Credit Card Transactions Fraud Detection & Handwriting A-Z alphabet Classification

Mohammad Rouintan

Computer Science, Shahid Beheshti University

Abstract

In this report, we explore two distinct topics: Credit Card Transactions Fraud Detection and Handwriting A-Z Alphabet Classification. We analyze the datasets related to these topics and discuss the methodologies employed in each case. For Credit Card Transactions Fraud Detection, we examine a simulated dataset consisting of legitimate and fraudulent credit card transactions spanning from 1st Jan 2019 to 31st Dec 2020. The dataset encompasses transactions made by 1000 customers with a pool of 800 merchants. We delve into the techniques used to detect and identify fraudulent transactions, which are crucial for financial institutions in preventing financial losses and protecting their customers. For Handwriting A-Z Alphabet Classification, we investigate a dataset comprising handwritten images of the English alphabet, divided into 26 folders (A-Z). Each image is centered and resized to a 28x28 pixel box, represented in grayscale. We explore the methods employed for classifying and recognizing handwritten alphabets, a task with applications in optical character recognition, language processing, and digitization of handwritten documents.

Introduction

- Credit Card Transactions Fraud Detection: Credit card fraud has become a significant concern in today's digital age, posing risks to both financial institutions and consumers. Detecting fraudulent transactions is crucial to minimize financial losses, protect customers, and maintain the integrity of the financial system. With the advent of advanced machine learning and data analysis techniques, financial institutions can leverage transactional data to build robust fraud detection systems.
  In this report, we examine a simulated dataset that encompasses legitimate and fraudulent credit card transactions. The dataset covers a two-year duration, starting from 1st Jan 2019 to 31st Dec 2020, and includes transactions made by 1000 customers with a pool of 800 merchants. We delve into the techniques employed to identify fraudulent patterns, such as supervised learning, and behavior analysis.
- Handwriting A-Z Alphabet Classification: Handwritten character recognition plays a vital role in various applications, including digitization efforts, automated document processing, and language understanding. The ability to accurately classify and recognize handwritten alphabets is a challenging task due to variations in writing styles, shapes, and sizes.
  In this report, we explore a dataset consisting of handwritten images representing the English alphabet. The dataset comprises 26 folders, each corresponding to a specific alphabet (A-Z). Each image is centered and resized to a 28x28 pixel box, represented in grayscale. We discuss the methodologies employed for alphabet classification, including traditional machine learning algorithms.

In the other hand we use GridSearchCv for tune our model's hyperparameters .We can also measure the result of the model using different metrics and adjust the model so that it is the best model according to our desired metrics. For example, we can use the following metrics:

- Accuracy Score
- F1 Score
- Recall Score
- Precision Score
- ROC AUC

According to the dataset and the purpose of the project, we can choose the appropriate metric. For example, in this dataset, our goal is to detect credit card transactions fraud, so our overall goal is to detect all frauds, which means it is not a problem even if a small amount of data is wrongly identified as fraud. But we must be careful that fraud does not get out of hand. In fact, we want to have a high Recall. In this project, we measure our models based on the all metrics and find best models.

By studying these two distinct topics, we aim to provide a comprehensive understanding of the methodologies, techniques, and challenges involved in Credit Card Transactions Fraud Detection and Handwriting A-Z Alphabet Classification. The insights gained from this exploration can contribute to the development of more effective fraud detection

systems and improved handwriting recognition algorithms, fostering advancements in financial security and character recognition technology.

Methodologies

Preprocessing Data

- Credit Card Transactions Fraud Detection: In data preprocessing, we first check whether there is a missing value in the dataset or not. Fortunately, our data does not have missing values. We have changed the gender attribute from categorical type to integer type in such a way that we have assigned the number 0 to females and the number 1 to males. In the following, we have removed features such as 'cc_num', 'first', 'last', 'trans_num', 'dob', 'trans_date_trans_time' because according to various tests and model training, the presence of these features was a weakness of the model. For some categorical features, we have used count encoder for example 'merchant', 'category', 'street', 'city', 'state', 'job'. Count Encoder is a popular technique used in data science for encoding categorical variables. It replaces each category with the count of occurrences of that category in the dataset. This encoding method captures the frequency information of each category, allowing machine learning models to utilize this valuable feature. Count Encoder is particularly useful when dealing with high-cardinality categorical variables, where one-hot encoding or label encoding may not be feasible or effective. By incorporating count information, this technique enhances the predictive power of categorical variables in machine learning models. Based on correlation

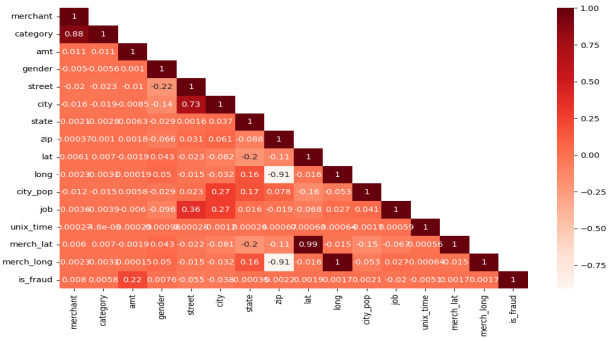

Figure 1: *Correlation Matrix*

matrix we remove 'long', 'lat' features beacause these two features have high correlation with 'merch_long' and 'merch_lat' featurs. We also remove 'unix_time' beacause this feature has very low correlation with target.

After these processes, in order to deal with the imbalance of data, we have used undersampling and oversampling techniques at the same time, so that first we have done with the SMOTE oversampling technique and we have increased the number of class 1 data to 150,000 and then with the RandomUnderSampling technique, we decreased the number of class 0 data to 150,000.

- Handwriting A-Z Alphabet Classification: For this, we have not done any pre-processing, just because we want to use traditional machine learning models, we used PCA technique to reduce the data dimensions and reduced the data dimensions from 784 features to 20 features.

Training Models

In this section, we check the best result of each model and the rest of the results are checked in the notebook. for models 'LogisticRegression', 'LinearSVC', 'KNN', 'GaussianNaiveBayes' we use StandardScaler. and for 'DecisionTree' model with GridSearchCv we find best model with hyperparameter ('max_depth': 14)

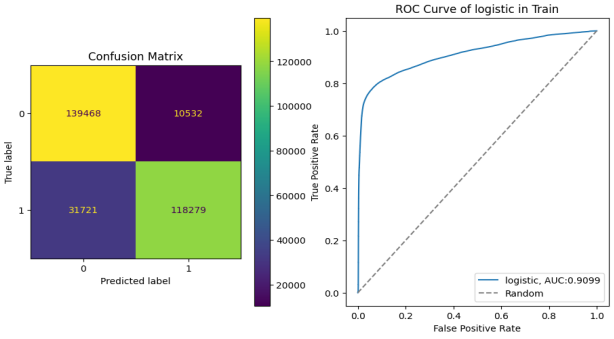

Figure 2: *LogisticRegression F1Score Train:0.85 , AUC Train: 0.91*

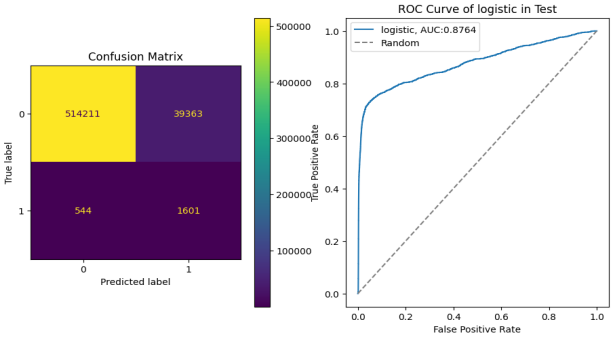

Figure 3: *LogisticRegression F1Score Test:0.07 , AUC Test:0.88*

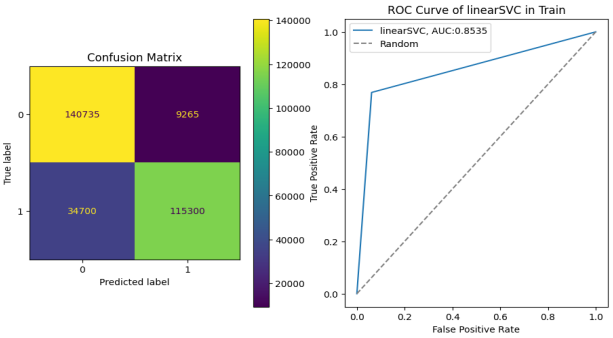

Figure 4: *LinearSVC F1Score Train:0.84 , AUC Train:0.85*

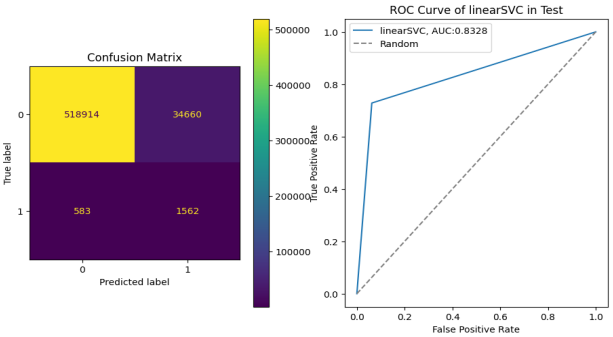

Figure 5: *LinearSVC F1Score Test:0.08 , AUC Test:0.83*

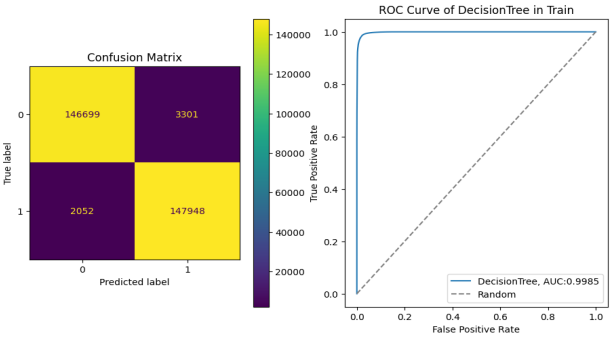

Figure 6: *DecisionTree F1Score Train:0.98 , AUC Train:0.99*

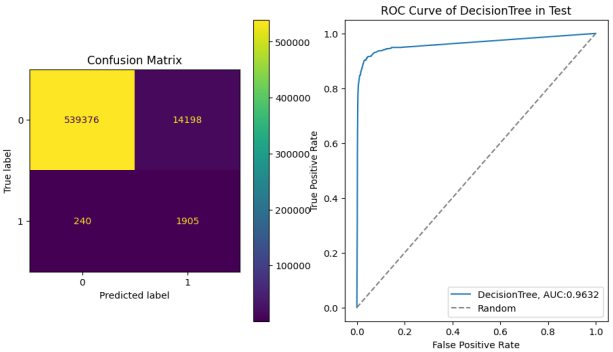

Figure 7: *DecisionTree F1Score Test:0.21 , AUC Test:0.96*

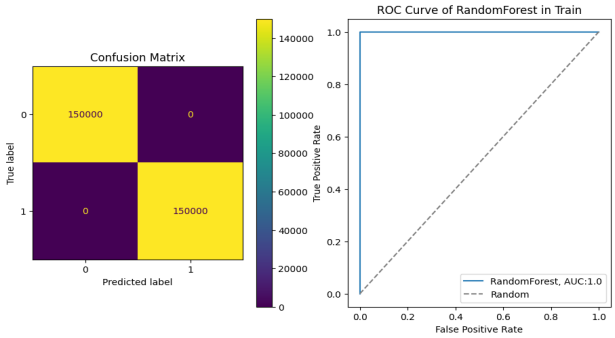

Figure 8: *RandomForest F1Score Train:1.0 , AUC Train:1.0*

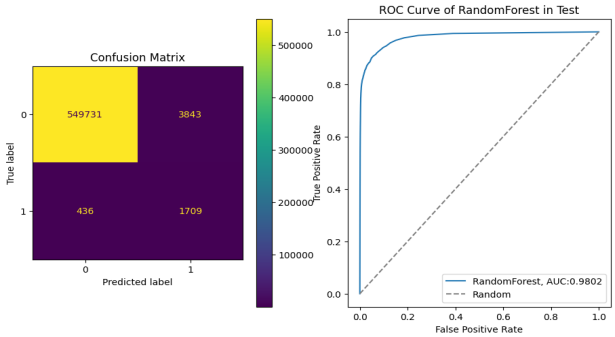

Figure 9: *RandomForest F1Score Test:0.44 , AUC Test:0.98*

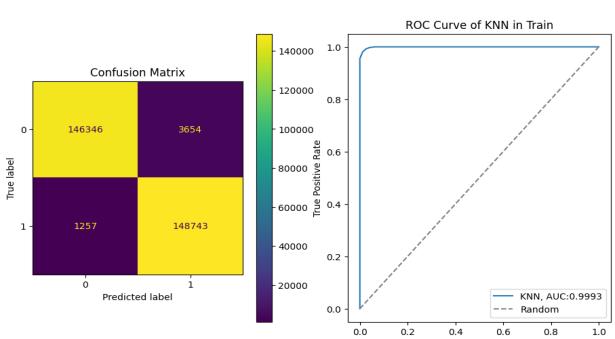

Figure 10: *KNN F1Score Train:0.98 , AUC Train:0.99*

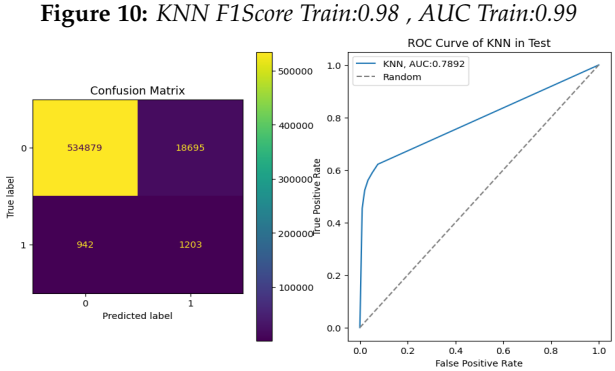

Figure 11: *KNN F1Score Test:0.11 , AUC Test:0.79*

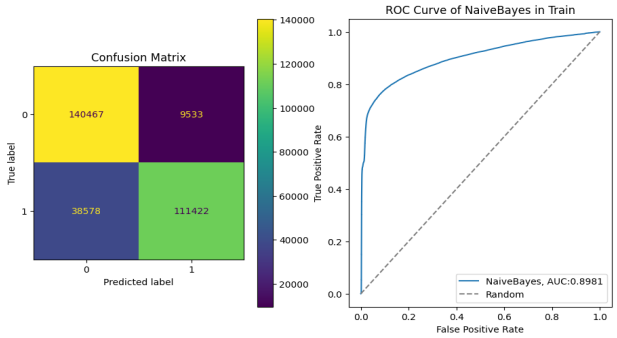

Figure 12: *GaussianNaiveBayes F1Score Train:0.82 , AUC Train:0.9*

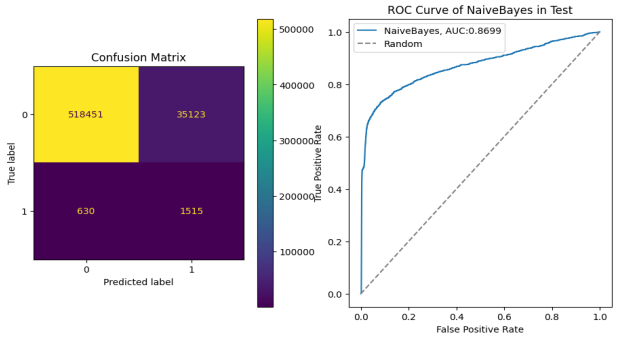

Figure 13: *GaussianNaiveBayes F1Score Test:0.08 , AUC Test:0.87*

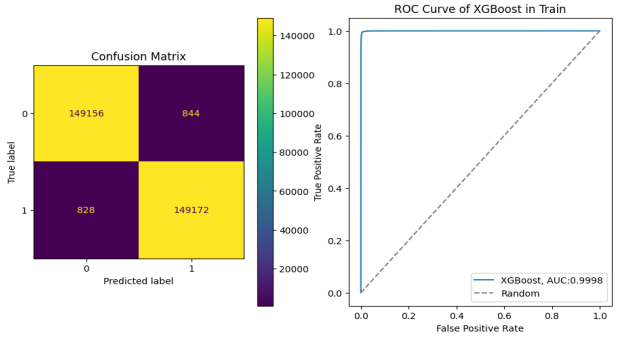

Figure 14: *XGBoost F1Score Train:0.99 , AUC Train:0.99*

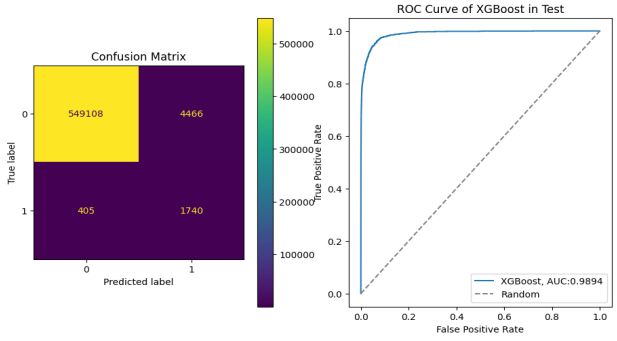

Figure 15: *XGBoost F1Score Test:0.42 , AUC Test:0.99*

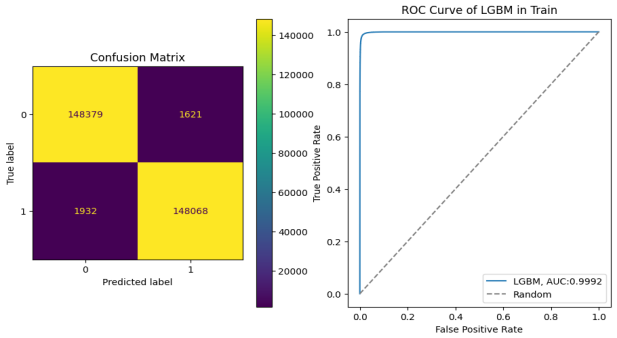

Figure 16: *LGBM F1Score Train:0.99 , AUC Train:0.99*

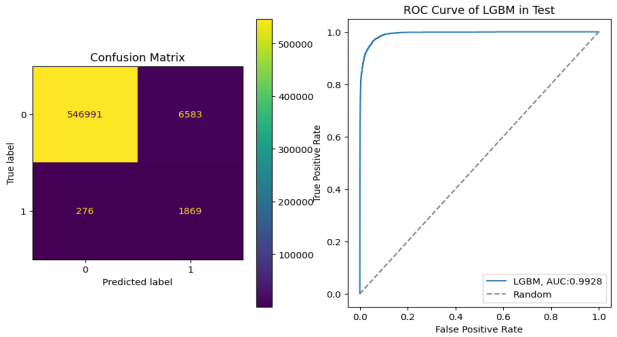

Figure 17: *LGBM F1Score Test:0.35 , AUC Test:0.99*

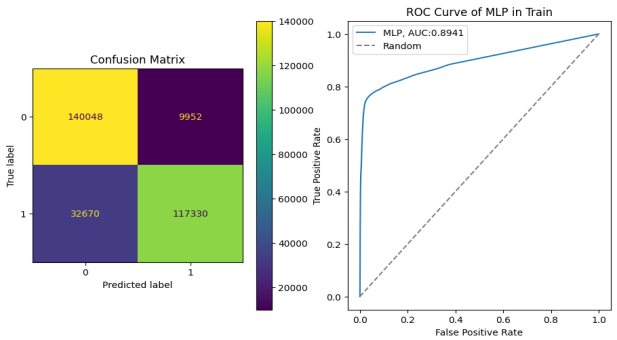

Figure 18: *MLPClassifier Train:0.85 , AUC Train:0.89*

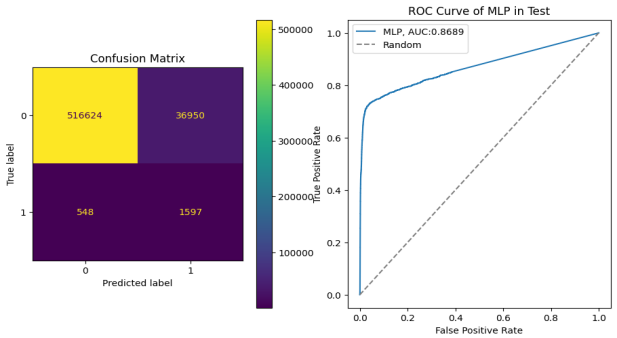

Figure 19: *MLPClassifier Test:0.08 , AUC Test:0.87*

for Handwriting A-Z alphabet (Classification) task we use RandomForest and XGBoost. These two models have same (n_estimators=200) and results of these model is very good. Accuracy of RandomForest and XGBoost models is 98%. for this task we also use deep learning model such as CNNs.

Results

The main purpose of the project is to examine a dataset on the subject of vehicle insurance fraud, which we should be able to detect between fraud and non-fraud using classification.

As I can see in the Training Model section, we have used 11 different classification models. We also said that we can use different metrics to measure the model and express the best results based on the desired metric. Based on the F1 score, the best model is RandomForest, which has an F1 score of 0.44 on the test data. After that, XGBoost is also good. Also if we measure based on Recall, the best model is Decision Tree that has 0.89 Recall in the test data.

In general, in order to get better results, Grid Search Cv should be done on all these models so that the best results can be obtained. Also, in the data preprocessing section, better results can be achieved by changing the procedure, for example, because this data has many categorical features, encoding these features in different ways can be effective.