**CS5228**

**FINAL PROJECT REPORT**



|  |  |  |
| --- | --- | --- |
| **LEE TUCK CHONG**  **A0214477B** | **NEO WEI JIE ESMOND**  **A0140417R** | **SHI LI MIN**  **A0133934E** |

Team Name: Big\_Brain

ABSTRACT

*A “charged off” loan is defined as a loan that is written off by the lender as the borrower is unable to repay the debt. The objective of this project is to predict whether a loan will be charged off. Using the data set from the US Small Business Administration (SBA), we performed an analysis of the variables available in the data set, and built 3 models to attempt to predict whether the loans would be charged off. Based on preliminary data, these 3 models had marginal differences in performance with accuracy more than 93%.*

# **Table of Content**

[**Introduction**](#_hrz599kvho54) **3**

[Technical Details](#_zgpzf7cbvhx6) 3

[**Data Preprocessing and Exploration**](#_ogjt6k2lg65j) **3**

[Overview](#_81avo7cbiwel) 3

[Data Preprocessing](#_1v96euneg8wn) 3

[Encoding](#_h4mfn19np7a5) 3

[Time and Currency Series](#_gfuzcr4szzup) 4

[Missing Value Handling](#_ly35k2by8jys) 4

[Data Exploration](#_2emja5fwjmiw) 4

[NAICS Code Series](#_yv0bvzv7vbj5) 4

[Data correlation analysis](#_c6stpqh9t0g3) 4

[Approved Loans](#_mnk5kuy97r2f) 5

[Term](#_b4nzruz93cno) 6

[Zip](#_68wvwit1oc9d) 7

[Jobs](#_tw4kgres09qr) 7

[Post-exploration Actions](#_nstcdhmcp377) 7

[**Model Selection**](#_fhqoahola4sn) **8**

[Initial Experimentation](#_xtoza64408r3) 8

[Model Tuning](#_2vx4ayn6dhd8) 8

[XGBoost Tuning](#_e6ngfn1wla46) 9

[MLP Tuning](#_pg5382bu7xr5) 9

[Final Hyperparameters](#_nc84am5mhgqj) 9

[Additional Data Preprocessing](#_qhpa24uf0zsc) 10

[**Model Evaluation**](#_qkl6hsauufg9) **10**

[Final submission](#_wbutk9xxm0nf) 11

[**Bibliography and Appendix**](#_b6x85xw8zz3e) **11**

[Bibliography](#_iqbvgw1k2d2t) 11

[Appendix](#_r8x57p3vltqt) 11

# 

# **Introduction**

A loan is the lending of money by one party to the other party, in which the borrower incurs a debt and is usually required to pay interest on top of the principal amount. For the institution who is providing these types of financial services, it is of their interest that the borrower is trustworthy enough such that he would be able to return the loan. However, that is not always the case, as the borrower might default on the loan and be unable to pay up. These loans are called charged-off loans, and financial institutions would seek to avoid it as much as possible.

As such, it is of their interest to be able to predict whether a loan would be likely to be charged off given the data provided by the prospective borrower. The financial institution could then choose to reject the loan should there be a risk of the loan being charged off .

The dataset that is used for this project is a census from the United States Small Business Administration (SBA), and is compiled by Min Li, Amy Mickel and Stanley Taylor of California State University in Sacramento. This dataset provides historical data from 1987 through 2014. Each observation represents a loan that was guaranteed to some degree by the SBA.

The data reside in a comma-separated values (csv) file. A header line contains the name of the variables. The following are all the variables available in the dataset:

## Technical Details

The code for this project is written as a Jupyter Notebook with Python 3.7.6. We used plotly, seaborn and autoviz library to conduct our data exploration by visualizing the variables in the data. We also used predominantly scikit-learn (sklearn) to construct the models and tune the hyperparameters, with XGBoost (XGB), LGBM and CATBoost (CATB) as external models used to construct the classifier.

# **Data Preprocessing and Exploration**

## Overview

|  |  |
| --- | --- |
| **Number of variables** | 24 |
| **Number of observations** | 50000 |
| **Rows contain missing value** | 409 |
| **Rows contain missing value (%)** | >1% |
| **Duplicate rows** | 0 |
| **Duplicate rows (%)** | 0% |

*Figure 1: Dataset Statistics*   *Figure 2: Distribution of “Charge-off” loans*

The proportionately low number of rows containing missing values and the absence of duplicate rows suggests that the data quality is high. Furthermore, the distribution of charge-off loans are well balanced. Other than F1, accuracy can be used as a metric to evaluate the model due to the well-balanced dataset.

## Data Preprocessing

### Encoding

For this project, target encoding is performed. As more than half of columns are categorical features, it would be inconvenient for us to perform any machine learning algorithm. Most machine learning algorithms require numerical input data. target encoding helps to convert all categorical variables into numerical value. Simplicity of implementation and ability to provide correlation between features and target (i.e. ChargeOff) are also part of considerations.

We used target encoding and numerical encoding by default except for CATBoost. As CatBoost has its own encoding method, target encoding is not required in preprocessing.

### Time and Currency Series

Simple transformation is performed on Time and currency series. Our team simply removes special characters ($) and converts currency into float numbers. For the datetime, we convert it into integers (i.e. multiplying the year by 100 and converting the month into quarter) which is convenient for the model to do fast processing. This method of encoding datetime may preserve both their ordinality and relationships. Eg: $10,000.1--->10000.1 , 09-Mar-05 ---> 20050309. We removed the days as intuitively it is difficult to justify.

### Missing Value Handling

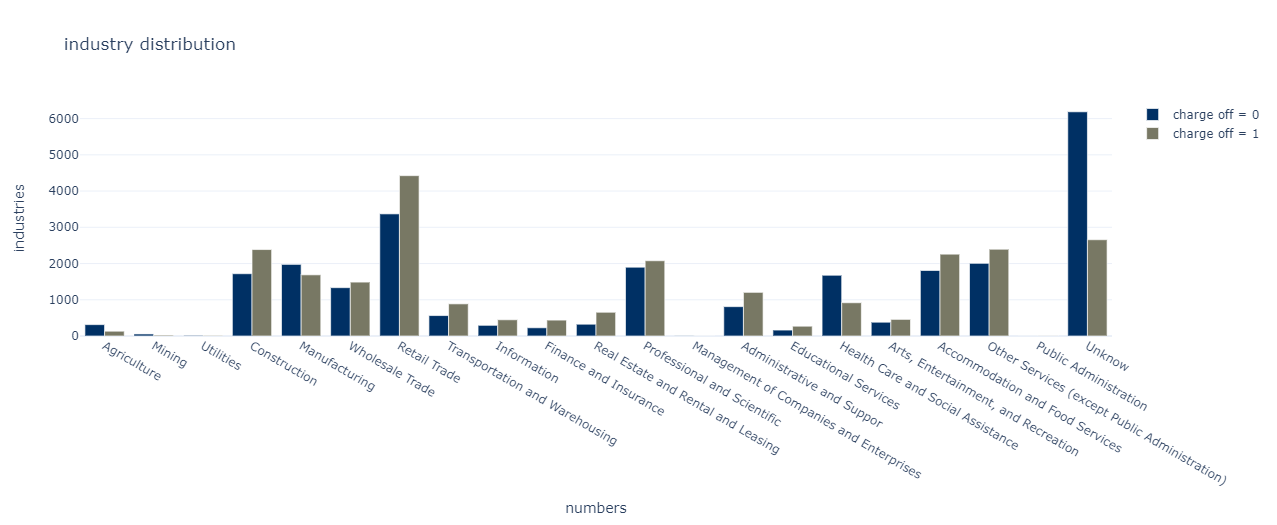
There are 485 missing cells in this dataset ( < 0.1% missing cells rate), which are considered as a very small portion of data. We apply three method to fill in the missing value:

* Label encoding auto-fill for categorical except CATboost
* Use interpolation fill up with numerical value for all models
* Fill ‘UNKNOWN’ to categorical value only for CATboost

## Data Exploration

### NAICS Code Series

The North American Industry Classification System (NAICS) is used by statistical agencies for the collection, analysis and publication of statistical data related to the US Economy. The NAICS Code is the identifier used to classify the business into groups. The most specific portion of the code is the first 2 digits, which contains the industrial sector that the business is in. The first two digits of NAICS code shows the detailed industry area as follows (United States Census Bureau, 2012):



*Figure 3: Histogram of charge off according to industrial sectors*

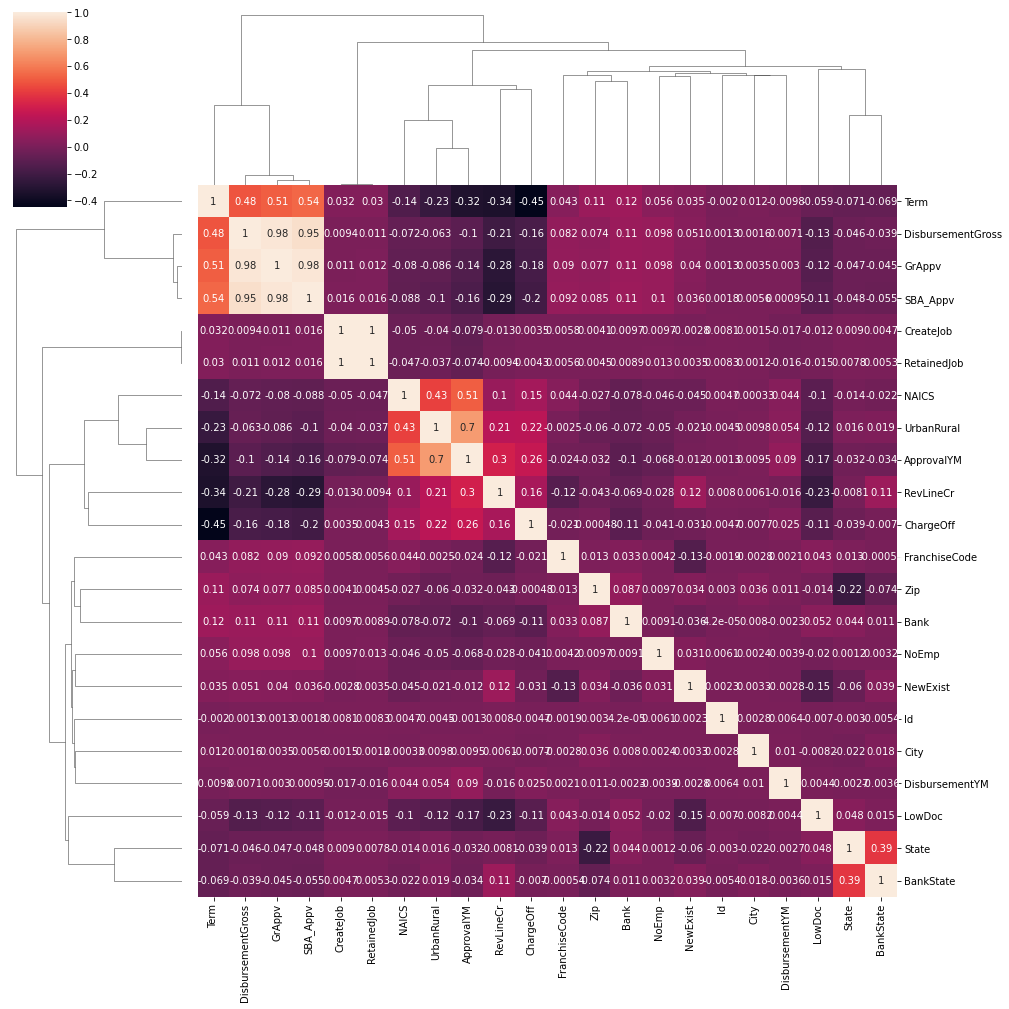
Based on the histogram, there is strong evidence that suggests that the industrial sector could be a strong predictor for a loan to be charged off. For example, the healthcare, social assistance and unknown industrial sectors have a higher proportion of charged off loans.

Due to the above result, we take the first two digits of NAICS code to help model generalize the result instead of full code.

### Data correlation analysis

By reading the heat map, ‘Term’ is the variable with the strongest correlation. This is followed by ‘DisbursementGross’, ‘GrAppv’, ‘SBA\_Appv’, ‘NAICS’, ‘UrbanRural’, ‘ApprovalYM’, ‘RevLineCr’, ‘Bank’ and ‘LowDoc’.

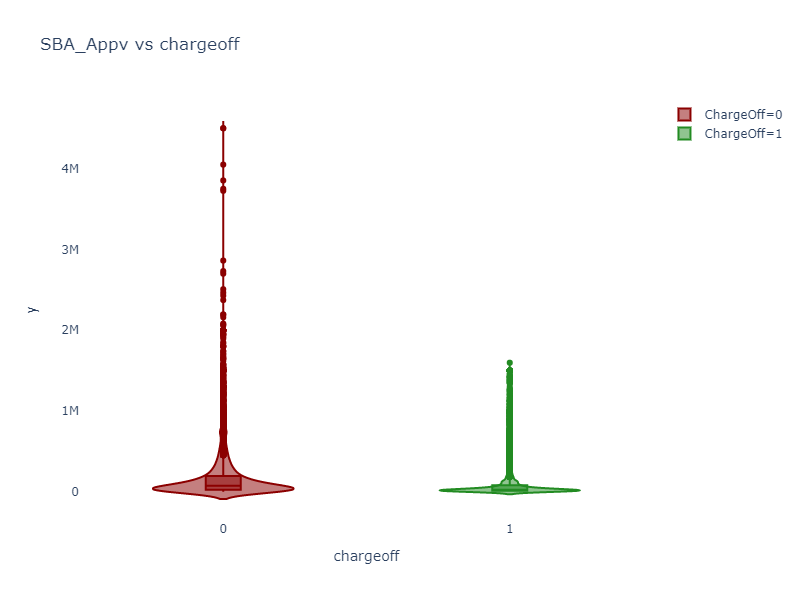
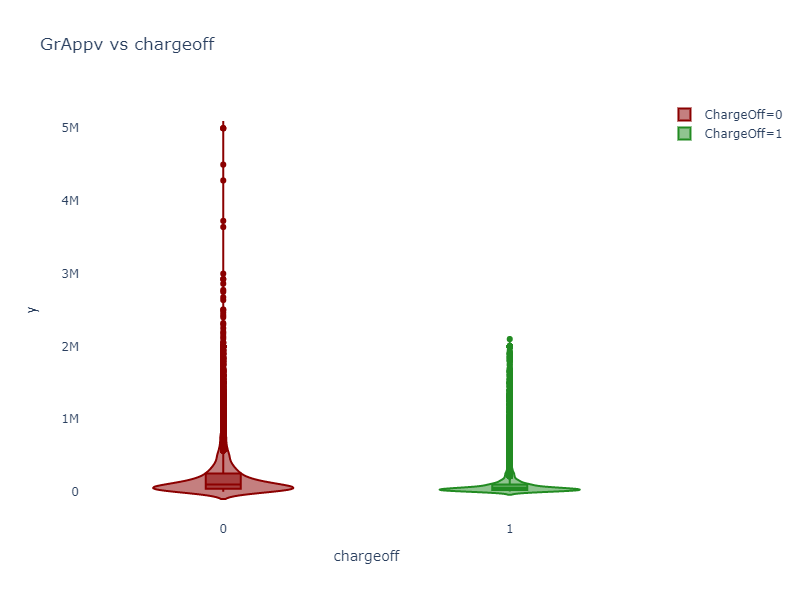
Some of these terms are also strongly correlated with each other (for example: ‘GrAppv’ and ‘SBA\_Appv’). We will seek to reconcile such issues in the next subsections.

****

*Figure 4: Heatmap of charge off and other features*

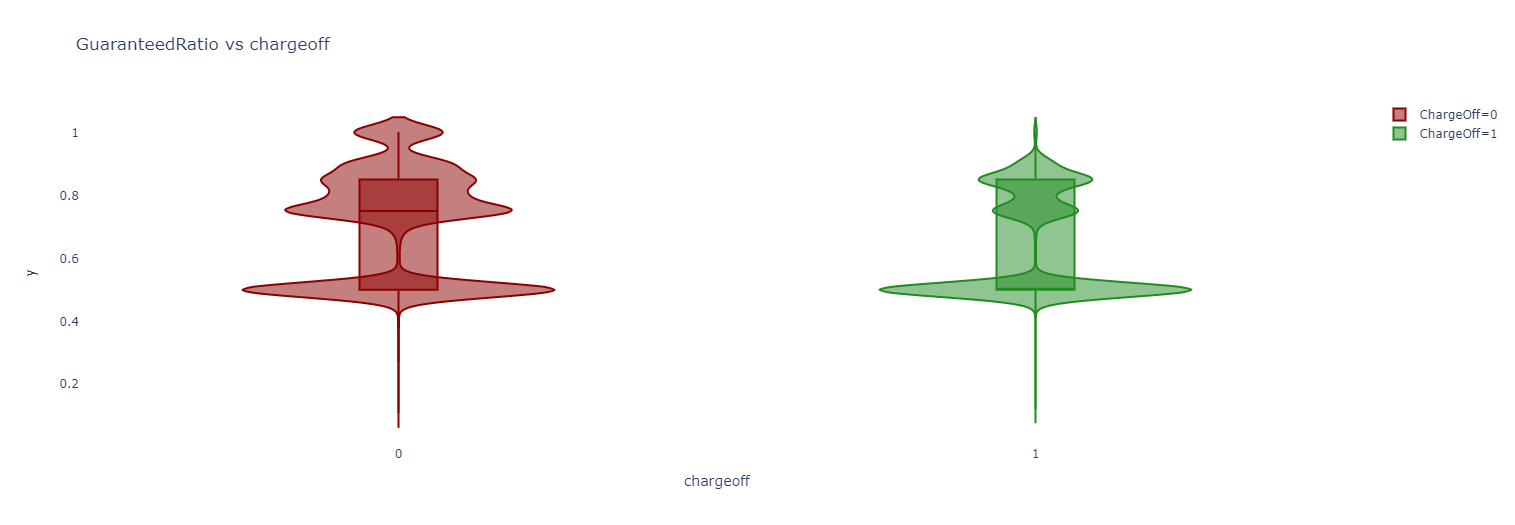
### Approved Loans

Below plots are the violin plots of ‘GrAppv’(Gross Amount of Loan Approved by Bank, left) and ‘SBA\_Appv’(SBA’s Guaranteed Amount of Approved Loan, right).The distribution is similar between ‘ChargeOff’ 0 and 1 in both figures. The only information we can get is that if the amount is higher than 2 million, the loan is more likely to be paid back. However, the majority of the amount is below half of a million, which means we need to find a more informative manner to evaluate the gross amount and SBA’s guaranteed amount.



*Figure 5: Violin plot of between charge off and GrpAppv (left), charge off and SBA\_Appv (right)*

Findings from SBA (U.S. Small Business Administration, 2011) suggests that SBA’s Guaranteed Portion of Approved Loan (‘GuaranteedRatio’) is another important risk indicator. This ratio could be calculated by dividing SBA\_Appv by GrAppv. The following violin plot shows the SBA’s Guaranteed Portion distribution in charge off and no charge off respectively. We could easily interpret that the two graphs’ distributions are different, especially for the ratio larger than 0.6. As a result, we adapt SBA’s Guaranteed Portion as our additional new feature to help us with the further classification.

****

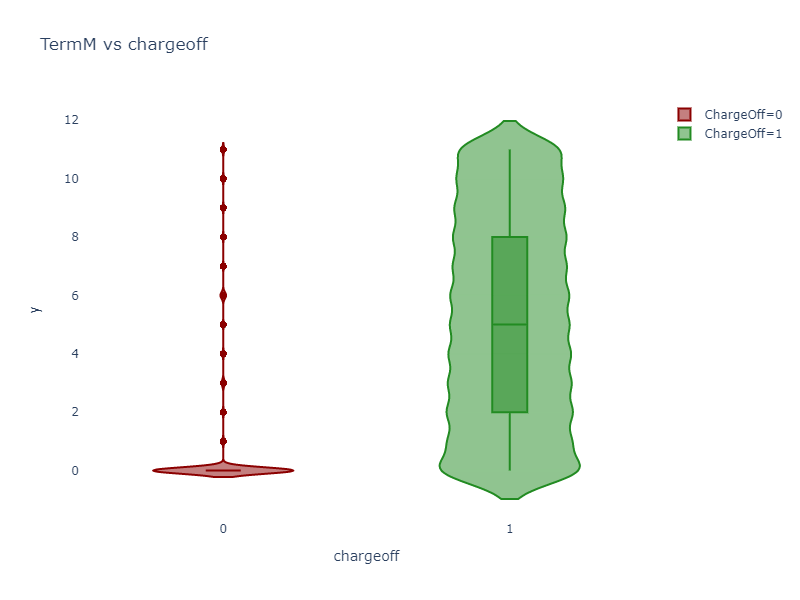
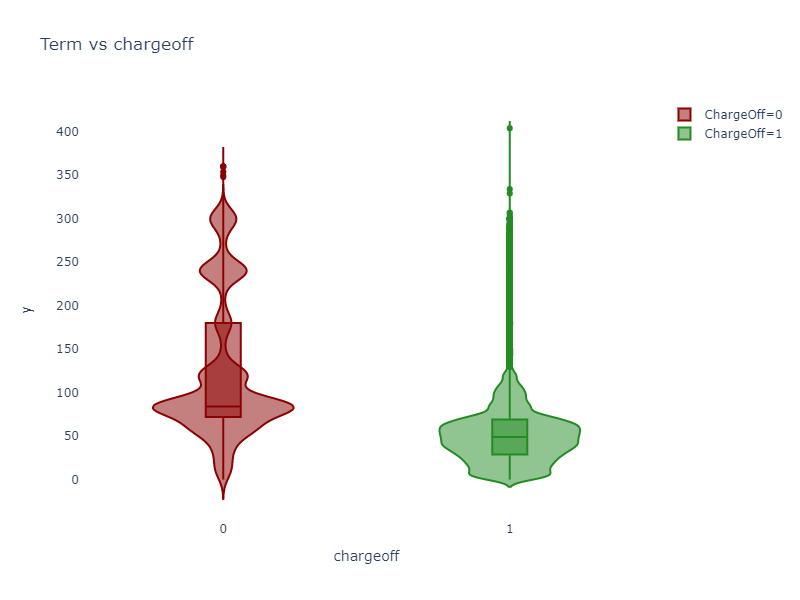
*Figure 6: Violin plot of between charge off and guarantee ratio*

### Term

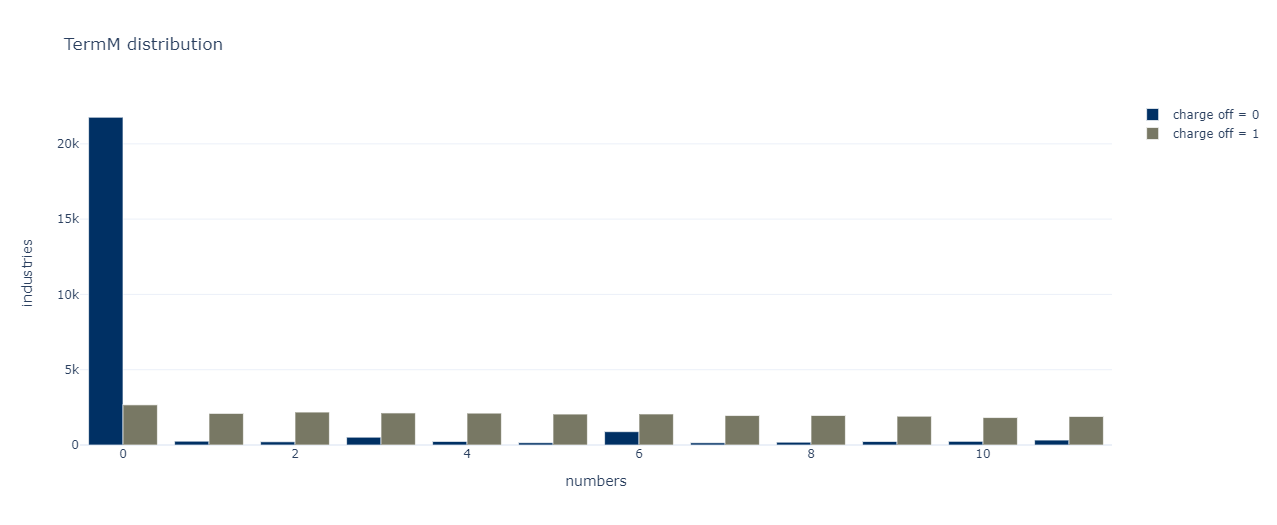
The ‘Term’ series shows to be the most relevant feature with ‘ChargeOff’ according to the heatmap plot shown above. The ‘Term’ refers to the loan term in months. We plot out the violin figure, and find out that most people tend to pay back for long term loan rather than short term. We discovered that the distribution of ‘ChargeOff = 0’ is more likely to be gathered around the month in multiples of 12, which is a year cycle.

As a result, we perform modulo operations towards ‘Term’. We define the series “TermM: as follows:

Based on the violin diagrams, loans with non-zero terms of ‘TermM’ are more likely to be charged off. One possible intuition is that non-zero term of ‘TermM’ is more likely for short-term loans rather than long term ones, which could result in a stress in cash flow for the businesses.

****

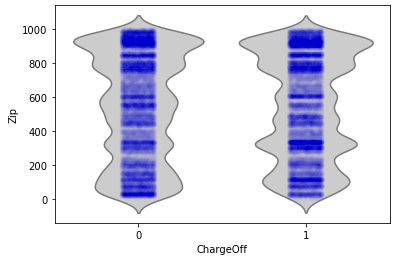
*Figure 7: Violin plot of between charge-off and Term (left), charge-off and TermM*

****

*Figure 8: Histogram of charge off according to termM*

### Zip

Five digits zip code is too specific for the classification. To reduce the probability of overfitting, our group decided to extract the first three digits of zip code. The first three digits give a more general location information. The Catplot further suggests that there is a possible clustering of charged off loans based on the first 3 digits. We decided to omit the last 2 digits of the Zip code.

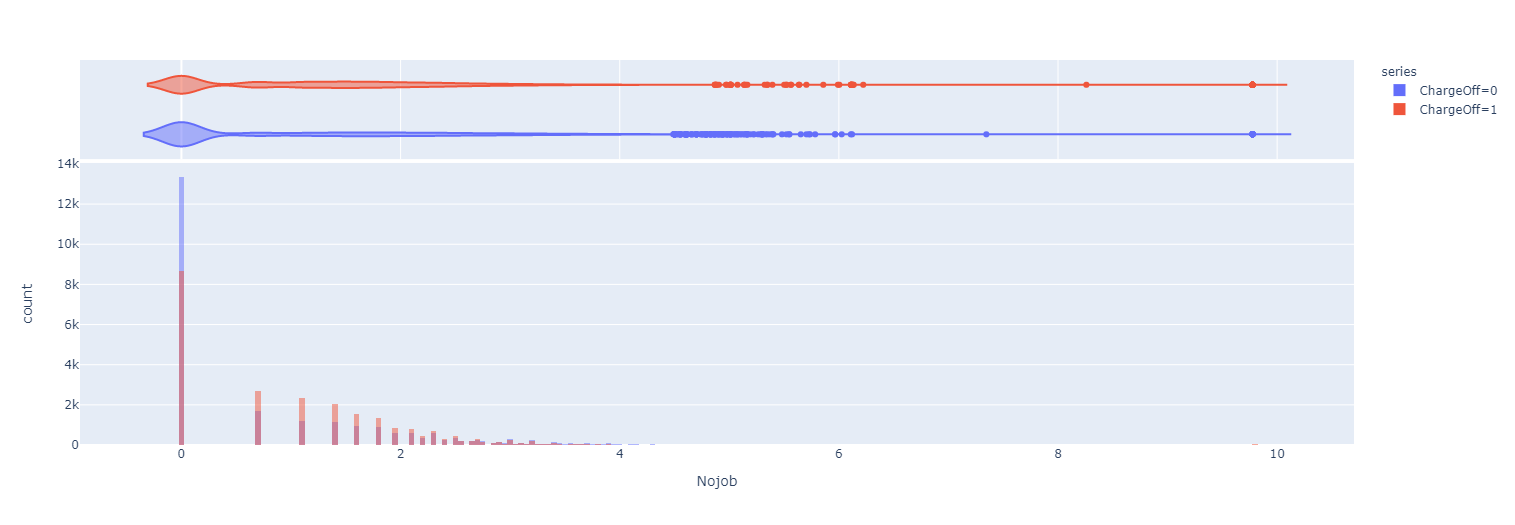


*Figure 9: Catplot between charge off and zip (front 4 digits)*

### Jobs

As ’RetainedJob’ and ‘CreatedJob’ are perfectly correlated. we merge these two columns into one column called ‘NoJob’ by addition operation. We plotted the histogram figure with the log operation applied to the x axis, and we also charted a violin chart showing the distribution of total number of employment.

Based on the figures, we found that loans that neither creates nor retains jobs are less likely to be charged off, and vice versa.



*Figure 10: Histogram figure of total numbers of jobs with the log operation applied to the x axis*

### Post-exploration Actions

After performing the data exploration, we dropped some of the columns to reduce data dimensionality without losing important information.

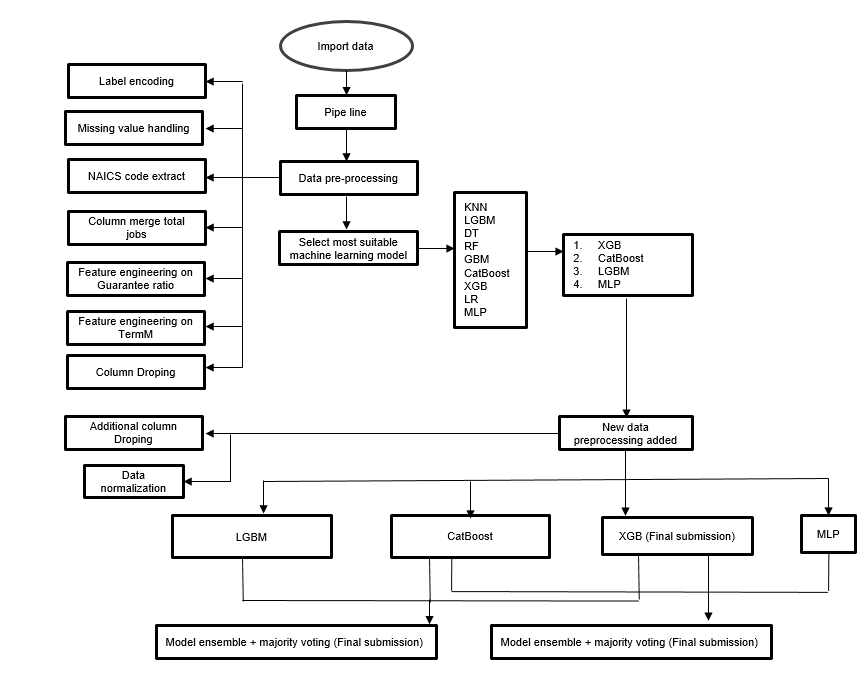
* **‘ID’, ‘Name’** and **‘BalanceGross’** are dropped. ID and Name have no contribution to the decision making of the final outcome in terms of correlation and intuition. BalanceGross column only has one distinct value ‘0’.
* **’RetainedJob’**  and **‘CreatedJob’** are dropped due to the collation of these 2 columns into the ‘NoJob’ column.
* **‘GrAppv’** and **‘SBA\_Appv’** are dropped. SBA’s Guaranteed Portion of Approved Loan (**‘GuaranteedRatio’**) is the combination of these two features and is more informative.
* ‘**ApprovalDate’** and **‘DisbursementDate’** are dropped in favour of **‘ApprovalDateYM’ and ‘DisbursementDateYM’**

Based on the data explorations, we created or modified the following columns:

* ‘**TermM**’ that stores the remainder when dividing ‘Term’ by 12.
* ‘**NoJob**’ which aggregates ’RetainedJob’ and ‘CreatedJob**’.**
* ‘**Zip**’ only considers the first 3 digits.
* ‘**NAICS Code**’ only considers the first 2 digits.
* **‘GuaranteedRatio’** which consist of information from ‘GrAppv’ and ‘SBA\_Appv’.
* ‘**ApprovalDateYM’** and **‘DisbursementDateYM’** contains the year and month information of ‘ApprovalDate’ and ‘DisbursementDate’.

# **Model Selection**

## Initial Experimentation

****

*Figure 11: Code structure figure*

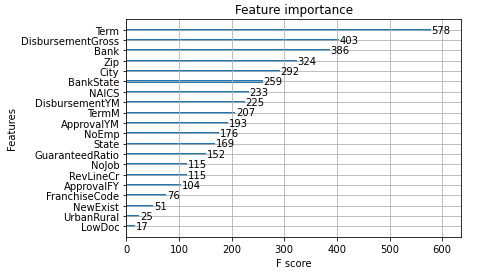
We built up a data pipeline for all models to have a performance preview based on their default hyperparameters. We would then choose some better-performing model to fine tune.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **KNN** | **LR** | **DT** | **RF** | **GBM** | **MLP** | **XGB** | **LGBM** | **CATB** |
| **Acc.** | 69.29 | 71.98 | 89.19 | 92.56 | 91.94 | 81.59 | 93.45 | 93.14 | 93.28 |
| **F1** | 69.28 | 71.76 | 89.19 | 92.56 | 91.94 | 81.40 | 93.45 | 93.14 | 93.28 |

Based on the metrics, we selected **XGBoost** for tuning. We also selected **MLP** as an alternative model for tuning as it is the best performing model that is not related to tree-based search. For the final model, we also considered the addition of LGBM and CATBoost into the model.

## Model Tuning

For XGB and LGBM, we dropped 4 feature columns of lowest scores (i.e. LowDoc, UrbanRural, NewExist, FranchiseCode) according to the importance plot generated by the XGBoost model to obtain a better mean and variance of the result in 10-fold cross validation (i.e. mean: 0.968; variance: 0.001). Usually those dropped features have their correlated others (such as LowDoc with RevLineCR).



*Figure 12: XGBoost importance plot*

Specifically for MLP, we apply standard scaler to data, and retain the columns without any other advanced data processing.

### XGBoost Tuning

XGBoost achieved the best score in the model comparison step. Furthermore, we start to fine tune the hyperparameter for XGBoost. Random search with 10-fold cross validation (Kohavi, 1995, 1137-1143) is applied to have a broader search range with a reasonable time cost. In total, it fits 10 folds for each of 100 candidates, totalling 1000 fits.

The strategy is wide search first, then narrow down through manually adjusting the variance, and fix the mean of the distribution close to the previously tuned parameters. Hyperparameter table could be found below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **colsample\_bytree** | **colsample\_bylevel** | **colsample\_bynode** | **gamma** | **learning\_rate** |
| uniform(0.9, 0.1) | uniform(0.9, 0.1) | uniform(0.9, 0.1) | uniform(0.4, 0.1) | uniform(0.2, 0.1) |
| **max\_depth** | **n\_estimators** | **subsample** | **reg\_lambda** | **reg\_alpha** |
| randint(5, 9) | randint(120, 170) | uniform(0.9, 0.1) | uniform(1, 1) | uniform(0, 1) |

The 3 most important parameters identified (Chen & He, 2020) is n\_estmators, max\_depth and learning\_rate as it plays an important role in controlling the model performance and minimizing the opportunities of overfitting or underfitting. We further improved the tuning by experimenting with additional hyperparameters of XGBoost.

### MLP Tuning

The strategy for MLP tuning is manually tweaking the neural network architecture. It consists of an input layer, 2 hidden layers of size (32,8) with 'tanh' as the activation function and an output layer. The rest of the parameters remain as default.

We decided to use 2 hidden layers and tuned the number of neurons to avoid overfitting and underfitting. We performed grid search on the activation function and found that tanh provides the best results.

### Final Hyperparameters

We used the default hyperparameters for CATBoost and LGBM.

|  |  |
| --- | --- |
| **Model** | **Hyperparameters (to 5 significant figures)** |
| **XGB** | **'colsample\_bylevel**': 0.92502, **'colsample\_bynode**':0.95492, **'colsample\_bytree'**: 0.97146, **'gamma'**: 0.46602, **'learning\_rate'**: 0.22799, **'max\_depth**': 8, **'n\_estimators'**: 158, **'reg\_alpha'**: 0.65711, '**reg\_lambda**': 1.4357, '**subsample**': 0.97300 |
| **MLP** | **Layer size:** (32,8), **Activation function:** tanh |

## Additional Data Preprocessing

We found the XGBoost has the best prediction score. We compare the three different data preprocessing methods on these two models by using accuracy and F1 score.

|  |  |  |
| --- | --- | --- |
| **Preprocessing method** | **MLP** | **XGBoost** |
| **Dropped additional columns** | Accuracy: 61.82  F1: 60.78 | Accuracy: 93.46  F1: 93.46 |
| **Standard scaler** | Accuracy: 90.7  F1: 90.7 | Accuracy: 93.18  F1: 93.18 |
| **Dropped additional columns & Standard scaler** | Accuracy: 91.1  F1: 91.1 | Accuracy: 93.45  F1: 93.45 |

For MLP, it requires features to be numerically represented in order to work as expected. Data normalization (i.e. we used StandardScaler) is necessary for ANN as the values for the features must be of similar scale so that the backpropagation algorithm can work properly/efficiently/effectively by updating the model parameters without any major problems (e.g. gradient vanishing/exploding).

For XGBoost (Chen & He, 2020), it requires minimal data preprocessing (for continuous variables) because it tests each of the thresholds (greedy algorithm) to determine the best split. No data normalization is needed. But reducing dimensionality is required for XGBoost, which removes noise and improves model results.

# 

# **Model Evaluation**

We mainly split the training dataset into training and validation with a ratio of 0.25. By comparing the accuracy and f1 score of the validation, we evaluate the model performance based on validation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **XGBoost** | **MLP** | **LGBM** | **CATboost** |
| **Accuracy** | 93.46 | 90.7 | 93.1 | 93.2 |
| **F1** | 93.46 | 90.7 | 93.1 | 93.2 |

From the table we could see XGBoost gives the best scores on the validation dataset.

## Final submission

We decided to candidate three models consisting of one single model and two assembled models. In assembled model, we perform majority hard voting (MV) techniques, which aggregate the outputs of a collection of 3 different models (i.e. XGBoost, CATBoost & MLP) by selecting the most occurred for each test instance. MV1 is the model with the 3 best performing models from our initial testing, and MV2 is the best 2 with MLP, with the logic that having a more varied ensemble to cover the shortcomings of tree-based models.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **F1** | **Leaderboard** |
| **XGBoost** | 93.46 | 93.46 | 0.93820 |
| **MV1 (XGBoost + LGBM + Catboost)** | 93.54 | 93.53 | 0.93940 |
| **MV2 (XGBoost + MLP + Catboost)** | 93.43 | 93.41 | 0.93745 |

# 

# **Bibliography and Appendix**

## Bibliography

Chen, T., & He, T. (2020, 9 2). *xgboost: eXtreme Gradient Boosting Package Version: 1.2.0.1*. http://cran.fhcrc.org/web/packages/xgboost/vignettes/xgboost.pdf

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *e International Joint Conference on Articial Intelligence (IJCAI)*, 1137-1143. http://ai.stanford.edu/~ronnyk/accEst.pdf

United States Census Bureau. (2012). *North American Industry Classification System.* https://www.census.gov/cgi-bin/sssd/naics/naicsrch?chart=2012

U.S. Small Business Administration. (2011). *The SBA Loan Guarantee Program: How it Works.* https://www.sba.gov/sites/default/files/SDOLoanFactSheet\_Oct\_2011.pdf

## Appendix

Violin plots for state, full zip code, bank, bank state, city, full NAICS code, approvalFY, NoEmp and RevLincr.

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |