Segmenting and Predicting Loan Repayment Probability of Lending Club Debtors

Project Presentation video link -
https://mediaspace.msu.edu/media/CSE+801B+-+Introduction+to+Data+Mining+-+Project+Presentation/1_er1zz5db

Table of Context

1. Context

One of the main problems of financial institutions is the non-payment of their debtors. Bank delinquency rates have been the subject of special attention for these entities because they have a direct relationship with the overall results of profitability and liquidity of the companies.

Due to the various ravages that the COVID-19 pandemic has left in society, the relationships between people and companies that provide services and products have changed dramatically. Particularly for debt placement banking activities, default rates have risen a lot because some of the people do not have the resources to meet this obligation. Many people have been left without work, many others have lost part of their income and some of them have even had to incur additional expenses to attend health emergencies due to the pandemic. In a way, the collection problem for financial institutions has become more acute.

2. Data

We will use information from an entity called LendingClub. This company allows people to borrow money in need in exchange for an additional interest payment. It is a peer-to-peer lending company.

The data is separated into 2 different files: accepted loans and rejected loans. There are more or less 151 features of every loan of the dataset.

The information can be found in the next link:
https://www.kaggle.com/wordsforthewise/lending-club

3. Objective

The objective of this project is the analysis of the loans in this database to predict their payment behavior. On the one hand, a segmentation model can be carried out to determine different clusters of debtors and identify distinctive characteristics of each one of them. On the other hand, a prediction algorithm can be carried out that allows knowing the probability of payment of each loan. This can be useful for risk management issues and to give more information to the investors who own this debt.

4. Data Cleaning and Feature Engineering

In its original state, the data contains 2260701 observations and 151 variables. There are a high number of observations and high dimensionality in the data. It contains records for loans issues in between 2007 and 2018. On observing the data, it was concluded that some variables are better described in the dictionary of information, so a match was done with this table to get a better description of the variables. This helped eliminate the variables that were not relevant as

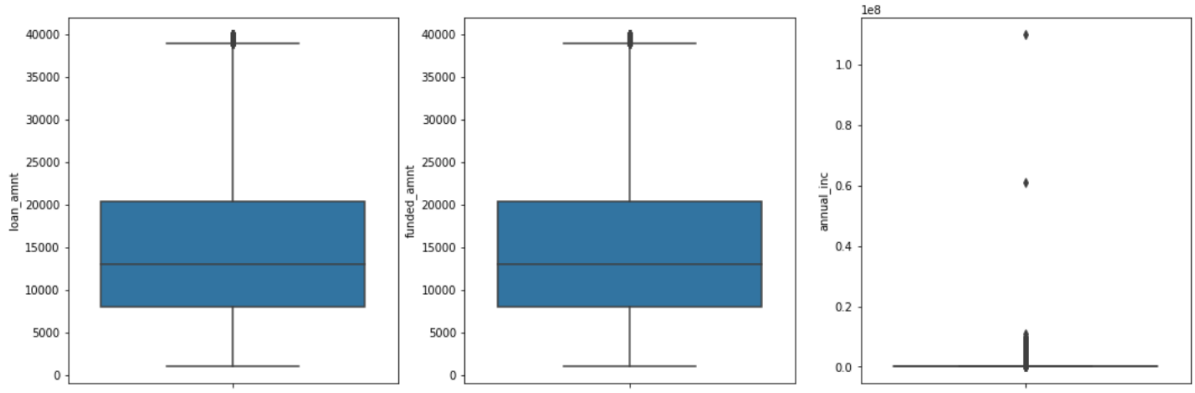
only the variables that were in both, the data and the dictionary, were picked. Moreover, the format of some variables containing dates was fixed and the ‘emp_length’ variable was transformed to a numeric variable.

The NA values were handled in the following manner:

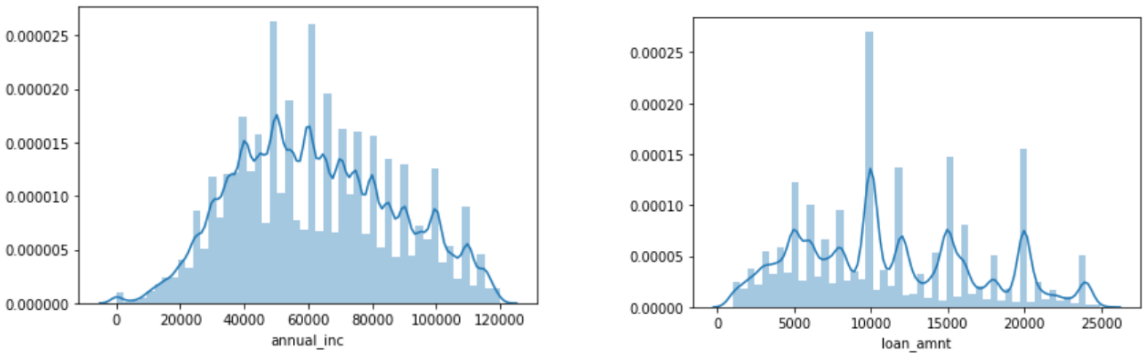
- 'emp_title' and 'verification_status_joint' variables were filled with ‘’.
- 'bc_open_to_buy', 'mo_sin_old_il_acct', 'mths_since_last_delinq',
- 'mths_since_last_major_derog', 'mths_since_last_record',
- 'mths_since_rcnt_il', 'mths_since_recent_bc', 'mths_since_recent_bc_dlq',
- 'mths_since_recent_inq', 'mths_since_recent_revol_delinq',
- 'pct_tl_nvr_dlq','sec_app_mths_since_last_major_derog' were filled with the max value of each column
- Rest of the columns were filled with the minimum value of each column.

In the end, the final dataset was left with 938821 observations and 102 variables.

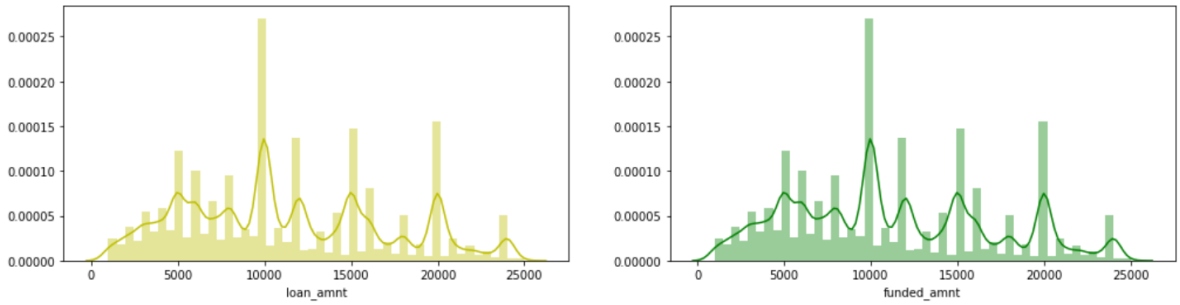
Some basic boxplots were made to see the distribution of data points in the categories of loan amount, funded amount and annual income which can be seen below.



Here we can see that most of the loan amounts and funded amounts fall in the range of 8000-20000, with some outliers. After removing the outlier’s frequency plots for loan and funded amounts were created which can be seen below.



After Outlier Treatment, we are still left with 72 % of data and we have sufficient information to proceed with Univariate Analysis.



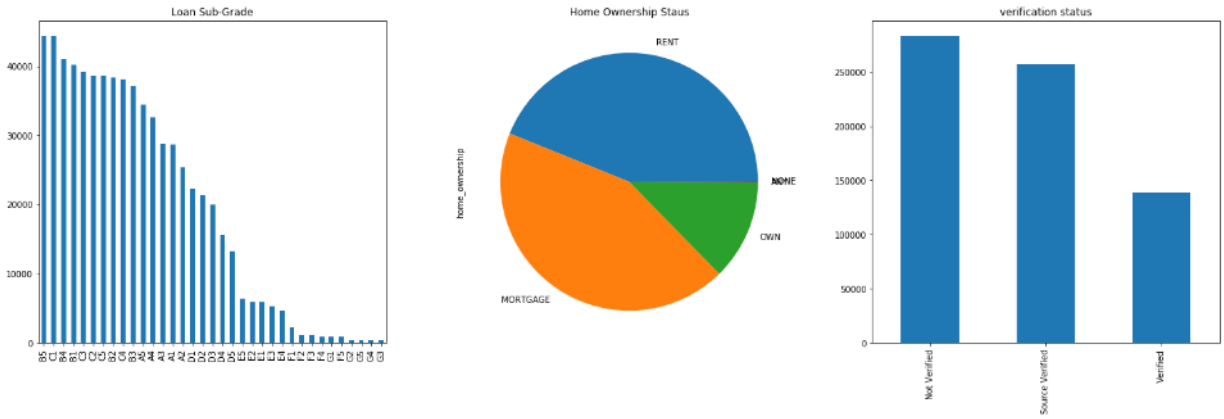
These variables are highly correlated, hence we remove funded_amount. Likewise, we reduced the features by doing correlation analysis.

Once the data is cleaned, we move on to some exploratory analysis to understand the data.

5. Exploratory Data Analysis

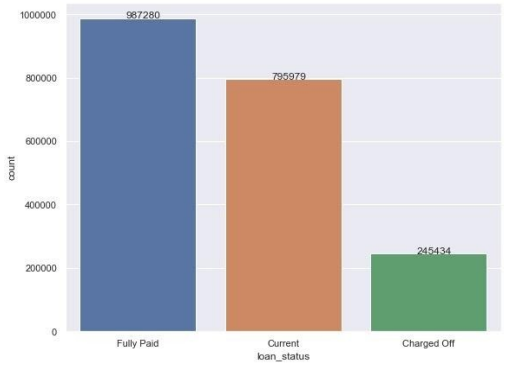
Loan Characteristics

- People who are taking loans have Home Ownership as Rent or in Mortgage.
- Most of loan applications do not have their income source verified, this is worth looking into as it might lead to defaulter loan



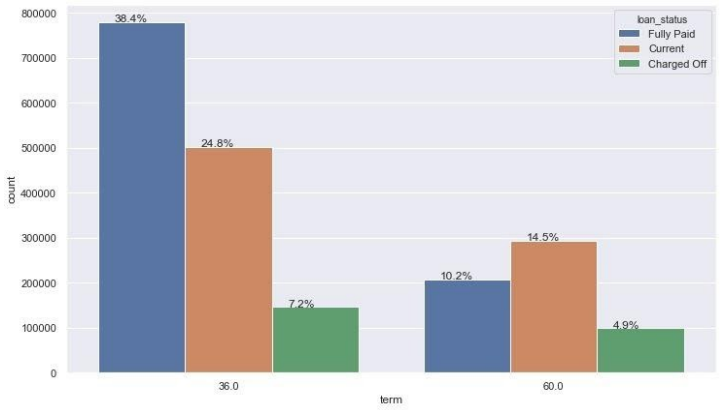
Loan Status

- We kept only 3 important loan statuses out of 7 present in the dataset, these are most useful
- We will focus on the Current and Charged off loans



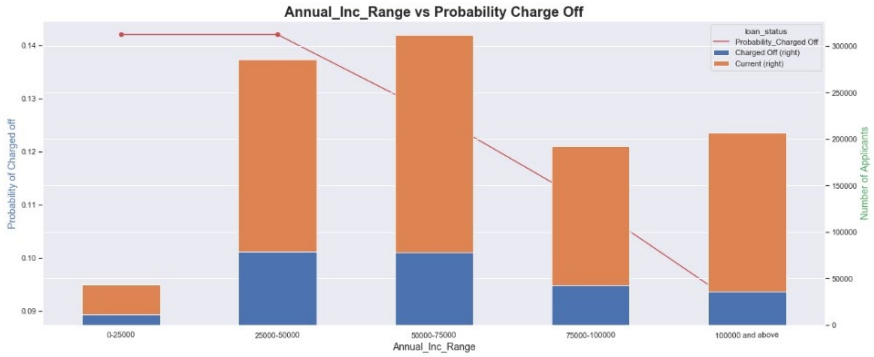
Loan Status Vs Term

- There are only 2 loan terms - 36 months and 60 months
- Smaller term loans are more likely to be charged off compared to longer term loans but majority loans are short term



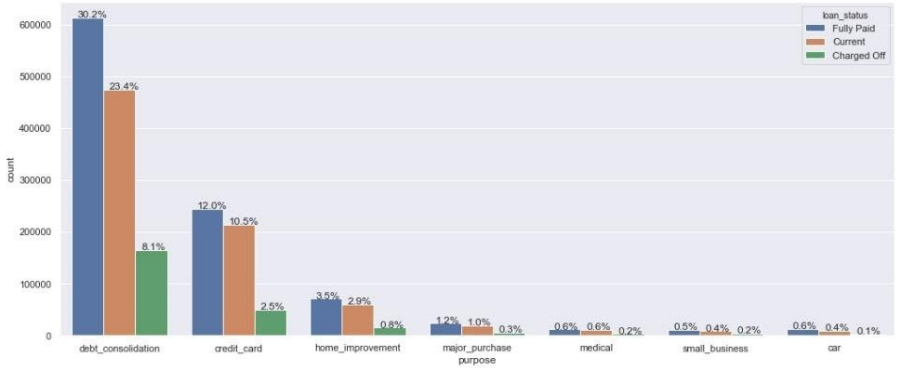
Annual Inc Range Vs Probability Charge Off

- Annual income should play a huge role in determining loan charge off probability
- With the income increases the probability of charge off decreases drastically
- This can be an important feature for the model



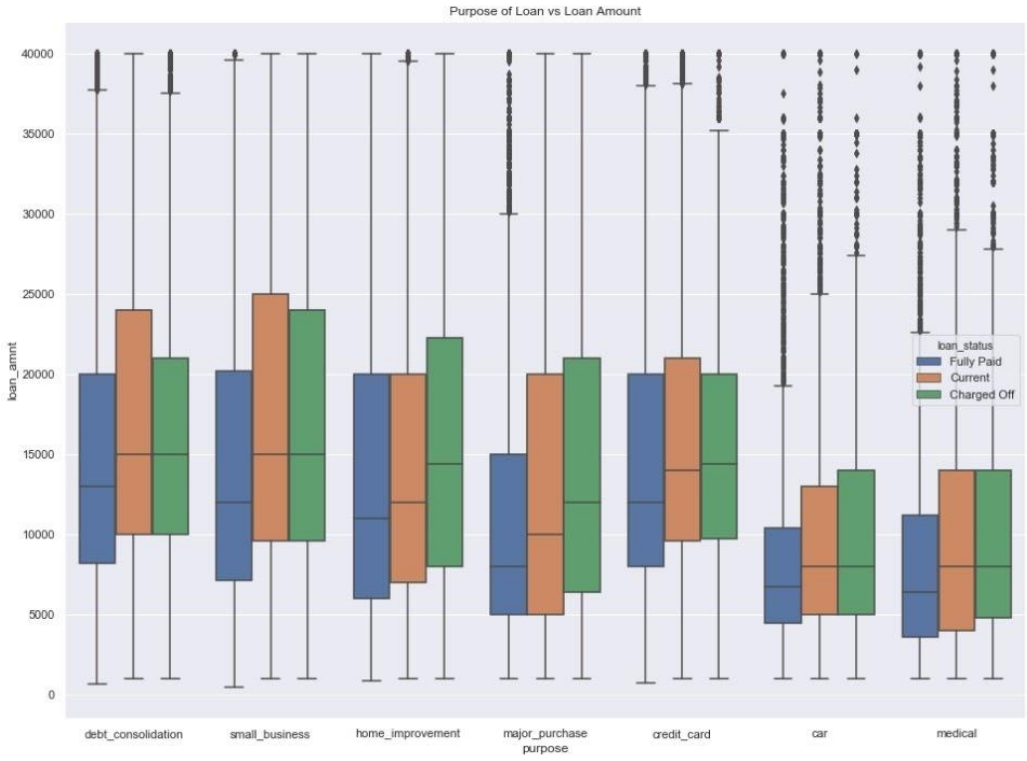
Purpose of Loan

- Most of the loans are taken for debt consolidation, credit card bills and home improvement and the charge off is also high for these loans



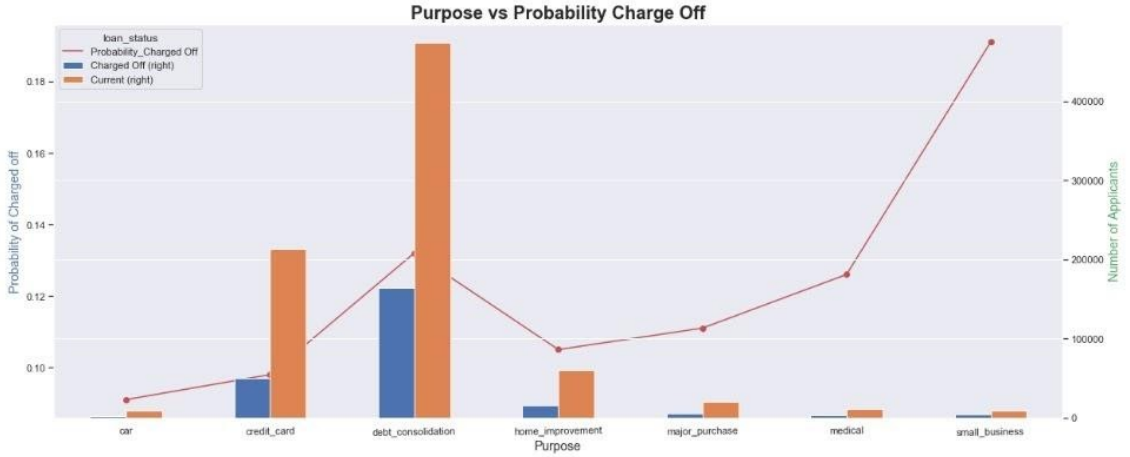
Purpose of Loan Vs Loan Amount

- For almost every loan purpose, the median loan amount for charged off loans is higher than the fully paid and current loans.
- Considering monitoring the loan amount would help reduce charge off probability



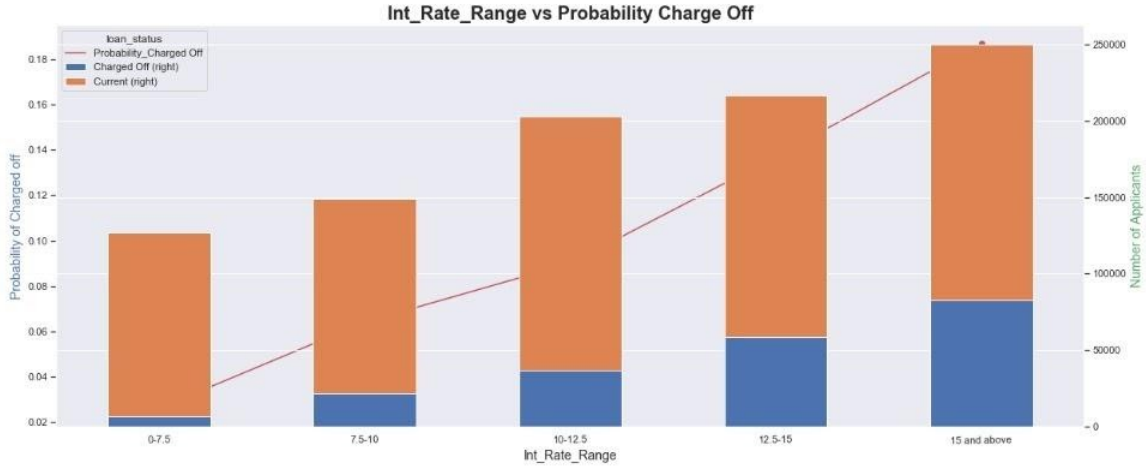
Purpose of Loan Vs Probability Charge Off

- Probability of charge off is really high for small business and debt consolidation loans.
- These types of loan should be monitored carefully



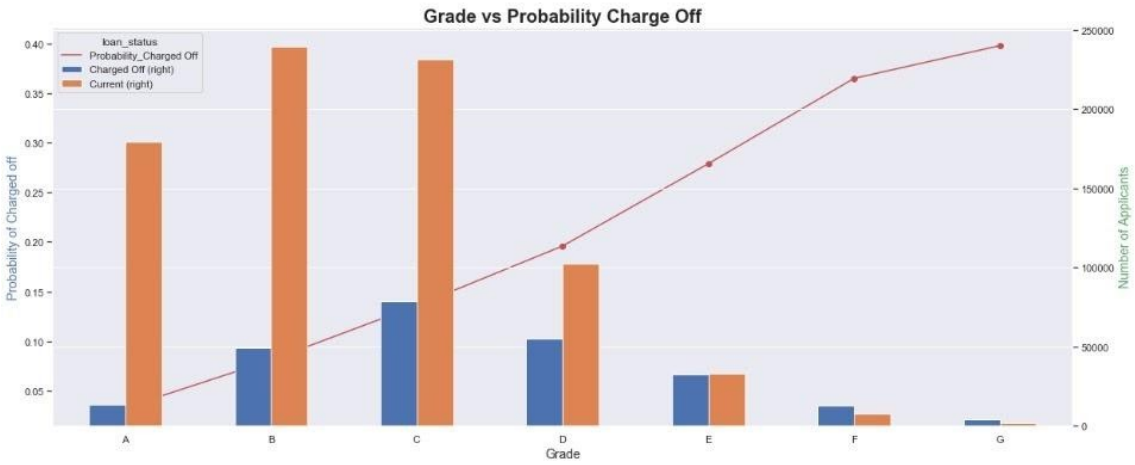
Interest Rate Range Vs Probability Charge Off

- Interest rate definitely affects the charge off loan.
- Loans with higher risk have high interest rate and in a way leads to charge off



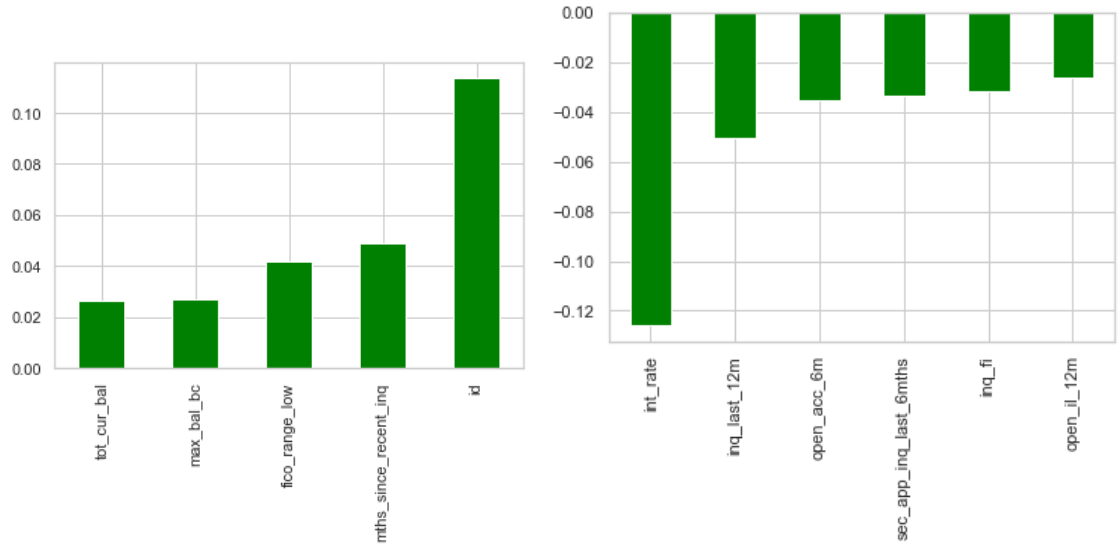
Grade Vs Probability Charge Off

- Loan grades are decided by the risk of each loan hence the probability of charge off increases with the grade
- Due to this reason, risky grades have low approval count



Highest Correlations with the outcome variable

- We computed the different feature correlations and saw which of them have the highest positive and negative correlations with the outcome variable.



Here features like id and months_since_recent_inquiry have the highest positive correlation and interest rate, inquiry_in_last_year have the highest negative correlation. This analysis helps understand the importance or weightage of the features with the Charge off probability. We can expect these variables to be in our model.

6. Modelling

Models Used

Project has 2 objectives - building a model to predict the probability of charge off on a current loan and identifying segmentations in the loan applicants to study behaviors of these segments. For the 1st objective, we built various Machine Learning classification models and compared their performance metrics (precision). For the 2nd objective, we built a clustering model to find the best number of clusters in the data.

- Classification Models
  - Naive Bayes Classifier
  - Random Forest
  - Logistic Regression
  - Neural Network

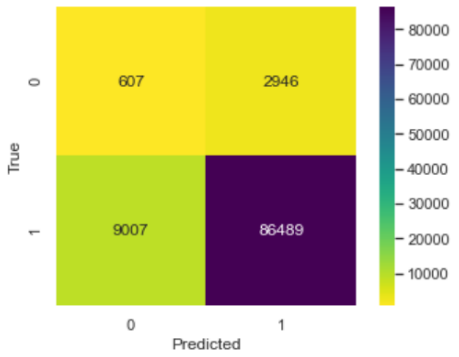
- Clustering Models
  - KMeans

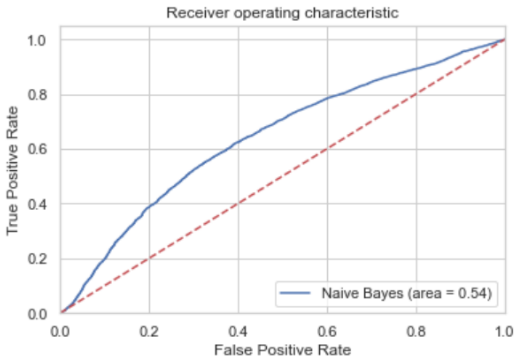
Classification - Naive Bayes Classifier

For these classifier algorithms, we used only information from 2018 because of the big running times that these types of models demand.

For Naive Bayes Classifier, we assume that there is independence among the different predictors.

- Model Accuracy: 0.879, although the accuracy is decent the precision is very low.





	precision	recall	f1-score	support
0	0.06	0.17	0.09	3553
1	0.97	0.91	0.94	95496
accuracy			0.88	99049
macro avg	0.52	0.54	0.51	99049
weighted avg	0.93	0.88	0.91	99049

Feature Importance - Naive Bayes Classifier

- We computed the feature importance of the Naive Bayes Classifier with the use of eli5 package in Python.

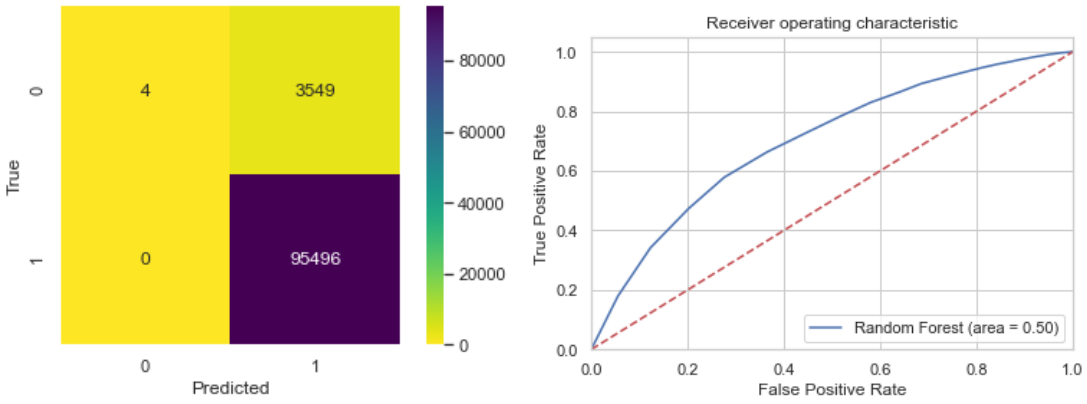
Weight	Feature
0.0023 ± 0.0005	sec_app_mths_since_last_major_derog
0.0012 ± 0.0003	bc_open_to_buy
0.0006 ± 0.0001	annual_inc_joint
0.0005 ± 0.0002	dti
0.0005 ± 0.0004	loan_amnt
0.0004 ± 0.0001	earliest_cr_line
0.0001 ± 0.0002	mths_since_recent_bc_dlq
0.0001 ± 0.0001	tot_coll_amt
0.0000 ± 0.0001	revol_util
0.0000 ± 0.0000	total_cu_tl
0.0000 ± 0.0000	grade_B
0.0000 ± 0.0002	emp_length
0.0000 ± 0.0001	delinq_amnt
0.0000 ± 0.0000	num_tl_90g_dpd_24m
0.0000 ± 0.0000	home_ownership_RENT
0 ± 0.0000	tax_liens
0 ± 0.0000	collections_12_mths_ex_med
0 ± 0.0000	grade_G
0 ± 0.0000	acc_now_delinq
0 ± 0.0000	grade_F
… 36 more …	

We can see that the most important features for predicting the repayment probability are:

- 'sec_app_mths_since_last_major_derog'
- 'bc_open_to_buy'
- 'annual_inc_joint'
- 'dti'
- 'loan_amnt'
- 'earliest_cr_line'

Classification - Random Forest

- 2018 info to achieve better running times
- Model Accuracy: 0.964, although accuracy is better than previous model, the precision is could be better

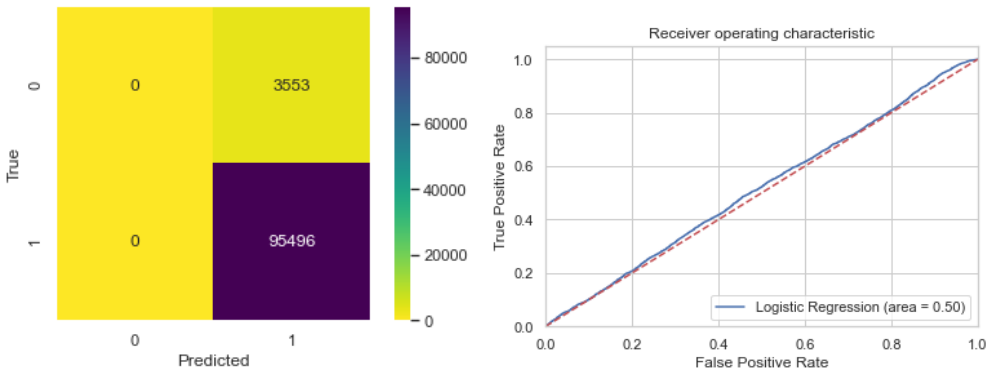


	precision	recall	f1-score	support
0	1.00	0.00	0.00	3553
1	0.96	1.00	0.98	95496
accuracy			0.96	99049
macro avg	0.98	0.50	0.49	99049
weighted avg	0.97	0.96	0.95	99049

Classification - Logistic Regression

Curiously, the running time for this algorithm was the highest of all Machine Learning models. Because of this, we picked only the most important features of the previous model to train the algorithm.

We achieved a Model Accuracy of 0.96, but the precision is very low again. This is the reason why we decided to create a Deep Learning Model to predict the repayment probability of the LendingClub loans.



	precision	recall	f1-score	support
0	0.00	0.00	0.00	3553
1	0.96	1.00	0.98	95496
accuracy			0.96	99049
macro avg	0.48	0.50	0.49	99049
weighted avg	0.93	0.96	0.95	99049

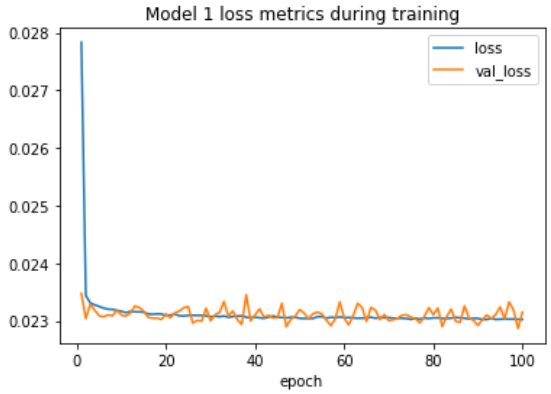
Classification - Neural Network Model

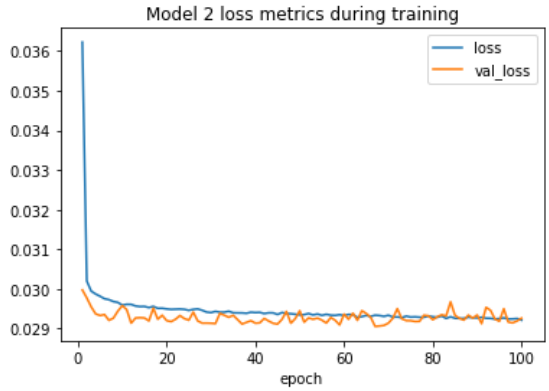
Since Machine Learning models did not give high precision, we tried implementing a Deep learning Neural network model. The model was built using several dense and dropout layers. We created 3 models based on different sets of features based on the feature engineering.

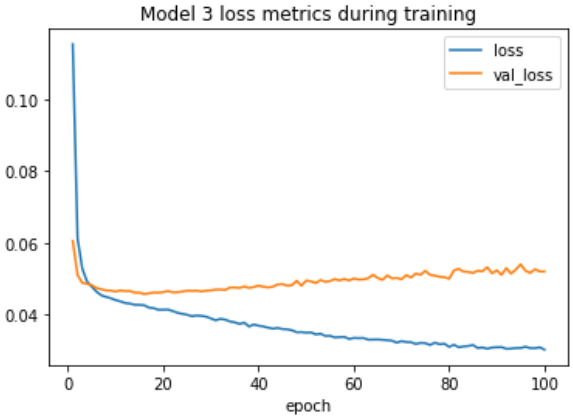
- Model parameters:
  - Activation function - ReLu
  - Optimizer - Adam
  - Loss function - mean_squared_logarithmic_error

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_8 (Dense)              (None, 64)                5376
_________________________________________________________________
dropout_6 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_9 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_7 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_10 (Dense)             (None, 16)                528
_________________________________________________________________
dropout_8 (Dropout)          (None, 16)                0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 17
=================================================================
Total params: 8,001
Trainable params: 8,001
Non-trainable params: 0
_________________________________________________________________
```

Classification - Neural Network Model Validation







Here is the model loss plot for training and validation data. As we can see model 3 has the highest loss where model 1 and 2 are comparable. So we checked the precision score for model 1 and 2 to pick the best model.

Classification - Neural Network Validation

The precision score for model 2 was the best compared to all the 3 models. The output metrics of the model is as follows.

```
Train Result:
================================================
Accuracy Score: 80.54%
________________________________________________
Classification Report:  Precision Score: 80.54%
                        Recall Score: 100.00%
                        F1 score: 89.22%
________________________________________________
Confusion Matrix:
 [[    30 172815]
 [     4 715287]]

Train Result:
================================================
Accuracy Score: 80.54%
________________________________________________
Classification Report:  Precision Score: 80.54%
                        Recall Score: 100.00%
                        F1 score: 89.22%
________________________________________________
Confusion Matrix:
 [[     9  43205]
 [     5 178816]]
```

```
Classification Report:
              precision    recall  f1-score   support

           0       0.64      0.00      0.00     43214
           1       0.81      1.00      0.89    178821

    accuracy                           0.81    222035
   macro avg       0.72      0.50      0.45    222035
weighted avg       0.77      0.81      0.72    222035


Confusion Matirx:
[[     9  43205]
 [     5 178816]]
```

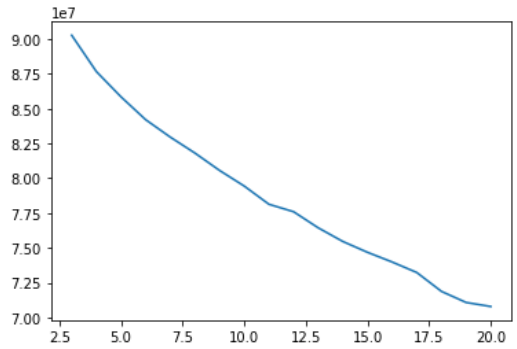
As seen the precision score for this model is much better compared to the machine learning model. Hence we finalize model 2 as the best model for this project.

Clustering - KMeans Model

After addressing the 1st objective of the project we move on finding customer segmentation. For this clustering algorithm is used. This is to find applicant segmentation to help understand different customer behaviors. Also identifying different segments will help lending organizations/ banks strategize better to handle the business operations.

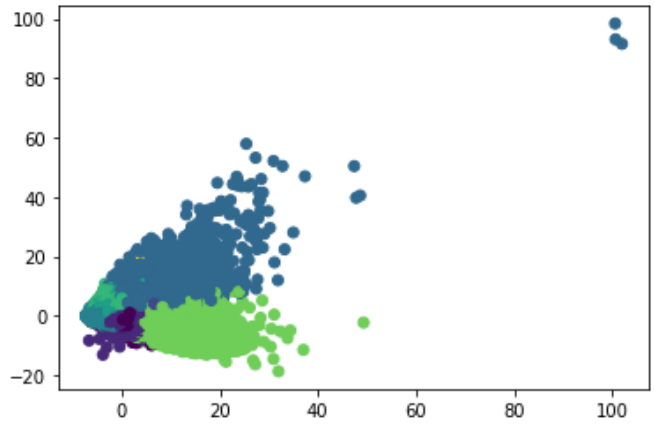
- Used PCA & KMeans algorithm to find clusters in the data

Clustering - KMeans Validation



- The loss graph does not flatten till k = 20
- Unfortunately, there are no evident clusters in the data
- Here is the cluster distribution for k = 10

```
79387945.44565827
```



Here we were expecting to find distant clusters based on the dataset but there are no obvious clusters here and hence we could not label every loan to a larger group.

7. Conclusion

From the analysis that was done over the LendingClub data, we can conclude the following

- Some of the EDA is very helpful understanding the data.
    - Most of the variables that were analyzed in the EDA were important features for the classification models.
- Precision is the best model performance metrics as we need to minimize False Positives.
    - Because we are working with unbalanced data (the number of paid loans was much higher than the number of unpaid loans), the precision is a much better accuracy measure.
- Machine learning models have much lower accuracy (& precision) compared to Deep Learning models.
    - Although Deep Learning Models have their limitations such as high computational complexity, in this case they did much better in the prediction of repayment probability when compared to machine learning algorithms.
    - A neural network can be a good model to find the probability of charge off.
- Further analysis can be done to achieve better accuracy and precision metrics:
    - Further analysis using CNN or RNN can help improve model accuracy.
    - Future analysis to improve precision can include extensive feature extraction.
- Finding customer segmentation based on customer background can help label new applicants faster and decide loan grades.

- After doing a process of feature selection, we could pick the best variables to use them in a new clustering model. It will surely give better results because there will be more distinction between the variables.
- An automated segmentation and labelling a new applicant to a cluster will help lending organizations quickly get an idea of what they are going to look into before even starting to study the loan application.