Sean Williams

Applied Data Science Project

Final Report

DSCI503

Introduction

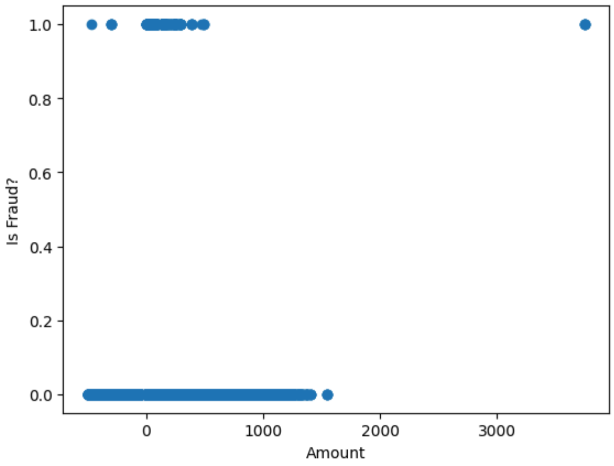
Automating the detection of credit card fraud can be difficult due to the number of possible factors and combinations of factors that may indicate a fraudulent transaction. That said, the binary classification nature of this problem allowed for a straightforward approach when building my model, and much of the process was spent working with the data itself.

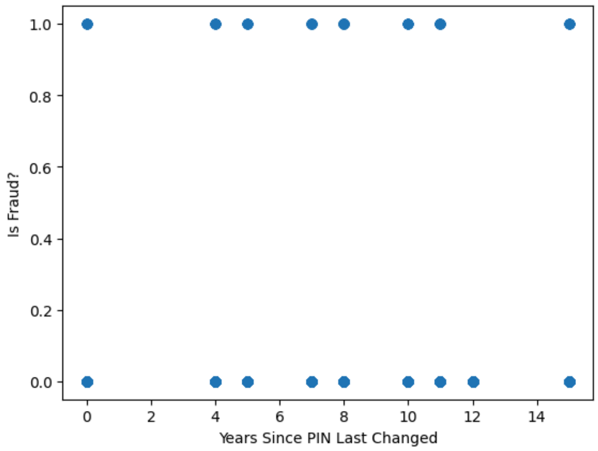
Data Processing

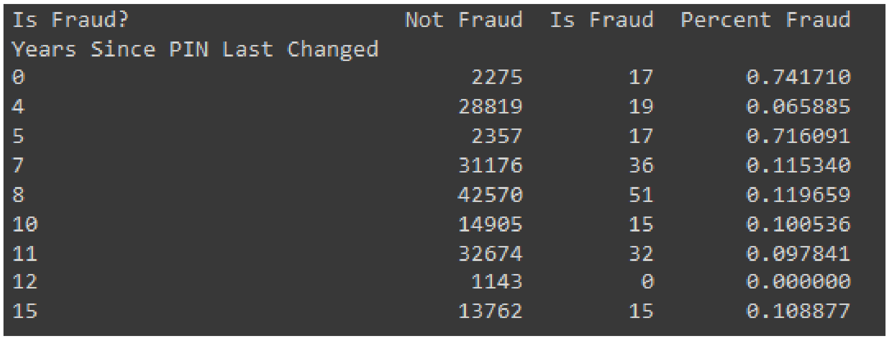
The dataset I utilized was synthetic, as legitimate sets of transactions are difficult to obtain online in any usable form due to their sensitivity. Still, as will be seen, it served as an adequate means for training the model. The set was divided into three different CSV files: one with card information, one with cardholder information, and another with transaction information. These sets needed to first be combined into a single dataframe for the model to make use of all available features. Because each cardholder could have more than one card and the transaction set had two columns for both the cardholder and card ID, I had to create a special ID column that could be shared across all three sets; this allowed me to join them into a single dataframe. I chose to drop certain columns from the data set, such as the card number and CVV, as I felt that the model should not be exposed to such sensitive and likely irrelevant information. In addition to all of that, I also converted numerical columns to floats or integers as needed, standardized them, and utilized one hot encoding on the categorical columns.

Exploratory Data Analysis

While attempting to visualize and extract trends from this dataset was not particularly helpful, it did help guide my thinking for the rest of the model. As can be seen in the following images, I first looked for a correlation between fraud and the number of years since the card's PIN had been changed, where 0 is an authentic transaction and 1 is a fraudulent one. Of more use was the proportion of transactions within a timeframe that were fraudulent, also pictured:

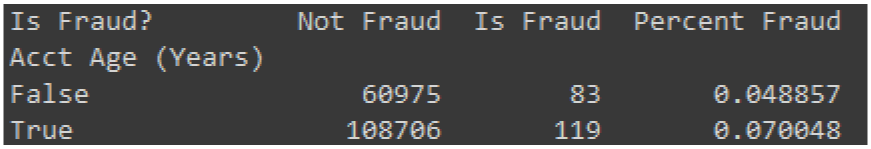




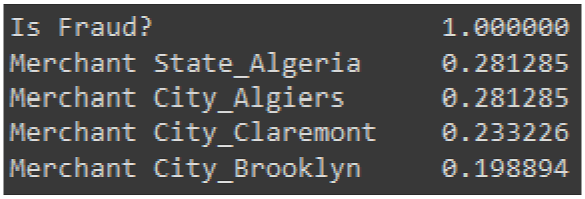


```
Is Fraud?                     Not Fraud  Is Fraud  Percent Fraud
Years Since PIN Last Changed
0                                  2275        17       0.741710
4                                 28819        19       0.065885
5                                  2357        17       0.716091
7                                 31176        36       0.115340
8                                 42570        51       0.119659
10                                14905        15       0.100536
11                                32674        32       0.097841
12                                 1143         0       0.000000
15                                13762        15       0.108877
```

These statistics suggest that transactions involving cards whose PINS were either recently changed or changed 5 years prior are more likely to be fraud. Additionally, I tested the card's age (less than or greater than 15 years old) and the transaction's location, which gave the following results:



```
Is Fraud?         Not Fraud  Is Fraud  Percent Fraud
Acct Age (Years)
False                 60975        83       0.048857
True                 108706       119       0.070048
```

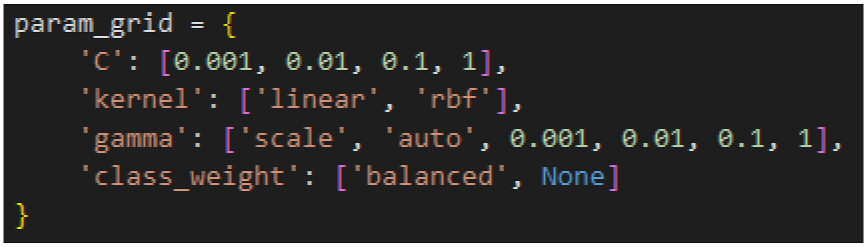


```
Is Fraud?                   1.000000
Merchant State_Algeria      0.281285
Merchant City_Algiers       0.281285
Merchant City_Claremont     0.233226
Merchant City_Brooklyn      0.198894
```

It is clear that these features are correlated to fraud, and that Algeria in particular appears to be a likely location for fraud to take place.

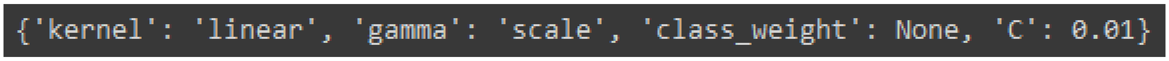
Model Development and Results

As this is a binary classification problem, I first tried training a simple logistic regression model. The default parameters for the applicable Python scikit-learn model immediately resulted in an accuracy score of 99.909%, which could not be improved through cross-validating different sets of parameters and in fact consistently returned worse results. It is for this reason that I disregarded the initial result, as I could not replicate it and doubted its seeming simplicity. I next tested a support vector machine—after running a grid search cross validation, I quickly realized that such a technique would take far too much time, so I instead opted for a randomized search using the below parameters:



```
param_grid = {
    'C': [0.001, 0.01, 0.1, 1],
    'kernel': ['linear', 'rbf'],
    'gamma': ['scale', 'auto', 0.001, 0.01, 0.1, 1],
    'class_weight': ['balanced', None]
}
```

The process took several hours to finish running, however I was eventually met with this combination of parameters:



```
{'kernel': 'linear', 'gamma': 'scale', 'class_weight': None, 'C': 0.01}
```

The final model had an accuracy of 99.8996%, however I am doubtful that this is a valid metric. According to the model, factors such as whether or not the card was on the dark web or the years since its PIN had been changed had little impact on whether or not a transaction was deemed fraudulent, whereas its location was a major indicator. As another student pointed out to me, the fact that my dataset had so few fraudulent transactions relative to the total number of entries would likely mean that any model trained on it would have a high accuracy score, as it could easily label every transaction as legitimate and only have a handful of false negatives. Unfortunately, this was the only set I had to work with, and thus despite my efforts I am unsure of whether or not I was successful.

Conclusion and Ethical Considerations

In any case, the processing steps I used would still be applicable to future datasets, it is simply a matter of gathering more instances of fraud. This will, of course, mean sourcing real information from credit card companies, which presents many ethical issues as transactions often contain private information. It should be the user’s right whether to opt in or out of any sort of transaction reporting, and the procedures to do so should be accessible. In addition, companies must be transparent with what data they are collecting, how they are using it and for what purpose. A solution I would propose is, when a transaction is found to be fraudulent, the user is asked if they would be willing to submit its information for the sake of catching more fraud in the future. This addresses both consent and transparency, hopefully allowing the model to become a legitimate tool for good. While I am disappointed that I did not achieve a satisfying product, I at least enjoyed the cleaning and training process and am much more knowledgeable about machine learning for any future projects.