St. Clair College of Applied Arts and Technology

DAB 300 Machine Learning 2

Group-45

Prediction of Credit Card Approval using Machine Learning

Date of Submission: 16th Dec 2020

Abstract:

Credit cards provide an efficient way to make payments. Banks approve or deny credit cards to the applicants based on certain characteristics of them. The objective of the project is to predict if the applicant is eligible for a credit card or is a defaulter based on the features provided. The dataset for the project is available to download from the Kaggle community. There are two csv files named 'application_record.csv' and 'credit_record.csv' which we used to create the final dataframe. Machine learning algorithms such as Decision Trees Classifier (DT), Random Forest Classifier (RF), Logistic Regression (LR) and Artificial Neural Network (ANN) have been used to predict a customer's eligibility for Credit Card. The highest accuracy is achieved by the Random Forest Classifier for default parameters i.e., 82.07%, followed by Decision Trees Classifier with default parameters (76.4%). But these models are overfitting for which hyper parameter tuning is performed. The lowest accuracy is 72.05% exhibited by the Artificial Neural Network. Overall, the classifier that yielded best results without overfitting is Random Forest classifier with best parameters and the testing accuracy is 76.03%.

Introduction:

The banking sector is considered as a field with numerous risks. One of the vulnerable risks is related to credit. Credit Card is issued to the customers by the bank to pay for the goods and services. A limit is imposed on credit cards to avoid risks of lending more money. A credit card should be issued only to customers who meet the criteria specified by the bank which helps the banks to reduce the credit loss. A credit card is different from debit card as debit card payments are directly from savings account whereas credit cards amount is being lent by the bank and need to be paid within the grace period. Most of the users prefer credit cards to debit cards as credit card payments benefit the user with more offers, cashbacks, easy transactions, etc. Usually, a bank uses set of characteristics such as address, demographic factors, income, and credit history to identify the applications that need to be approved or denied. Using the credit card but failing to pay off the monthly debts is harmful for the banks as bad debt increases. Finding and avoiding defaulters is beneficial to the banks. The prediction of ability of an applicant to repay the debt monthly is one of the crucial tasks in the banking sector. The task for this study is the classification of eligibility of applicants for their credit card approval. This classification is done by using machine learning algorithms namely; Decision Trees Classifier, Random Forest model, Logistic Regression Classifier, Artificial Neural Network.

Benchmarks:

This data was created by Xiao Song recently which is why, any research was not published by data analysts or data science students. A jupyter notebook created by Xiao Song is available in the Kaggle community which has been chosen as a benchmark for this study. The performance of classifiers such as Logistic Regression, Decision Trees Classifier, Random Forest Classifier, Support Vector Machine Classifier, Light Gradient Boosting Machine, XGBoost and CatBoost was shown in his notebook. The author used oversampling technique called SMOTENC () on the data before splitting, to overcome the problem of class imbalance. The highest accuracy was reported for the XGBoost classifier i.e., 93.79%. The lowest accuracy was 50.08% for the CatBoost classifier. The accuracies of Logistic Regression model, Decision trees classifier model, Random Forest Classifier model, SVM Classifier model, and LightGBM model are 61.22%, 82.9%, 89.46%, 59.38%, and 90.36% respectively. The author also evaluated confusion matrix for each of these models. The False Negatives and False Positives were higher in the case of CatBoost model.

Dataset:

For this study, the Credit Card Approval dataset has been chosen to classify the eligibility of the applicants. This data was created by Xiao Song and is available in the Kaggle Community. The data consists of two datasets; application_record.csv and credit_record.csv. The application_record contains 438557 samples with unique ID for each sample. There are 17 features in this dataset that represent the characteristics of each applicant. These characteristics are; gender, if the applicant owns a car, property, number of children, annual income, source of income, education, marital status, type of living (apartments/rentals/with parents), days since birth (contains negative values), days since employment (contains negative values), if the applicant has a mobile phone, work phone, landline, if the applicant has an email, type of occupation, and number of family members. The credit_record contains 1048575 samples with repetitive IDs for each. It has two features of which one is the target feature which represents the applicant's eligibility for a credit card based on the number of days the applicant took to pay his/her debts. It has 8 unique classes; C, X, 0, 1, 2, 3, 4, and 5. C, X states that the applicant has either paid the debt or has not borrowed loan. 0, 1, 2, 3, 4, and 5 represent the applicant has paid the debt numerous days past due date. Other feature is the number of months till the data extraction (contains negative values) i.e., extracted month is 0 and the previous month is -1 and so on.

Data Preprocessing:

The two datasets; application_record and credit_record, are read into two dataframes. The MONTHS_BALANCE feature in the credit_record dataframe has all negative values. In order to convert it to positive, it is multiplied by -1, then it is aggregated by grouping by ID column. Now there are 45985 unique IDs and aggregated MONTHS_BALANCE feature in credit_record dataframe. This dataframe is merged with application_record on ID column.

There are 36457 unique IDs common in both the dataframes i.e., the number of samples get reduced to 36457. The original credit_record dataframe contains STATUS feature which is the target feature. The multi-class feature is converted to binary by converting the C, X to 1 and remaining classes to 0. Here 1 represents good customer, the application can be approved and 0 for bad customer, the application needs to be denied. Then the MONTHS_BALANCE feature is dropped from the dataframe as it is already present in the merged dataframe and the ID column is made unique by dropping duplicates in the column. The dataframe that is merged before (that contains MONTHS_BALANCE) and the dataframe that contains ID and TARGET are merged on ID column. This is the complete dataframe. This dataframe is checked for missing values. It is found that there are 11323 missing values in the OCCUPATION_TYPE column. These missing values are filled with the most frequent type in the OCCUPATION_TYPE. This is the cleaned dataframe. Now, the ID column is dropped and the dataframe is checked for duplicate samples. 3581 duplicate samples are found and dropped from the dataframe. Now, there are 32876 samples in the dataframe. The categorical variables are identified, and encoding is done to convert them to numeric. For example, CODE_GENDER feature outlines the gender of the applicants i.e., it contains M (male) and F (female). These M and F are replaced by 1 and 0. Similarly, features such as FLAG_OWN_CAR, FLAG_OWN_REALTY, NAME_INCOME_TYPE, NAME_EDUCATION_TYPE, NAME_FAMILY_STATUS, NAME_HOUSING_TYPE, and OCCUPATION_TYPE are encoded the same way as CODE_GENDER. The feature DAYS_BIRTH and DAYS_EMPLOYED comprise of negative values for days. These columns are converted to positive and each value of them is divided by 365 to convert them to years. It is found that there are negative values in DAYS_EMPLOYED which means that the applicants with those values are unemployed and set to 0. DAYS_BIRTH and DAYS_EMPLOYED are renamed to AGE and EXPERINECE respectively.

Exploratory Data Analysis (EDA):

A pie chart is plotted depicting the gender proportion. There are 66.94% female applicants and 33.06% male applicants in the data. The distribution of types of living is plotted in a bar chart. It is observed that most of the applicants are living in their house/apartment and least number of applicants live in co-op apartments. The histogram that illustrates the Age of applicants is plotted and it is found that majority of the applicants age between 25 to 48 years and applicants with age 20 to 24 years are less in number. A boxplot has been plotted for the distribution of Income of applicants. The perception is there are applicants with income between $120k to $210k. Another histogram that depicts the work experience of the applicants is plotted and it is noticed that applicants with none to 8 years of experience are more in number and applicants with 25 years and above are very less in number. Finally, a correlation matrix is plotted, and it is observed that there is a strong positive relationship between CNT_CHILDREN and CNT_FAM_MEMBERS.

Methods:

Before building the classification models, the dataframe is divided into two sets; one set contains all the features and the other set contains the target/class features. Class imbalance is observed, and an oversampling technique called SMOTENC (Synthetic Minority Over-sampling Technique for Nominal and Continuous) to increase the samples of the minority class is used to overcome the class imbalance problem. Then, both the features set, and target set are split into training and testing sets. Scaling is done on using the MinMaxScalar() before building the models. In this section, the steps followed to build the classification models are discussed.

Decision Trees Classifier:

The decision trees classifier is one of the supervised machine learning algorithms that works on if-else condition. The decision trees classifier is imported, and a model is built using default parameters. The training set accuracy, testing set accuracy, precision, recall, and F-measure are computed. Then a confusion matrix is plotted. After computing the above metrics, another decision trees classifier using hyper parameter tuning is built for improving the accuracy of decision trees model with default parameters. A parameter grid with parameters such as max_depth, max_leaf_nodes, min_samples_split, are used with list of values for each of them. 5-folds are used for cross validation in the GridSearchCV(). The train and test accuracies are computed, and the best parameters have been obtained for the model.

Random Forest Classifier:

Random Forest Classifier is one of the ensemble machine learning algorithms that generates a set of decision trees from a randomly selected samples of training data. First, a random forest classifier model is built with default parameters. The metrics such as train accuracy, test accuracy, precision, recall, F-measure are computed, and a confusion matrix is plotted. In order to improve the accuracy, random forest classifier with best parameters is also built by passing parameter grid created with parameters and their respective list of values, and cross validation of 5 folds in GridSearchCV(). Then the train and test accuracies are computed for the best parameters.

Logistic Regression:

Though there is regression in its name, logistic regression is a classification machine learning algorithm. It is good for binary classification problems. The logistic regression model is built using default parameters. Train accuracy, test accuracy, precision, recall, F-measure are computed for the model. A confusion matrix is also plotted for the model. Another logistic regression model is built using hyper parameter tuning to enhance the accuracy of the model. A parameter grid is created with two parameters; C containing values list [0.01, 0.1, 1, 2, 10, 100] and penalty with values list ['l1', 'l2']. This parameter grid and cross validation of 5 folds

is passed in the GridsearchCV(). After that, the train and test accuracies are computed, and best parameters are obtained for the model.

Artificial Neural Network:

Artificial Neural network mimics human nerve cells to analyse and process the data. An artificial neural network is built with 3 layers; 1 input layer, 1 hidden layer, and 1 output layer. The input layer consists of 10 neurons, Rectified linear activation function, a kernel initializer and regularizer. The one and only hidden layer contains 6 neurons, rectified linear activation function, a kernel initializer and a regularizer. The output layer has one neuron and sigmoid activation function is used. The 'Adam' optimizer is used and 'binary crossentropy' is used as loss function in the compilation of the network. The network is fit with the training data with 10 epochs, 128 as batch_size and 20% validation split. The training and testing accuracies have been computed and accuracies and losses are plotted.

Results:

The below screenshot comprises of the training and testing accuracies of the models built for the project sorted by alphabetical order.

	Model	Training Accuracy	Testing Accuracy
6	Artificial Neural Network	72.208303	72.070533
0	Decision Tree	98.781082	76.401755
1	Decision Tree with Best Parameters	75.971071	75.613522
4	Logistic Regression	73.105260	72.574354
5	Logistic Regression with Best Parameters	73.942250	73.549488
2	Random Forest	98.781082	82.073785
3	Random Forest with Best Parameters	76.491143	76.027954

The Artificial Neural Network has a training accuracy of 72.21% and a testing accuracy of 72.07%. The training and testing accuracies of the decision trees classifier are 98.78% and 76.40% respectively which exhibits overfitting. The precision, recall, F-measure of the model are 77.05%, 74.81%, 75.91% respectively. To overcome the overfitting problem, hyper parameter tuning is performed on decision trees classifier. The training and testing accuracies of that model are 75.97% and 75.61% respectively and the best parameters are 'max_depth': 10, 'max_leaf_nodes': 30, 'min_samples_split': 20. Random forest classifier has faced the same problem as decision trees classifier (overfitting). Random Forest Classifier with default parameters has a training accuracy of 98.78% and a testing accuracy of 82.07%. The precision,

recall, F-measure of the model are 81.57%, 82.61%, 82.08% respectively. The accuracies of train and test sets for random forest classifier with best parameters are 76.49% and 76.03% and the best parameters are 'max_depth': 30, 'max_leaf_nodes': 22, 'min_samples_split': 50. Logistic Regression model with default parameters is built, training accuracy is 73.11% and testing accuracy is 72.57%. The precision, recall, F-measure of the model are 73.44 %, 70.23%, 71.8% respectively. To improve the accuracy, logistic regression model with best parameters is built. There is slight increase in the accuracies, the training and testing accuracies of the model are 73.94% and 73.55% respectively. The best parameters are 'C': 10, 'penalty': 'l2'.

Discussion:

The objective of this study is to classify the good applicants i.e., who pay the debts in time, and bad applicants i.e., who pay the debts past the due time, based on their features for credit card approval. To achieve this objective, models using different machine learning algorithms have been built, evaluated and their accuracies have been reported. The random forest classifier with default parameters has the highest testing accuracy of 82.07% but the model is overfitting. But after performing hyper parameter tuning, it is observed that overfitting is reduced and the accuracy of that model is 76.03% whereas the lowest accuracy is reported for Artificial Neural Network i.e., 72.05%. One of the challenges faced during the study is merging the application_record (contains features) and credit_record (contains target) dataframes. These dataframes are to be merged on the ID column present in both the dataframes. But the IDs are repetitive in the credit record dataframe and only 36457 IDs are common for both dataframes which made it quite challenging. Dealing with the class imbalance is another challenge. After some research, an oversampling technique called SMOTENC has been utilized to increase the number of samples for the minority class to overcome this problem. Please find the link to the article about SMOTENC in the references. Another challenge is to overcome the overfitting problem in random forest classifier and decision trees classifier. Hyper parameter tuning has been used for both the classifier models and different ranges of values have been passed for the parameters in the parameter grid to reduce overfitting. One failure we have encountered was in the case of a feature called ‘DAYS_EMPLOYED’. Initially, this feature contained almost negative values i.e., the number of days since the applicant’s employment counted from present. After converting the days into positive and the years, it is observed that there were still negative values which were positive before conversion. These values are replaced with 0 indicating that they are not employed.

Conclusion:

Though the model with highest testing accuracy is Random Forest Classifier with default parameters, it is not considered to be the best because of overfitting. The best classifier model for predicting the eligibility of the applicants for credit card approval is Random Forest Classifier with Best Parameters. It has obtained a training accuracy of 76.49% and testing accuracy of 76.03%. Other classification models like Support Vector Machines classifier, and

XGBoost classifier can be built and evaluated. This dataset has class imbalance problem. Instead of using the oversampling techniques, real time data can be collected for the minority class and used for classification. The categorical variables can be encoded using one hot encoding.

Contributions:

Student Name	Student Number	Section	Contributions
Hari Sai Palem	0747511	001	I have performed Exploratory Data Analysis and built the logistic regression model with default parameters and with best parameters using GridSearchCV. I have documented the Abstract, Introduction, Benchmarks, Dataset, Exploratory data analysis, logistic regression part of methods section of the final report.
Manoj Reddy Lenkala	0753802	001	I have performed encoding for the categorical features and binary features, solved the class imbalance problem and built the Artificial Neural Network (ANN). I have also documented the part of ANN in the methods section of the final report.
Naveen Kumar Tedla	0753623	004	I have merged dataframes, dealt with missing values and duplicates, and built Random Forest Classifier with default parameters and found that the model is overfitting. I have used hyper parameter tuning on the random forest classifier. I have documented the part of random forest classifier model in the methods section the final report.
Sunil Kumar Vidam	0735027	005	I have built the decision trees classifier with default parameters and observed there is overfitting. To overcome the overfitting, I have performed hyper parameter tuning on decision trees classifier. I have documented the decision trees classifier model part of methods section, Results, Discussion and Conclusion sections of the final report.

References:

- Oversampling technique SMOTENC: *https://imbalancedlearn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTENC.html#imblearn.over_sampling.SMOTENC*.
- For building the fully connected Neural Network: *https://keras.io/guides/sequential_model/*.
- Dataset Source: *https://www.kaggle.com/rikdifos/credit-card-approval-prediction*.
- GridSearchCV for hyper parameter tuning: *https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html*.
- Scikit learn: *https://scikit-learn.org/stable/*.
- Keras documentation link: *https://keras.io/about/*.
- This article was helpful in gaining credit card information for this report: *https://www.themint.org/teens/credit-card-facts.html*.
- Benchwork model: *https://www.kaggle.com/rikdifos/credit-card-approval-prediction-using-ml*.

Appendices:

1. DAB 300 - Machine Learning 2 - Group 45 - Final Project.ipynb

   - This file contains all the code for Reading the data, Data Preprocessing, Exploratory Data Analysis, Model building and Evaluation.

2. Datasets:

   - This .csv file contains the applicants features.

   - This .csv file contains the ID, number of months before data extraction, Target feature.