Building a Fraud Detection Model

Mitchell Seiter

Abstract

The goal of this project was to use classification models to predict whether a certain credit card transaction was fraudulent based on user data. Credit card fraud is a major issue for all major financial institutions and can be very costly to the company, whether in the form of missing fraud or detecting fraud when it is NOT actually occurring. Using sklearn and

Design

The purpose of this project is to help financial institutions and their customers to have a better experience when dealing with credit card transactions. This will prevent banks from losing money and help customers have a better sense of mind, knowing their money is safe. The data comes from a simulated data set that mimics real credit card transactions.

Data

The dataset contains 555,719 credit card transactions with 12 features for each. This was an imbalanced data set with only 2,145 rows containing the positive class. Some feature engineering involved creating a few extra rows that included time since last transaction, total amount of fraud per person, and distance from merchant. These features along with amount and category of purchase were the main ones looked at.

Algorithms

Feature Engineering

1. Creating new columns that included:
    a. purchase distance which used coordinates of person and location of merchant
    b. Time since last purchase which took the difference of the unix time given
    c. Running total of how much fraud a person had associated with them

2. Converted purchase category to dummy variables

I built multiple classification algorithms that included KNN, logistic regression, decision trees, random forest, and gradient boost.

After a base model for all of the above and checking the precision/recall curves it was apparent that the random forest model would be the best fit.

One hyperparameter that was tuned was the n estimator which gave a better fit for the model. After building a confusion matrix and running a cross validation I determined that the random forest model was maximizing the amount of true positive cases while keeping false negatives/positives at a minimum.

Tools

- Numpy and Pandas for data manipulation
- Sklearn for modeling
- Matplotlib and Seaborn for plotting

Communication

This information will be presented in slides format and will be uploaded to my github repository.