Abstract

The database is provided by Home Credit, a company which provide a loans to people with less credit history or no history [1]. The company want us to study this database so we design a model which can do classification be defaulter or not. The primary task is to understand the data and then built a model which can classify that the applicant will pay the loan without defaults or not. In the database the table application_train dataset has a column name TARGET which is result the applicant was defaulter or not. If the target is 0 that means the applicant paid his loan properly without any default and if target is 1 then the applicant was a defaulter. The company "Home Credit" is hosting this competition [2]. So, that they can develop a good machine learning model to predict the defaulter with the details from their application.

Introduction

One of the biggest problem Financial institution have credit risk assessment from decades which is a binary classification problem in which they should sanction a loan or not. There are so many Institution got bankrupt just because they got into the debts which they can't repay. To solve this problem we are implementing machine learning models and compare them. To find which one is good to predict that a applicant is faithful or delinquent.

We are implementing multiple algorithms with different features and comparing them with each other. Trying to understand which one works good for the dataset. We have implemented Logistic regression, Naive Bayes and support vector machine. I think that logistic regression is good for time complexity. It doesn't have high accuracy but it give the classification faster then other algorithms. Also I am using the experimental features on the algorithm rather then the old features to get more accuracy on the prediction. I have even tried the transfer learning on this dataset I used some other features to train the dataset and while testing the data I am using some other features which increased my accuracy by 8%.

I have implemented transfer learning on logistic regression with polynomial features which increases the AUC compares to the other models AUC. I found out that models sometimes perform well if we give them a related input to train the data and give them twist input to test it. There are so many companies which are trying transfer learning on different dataset. As Google is trying transfer learning on image recognition and there are more other companies which tries this transfer learning to save the time of the training a model to get result they find a trained model with some different dataset and gives them other dataset to test and it sometimes give better accuracy compares to the dataset it was trained with. Which is why transfer learning is really important this can save lots of time to train any data if we can find related models to our dataset.

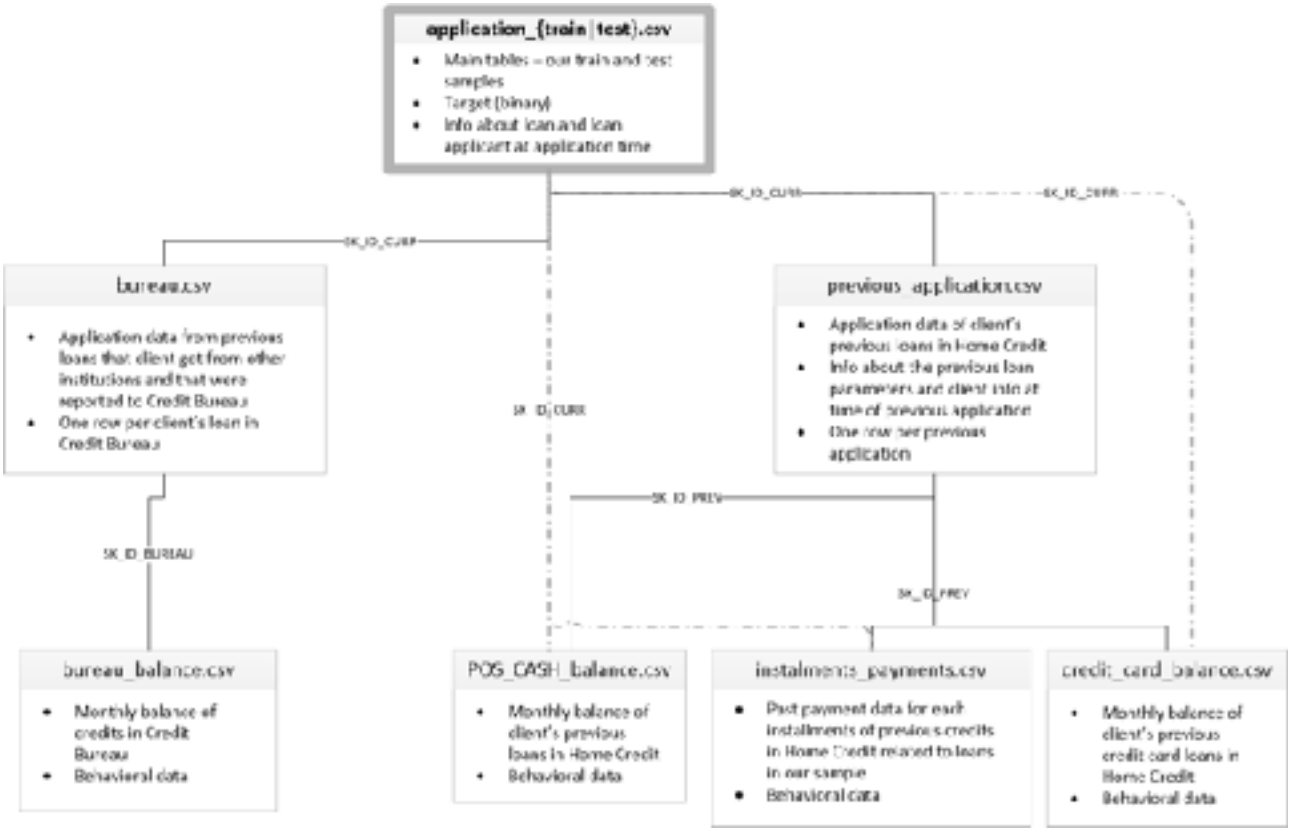

Fig. 1

Related work

Literature

In this paper they are comparing classification between Decision tree and logistic regression for credit risk. They are designing a model which can classify farmers will able to repay the debts or not. Using multiple features from the dataset got from two banks in India pertaining to the agricultural production loans given to farmers in and around Honavar, a backward block in Karnataka, India. [3] In the dataset there are total of 25 features in which they are using top 20 features. Some of them are-Crop for which the loan was taken, Procured inputs, Spent for irrigation, Spent for miscellaneous purposes, Repaid advance money, Income, etc. After running the dataset on logistic regression and decision tree they conclude that the decision tree better they evaluate this he algorithms on the basis of accuracy and complexity of trained classifier.

One of the biggest issue in classifier is never define the rule which caused that classification specially in neural networks. In this paper they are trying to get exact the rules known as neurorules, Trepan and Nefclass which can explain the classification [4]. They are using 3 real life dataset: German Credits, Bene 1 and Bene 2. The features used for German Credits are-Checking Account, Terms, Credit History, Purpose, Credit Account, Saving Account, Present employment since, Installment rate and Personal status and sex. For Bene 1 features are-Identification number, Amount of loan, Amount of purchase invoice, Percentage of financial burden, Term, Personal loan, Purpose, Monthly payment, Saving Account and Other loan expenses. They used the decision tables to extract the rules of the reactions of any application which was built on the 3rd layer of the feedforward neural network.

Some people are using trees as a classifier as they are more accurate as compare to other models. LAD tree classifier and REP tree classifier are algorithms used by the people on the Greman credit data [5]. It has 20 attributes, namely, Duration, Credit History, Checking Status, Purpose, Credit Amount, Employment, Own Phone, etc. The data set 1000 rows of credit application with class detail. It discriminates the records into two classes, namely, good and bad. After comparing the results of both the algorithms it is concluded that REP is better in Classification accuracy and Time taken to build the model.

Ensemble models are usually good for classification of data. In recent studies they actually performed better then classical classifiers. Artificial neural network (ANN) (Lai et al., 2006b; Malhotra & Malhotra, 2003; Smalz & Conrad, 1994) and support vector machine (SVM) (Huang, Chen, Hsu, Chen, & Wu, 2004; Van Gestel, Baesens, Garcia, & Van Dijcke, 2003) are advantageous to statistical models and optimization techniques for credit risk evaluation. They are designing a ANN on Japanese consumer credit card application approval obtained from UCI Machine Learning Repository [6]. In this experiment rather than using ‘‘one-member-one-vote’’ or ‘‘majority-rule’’ ensemble, the novel neural network ensem- ble aggregates the decision values from the different neural ensemble members, instead of their classification results directly. The new ensemble strategy consists of two critical steps: scaling, which transforms decision values to degrees of reliability, and fusion, which aggregates degrees of reli- ability to generate final classification results.

Kaggle

People are using different kinds of algorithms as the data for this completion is really complex and it also has different types of features in this database. Column types are Float, Integer and Categorical[9]. Also there are 67 columns have missing values in it. Some of them are-COMMONAREA_MEDI, COMMONAREA_AVG, COMMONAREA_MODE, NONLIVINGAPARTMENTS_MEDI, NONLIVINGAPARTMENTS_MODE, NONLIVINGAPARTMENTS_AVG, etc.

As mentioned there are Categorical values in the dataset which most the algorithm can’t support .Usually categorical values are been encoded to integers using One-hot encoding or label encoding but LightGBM is one of the algorithm which can handle categorical values. That’s why many people are implementing LightGBM as there are 16 categorical values. As One-hot encoding is hard to implement.

One of the Kaggle kernel Good_fun_with_LightGBM [8] has implemented LightGBM and run it on the whole database by building the dataset 1st and then it send the dataset to train the model. Some of the top Features of output are-EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3 and DAYS_BIRTH. He used ROC for the evolution of the model and the accuracy is 0.7825 ± 0.0033.

In one kernel they are using imputing [7] for the missing values in the database. In imputing the missing values are replaced with the other values such as mean, mode whatever makes sense for the column. But some people are trying to implement XGBoost on this data so they don’t need to worry for the missing values. One of the implementation of XGBoost on all tables it has the accuracy of .796 [10] in that implementation he is just joining all the tables and send the table to the training model of XGBoost. The output comes in a form of histogram in which the top 5 Features are-EXT_SOURCE_3, EXT_SOURCE_2, ANNUITY_LENGTH, NEW_EXT_SOURCES_MEAN, SOURCES_PAOO. Most of the public kernels are for learning purpose. They are designed so other can understand the database and what kinds of features are important also which algorithms can be helpful for this dataset.

Method

- **Data Analysis**

In Data Analysis I started checking the data that it is balanced or not. I plot a pie chart and found that only 8% has Target == 1 which means there is only 8% data of rejected applications. There are total of 7 tables in this database which has 3 types of data in them-Integer, Numeric and Text. In which there are 65 columns of Numeric, 41 columns of Integer and 16 columns of Text. Also there are columns which has missing values in them after running the data it turned out to be 67 columns out 122 column in application train has missing values.

Then I started running information gain on each table to find good Features which I can choose to train my model on. First I run the information gain on Application train to find highest information gain and also the frequency of the feature if the feature has missing values then it can be biased or the feature may not that useful as it is not filled by everyone. So we can't train using those features which have missing values. I have used some of the features which are related to the Target directly like DAYS_BIRTH that is the age of the person. In one of the kaggle kernel they plotted a histogram and found that as the person's age increases it is less likely that they will default something in there loan instalments. I also have run Empirical cumulative distribution function(ECDF) on some particular features which I found relative because of there high Information Gain. As you can see in Fig. 1.

I also perform polynomial degree dataset on the features and plotted ECDF of them and found they work better with the polynomial features. I have selected 2 polynomial Features to train my logistic regression model.

	Feature Name	Information Gain	Frequency
0	TARGET	[illegible]	1.0
1	EXT_SOURCE_3	[illegible]	[illegible]
2	EXT_SOURCE_2	[illegible]	[illegible]
3	OCCUPATION_TYPE	[illegible]	0
4	FLAG_DOCUMENT_3	[illegible]	[illegible]
5	ORGANIZATION_TYPE	[illegible]	0
6	DAYS_BIRTH	[illegible]	1.0
7	NAME_INCOME_TYPE	[illegible]	0
8	NAME_EDUCATION_TYPE	[illegible]	0
9	CODE_GENDER	[illegible]	0
10	DAYS_LAST_PHONE_CHANGE	[illegible]	[illegible]
11	REG_CITY_NOT_WORK_CITY	[illegible]	1.0
12	DAYS_EMPLOYED	[illegible]	1.0
13	FLAG_EMP_PHONE	[illegible]	1.0
14	REGION_RATING_CLIENT_W_CITY	[illegible]	1.0
15	FLOORSMAX_MODE	[illegible]	[illegible]
16	FLOORSMAX_AVG	[illegible]	[illegible]
17	FLOORSMAX_MEDI	[illegible]	[illegible]
18	REGION_RATING_CLIENT	[illegible]	1.0
19	DAYS_ID_PUBLISH	[illegible]	1.0
20	FLAG_DOCUMENT_3	[illegible]	1.0
21	AMT_GOODS_PRICE	[illegible]	[illegible]
22	EMERGENCYSTATE_MODE	[illegible]	0
23	REG_CITY_NOT_LIVE_CITY	[illegible]	1.0
24	ENTRANCES_MEDI	[illegible]	[illegible]
25	ENTRANCES_AVG	[illegible]	[illegible]

Fig. 2

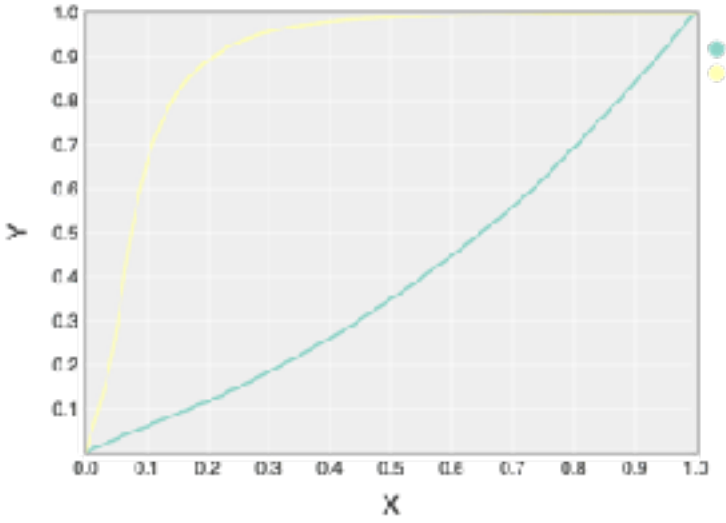

Fig. 3

As they both have great ECDF curves which implies that they are better features to train a dataset. I even tried more negative polynomial of division of 2 features which also gave a great ECDF curve which as there is a big difference between the values of the values which can help them to classify the test case easily. I used the feature EXT_SOURCE_2/EXT_SOURCE_1 which gave a very great distribution in the feature which can help us in the classification. The plot you are seeing in the Fig 2 is ECDF of EXT_SOURCE_2/EXT_SOURCE_1. In which you can see the distribution has great curve though which we can assume the classifier will work great.

- **Algorithms**
  - **Transfer Learning on Logistic Regression with different features**

In logistic regression I have implements transfer learning as well polynomial features. Logistic regression is one of the fastest algorithm in all machine learning models. And using some polynomial features increases the accuracy of the classification. For testing the model we are using ROC AUC and it is better then the random guess as the random guess classifier is a straight like which has a 50-50 chances of both the events occurring. But as we use the polynomial features it increases the accuracy of the model. We even have tried a little bit of transfer learning in this model as we are

$$Likelihood = P(y|X, w) = \prod_{i=1}^{N} \sigma(w^T x_i)^{y_i} \times (1 - \sigma(w^T x_i))^{1 - y_i}$$

$$NLL = -\sum_{i=1}^{N} \{ \; y_i ln(\sigma(w^T x_i)) + (1 - y_i) ln(1 - \sigma(w^T x_i)) \; \} = \sum_{i=1}^{N} (1 - y_i)(w^T x_i) + ln(1 - e^{-w^T x_i})$$

$$\nabla NLL = x_i(\sigma(w^T x_i) - y_i)$$

training this data using one different feature. And then testing the AUC with a complete new feature it actually works better with the transfer learning. Turns out the Polynomial Feature EXT_SOURCE_2* EXT_SOURCE_3 works very well for training the data and if we use a complete different feature DAYS_BIRTH instead of the polynomial feature the model works more accurate.

This indicates that some of the features can be co-related to each other and if we use some feature engineering on them and try to use the transfer learning[11]. We can get much more better results as compare to the classical modelling technique. In which we only use the same features to train and test the data. The home credit database is a very big dataset and we can't use all the columns to train a data so we can reduce the columns but applying polynomial featuring and also can reduce more by using the transfer learning which can make the model more efficient in compare to time and accuracy. As the features will mixed so the accuracy will increases and the number of columns will reduced which will decreased the time to train a model.

- **Random Forest**

Random forests or **random decision forests** are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.[13][14] The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ with responses $Y = y_1, ..., y_n$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1, ..., B$:

1. Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.
2. Train a classification or regression tree f_b on X_b, Y_b.

After training, predictions for unseen samples $\boldsymbol{x'}$ can be made by averaging the predictions from all the individual regression trees on $\boldsymbol{x'}$:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x')$$

The entire dataset is divided into 20 parts, 19 of those divisions are used to train 19 decision trees on random bootstrapped feature set. The AUC of this model reached 0.7314845685858571 which is considerably higher than logistic or a single decision tree classifier.

- **Analysis**

Most of the algorithms are analysis on the basis some complexities in which time complexity is most important but at the same time accuracy of the result is also important. This are the two main factors to analysis any kind of algorithms. I am also analysis my models using this two top requirements. I implemented Logistic regression to get a faster solution as compare to all other algorithms I am using. Which can give the result in less then 1 minute AUC is used to check the accuracy of the models. this dataset contains 122 features which are lot to train any model In analysis I also changed features to get more accuracy from the model.

Experiment and Discussion

- **Experiment Methodology**

I am using some features of the data to train the data and I also did the polynomial features there to train and test the data. I also used multiple columns to train the naive bayes and It still has less accuracy of the model. I experiment the transfer learning on logistic regression I trained the model using one different feature form the testing data which gave a very good accuracy after the training with different features. So the transfer learning is really good on logistic regression.

- **Evaluation Criteria**

In evaluation criteria the data is on hyper plane so I am using ROC AUC for the evaluation criteria. We can use confusion matrix as we have True positive and False positive for ROC we can evaluate using confusion matrix.

- **Experimental Evaluation**
  - **Logistic Regression with EXT_SOURCE_***

Logistic regression is a very fast model to get result with less accuracy but if we tuned the features right we can increase the accuracy of the models. As we are comparing the Logistic Regression with EXT_SOURCE_* which gave the accuracy of 0.48 which is near to the Radom classifier but when we start tuning the features. We start to get more accuracy when I changed the features from EXT_SOURCE_* to AMT_CREDIT, AMT_ANNUITY, DAYS_BIRTH and EXT_SOURCE_2/EXT_SOURCE_1 which increased the accuracy from .48 to .52 but after using the Transfer Learning on this model increase the accuracy by 4% more when I changed the training feature from DAYS_BIRTH to EXT_SOURCE_2*EXT_SOURCE_3 but then we use the DAYS_BIRTH to test the model. It increased the 48% to 56% of the classifier to logistic regression.

In Random Forest the decision is made by using multiple trees which later on being done a voting or majority to find the result as this is a decision tree it is much more accurate as compare to all other methods. In this way we can cover more features to get better result and Better accuracy for the dataset.

  - **Naive Bayes**

Naive bayes is a great model which is also faster and more accurate but it doesn't support missing values in the model. We tried to choose some feature to create a model but it doesn't do very good on the AUC. But the transfer learning of logistic learning beat the AUC of naive bayes with polynomial features. If we tries transfer learning on Naive bayes which fails. Probably transfer learning won't work on naive or I chose the wrong features.

  - **Random Classifier**

Random Classifier is nothing just classifying the result randomly. I choose logistic regression which is as fast random classifier and also give a better result with the transfer learning and polynomial features. In logistic regression the AUC is 0.56 but in the random classifier is 0.51 so which conclude the logistic regression with EXT_SOURCE_* but with some polynomial features with transfer learning increased the 0.56 while the classical regression was lesser then 0.48.

  - **Classifier with fixed output**

Fixed Classifier is always giving the fixed value which can be 50 percent right and vice-versa. The AUC is 0.49 of Fixed Classifier and logistic regression with transfer learning gives the AUC 0.56. The classical logistic classifier is lesser then the fixed classifier. But the logistic regression with transfer learning increases the AUC.

- **Additional experiments**

In this experiments I have done some different kind of polynomial features I did multiplication division on the features which give great ECDF and Information Gain. I multiple EXT_SOURCE_2*EXT_SOURCE_3 which gives a great Information gain and for EXT_SOURCE_2/EXT_SOURCE_1 ECDF has a great distribution. I also tried Transfer Learning in the logistic regression which gives great accuracy as compare to the EXT_SOURCE_* features used to train logistic regression model.

We can even try some different approach to do data analysis which can give us more brief about the data such as Pearson Correlation and Chi-2.

Conclusion

I have tried transferred learning on logistic regression with different features and it gives better ROC AUC compares to other modelling algorithms there are more different types polynomial features we can implement and check they might works better with the transfer learning compare to the features I am using right now. The SVM can also be made kernelise to get more accurate result. I have implemented 2 algorithms one is faster which can use different polynomial feature from 122 columns and get more accuracy and the other is SVM which can get slower by implementing kernalize version to get more accurate classifier which can give better prediction compares to all the algorithms we have implemented so far. We can even do an Ensemble SVM which can increase the AUC as we will use different modelling features on different SVM and then combine there knowledge together using ‘‘one-member-one-vote’’ or ‘’majority-rule’’ those can also give better accuracy as there will be multiple models working with different angles to understand a dataset.

References

1. http://www.homecredit.net/about-us.aspx

2. https://www.kaggle.com/c/home-credit-default-risk

3. S S Satchidananda, Jay B.Simha, 2006. Comparing decision trees with logistic regression for credit risk analysis.

4.

5. Lakshmi Devasena C, 2015. PROFICIENCY COMPARISON OFLADTREE AND REPTREE CLASSIFIERS FOR CREDIT RISK FORECAST.

6. Amir E. Khandani, Adlar J. Kim, Andrew W. Lo, 2010. Consumer credit-risk models via machine-learning algorithms.

7. Lean Yu , Shouyang Wang , Kin Keung Lai, 2008. Credit risk assessment with a multistage neural network ensemble learning approach.

8. https://www.kaggle.com/kmanojus/classification-regression-problem

9. https://www.kaggle.com/ogrellier/good-fun-with-ligthgbm/code

10. https://www.kaggle.com/willkoehrsen/start-here-a-gentle-introduction/notebook

11. https://www.kaggle.com/kailex/tidy-xgb-all-tables-0-796/code

12. Farid Beninel, Waad Bouaguel, Ghazi Belmufti, 2000. Transfer Learning Using Logistic Regression in Credit Scoring

13. Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016

14. Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601.