Data mining on Charged-off Prediction Competition

[redacted]

Nov 2020

[redacted]

[redacted]

[redacted]

Contents

1 Introduction

1.1 Problem Statement

This Kaggle Competition aims to help decide whether should approve certain loan or not, depending on the multi information given by the historical data provided by Small Business Administration (SBA). The training data's label is ChargeOff(1-yes,0-no). To reproduce our results, please follow 'README' in the zipped codes file.

1.2 Dataset

The training datasets has 50000 samples(rows) and each sample is a loan record. For each record, it has 23 attributes represents different information. The set of attributes are explained here:

- Name:Borrower Name [categorical]
- City,State,Zip: Borrower's Geographic information [categorical]
- Bank,BankState:Bank Name and State[categorical]
- NAICS:North American Industry Classification System code[categorical]
- ApprovalDate: Date SBA Commitment Issued[categorical]
- ApprovalFY: Fiscal Year of Commitment[categorical]
- Term: Loan term in months[continuous]
- NoEmp:Number of Business Employees[continuous]
- NewExist:1 = Existing Business, 2 = New Business[categorical]
- CreateJob,RetainedJob: Number of jobs created/retained [continuous]
- FranchiseCode: Franchise Code 00000 or 00001 = No Franchise[categorical]
- UrbanRural: 1= Urban, 2= Rural, 0 = Undefined[categorical]
- RevLineCr: Revolving Line of Credit : Y = Yes[categorical]
- LowDoc: LowDoc Loan Program: Y = Yes, N = No. Related to interest rate[categorical]
- DisbursementDate: Disbursement Date [categorical]
- DisbursementGross: Amount Disbursed [continuous]
- BalanceGross: Gross amount outstanding [continuous]
- GrAppv: Gross Amount of Loan Approved by Bank [continuous]
- SBA Appv: SBA's Guaranteed Amount of Approved Loan [continuous]

2 Exploration Analysis and Data Preprocessing

In this part, we aim to extract patterns and characteristics from the original messy attributes. We want to make each attribute tractable and meaningful for the machine learning models, which must be numerical. Noticed that many attributes given have different noise except the missing value, which do not match the format in description, we should clean these samples as well.

2.1 Data Preprocess for Borrower's information

This part we apply data preprocess to the attributes related to the borrower's personal information.

2.1.1 Names

There are 48757 unique borrower names and only 85 Names appear more than 3 times, who occupy 659 records. We find that for Name first seen, there is weak link in ChargeOff. But it is slightly partial to 0 for the Names appear 3 or more times, which we call them 'familiar names' Following figures is an illustration of our analysis and assumptions. We will apply similar technique to other Name attributes as well, which namely establishes a black list or white list for those familiar names.

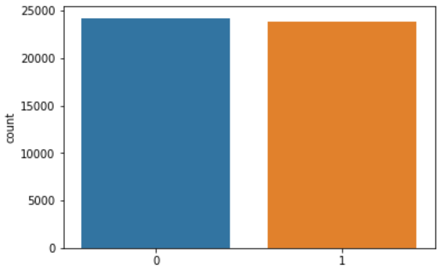

Figure 1: (a) Unique Names

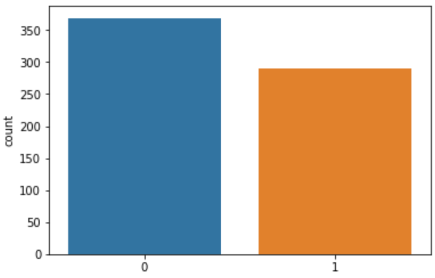

Figure 2: (b) Familiar Names

2.1.2 Geographic information

In order to better understand the information about the Cities, States, Zips in the data set, we also conduct a preliminary visualization of the data. Try to explore whether there is a close relationship between Chargeoff and the geographic information.

We did three sets of experiments to explore the relationship between city, state, zip and Chargeoff and the result shows that Zip have no influence on chargeOff, we can simply delete it. "State" seems have a strange/slight bias around frequence 2000 but we think it will not influence. For city, more frequent certain city appears, more likely chargeOff turn to 1. The following part are

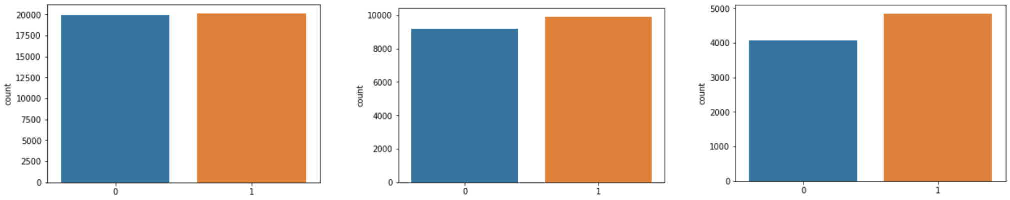

Figure 3: The Chargeoff change with City appears more than 3,30,100 times(from left to right)

2.2 Data Preprocess for Bank's information

We divide the bank information in to four part. The number of certain bank name in each bins is 0-20, 21-200, 201-500, >501. The number of samples in each interval are 6413, 9089, 9709, 24789. In addition, We change the 52 kinds of bank state to number 0-51 and delete the missing value and finally get 49923 samples for further work.

2.3 Data Preprocess for Bussiness' information

This part we apply data preprocess to the attributes related to the borrower's bussniess/company.

2.3.1 Basic Information

The NAICS part has 8000 samples with unknown NAICS. Excluding these 8000 examples, we did not find that charge-off has a clear preference for NAICS.

The ApprovalDate does have influence on the charge-off but we can't also impossible, figure out which day is more possible to get pass. Thus, we ignore this feature.

The ApprovalFYs also have influence on the charge-off. The value interval is 1969 2014 and we do normalization.

The Term feature has an obvious relationship between term and result. We found that in the five terms of "80", "60", "240", "120", "300", the charge-off rate is extremely low. Thus, we add a blacklist. For term value in blacklist("80", "60", "240", "120", "300"), set 0, otherwise 1.

The UrbanRural has three values. 0, 1, 2 have 13877, 30059, 6064 samples respectively. We directly use the original data format.

2.3.2 Business Scale

For this part of the data, we find that the main reason that affects the results is the commercial scale.

The NoEmp shows that there is no bias for small company, tendency to 0 for middle company, tendency to 1 for big company. Thus, we divide NoEmp by company size(0-10, 10-100, >100)

The CreateJob shows that if it creates a lot of jobs(>80), ChargeOff tend to be 1. For very low job create(<10), no bias. For 10-100, tend to be 0. Thus, we divide CreateJob by jobs number(0-10, 10-80, >80).

The RetainJob shows the same feature as CreateJob. We directly normalize the value by dividing all the values in the data set by 80. (eg. $1 \rightarrow 0.0125$)

2.3.3 Special Points

The connection between FranchiseCode and result cannot be found through preprocessing. For RevLineCr and LowDoc, we convert boolean values into numbers(0,1) and train in the model.

2.4 Data Preprocess for Loan′ information

This part we apply data preprocess to the attributes related to the loan, mainly the currency amount, and mine new features.

2.4.1 Business Side

Because the bank has a mandatory repayment date, the repayment date is largely related to the final result. According to the data, we found that there is a high probability of charge-off on the date when many people make repayments at the same time. The figure shows that more frequently disbursement appears, more possible ChargeOff to be 1. The realistic logic of this is that it may be more likely to be the repayment date imposed by the bank. Thus, we rank the date by disbursement frequency. For example, there are 560 disbursment on 31-Aug-07, we change the date 31-Aug-07 to $\frac{560}{591}$(591 is the largest disbursement value in the Data Set)

2.4.2 Institution Side

GrAppv and SBAAppv stand for the amount of loan approved by the bank and SBA and the latter is the final amount. Intuitively, they should be linked to

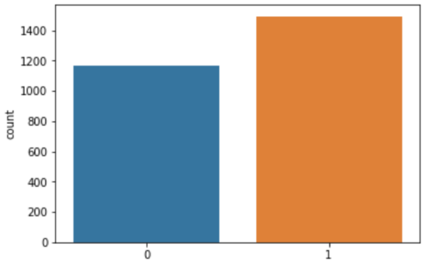

Figure 4: (a)Days with a small number of Disbursement

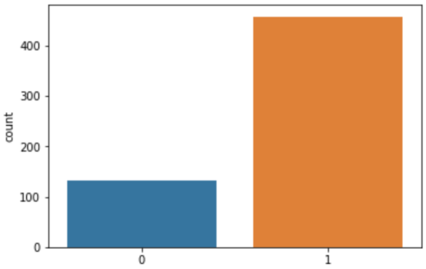

Figure 5: (b) Days with a large number of Disbursement

ChargeOff strongly, we choose to translate it into int type and retain the original amount. For attribute BalanceGross, it is always zero so we simply abandon it.

Additionally, we find that there are latent attribute Disbursement-ratio (DisbursementGross/SBA) and SBA-Appv-ratio(SBA/GrAppv) that may have strong link with whether it is expected to be charged-off. After generating this new attribute, we do find a high correlation coefficient which proves our proposal.

3 Data mining

In this section, we aim to explore suitable model or combination of models to apply data mining on the pre-processed training data. We also delete relatively useless attributes to streamline features that model needs to learn. Our plan of data mining is following:

1. Prepare the pre-processed training data and split it into train/validation set.
2. Choose attributes combination which will be fed into model(Initialized with complete attributes.)
3. Choose training model and fine-tune the parameter.
4. If not good enough, return to (2) and reduce attributes according to correlation.

3.1 Model Selection

3.1.1 Model1 - sklearn built-in models

Scikit-learn (sklearn) is a commonly used third-party module in machine learning, which encapsulates commonly used machine learning methods, including regression, dimensionality reduction, etc. We first try multi built-in model used in sklearn.

We find that Decision Tree and random forest works better than other methods like linear regression. Random Forest works by training numerous decision trees each based on a different resampling of the original training data, which is known as Bagging (Bootstrap Aggregating). In Random Forest the bias of the full model is equivalent to the bias of a single decision tree, which itself has high variance. By creating many of these trees, in effect a "forest", and then averaging them the variance of the final model can be greatly reduced over that of a single tree. Intuitively, we infer that feature selection method works better is that despite preprocessed, multi-features still show excessive diversity in numerical range and association form with the predict label, which makes trouble for the models requiring smooth features. Random Forest will not be influenced by this and thus performances better in training among them naturally.

3.1.2 Model2 - latest external models

As to the provided data, sklearn built-in models cannot achieve ideal performance on testing set, so we turn to latest external models, which performs better and is easier to use.

CatBoost (categorical boosting) is a gradient boosting algorithm library that can handle categorical features well. In this competition, most attributes are categorical, leading to difficulty in data preprocessing. Over-processing the original data will cause overfitting during training and validation phase while under-processing rises the incomprehension of the model, which also leads to a bad result. However, CatBoost automatically handles categorical features. It calculates the frequency of occurrence of a certain category feature (category), and then add hyperparameters to generate new numerical features. it is quite similar to the idea how we preprocess the categorical attribute. The difference is that these all will be done by CatBoost. It also uses combined category features, which can take advantage of the relationship between features.

The base model of CatBoost is the symmetric tree and it optimizes the algorithm of leaf-value calculation, which prevents the model from overfitting. This point is especially important since we met overfitting now. The training phase of CatBoost is visualized in Figure6.

3.2 Attributes Reduction

The main strategy of attributes reduction is that:

- Before and after feature mapping/preproessing, we calculate the correlation. For certain attribute, weaker the correlation with ChargeOff, higher the priority we eliminate it.
- After model training, we check the feature importance. We set higher priority to eliminate attributes with less importance.

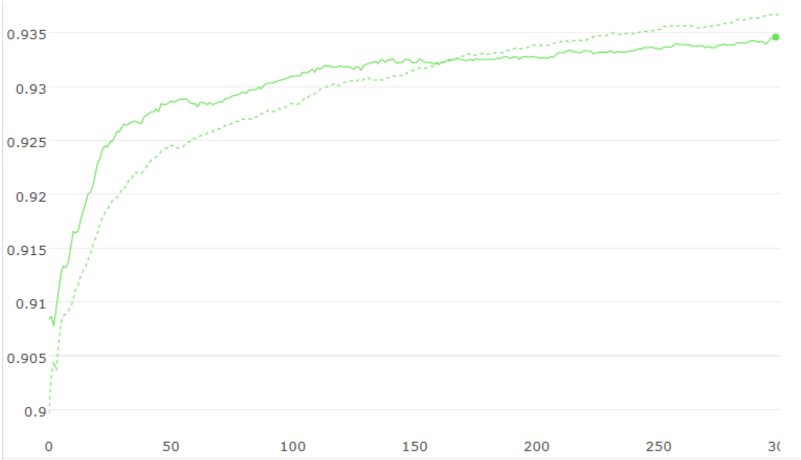

Figure 6: The training phase of CatBoost. The dashed line is train accuracy, the solid line is validation accuracy

- We manually try different combination of attributes according to the elimination priority.

4 Evaluation and Interpretation

In this part, we give the experiment results on attribute Reduction, model selecting, and final results.

4.1 Correlation Analysis and feature importance

We check the correlation between various attributes and labels before tuning the hyper-parameters on our models, which is supposed to be most influential to decision for certain loan. We first use clusterMap() to check correlation of our manually-processed features with the label ChargeOff. We can clearly find that the correlation becomes closer in preprocessed features than the original one in the following figure.

We also check the feature importance after the built-in preprocess of CatBoost by calling interface and visualizing it using matplotlib.

When performing correlation analysis, we eliminate attributes that contributes less or even have negative effect on the model prediction with respect to the correlation coefficients. Our result is better after removing redundancy relies on both the correlation coefficients and feature importance before and after fed into the model.

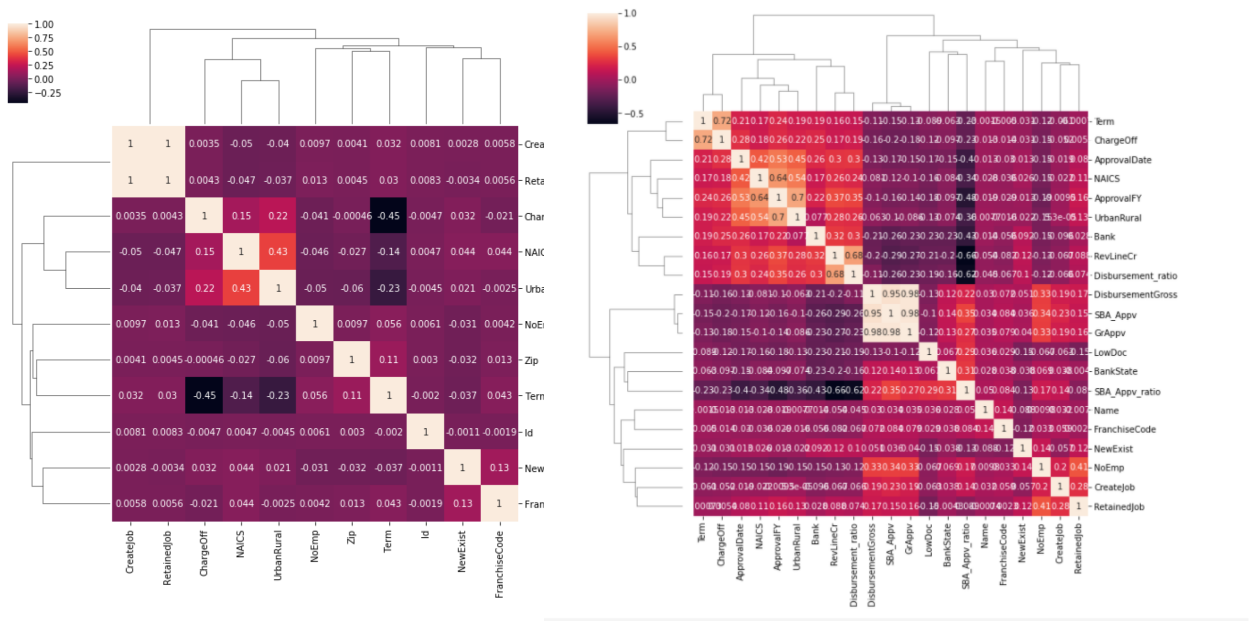

Figure 7: ClusterMap. Left: original features. Right: preprocessed features. Lighter means higher correlation thus we can see that after preprocessing, attributes like Term, ApprovalDates, which has weak relation with ChargeOff, improves significantly.

4.2 Approach Comparison

When performing correlation analysis, we train model with different feature combination. We get a result as following.

Table 1: Performance comparison with different models. Mixed means mixture of handcrafted and automatic preprocessed training data.

Model	KNN	LR	RF	GB	DT	CatBoost	CatBoost(mixed)
Accuracy(Val)	76.03	65.12	99.98	88	99.98	93.2	**93.46**
Accuracy(Test)	-	-	-	84.2	65.6	93.272	**93.422**

Twenty percent of the original training dataset is split into validation set to see the result. CatBoost(mixed) model achieves the highest point on public board in performance. We can find that our handcrafted preprocess causes overfitting on validation set but we still evaluate the feature reduction does relieve this overfitting. The latest external CatBoost achieve better performance and solve the overfitting problem. The performance is further improved by combining handcrafted preprocess with automatic preprocess.

Final Best Result We submit the TOP results on validation set, check the performance on Kaggle test set and choose the model that performs best on public board. The training data we use finally combines our handcrafted preprocessing attributes and the original attributes. They are chosen according the feedback of CatBoost Model training results.

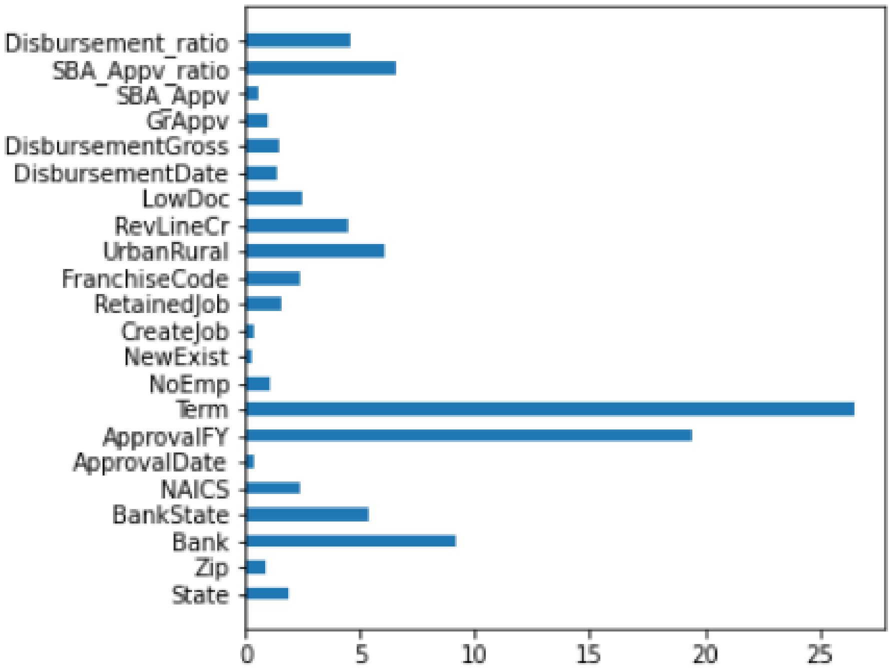

Figure 8: Feature importance given after training of CatBoost.

5 Conclusion

Our over-preprocessing of original data should be a rough version of existing published algorithm. Since we haven't foreseen the overfitting caused by over-preprocessing, the performance achieved is not good enough with multi-models even with feature reduction.

After switching to the CatBoost library,which have automatic data preprocessing, we get much better performance. We further fine-tune the hyper-parameters and try different feature combination selected from handcrafted and automatic preprocessing data. The preprocessing phase does have a long-term influence during the whole data mining processing, which need special attention.