Part one and two model selection exercises

Reza Mousavi

Computer Science Faculty, Shahid Beheshti University

18 December 2023

ABSTRACT

This data science lesson report explores a simulated credit card transaction dataset spanning January 1, 2019, to December 31, 2020. The dataset, generated using the Sparkov Data Generation tool by Brandon Harris, encompasses legitimate and fraudulent transactions involving 1000 customers and 800 merchants. The simulation process involves predefined lists of merchants, customers, and transaction categories, utilizing the ”faker” Python library to simulate transactions.Simulation profiles, such as ”adults2550femalerural.json,” define transaction properties based on age, gender, and location. These properties include parameters like daily transaction limits, distribution across days of the week, and normal distribution properties for transaction amounts. The report outlines the simulation methodology, including the use of different profiles and the merging of transactions to create a realistic representation.Acknowledging Brandon Harris for his tool, the report underscores its utility in generating authentic fraud transaction datasets. The analysis of this dataset offers insights into transaction patterns, anomaly detection for potential fraud, and the evaluation of fraud detection algorithms in the credit card transaction domain.

1. Introduction

The primary challenge encountered while working with this dataset revolves around the pronounced class imbalance between the two categories. Class 0 comprises an extensive dataset of nearly 1.2 million records, while Class 1 is represented by a significantly smaller dataset of only 7,500 records. Addressing this imbalance is crucial for building an effective predictive model. To tackle this issue, various methods have been employed, including undersampling techniques.In approaching the predictive task, classic machine learning methods are applied. The significance of feature engineering and feature selection methods is underscored to enhance the model’s efficacy. The analysis encompasses a spectrum of tree-based models, such as decision trees, XGBoost, and Random Forest, along with other methodologies like Support Vector Machines (SVM), logistic regression, among others. The selection of an appropriate model is a critical step in ensuring the accuracy and reliability of the predictive outcomes.

2. Exploratory Data Analysis

2.0..1. Overview of the data

First, we will examine the proportion of is fraud classes in a general way:

The imbalance of the data of each class can be easily understood from the above diagram that the number of data of class one is less than one percent of the data1.Now we will examine the amount of the transaction against the possibility of fraud: Based on the above diagram, it is quite clear that if the amount of transaction increases from 250, the probability of fraud should increase greatly, and the distribution of these two classes based on the amount of transaction are at a suitable distance from each other. According to the diagram, we can see that if the amount of transaction increases from 250 We should expect cheating.2

[H]

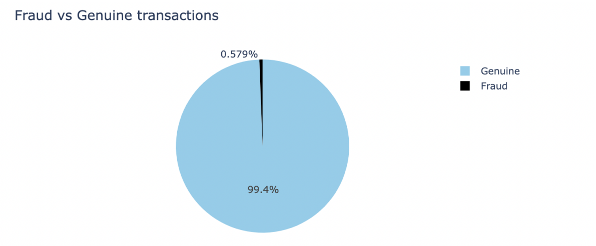


Fig. 1. The percentage of the number of data in each class

Now we are going to examine the age distribution of people according to their cheating: Based on the above trend, it is clear that most of the frauds occurred in the game between the ages of 30 and 55. Based on the previous results, this type of fraud is expected to have a very good relationship with the amount of transactions.3

[H]

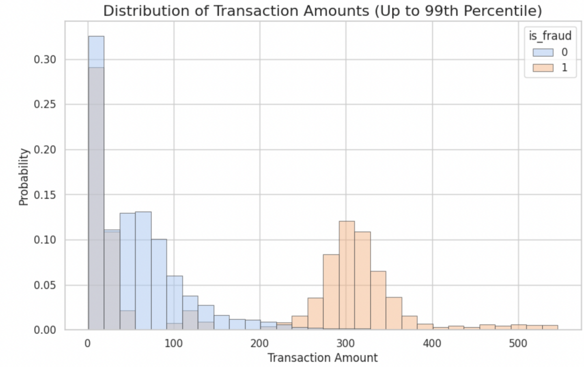


Fig. 2. Fraud probability distribution according to transaction amount

[H]

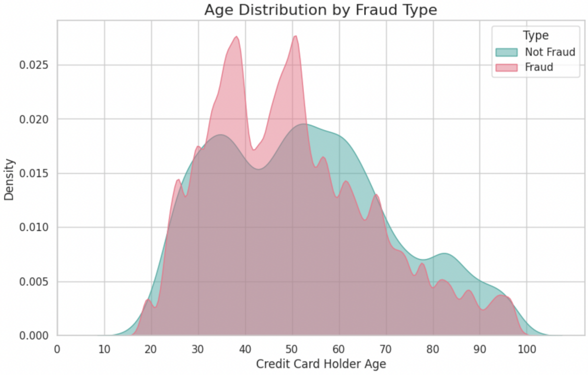


Fig. 3. Age distribution of people based on cheating

2.0..2. Data Preprocessing

Like the age feature introduced above, we extract the month and hour features from the trans date trans time column. Then we log the data related to the amt column and put it in the data. We use this procedure to scale this column.Finally, we WOEEncoder the features ”category”, ”state”, ”city”, ”job”.

2.0..3. Undersampling

In the undersampling approach, we have systematically reduced the dataset to 82,000 data lines with the goal of achieving a more balanced representation of both classes. This reduction is a deliberate effort to mitigate the substantial class imbalance observed in the original dataset, where Class 0 is predominant with nearly 1.2 million records, while Class 1 is underrepresented with only 7,500 records. By downsizing the dataset, we aim to create a more equitable distribution between the two classes, providing a foundation for more robust and unbiased model training and evaluation.

[H]

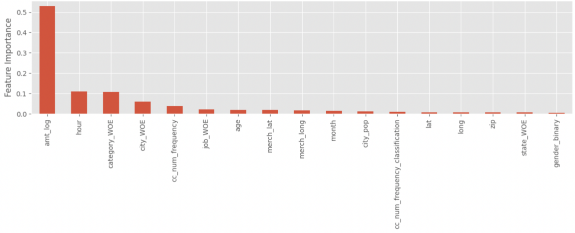


Fig. 4. Selection of the most important features

2.0..4. Feature Selection

In the feature selection process, we trained a random forest model using all the numerical features extracted from the dataset, including those that were originally present in the data. The model’s evaluation allowed us to determine the importance of each feature. Based on this evaluation, we obtained a ranked list of features, where each feature is assigned an importance score.This approach enables us to identify the most influential features for the predictive task. The feature selection step is crucial for enhancing model performance, as it allows us to focus on the most relevant aspects of the data and discard less informative variables, thereby streamlining the model training process and potentially improving its accuracy.4 Finally, according to the above explanation, we select the first 8 important features that are the most important.

2.1. Results Of The Models

2.1..1. logistic Regression

5

2.1..2. SVM

6

2.1..3. KNN

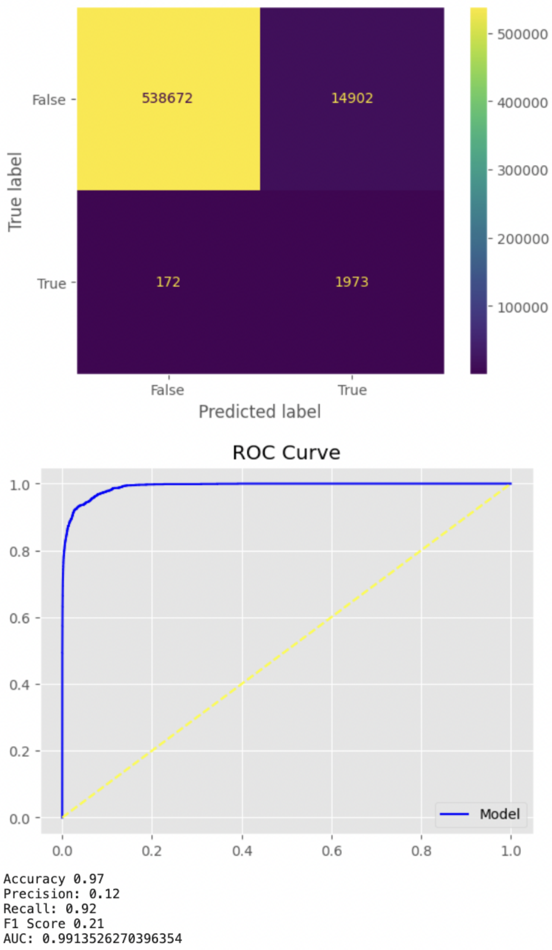
7

2.1..4. Random Forest

8

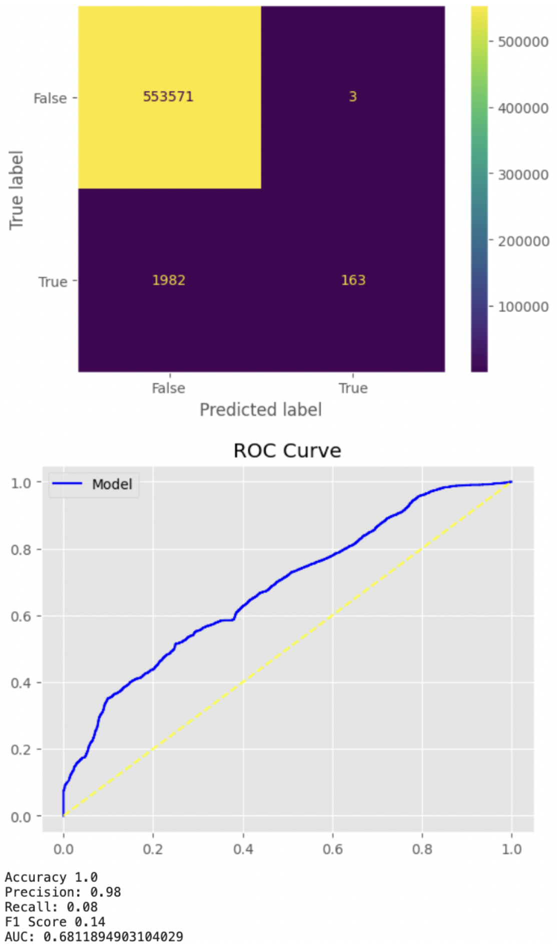
2.1..5. Navy Base

9



```
Accuracy 0.97
Precision: 0.12
Recall: 0.92
F1 Score 0.21
AUC: 0.9913526270396354
```

Fig. 5. logistic Regression



```
Accuracy 1.0
Precision: 0.98
Recall: 0.08
F1 Score 0.14
AUC: 0.6811894903104029
```

Fig. 6. SVM

2.1..6. Design Tree

10

2.1..7. Light Gradient Boosting

11

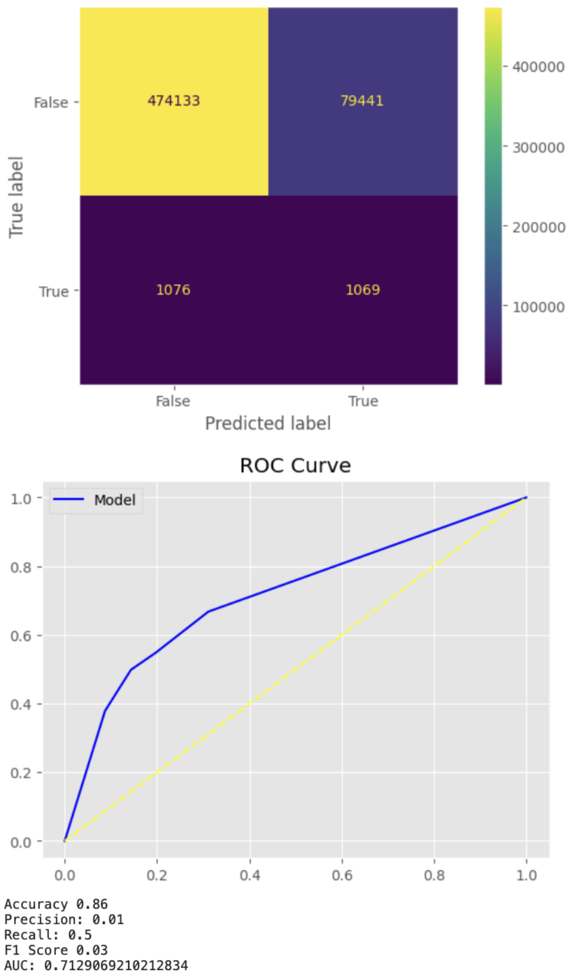
3. Conclusion of part one

The LGB (Light Gradient Boosting) model emerged as the most successful among all the models evaluated. Notably, many models demonstrated high recall for Class 1, indicating their ability to effectively identify instances of this minority class. However, what sets the LGB model apart is its ability to strike a balance between precision and recall, as evidenced by its favorable F1 score.The F1 criterion, which considers both precision and recall, is a particularly important metric for tasks where imbalanced classes are present. In this context, the LGB model achieved an F1 score of 48, signifying a well-balanced performance that avoids overly favoring one class over the other. This balance is crucial to ensure that the model does not overly generalize or misclassify instances, making it a more suitable choice compared to other models evaluated in this study.

4. Part II

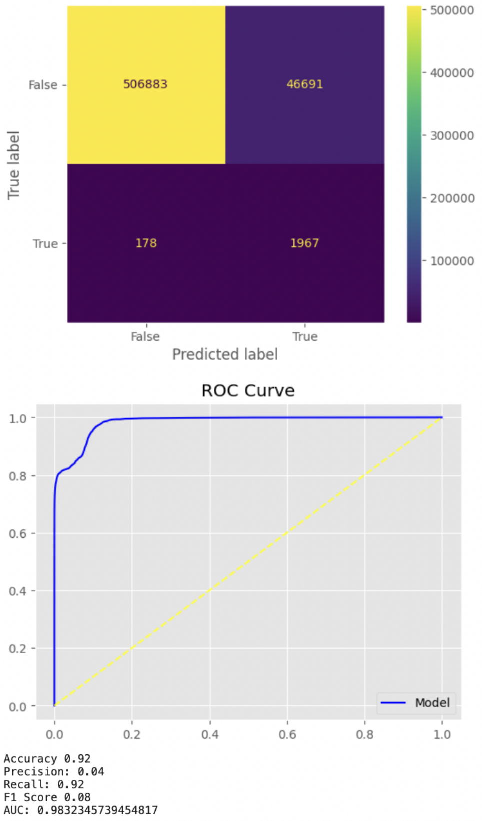
In this section, a preprocessing step involves flattening each 28 x 28 image and concatenating the pixel values into a single row. This process results in a data frame with dimensions of 249,540 rows and 784 features, where each row represents an image. The purpose of this transformation is to enhance the efficiency of the PCA (Principal Component Analysis) algorithm by reducing the computational burden and accelerating the learning speed.

Subsequently, the PCA algorithm is applied to reduce the dimensionality of the dataset to 20 features. This reduction aims to retain the most significant information while minimizing the



```
Accuracy 0.86
Precision: 0.01
Recall: 0.5
F1 Score 0.03
AUC: 0.7129069210212834
```

Fig. 7. KNN



```
Accuracy 0.92
Precision: 0.04
Recall: 0.92
F1 Score 0.08
AUC: 0.9832345739454817
```

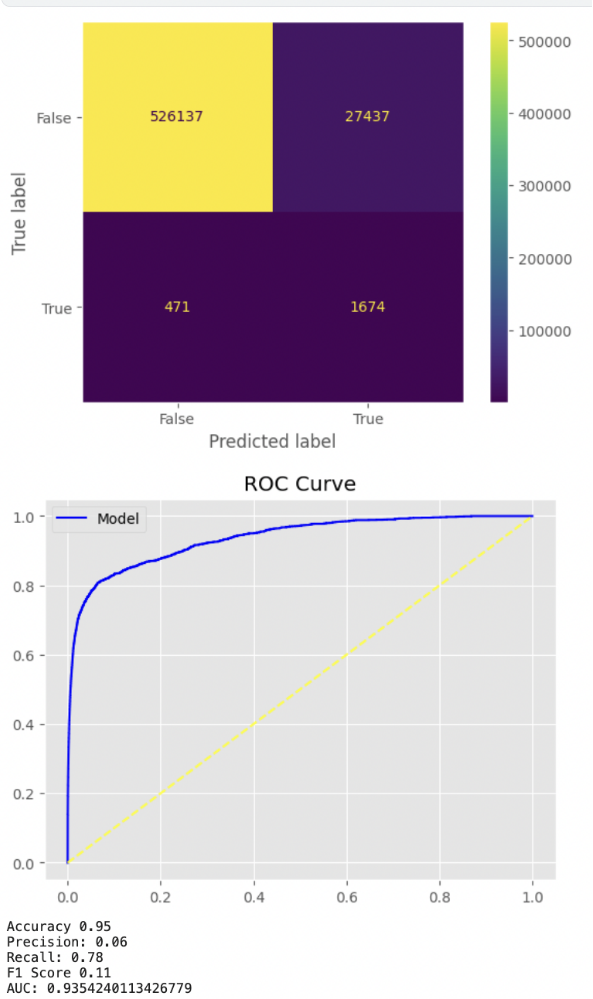
Fig. 8. Random Forest

computational complexity. After implementing the PCA, a random forest model is trained on the transformed data.

The outcomes of the random forest model are then assessed for performance metrics, providing insights into the model's ability to capture relevant patterns and relationships in the image data. The application of PCA, coupled with the random forest model, offers a streamlined approach for feature reduction and efficient image classification.12

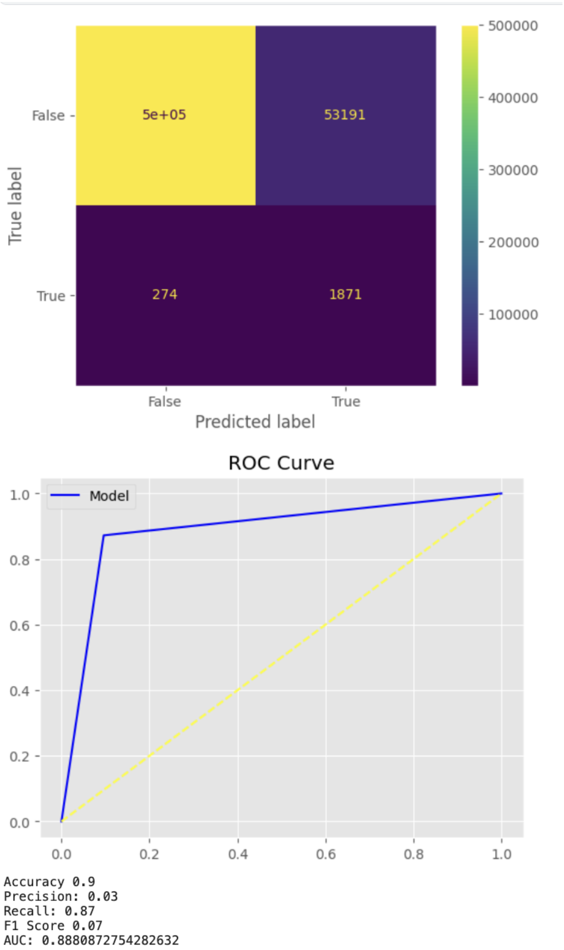
References

1. Kaggle
2. matplotlib



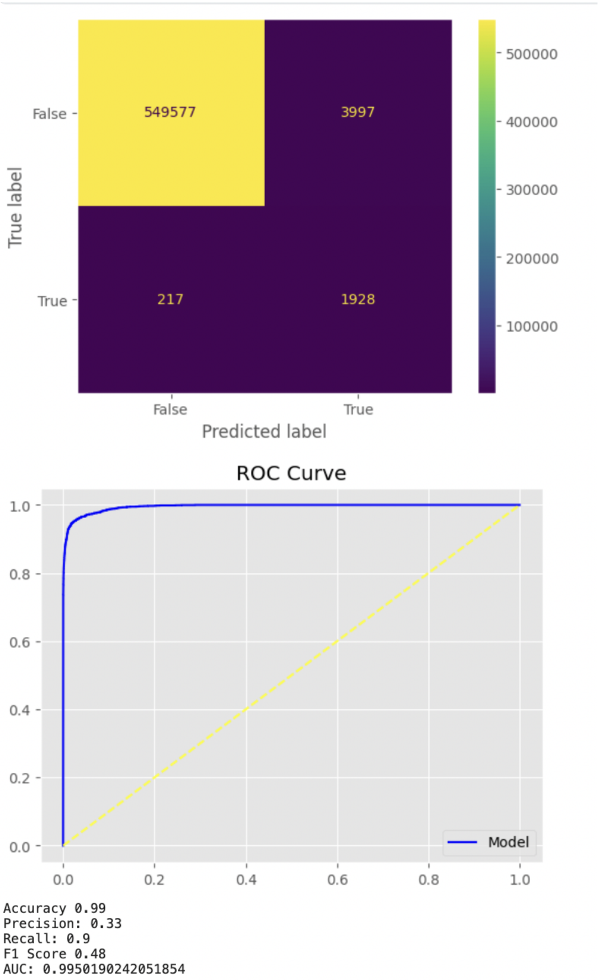
```
Accuracy 0.95
Precision: 0.06
Recall: 0.78
F1 Score 0.11
AUC: 0.9354240113426779
```

Fig. 9. Navy Base



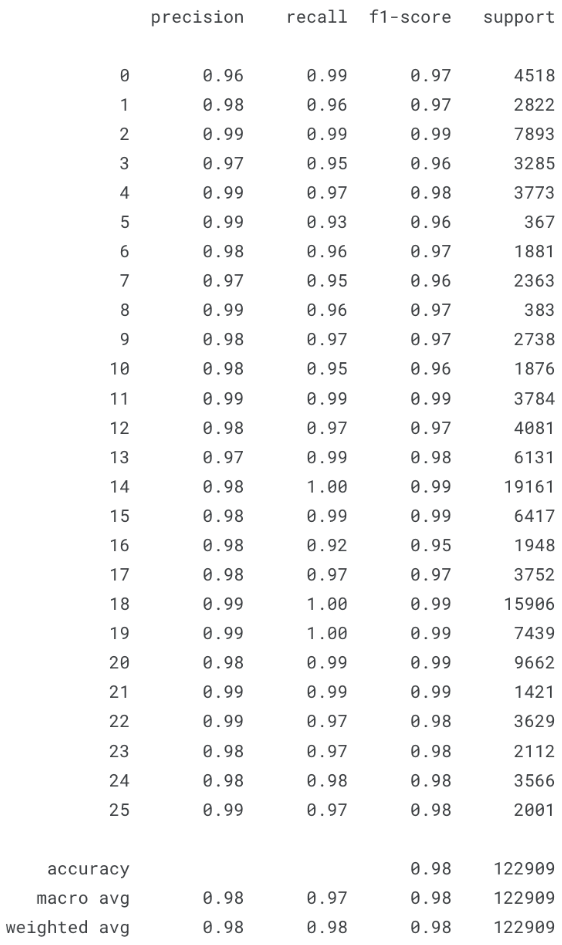
```
Accuracy 0.9
Precision: 0.03
Recall: 0.87
F1 Score 0.07
AUC: 0.8880872754282632
```

Fig. 10. Design Tree



```
Accuracy 0.99
Precision: 0.33
Recall: 0.9
F1 Score 0.48
AUC: 0.9950190242051854
```

Fig. 11. Light Gradient Boosting



```
              precision    recall  f1-score   support

           0       0.96      0.99      0.97      4518
           1       0.98      0.96      0.97      2822
           2       0.99      0.99      0.99      7893
           3       0.97      0.95      0.96      3285
           4       0.99      0.97      0.98      3773
           5       0.99      0.93      0.96       367
           6       0.98      0.96      0.97      1881
           7       0.97      0.95      0.96      2363
           8       0.99      0.96      0.97       383
           9       0.98      0.97      0.97      2738
          10       0.98      0.95      0.96      1876
          11       0.99      0.99      0.99      3784
          12       0.98      0.97      0.97      4081
          13       0.97      0.99      0.98      6131
          14       0.98      1.00      0.99     19161
          15       0.98      0.99      0.99      6417
          16       0.98      0.92      0.95      1948
          17       0.98      0.97      0.97      3752
          18       0.99      1.00      0.99     15906
          19       0.99      1.00      0.99      7439
          20       0.98      0.99      0.99      9662
          21       0.99      0.99      0.99      1421
          22       0.99      0.97      0.98      3629
          23       0.98      0.97      0.98      2112
          24       0.98      0.98      0.98      3566
          25       0.99      0.97      0.98      2001

    accuracy                           0.98    122909
   macro avg       0.98      0.97      0.98    122909
weighted avg       0.98      0.98      0.98    122909
```

Fig. 12. Results