Concordia University

Final Report

SOEN6111

Credit Card Score Forecaster with Supervised Learning

Author: Somayeh Ghahary — *Author ID:* 40106359

Mehrnoosh Amjadi — 40091264

Submitted to: Dr. Tristan Glatard

April 18, 2021

1 Abstract

The role of credit cards in facilitating transactions is undeniable. Banks provide such services to their customers and try to surpass their competitors in this field. However, it is also very crucial for banks to offer such services to those who can pay their statements on time. In this project, we investigate our time to tackle this challenge. In fact, we propose an automatic system that can predict customers who may not be able to pay their statements on time and vice versa. Applying supervised learning leads to ideal consequences in similar prediction projects. Our primary goal is to learn how we can apply different machine learning algorithms for predicting good or bad customers and the secondary goal of the project is to observe the impact of INCOME feature in our predictions.

2 Introduction

In the business world, profit and interest are very important. Banks and financial institutions are looking for plans that can be more profitable and prevent potential losses. Therefore, providing services such as credit cards to those who are able to pay their statements on time is vital for banks managers. By applying this system, banks and financial institutes would be able to decide to accept or reject the credit card application of a customer by categorizing that customer to a "good" or "bad" client.

On the other hand, with the unprecedented growth of the data in the financial industry, banks are interested to have an automated mechanism to decide about issuing credit cards to their costumers, instead of old manual methods. Machine learning models facilitate the process of predicting the score of credit cards in compare to traditional methods. Although many literature and competitions have valuable results in this field like [4] and [2], we want to propose and implement an idea for forecasting the credit card scores with multiple models in supervised learning.

The primary goal of this project is to apply Machine Learning methods on the chosen datasets to implement a supervised learning for predicting the score of credit card of customers according to their historical and financial records and the secondary goal is to analyze the effect of one special feature on the result of predictions.

The rest of the report is organized as follows: Section 3 provides a discussion on relevant information about methodology including datasets selection, preprocessing, visualization, splitting and labeling the data, algorithm selection and required tools and technologies. Then the conclusion of our project is provided in section 4 describing the evaluation and results. Finally, discussion, limitation and future work are mentioned in section 5.

3 Materials and Methods

We use general framework of the machine learning to go through implementing this project. We follow these phases: dataset selection and preparation, preprocessing, visualizing, splitting dataset to train, validate and test sets, algorithm selection, training, predicting and finally evaluating. Details of each phase are described bellow:

3.1 Dataset Selection and Preparation

We decide to use "Credit Card Approval Prediction" data [1] as the dataset of the project since it seems to have features that help the progress of the project. It includes two datasets, which are connected by ID. The "application_record.csv" file consists of personal information of over 440,000

clients with 18 features like family status, number of members in the family, age, income type, amount of income, properties owning, etc. The other dataset "credit_record.csv" records history of credit card of all the costumers such as status containing over 1,000,000 records in 3 columns. The both dataset files contain a variety of heterogeneous data recorded in the text and numeric formats. They might have empty, duplicate values and outliers. We merge this two datasets and follow preprocessing. We come to the idea of analysing the effect of INCOME AMOUNT feature by making two different datasets. One of them includes all the features and the other dataset excludes INCOME AMOUNT feature. All the next phases execute on both of these datasets. Finaly we evelaute the effect of this feature.

3.2 Dataset preprocessing

To get valuable results, we need a comprehensive preprocessing phase. First we drop unnecessary features and handling null and empty values. We drop OCCUPATION TYPE, since we noticed that we have lots of missed values in this column. Meanwhile, we have other features like INCOME TYPE and AMOUNT of INCOME that can compensate the missing of OCCUPATION TYPE feature, because they have more accurate information. While, all the costumers have a mobile phone and HAVING MOBILE flag is set to 1 for all of the them, we decide to drop it.

We use LabelEncoder to convert text values to numerical values. Also, we convert types of features from object to integer and float. Scikit-learn models need features in numeric type. After visualization, we noticed that NUMBER of CHILDREN, NUMBER of FAMILY MEMBERS and INCOME AMOUNT have outliers 1, we remove outliers from them. For important STATUS feature that expresses the history information of the debt payment of the costumers, we convert STATUS values to two values 0 for who paid off that month or didn't have loan and 1 for who passed due date to pay their debts. We get some idea for preprocessing from [5].

Finally we merge two datasets based on their unique IDs and then drop ID feature. Since each feature has a variant range, we normalize all the columns to scale them. Moreover, by oversampling the minority class we overcome the imbalance issue of the data.

3.3 Defining Train, Validate and Test Datasets

We consider STATUS feature as a label or y axis, the feature or X axis consists all the other features. Then we split dataset to train, validate and test datasets, each one comprises respectively 60%, 20%, 20% of original data.

3.4 Visualization

We comp up with a bunch of plots and charts to visualize the features and get a better sense of them as it is usual at most of the machine learning projects. We plot feature distribution 1 and find outliers in some columns. The credit card STATUS plot shows that data is imbalance. Also, we add correlation matrix of features. In addition, for conclusion we utilize plotting for evaluation phase on the prediction results.

3.5 Algorithm Selection

In this project, we utilize six famous machine learning algorithms including Decision Tree, Random Forest, K-Nearest Neighbors, Logistic Regression, Support Vector Machine and Neural Network MLPerceptron Model. For each of them, we set different stochastic parameters (For example: for

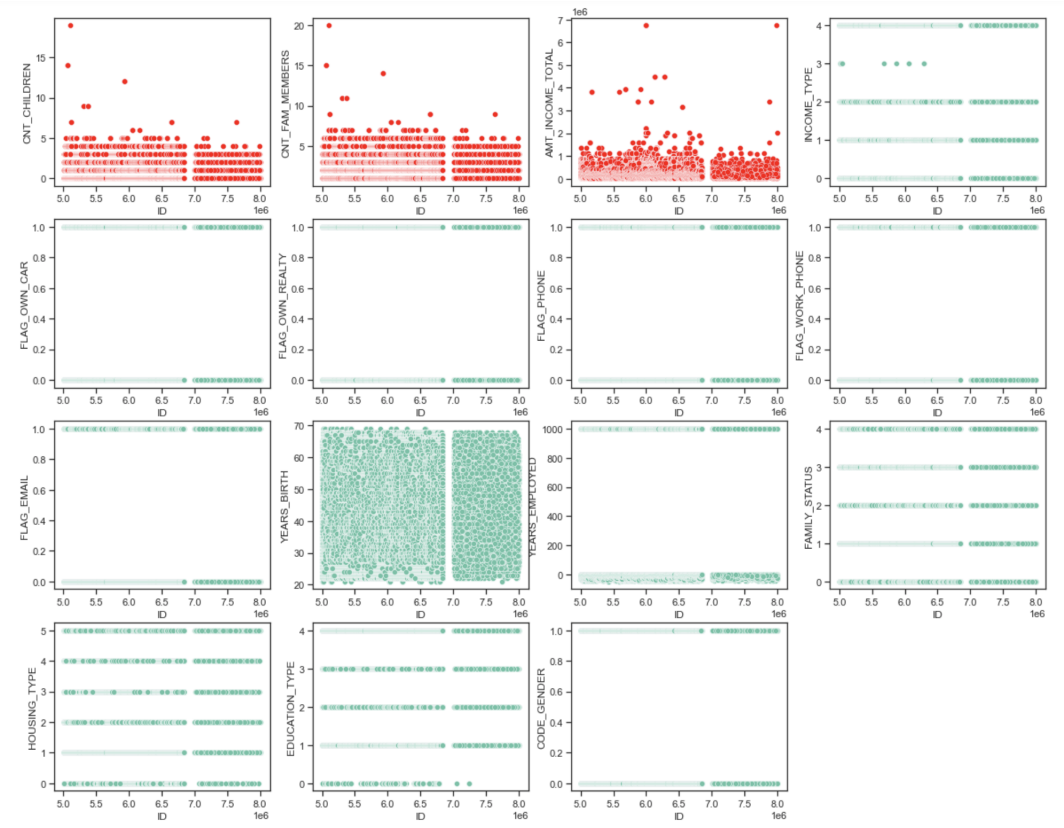

Figure 1: feature distribution

Decision Tree, we try six different max-depth) to find the best estimator among them. The best estimators are obtained based on the best results from accuracy score of validation set. We accomplished these training phase for both datasets (including and excluding INCOME). Then, we predict the test data using the best estimator we obtain for each of the classifiers. Also, for comparing the results and find the best classifiers, we compare the results from accuracy scores, precision, recall and f1-score perspectives. Finally, we visualize the results of predictions using Confusion Matrix (from Scikit-learn) to have a better sense of True positive, True negative, False positive and False negative predictions of predicted labels.

Tools	Purpose
Python3.6, Anaconda, Jupyter Notebook	Programming Language and IDEs
Numpy, Pandas	Numerical Analysis
Matplotlib, Seaborn	Visualization and Plotting
Scikit-learn [3]	Machine Learning Packages

Table 1: Tools and technologies used

3.6 Tools and Technologies

During implementing different phases of the project, we apply the following tools and technologies in table 1.

4 Evaluation and Results

As it is shown in the Figure 2 and 3, by comparing the results obtained, Random Forest has the best performance (best F1 Score) relative to five other algorithms. Decision Tree and K-Nearest Neighbors are the second and third classifiers which have good performances. However, in terms of total running time (including training, predictions and plotting the confusion matrix and accuracy score), K-Nearest Neighbors has the longest running time with 1822 seconds for dataset1 and 1654 seconds for dataset2. Decision Tree is the fastest classifier with 6.5 seconds running time for dataset1 and 5.72 seconds for dataset2. Also, if we want to compare the results from two datasets (dataset1 including INCOME and dataset2 excluding INCOME), we come to this conclusion that INCOME feature has a slight impact on predictions. As you can see, the f1 scores are only improved 1 or 2 percent in all the classifiers in dataset1. Another observation is that the total running time of classifiers of dataset2 is approximately lower than the total running time of classifiers of dataset1 and this is not far from the expectation because dataset1 has one more feature which can take more time during training and prediction phases. Finally, we plot confusion matrix for each classifier to have a better sense about true and false predicated class labels. Figure 4 indicates that Random Forest classifier predicts 43631 True Positive (good customers) class labels in dataset1. While True Negative contains 40000 labels. Therefore, we can have an overview about True Positive, True Negative, False Positive and False Negative predicted labels.

	Estimator	Accuracy	Precision	Recall	F1_Score	Running Time (s)
0	DecisionTreeClassifier	84.71	98.04	70.84	82.25	6.5
1	RandomForestClassifier	91.12	93.93	87.92	90.83	293.8
2	KNeighborsClassifier	80.44	84.38	74.71	79.25	1822
3	LogisticRegression	56.18	56.43	54.24	55.31	9.063
4	SVC	52.18	51.26	88.85	65.01	433
5	MLPClassifier	56.94	58.56	47.48	52.44	257.8

Figure 2: Dataset1 Evaluation

5 Discussion

To discuss about relevance of the solutions, we can say that all of the models which we use are supervised classifiers which needs labels. In addition, we utilize implemented models in Scikit-learn library to train and predict. These models are limited to the size of the memory, which means, we cannot test a dataset which is larger than the memory size. Since, Scikit-learn library is a single-thread method, these models cannot train and predict in multi-threaded and parallelized way.

In this project, we utilize some stochastic parameters for each classifiers and find the best estimator among them. However, there may exist other parameters which can perform better. For

	Estimator	Accuracy	Precision	Recall	F1_Score	Running Time (s)
0	DecisionTreeClassifier	84.64	97.92	70.78	82.17	5.72
1	RandomForestClassifier	90.02	93.42	86.11	89.62	234.5
2	KNeighborsClassifier	80.08	83.98	74.35	78.87	1654
3	LogisticRegression	56.27	56.50	54.53	55.50	7.656
4	SVC	52.21	51.50	75.79	61.33	446
5	MLPClassifier	60.86	63.25	51.82	56.97	180.77

Figure 3: Dataset2 Evaluation

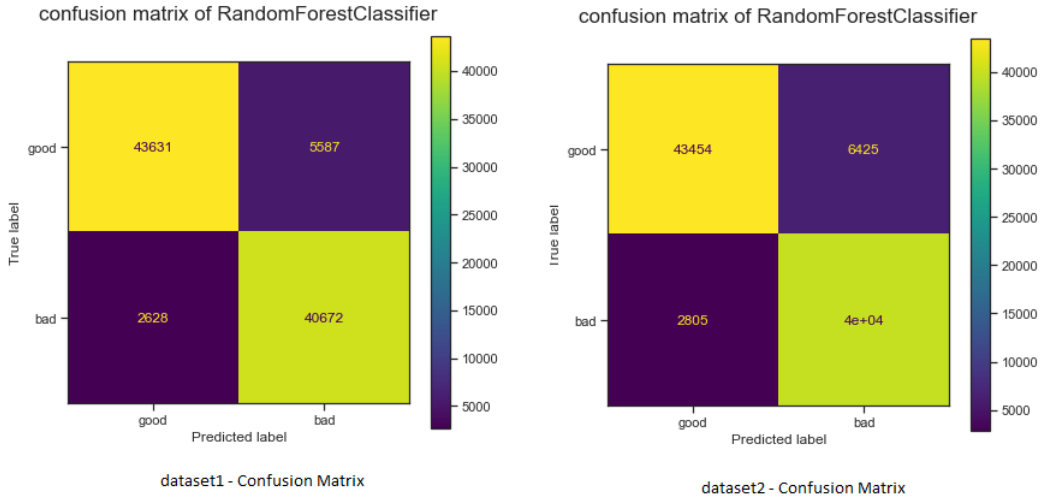

Figure 4: Two datasets Random Forest Confusion matrix

example, there may exist other parameters for C (regularization value) which can results better performance for Logistic Regression classifier. In the next step of the project, we may use randomized search to find hyper-parameters for the classifiers.

By training the models on different datasets, we can analyze if we reach to same performance, which shows our results are dependent to data or not. As, K-Nearest Neighbors is the classifier which has a good performance but longest running time. In the next phase, we can implement K-Nearest Neighbors from scratch with Pyspark and Dask and compare the results with Scikit-learn. Ultimately, since we have a huge dataset, we can take advantages of parallelization of the data using Spark and Dask to reduce the running time.

References

[1] Credit card approval prediction.

[2] Omareltouny. Credit card approval, Aug 2020.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[4] Rikdifos. Credit card approval prediction using ml, Nov 2020.

[5] umerkk12. Credit card predictive analysis, Jan 2021.