# Final Project

**Kai Xu (A0198855E)**
Department of Computer Science
National University of Singapore
e0403945@u.nus.edu

**YuTing Tseng (A0212195L)**
Department of Computer Science
National University of Singapore
e0503474@u.nus.edu

Team: The Phantom of the Cash

**Abstract**

*This is a report for the final project of CS5228, a Kaggle competition. In this competition, we get the data only from https://www.kaggle.com/c/cs5228-2021s1/data and manage to predict whether the loan should be approved or denied. This report contains the steps and approaches we take, including: 1. data preprocessing (transform the data format in willing ways); 2. data analysis (observe the relationship between features and outcome); 3. model analysis (give a trial on various models and evaluate those algorithms); 4. conclusion (interpret of our methods).*

## 1 Data Preprocessing

In order to process the data set, we first need to understand the format of each attribute and the meaning they represent. The following table provides variable names, data format and brief descriptions for 24 attributes.

| Variable Name | Format | Description | Example |
|---|---|---|---|
| Name | Text | Borrower name | EAST L R |
| City | Text | Borrower city | LOS ANGELES |
| State | Text | Borrower state | CA |
| Zip | Int | Borrower zip code | 90022 |
| Bank | Text | Bank name | BANK OF AMERI |
| BankState | Text | Bank state | NC |
| NAICS | Int | NAICS Code | 811430 |
| ApprovalDate | Date | Date SBA commitment issued | 29-Jul-02 |
| ApprovalFY | Date | Fiscal year of commitment | 2002 |
| Term | Int | Loan term in months | 60 |
| NoEmp | Int | Number of business employees | 32 |
| NewExist | Int | 1: Existing business 2 :New business | 1 |
| CreateJob | Int | Number of jobs created | 12 |
| RetainedJob | Int | Number of jobs retained | 20 |
| FranchiseCode | Int | Franchise or Not | 1 |
| UrbanRural | Int | 1: Urban, 2: rural, 0: undefined | 1 |
| RevLineCr | Text | Revolving line of credit. | Y |
| LowDoc | Text | LowDoc Loan Program | N |
| DisbursementDate | Date | Disbursement date | 31-Aug-02 |
| DisbursementGross | Currency | Amount disbursed | $17,000.00 |
| BalanceGross | Currency | Gross amount outstanding | $0.00 |
| GrAppv | Currency | Gross amount of loan approved | $17,000.00 |
| SBA_Appv | Currency | Amount of approved loan | $8,500.00 |
| Outcome | Int | ChargeOff or Not | 0 |

### 1.1 Text Data

We convert State into state number, and "Y" and "N" to 0 and 1 for RevLineCr and LowDoc. For other text data, we drop them as they are not correlated to the outcome.

### 1.2 Numerical Data

Currency: we convert it to float by regular expression. For example:

```
data["DisbursementGross"].replace("[\$,]", "", regex = True).astype(float)
```

### 1.3 Data/Time Data

Date: we convert it to int by regular expreesion and Timestamp. For example:

```
pd.to_datetime(data["ApprovalDate"], format="%d-%b-%y").astype(int) / (10**11)
```

## 2 Data Analysis

### 2.1 State

As the economic situation of each state is different, the loan default situation is also different. The figure below shows the default ratio of loans represented by the state.

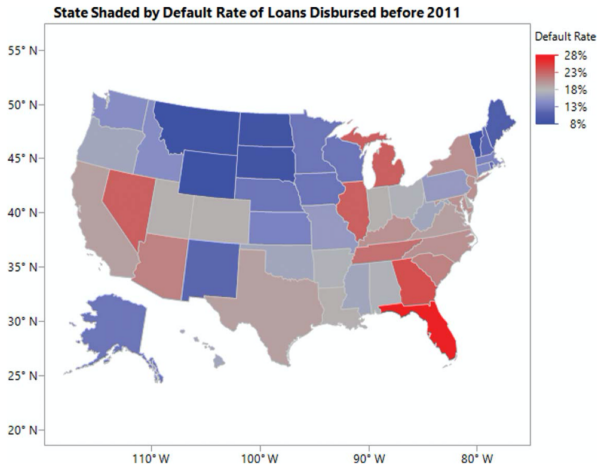


Figure 1: State Default Rate

### 2.2 ApprovalFY

The above table represents the loan disbursement situation in each year, including the total number of loan disbursements and the disbursement rate. Taking the Great Depression (2007 to 2009) as a division, we can observe: 1. Prior to 2007, loans to enterprises had been increasing. In 2007, there was a cliff-like decline, after which the number of loans issued has been declining (until 2014 in this data set). 2. Before 2004, banks were relatively conservative in issuing loans, but the proportion of loans was increasing year by year. After 2005, even during the Great Depression, their loan issuing rates gradually increased. (This should be due to the government's support for specific industries. The picture below also represents this tendency.) We therefore made a handcrafted feature `Recession` identifying whether the loan is active during Recession.

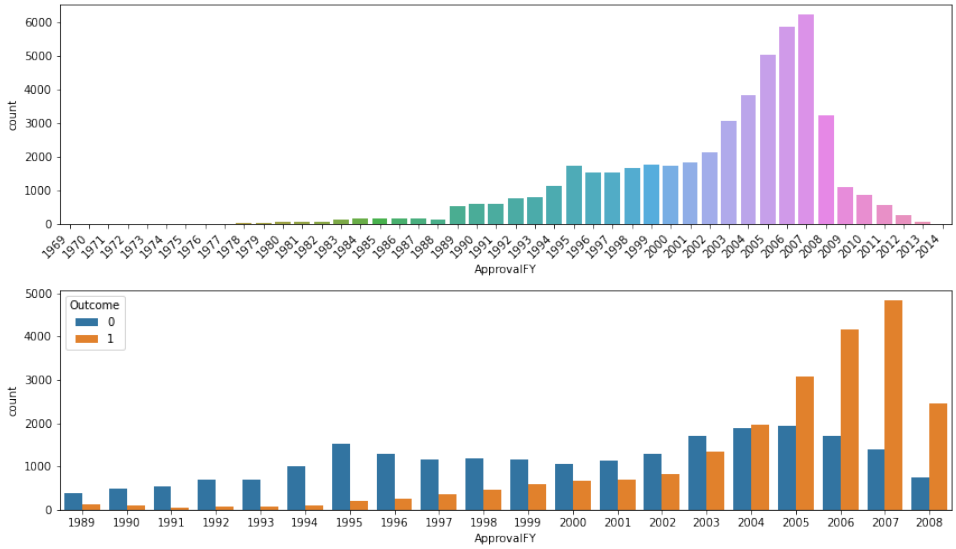


Figure 2: Loan proposed corresponding to ApprovalFY

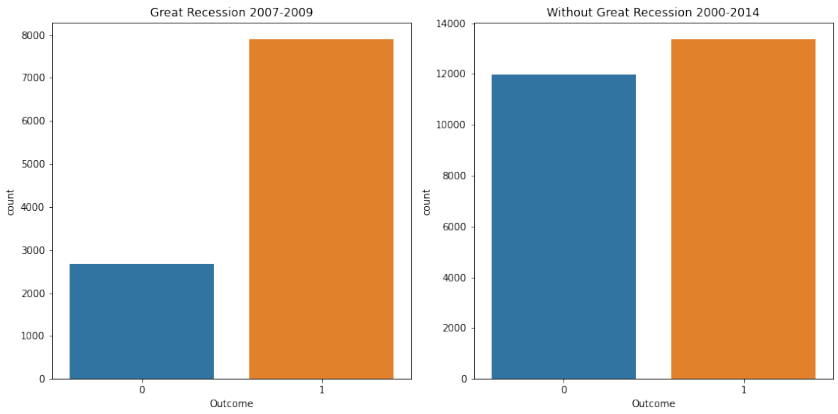


Figure 3: The impact of Great Recession on the charge-off rate.

## 2.3 NAICS

**Table 3.** Industry default rates (first two digit NAICS codes).

| 2 digit code | Description | Default rate (%) |
|---|---|---|
| 21 | Mining, quarrying, and oil and gas extraction | 8 |
| 11 | Agriculture, forestry, fishing and hunting | 9 |
| 55 | Management of companies and enterprises | 10 |
| 62 | Health care and social assistance | 10 |
| 22 | Utilities | 14 |
| 92 | Public administration | 15 |
| 54 | Professional, scientific, and technical services | 19 |
| 42 | Wholesale trade | 19 |
| 31–33 | Manufacturing | 19, 16, 14 |
| 81 | Other services (except public administration) | 20 |
| 71 | Arts, entertainment, and recreation | 21 |
| 72 | Accommodation and food services | 22 |
| 44–45 | Retail trade | 22, 23 |
| 23 | Construction | 23 |
| 56 | Administrative/support & waste management/remediation Service | 24 |
| 61 | Educational services | 24 |
| 51 | Information | 25 |
| 48–49 | Transportation and warehousing | 27, 23 |
| 52 | Finance and insurance | 28 |
| 53 | Real estate and rental and leasing | 29 |

NAICS stands for North American Industry Classification System. It is a 2 to 6-digit hierarchical classification system used by the Federal Statistics Agency to classify commercial organizations. The first two digits represent different economic sectors. Table 3 shows the description and also the default rate of different industry codes.

Changes in industry default rates are usually due to the periodicity of product or service demand. Some industries expanded sharply during a specific period and contracted after that end, such as construction and hotel industries. In contrast, income fluctuations in some sectors were much more stable, such as the medical industry and service industries. The default rate represents credit risk, which affects whether loans should be extended. The default rates of different industries are provided by SBA.

We also made a handcrafted feature called `NAICS_sector_default_rate` base on the above table to

further improve the accuracy of the classification.

The proportions of different industries in the dataset and their loan issuance success rates are shown in the figure below. The retail industry accounts for the largest proportion, followed by construction, food,

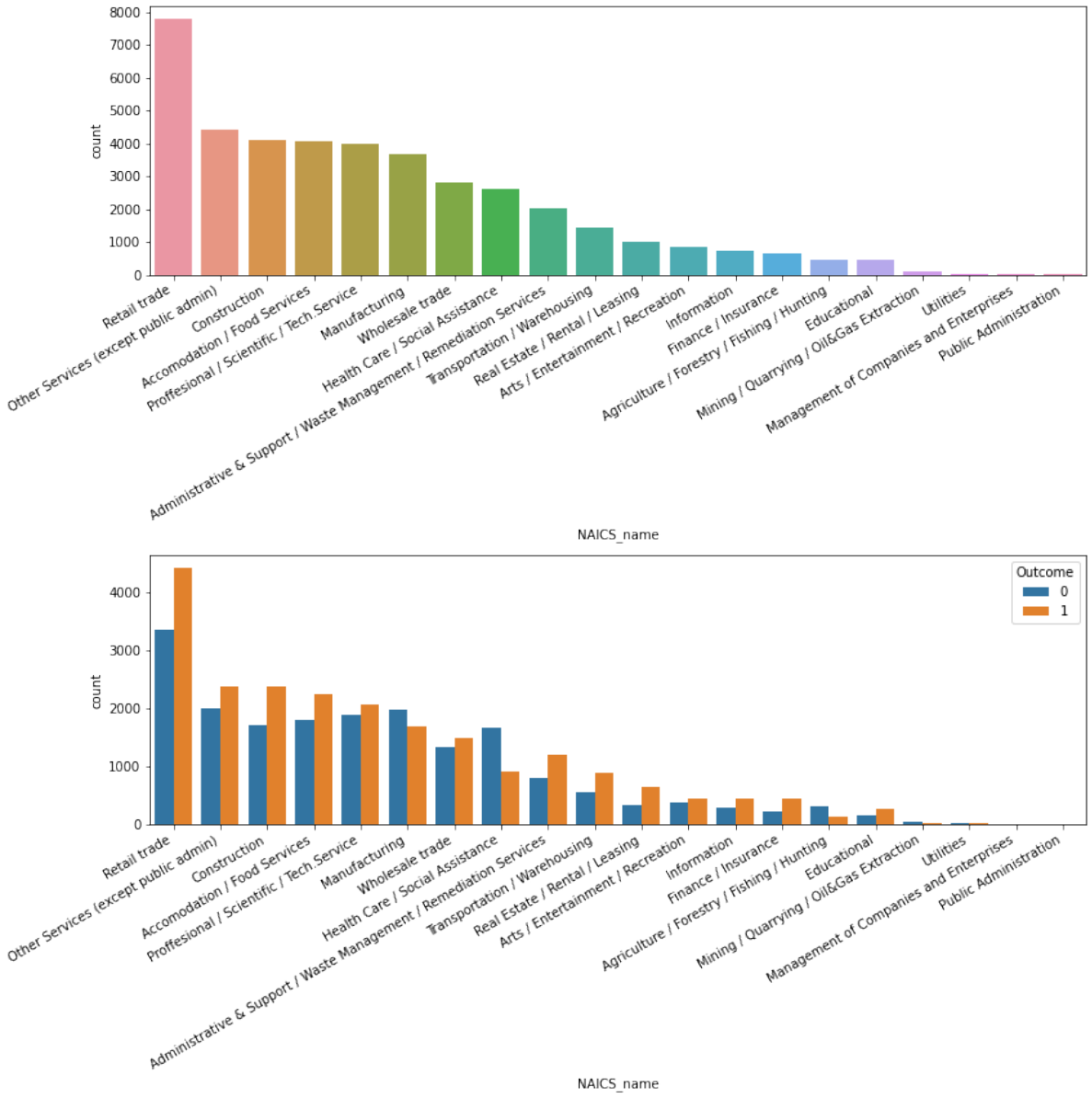


Figure 4: Loan proposed corresponding to NAICS

and factories. In addition, the bank's decision of whether to grant loans is related to various industries. The most obvious is that banks have a negative willingness to lend to manufacturing, health care/social assistance, agriculture/forestry/fishing/hunting industry.

## 2.4 DisbursementGross, GrAppv, SBA_Appv

Most of the values of these three features are in range between 0 and 1. We replace the original values with them after logarithmic transform.

## 2.5 Term, NewExist, FranchiseCode, UrbanRural, RevLineCr, LowDocs

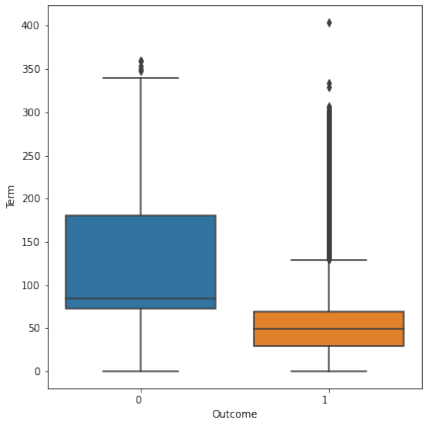


Figure 5: The impact of the loan term on charge-off. A shorter loan period will result in a higher charge-off rate. The term more significant than 240 months means the loan is backed by real estate and will tend to have a significantly higher charge-off rate; the rationale is the value of the land is often large enough to cover the amount of any principal outstanding. We also made this as a handcrafted feature called ``RealEstate''.

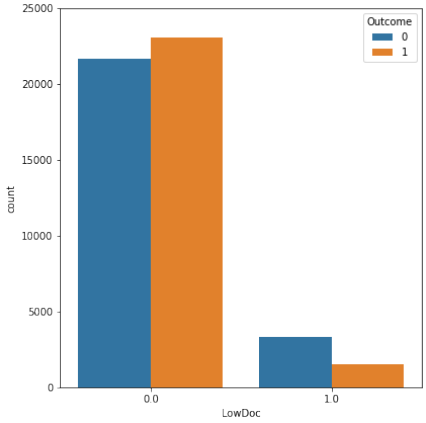


Figure 6: The impact of whether the business is in the LowDoc Loan program on charge-off rate. 0 means NO, 1 means Yes.

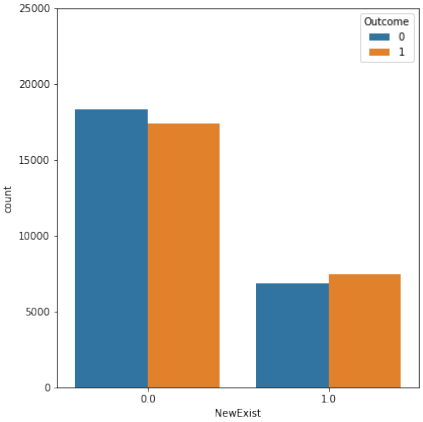


Figure 7: The impact of new business or existing business on the charge-off rate. 0 is existing business, 1 is new business.

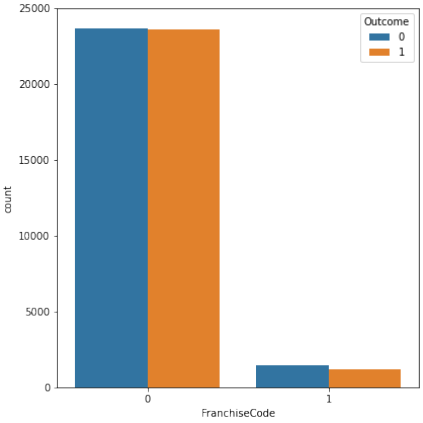


Figure 8: The impact of franchise on the charge-off rate. 0 means there is no franchise code, 1 means franchise code exists.

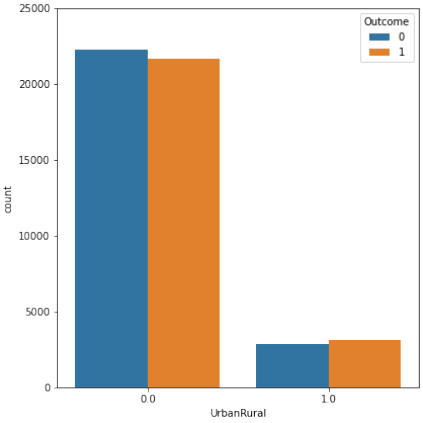


Figure 9: The impact of the location on charge-off rate. 0 is urban, 1 is rural

## 2.6 Correlation Heatmap

Below shows the correlation of the data attributes as well as the handcrafted features.

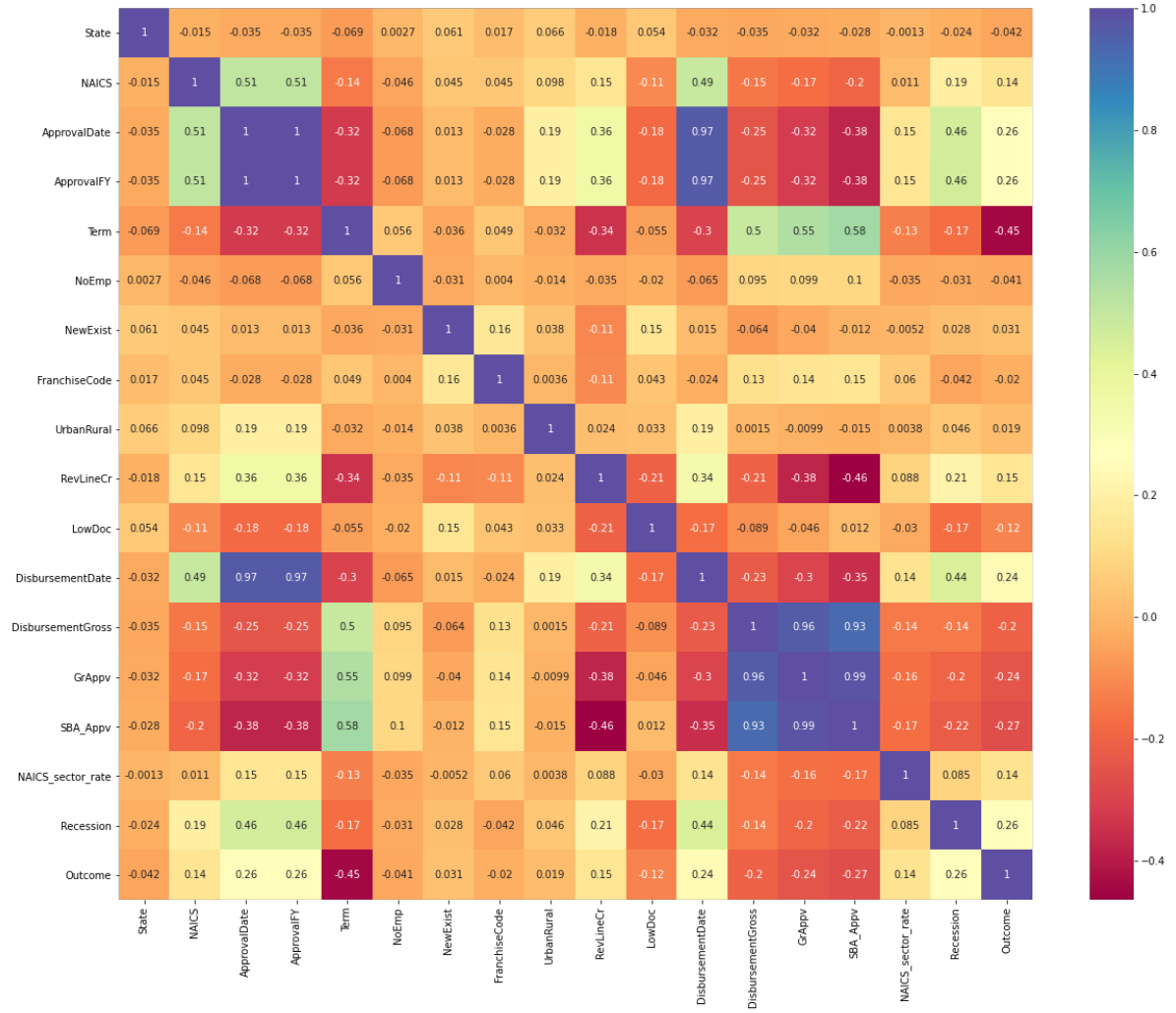


Figure 10: Correlation Heatmap between Features

```
Irrelevant to the outcome:
    Zip (Borrower zip code), CreateJob, RetainedJob.

Have low correlation with the outcome:
    NoEmp, NewExist, FranchiseCode, UrbanRural.

Related to the outcome :
    NAICS, ApprovalFY, RevLineCr, LowDoc, DisbursementGross, GrAppv,
    SBA_Appv, NAICS_sector_rate, Recession, RealEstate.

Highly correlated to the outcome: Term
```

# 3 Modeling

## 3.1 Algorithm Selecting

We firstly examine several model algorithms by cross-validation score on different datasets mentioned in the above section, including K Nearest Neighbor (KNN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Gradient Boosting (GBM), Light Gradient Boosting (LGBM), AdaBoost (ADA), XGBoost (XGB), CatBoost (CAT).

| Model | whole1 | whole2 | whole3 | recn1 | recn2 | recn3 | none1 | none2 | none3 |
|---|---|---|---|---|---|---|---|---|---|
| KNN | 69.00 | 68.98 | 70.59 | 72.78 | 74.73 | 74.85 | 67.94 | 73.71 | 74.68 |
| LR | 75.45 | 75.60 | 76.68 | 75.15 | 77.42 | 77.53 | 72.24 | 76.46 | 77.12 |
| DT | 88.07 | 88.12 | 88.22 | 89.80 | 89.37 | 88.55 | 87.72 | 87.78 | 87.55 |
| RF | 90.15 | 89.83 | 90.15 | 91.32 | 91.29 | 91.62 | 89.35 | 89.65 | 89.84 |
| GBM | 89.67 | 89.70 | 89.89 | 92.78 | 92.77 | 92.74 | 89.18 | 89.29 | 89.30 |
| LGBM | 92.25 | 92.28 | 92.16 | 93.68 | 93.56 | 93.59 | 91.81 | 91.81 | 91.62 |
| ADA | 87.65 | 87.65 | 87.59 | 91.36 | 91.47 | 91.35 | 86.81 | 86.82 | 86.94 |
| XGB | 92.60 | 92.65 | 92.35 | 93.85 | 93.66 | 93.70 | 92.08 | 91.98 | 91.65 |
| CAT | 92.63 | 92.57 | 92.43 | 93.98 | 93.64 | 93.55 | 92.25 | 92.02 | 91.91 |

According to the results above, we observed that the data containing the whole training dataset has higher accuracy than the combination of recession and non-recession data. Therefore, we choose to focus more on training the model over the entire dataset rather than separated into two parts. In those various models, LightGBM, XGBoost, and CatBoost perform best.

## 3.2 Hyperparameter Tuning

From this section I would renamed the original data as *data0*, the whole data with `["Zip", "CreateJob", "RetainedJob"]` removing as *data1*, and that with `["FranchiseCode", "UrbanRural"]` removing as *data2*. We find the best model with the highest f1 score over there two types of data through a package named "GridSearchCV". To avoid the high cost of the computational resource, we tune each hyperparameter with greedy.

Taking the tuning process for LightGBM model as an example: The hyperparameter `learning rate` and `n_estimators` are highly correlated; thus, we use GridSearchCV to find the best combination of them and leave other parameters as default values.

We then narrow down the suitable range of those hyperparameters; subsequently, we tune the others, such as `max_depth` based on the fixed smaller range of decided hyperparameters. Eventually, all the other hyperparameters are find in this method as well.

### 3.2.1 LightGBM

- `n_estimators`: defines the number of sequential trees to be modeled; higher value is fairly robust but possible to be overfitting.
- `learning_rate`: determines step size shrinkage as well as the impact of each tree on the final outcome; lower value makes the model more robust to the specific characteristics of tree but is computationally expensive.
- `max_depth`: defines the maximum depth of the tree, but has no effect on tree width; can be used to control over-fitting as higher value allows model to learn relations very specific to a particular sample.
- `min_data_in_leaf`: defines the minimum number of samples (or observations) which are required in a node.

#### 3.2.2 XGBoost

- `n_estimators`: defines the number of sequential trees to be modeled; higher value is fairly robust but possible to be overfitting.
- `learning_rate`: determines step size shrinkage as well as the impact of each tree on the final outcome; lower value makes the model more robust to the specific characteristics of tree but is computationally expensive.
- `max_depth`: defines the maximum depth of the tree, but has no effect on tree width; can be used to control over-fitting as higher value allows model to learn relations very specific to a particular sample.
- `min_child_weight`: defines the minimum weights of samples (or observations) which are required in a node.
- `gamma`: defines the minimum loss reduction required to make a further partition on a leaf node of the tree; larger value keeps the algorithm more conservative.
- `colsample_bytree`: defines subsample ratio of columns when constructing each tree; subsampling occurs once for every tree constructed.

#### 3.2.3 CatBoost

- `n_estimators`: defines the number of sequential trees to be modeled; higher value is fairly robust but possible to be overfitting.
- `learning_rate`: determines step size shrinkage as well as the impact of each tree on the final outcome; lower value makes the model more robust to the specific characteristics of tree but is computationally expensive.
- `max_depth`: defines the maximum depth of the tree, but has no effect on tree width; can be used to control over-fitting as higher value allows model to learn relations very specific to a particular sample.
- `l2_leaf_reg`: defines coefficient at the L2 regularization term of the cost function.

#### 3.2.4 Cross-Validation Score

| Model | Data | Hyperparamter | F1 Score |
|---|---|---|---|
| LightGBM | data0 | `n_estimators`: 450, `learning_rate`: 0.10, `max_depth`: 6, `min_data_in_leaf`: 20 | 0.92644 |
| | data1 | `n_estimators`: 600, `learning_rate`: 0.10, `max_depth`: 6, `min_data_in_leaf`: 20 | 0.92531 |
| | data2 | `n_estimators`: 600, `learning_rate`: 0.05, `max_depth`: 7, `min_data_in_leaf`: 20 | 0.92527 |
| XGBoost | data0 | `n_estimators`: 300, `learning_rate`: 0.10, `max_depth`: 7, `min_child_weight`: 3, `gamma`: 0.20, `colsample_bytree`: 0.6 | 0.92742 |
| | data1 | `n_estimators`: 300, `learning_rate`: 0.10, `max_depth`: 7, `min_child_weight`: 3, `gamma`: 0.25, `colsample_bytree`: 0.6 | 0.92629 |
| | data2 | `n_estimators`: 300, `learning_rate`: 0.15, `max_depth`: 5, `min_child_weight`: 3, `gamma`: 0.25, `colsample_bytree`: 0.6 | 0.92600 |
| CatBoost | data0 | `n_estimators`: 1000, `learning_rate`: 0.10, `max_depth`: 6, `l2_leaf_reg`: 3 | 0.92704 |

| Model | Data | Hyperparamter | F1 Score |
|---|---|---|---|
| CatBoost | data1 | `n_estimators`: 1200, `learning_rate`: 0.10, `max_depth`: 6, `l2_leaf_reg`: 4 | 0.92613 |
| | data2 | `n_estimators`: 1200, `learning_rate`: 0.10, `max_depth`: 6, `l2_leaf_reg`: 2 | 0.92635 |

### 3.3 Results Ensembling

In order to get better accuracy, we assume that ensembling the results of several models would be a great choice. We have tried different ensembling methods, inclusive of combining the probabilities of the same model with different datasets, merging the predictions of various models with the same dataset, and, without doubt, the results of distinct models with diverse datasets. Moreover, we pick up a great numbers of models with best hyperparameters, add their predicted probabilities, and eventually get the final results with Kaggle public score 0.93685.

| Model | Kaggle | Data | Kaggle | All(Ensemble) | Kaggle |
|---|---|---|---|---|---|
| LightGBM | 0.92865 | data0 | 0.93230 | One Time | 0.93095 |
| XGBoost | 0.93032 | data1 | 0.93105 | Two Times | 0.93130 |
| CatBoost | 0.93042 | data2 | 0.93107 | Best | 0.93685 |

## 4 Conclusion

For the data prepossessing, we analyzed different data formats, performed different pre-processing for different data formats, and added handcrafted features according to the background knowledge. For modeling, we thoroughly analyzed the advantages and disadvantages of various models, and finally determined three excellent classifier algorithms. After adjusting the parameters for them, we found a series of parameter sets. Combining different models, as well as the same model with different parameters, we achieved a public test score of 93.68.