Credit Card Fraud Detection System

An Exercise in Card-Present Fraud

Capstone Project
Kendall T. Herron

Two conditions must meet for an act to constitute fraud, the perpetrator must be aware that the statement or claim is false or altered, and there is an intent to deceive for economic benefit.

Did you know credit card fraud is one of the most common types of fraud? It remains a significant and evolving concern in the world of financial services. It poses financial losses to both cardholders and financial institutions and erodes trust in the financial system. Fraudsters continue to find new ways to exploit vulnerabilities in payment systems.

Having this concern, I will create a credit card fraud detection system for the Kaggle data set credit that will help financial services identify some fraudulent transactions.

Audience

Any financial service provider would reap the rewards of being able to provide a credit card fraud detection system for their clients.

The financial service providers around the world have a large population of customers that would make fit for the credit card fraud detection system.

Data Source

For the data, I choose a Kaggle data set of previous outcomes of credit card transactions. With over 37 thousand different transactions through out 7 different cities.

To upload this dataset, I utilized a pandas dataframe to read the credit csv file.

About the Data

All this data is online and easy to understand. There wasn't a lot to clean up as the dataset wasn't prone to discrepancies.

Data Wrangling

Below is an overview of the main issues I ran into while cleaning the data:

- Problem1: This dataset has some missing values in one of columns and naming issues.

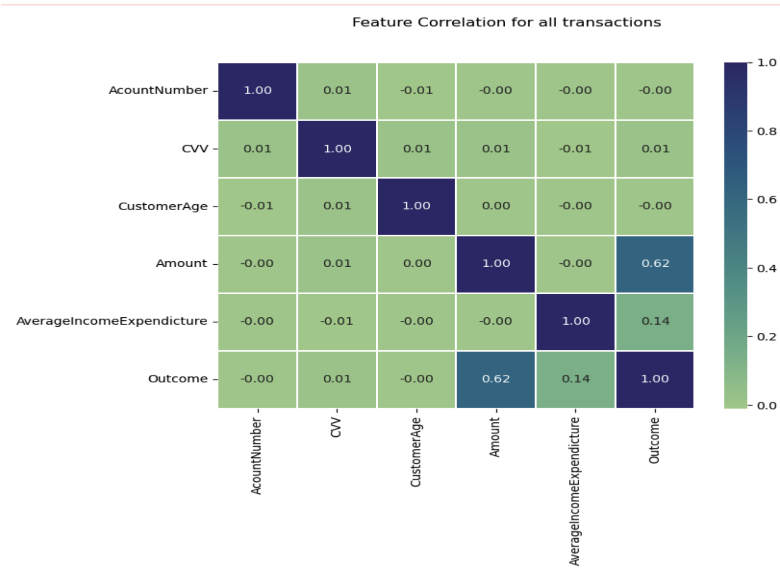
- Solution part 1: normalize the data
    - Sort the values by count to find the number of missing rows and fill with the mean average.
    - Renaming the columns for data cleanness.

Exploratory Data Analysis

In the EDA, I was able to identify that the dataset will be sufficient for the fraud detection system.

Below are a few couple pertinent findings:

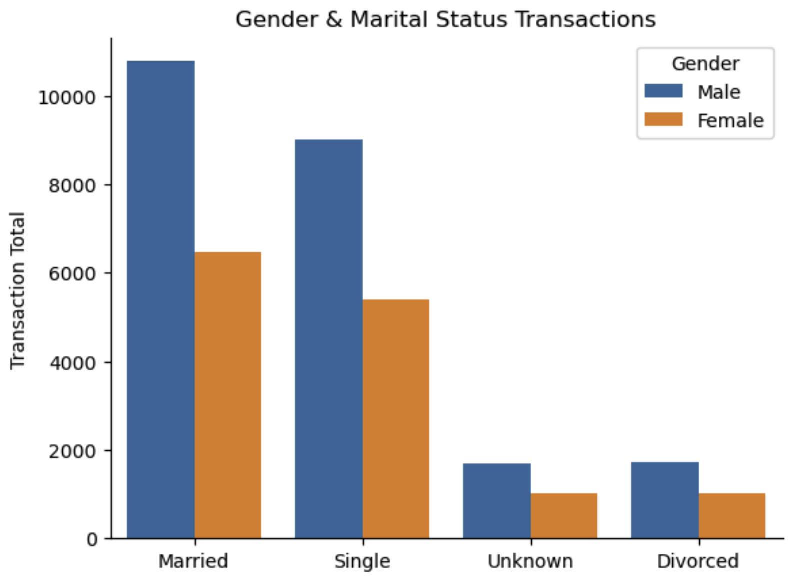
- Imbalanced data
    - For every transaction there is roughly a 74% chance the transaction will be fraudulent.
    - The number of normal transactions was 9727.
    - The number of fraudulent transactions was 27370.
    - There is a 281% chance the transaction is fraudulent.
- Transaction amount correlates with outcome
    - There is a moderately strong correlation between the amount and outcome variables.



Hypothesis Testing

Although it doesn't pertain to the credit card fraud detection system. I wanted to explore an intriguing question about the credit data.

1. *Does gender or marital status give any insights on who spends more?*
    a. I was curious to see if males or females with different marital status spent more money.
    b. Results: According to this dataset, males spend more money and married people spend the most.



Algorithms

I chose to work with the Python library scikit as well as PyCaret for training my credit card fraud detection system.

I tested the credit cleaned dataset with three different classifiers using scikit learn. Logistic Regression, Random Forest, and Gradient Boosting all had a F1 score above 80%.

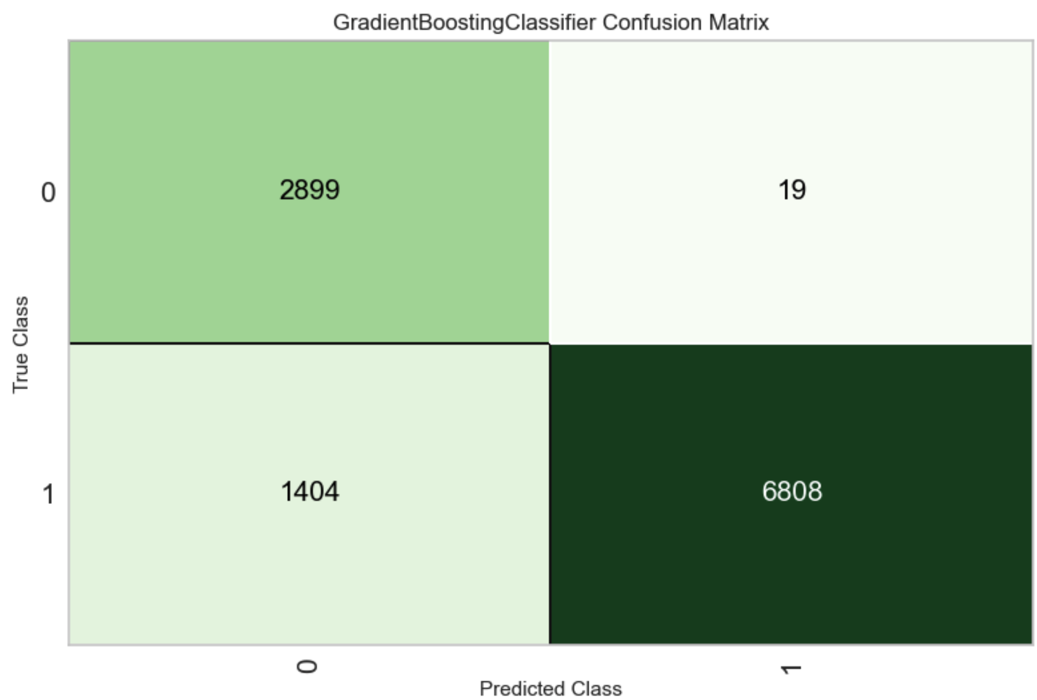
I used PyCaret to train multiple models simultaneously while comparing their model performances. There were 15 different models and Gradient Boosting performed the best. It should be noted that this algorithm, although is the most accurate it's also computationally expensive compared to the other 15 models.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.8629	0.9452	0.8360	0.9745	0.9000	0.6860	0.7067	4.6810
lightgbm	Light Gradient Boosting Machine	0.8592	0.9454	0.8632	0.9411	0.9005	0.6616	0.6689	0.4760
rf	Random Forest Classifier	0.8569	0.9432	0.8792	0.9232	0.9006	0.6454	0.6480	2.1210
catboost	CatBoost Classifier	0.8564	0.9439	0.8750	0.9263	0.8999	0.6466	0.6499	7.4360
et	Extra Trees Classifier	0.8521	0.9363	0.9007	0.8991	0.8998	0.6170	0.6171	1.7660
dt	Decision Tree Classifier	0.8513	0.8079	0.8993	0.8992	0.8993	0.6157	0.6158	0.2380
ada	Ada Boost Classifier	0.8497	0.9369	0.8686	0.9232	0.8950	0.6311	0.6349	1.1250
qda	Quadratic Discriminant Analysis	0.8345	0.9149	0.8377	0.9315	0.8819	0.6077	0.6179	0.1060
nb	Naive Bayes	0.8266	0.9185	0.8029	0.9549	0.8723	0.6085	0.6311	0.0820
lr	Logistic Regression	0.8210	0.9165	0.8096	0.9393	0.8697	0.5888	0.6057	1.0330
ridge	Ridge Classifier	0.8192	0.0000	0.7850	0.9631	0.8650	0.5999	0.6293	0.0810
lda	Linear Discriminant Analysis	0.8192	0.9222	0.7850	0.9631	0.8650	0.5999	0.6293	0.1910
knn	K Neighbors Classifier	0.8125	0.8862	0.8013	0.9353	0.8631	0.5711	0.5886	0.4950
svm	SVM - Linear Kernel	0.4049	0.0000	0.3000	0.2213	0.2547	0.0000	0.0000	0.3230
dummy	Dummy Classifier	0.2622	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0950

I adopted the F1 metric over just accuracy because I wanted to get the actual positive cases that are correctly identified. However, I chose gradient boosting as it provided high recall at the same time with great accuracy, precision, and F1 score.

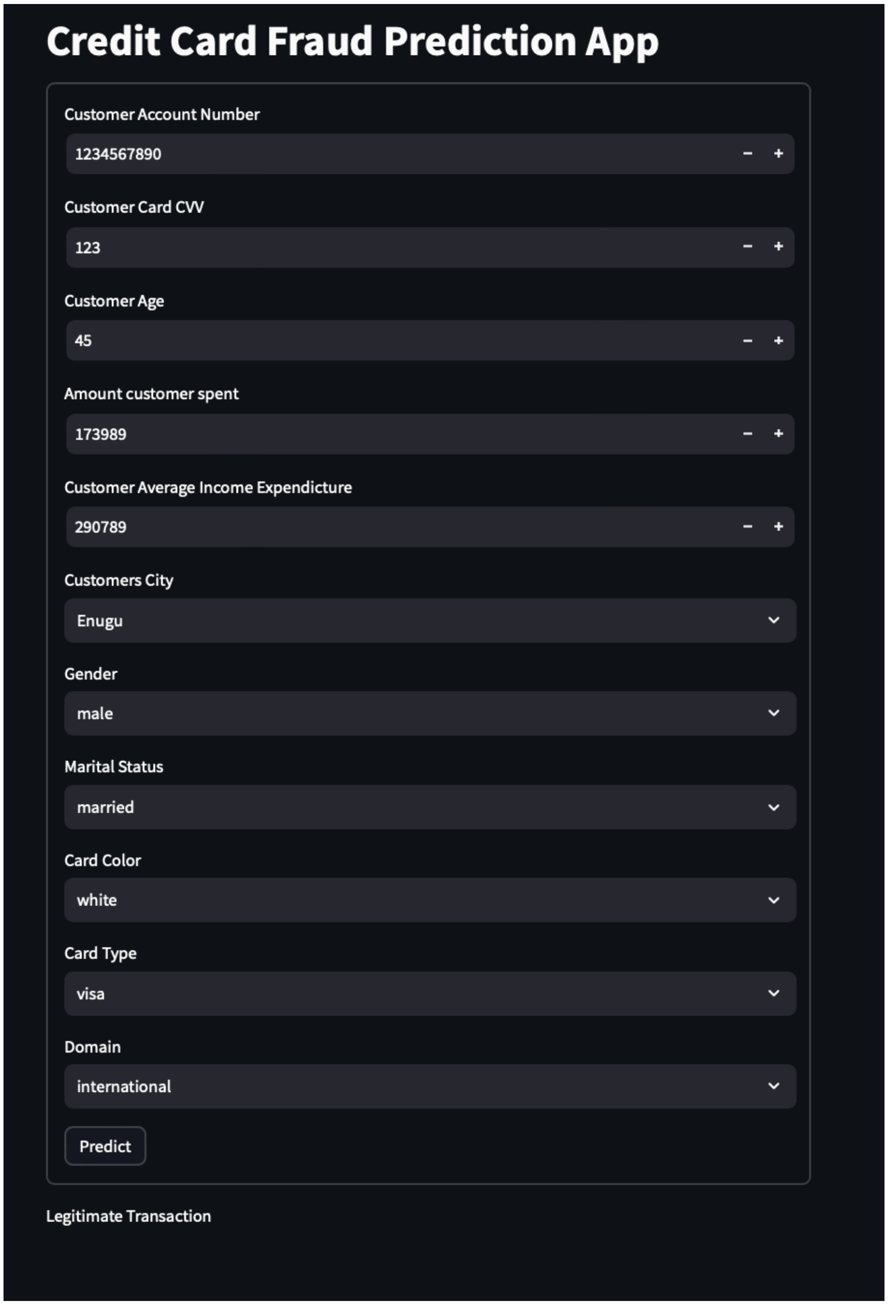
$$F_1 = \left(\frac{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}}{2}\right)^{-1} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

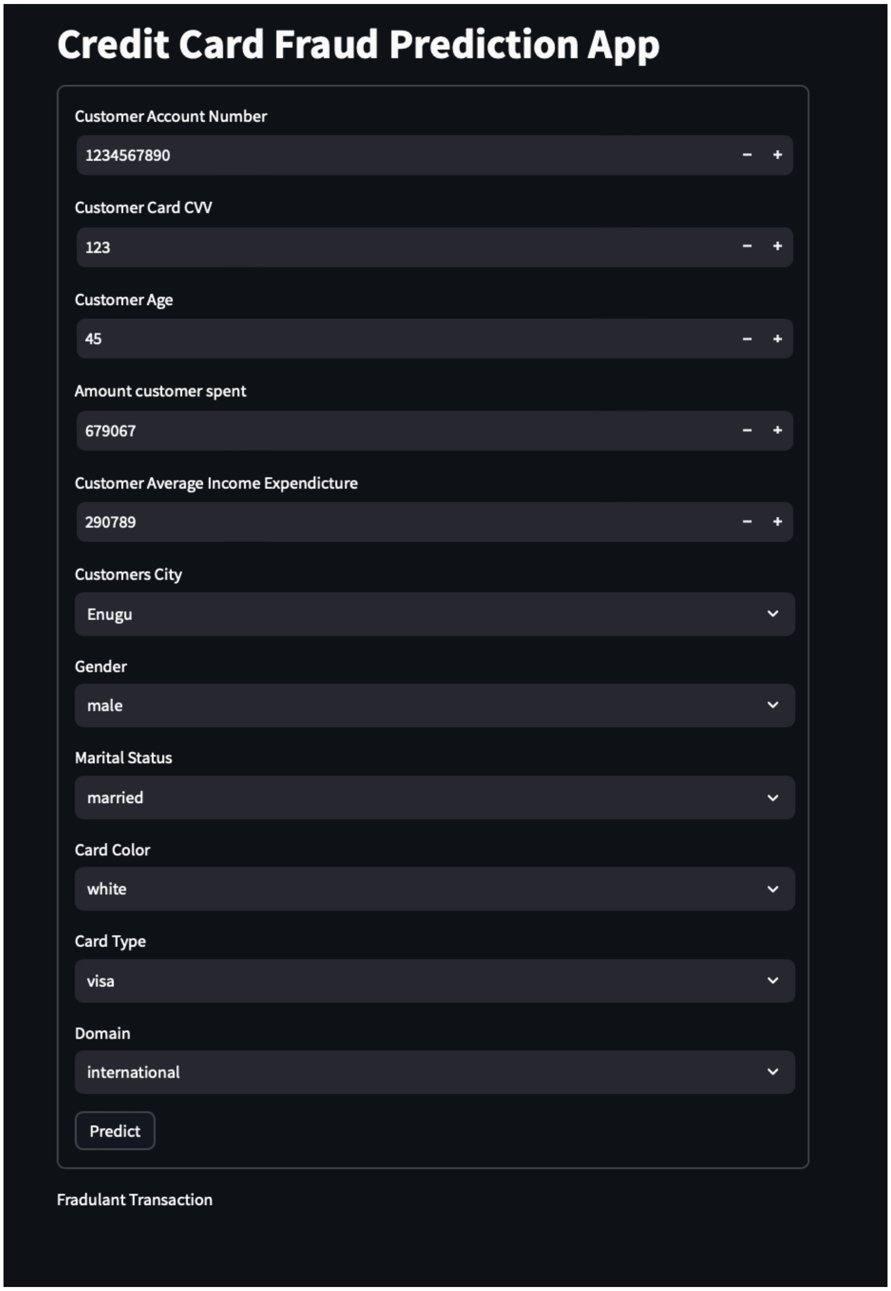
You can see in the below diagram there was only 19 False Negatives. The cost of False Negatives is much higher than the cost of False Positives.



Predictions

In the final prediction application, the user can enter all the account information attached to the transaction. The user can then select the predict button to see this transaction is either fraudulent or legitimate.


Credit Card Fraud Prediction App
Customer Account Number
1234567890
Customer Card CVV
123
Customer Age
45
Amount customer spent
173989
Customer Average Income Expendicture
290789
Customers City
Enugu
Gender
male
Marital Status
married
Card Color
white
Card Type
visa
Domain
international
Predict
Legitimate Transaction



Future Improvements

In the future, I would love to spend more time creating a batch system, wherein a user could import multiple transactions at a time to predict if the transactions are legitimate or fraudulent. This credit card detection system would also be improved by connecting to a reporting automation, where a user could receive a report of transactions as fraudulent or legitimate to alert the card holders to prevent any loss money. This credit card detection system would also be improved by connecting to a reporting automation, where a user could receive a report of transactions as fraudulent or legitimate to alert the card holders to prevent any loss money.