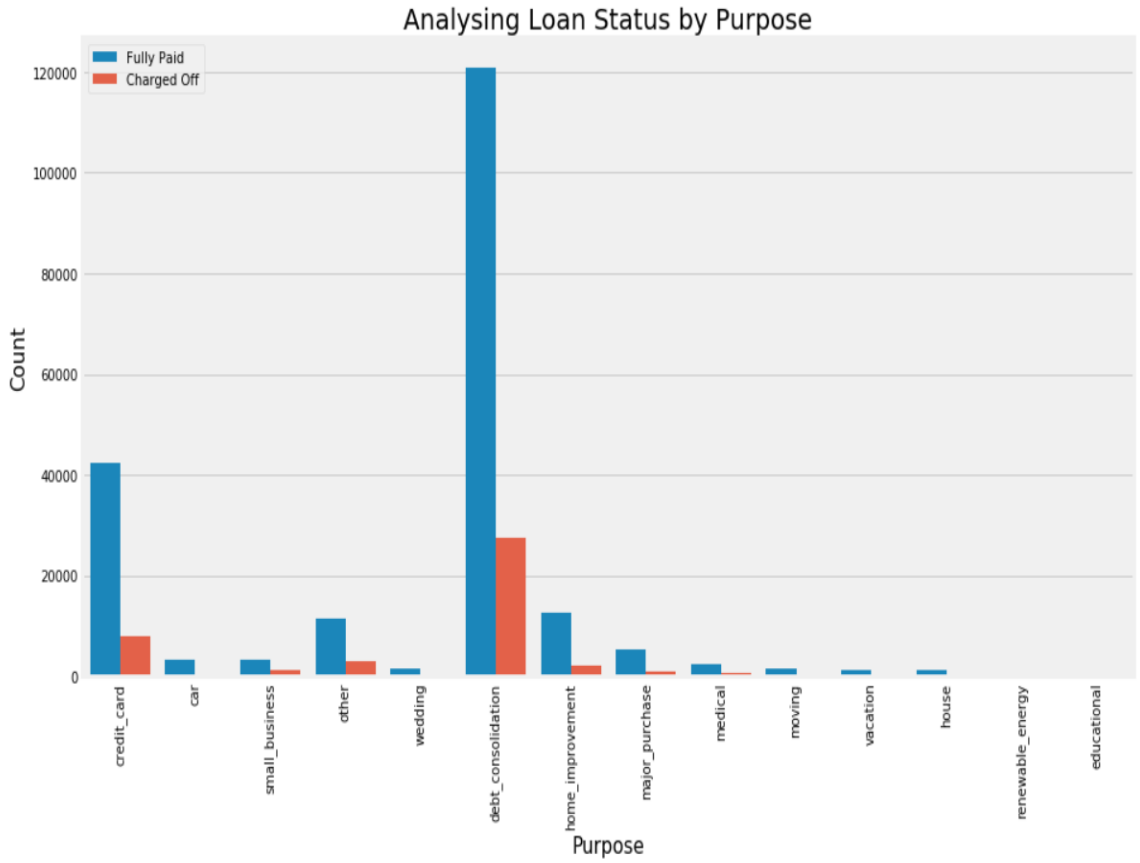

Fig. 36: ***Count of simplified loan status by purpose***

All categorical values were converted into numeric data types.

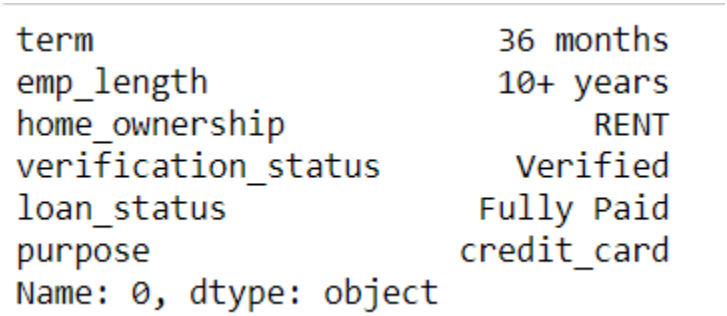

Fig. 37: ***List of categorical features***

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. 38: ***List of input features for the model***

4.4 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.

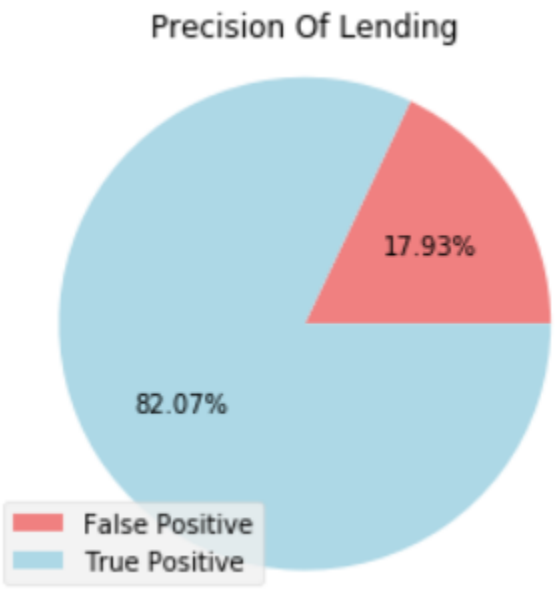

Fig. 39: ***Pie chart of True positive rate vs false positive rate***

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.

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.

Stratified K-fold type of Cross- validation was used for the project. Stratification is the process of rearranging the data as to ensure each fold is a good representative of the

whole. For example, in a binary classification problem where each class comprises 50% of the data, it is best to arrange the data such that in every fold, each class comprises around half the instances. The dataset was divided into K folds. K was set as 3 for the project. In the following steps, the second part of that particular step describes the model with K fold cross validation where the number of folds(K) are set to 3.

The results for the model with no weight changes and with the use of K-fold cross validation are as follows:

Precision: 0.8206

Accuracy: 0.8199

T.P.R.: 0.99

F.P.R.: 0.99

Step 3: a) Weight was added to mistakes in order to penalize the model when it overfits for improving the performance of the model. The weight was balanced.

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

	precision	recall	f1-score	support
0	0.29	0.64	0.40	13527
1	0.89	0.65	0.76	62278
avg / total	0.79	0.65	0.69	75805

```
array([[ 8721,  4806],
       [21576, 40702]], dtype=int64)
```

As a result, a better precision of 0.89 was obtained.

T.P.R.: 0.65

F.P.R.: 0.35

Accuracy: 0.65

b) Stratified K- fold was applied to the above model and the results were as follows:

Precision: 0.82

Accuracy: 0.55

T.P.R.: 0.58

F.P.R.: 0.58

Step 4: a) A different weight was added to see if a better precision is obtained. A manual weight of 10 for charged off and 1 for fully paid was added. The following images show the classification report and confusion matrix of the result:

	precision	recall	f1-score	support
0	0.19	0.96	0.32	13527
1	0.94	0.13	0.23	62278
avg / total	0.81	0.28	0.25	75805

```
array([[13007,   520],
       [53982,  8296]], dtype=int64)
```

As a result, a better precision of 0.94 was obtained.

T.P.R.: 0.13

F.P.R.: 0.03

Accuracy: 0.28

b) Stratified K- fold was applied to the above model and the results were as follows:

Precision: 0.82

Accuracy: 0.25

T.P.R.: 0.12

F.P.R.: 0.11

Step 5: a) A different weight was tried. A manual weight of 8 for charged off and 1 for fully paid was added. The following images show the classification report and confusion matrix of the result:

	precision	recall	f1-score	support
0	0.21	0.92	0.34	13527
1	0.93	0.23	0.37	62278
avg / total	0.80	0.35	0.36	75805

```
array([[12459,  1068],
       [47915, 14363]], dtype=int64)
```

As a result, a precision of 0.93 was obtained.

T.P.R.: 0.23

F.P.R.: 0.07

Accuracy: 0.35

b) Stratified K- fold was applied to the above model and the results were as follows:

Precision: 0.82

Accuracy: 0.30

T.P.R: 0.18

F.P.R: 0.17

The model with a manual weight 8 for class 0 and 1 for class 1 with a high precision rate (0.93) and decent accuracy (0.35) is a profitable model for the goal of the project. It would work well because, it has a good balance between accuracy and precision.

If there are 100 loan applications, by using this model, the investor would lend money to 35 borrowers (assume the interest rate is 10% and the loan amount to be 1000$ per application).

By having a precision of 0.93, 33 out of 35 borrowers would pay the amount in time (+ $3300), whereas two would default (- $2000). Thus, by using this model the investor would be in a profit of 1300$.

Chapter 5: Summary

5.1 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://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html
6) http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_predict.html
7) http://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
8) https://www.lendingclub.com/public/credit-score-101.action
9) https://www.openml.org/a/estimation-procedures/1