Project 1: Detecting Credit Card Fraud

Abstract

The enclosed data analysis project aims to build a machine learning model that has the capability to predict, with great accuracy, when a credit card transaction presents as being fraudulent. Beginning with exploratory data analysis, I will use machine learning algorithms to understand and work through these data. By exploring the data set and understanding trends, I will build and train multiple models to predict which transactions are fraudulent, ultimately aiming to test and select the most accurate model for this set of data. The data being used comes from the organization Worldline and the Machine Learning Group [http://mlg.ulb.ac.be/]. This is a Card Transactions dataset that consists of a combination of fraudulent and non-fraudulent transactions. There are nearly 285,000 rows of data and 31 features. 28 of these features are nondescript for the public for security reasons, but the three I am interested in are Time, Amount, and Class (fraudulent or non-fraudulent). This data structure presents us with a classification project opportunity. Classification is a technique which works with data that is categorized into a certain number of classes. The goal of such a project is to identify which category or class a new data piece will fall into. These data are important to analyze as our world becomes increasingly automated and electronic. With credit card fraud on the rise, we must continue to be diligent about finding ways to cut through the falsehoods. The produced model(s) could be immensely useful in today's increasingly electronic world, where millions of transactions occur daily, many of them fraudulent.

Detecting Credit Card Fraud

Initial Exploratory Analysis

The data set I have used contains a collection of fraudulent and non-fraudulent credit card transaction information. There are 284,807 transactions in the set. Many of our columns are kept anonymized for security reasons, so we are required to work with what we have. The only two columns that we know are transaction and amount. However, we do know that the remaining, unknown, columns, have already been scaled. Through initial analysis we find that the mean of the data set is 88.35. We find no Null values, which is nice because this means that we do not need to work through manners of accounting for those values. Interestingly, we also find that the vast majority of the transactions are non-fraudulent, with fraudulent transactions only occurring 0.17% of the time. Our maximum transaction is for an amount of $25,691.16 but combining that with our mean of 88.35 it is easy to suspect a strong skew to the right with the majority of transactions having a much smaller value than our maximum transaction (Figure 1).

Figure 1: Heavy right skew

Detecting Credit Card Fraud

To begin to familiarize myself with this data set, I first initiated some exploratory data analysis techniques. Exploratory data analysis is an important first step of investigating your data so that you can begin to understand it as well as identify any emerging patterns to better help navigate further analysis of the data. This process is valuable because it allows us to refine our data prior to spending too much time in the analysis stage.

Our time variable is recorded in the number of seconds passed since the very first transaction in the data set. Because we have approximately 175,000 seconds worth of data, we can conclude that our data set was recorded over about 48 hours or two days' time. By visualizing this time period, we are able to quickly see something interesting. We see that just about halfway through the data collection, the volume of transactions decreases significantly, followed by a quick increase shortly thereafter. Without knowing the exact time of day, we can realistically make the assumption that this drop in activity occurs during nighttime hours.

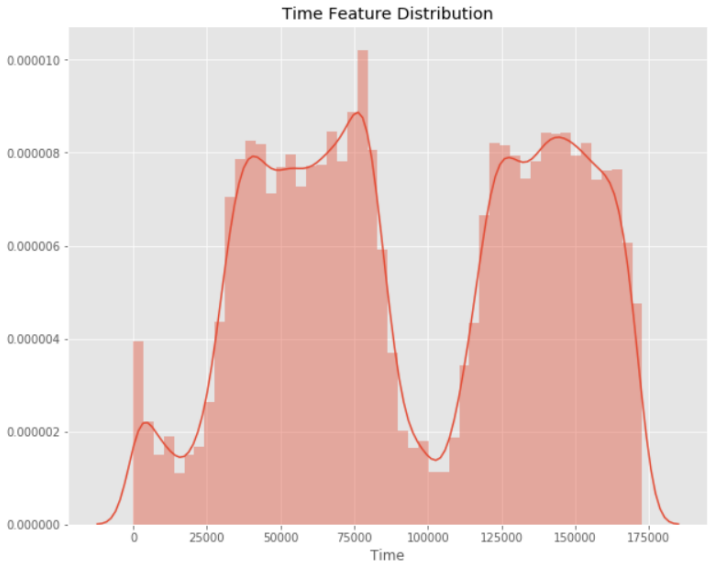

Figure 2: Distribution of time

Detecting Credit Card Fraud

Data Preparation

We know through initial discovery that our anonymized columns have already been scaled for us. Before moving on into full analysis of our data set, we will need to scale our Time and Amount features as well. Without scaling these features, we risk poor performance of our machine learning algorithms. To standardize these features, we can use `StandardScaler` from `sklearn` *(see Appendix A for details on specific methods used throughout).* Using this tool transforms the data such that its distribution will have a mean value of 0 and a standard deviation of 1. This allows our features to have a common scale for us to be able to build machine learning algorithms.

Recall that our original data set is greatly imbalanced, with the vast majority of transactions being non-fraudulent. If we were to continue with this data set as-is without adjusting for this imbalance, we run the risk of overfitting with our algorithms since the assumption will likely be made that most transactions are not fraudulent. We are looking for a model that will *detect patterns* rather than *assume outcomes*, so we want to account for this imbalance before continuing on with our analysis. We need to create a training data set to counteract the great imbalance of transaction type presented in the original data set. This will allow us to build a model that can detect fraudulent transactions and label them correctly in this way rather than making the incorrect assumption that "most" presented transactions will be fraudulent. To do this, we can use the random under-sampling method, which essentially removes data to create a training data set which has a more balanced distribution of transaction type (fraudulent or non-fraudulent). This will force our algorithms to correctly detect when a fraudulent transaction exists.

To create the balanced training set to be used, first we can count the number of fraudulent transactions that exist in the original data set. By randomly selecting the same number of non-fraudulent transactions from the data set, we will end up with a 50/50 mixture of transaction class. We

want to then concatenate the two "lists" and shuffle to randomize the class type. Since we have 435 fraudulent transactions in our list, and thus have selected the same number of non-fraudulent transactions, our new data set contains 870 total transactions. By visualizing the class distribution in the original dataset compared to the newly created subsample, we can easily see how much more balanced it has become.

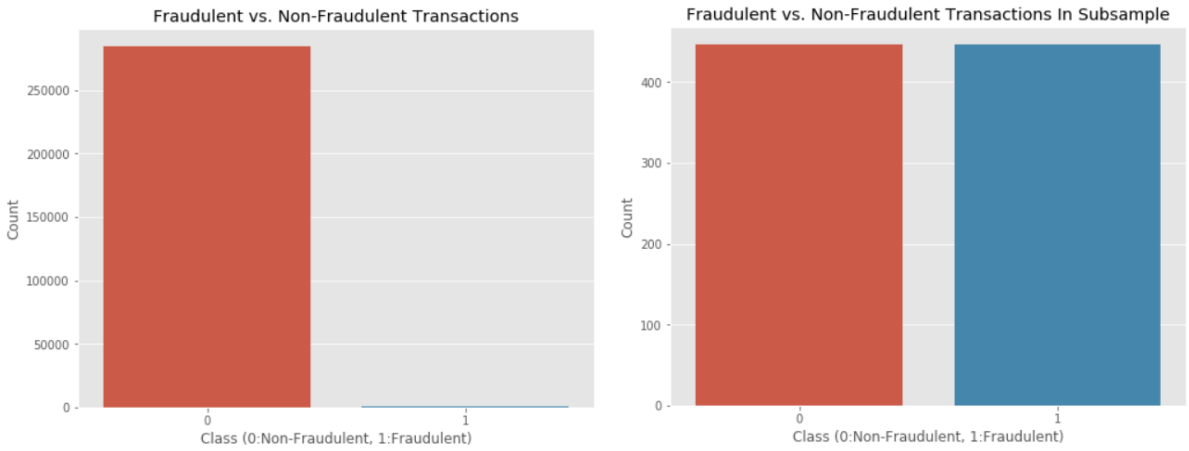

Figure 3: Class Distribution in original vs. subsample

When we take a look at performance, rather than using metrics of accuracy, instead we are going to use the measure of Receiver Operating Characteristics-Area Under the Curve (ROC-AUC) performance. The ROC-AUC is a very useful metric for checking performance of classification models, which is what we are going to be building. "The ROC is a probability curve while the AUC represents the degree or measure of separability." (Narkhede, 2019). Ultimately this method of measurement gives us a value between 0 and 1 with 1 being a perfect score and 0 being the opposite. The higher the score, the better prediction ability of the model. We can even visualize this performance by looking at the AUC curve. A confusion matrix is not a meaningful check for accuracy for this type of data set with unbalanced classification.

Detecting Credit Card Fraud

One important step to analyzing data is to identify and work with any outliers that may exist. The way you deal with outliers can have a significant effect on your data and, ultimately, the power of your model's predictions. For outlier removal with this data set I chose to look at features with a correlation of 0.5 or more with the class variable (fraudulent or non-fraudulent). First we can visualize the positive and negative correlations with different features before we go ahead and actually remove any outliers. We calculate our Interquartile Range (IQR) and use box plots to display our data. Box plots are a good tool for this because we can easily see the 25^{th} and 75^{th} percentiles as well as identifying any extreme outliers. We have to be somewhat careful with how we define our thresholds for removing outliers. A lower threshold will remove more outliers, however, perhaps we want to limit those removals only to *extreme* outliers in order to retain most of our data thus limiting the risk to our model accuracy in both directions. A data point is typically considered to be an outlier if it lies outside of 1.5 x IQR, but with our data set using this threshold would significantly reduce our training data size. For this reason, we will elect to only remove those points sitting outside of 2.5 x IQR. By using these specifications, we reduce our subsample from 870 transactions to 612 transactions.

Since we are not able to visualize our classes multi-dimensionally at this point, we can use a dimensionality reduction method to project higher dimensional distributions into lower dimension visualizations. The t-distributed stochastic neighbor embedding (t-SNE) algorithm offers a way to get a feel for how our data is arranged in a high-dimensional space. This algorithm calculates a similarity measure between pairs of instances in the high-dimensional and low-dimensional spaces, and then attempts to optimize the two measures by quantifying the error between predicted and expected values. T-SNE can do a good job of clustering the transactions that were fraudulent and non-fraudulent in our data set. By using t-SNE and showing our data on a two-dimensional space we can see a scatterplot of the resulting clusters.

Detecting Credit Card Fraud

Training Algorithms

Now that we have prepared and cleaned our data set so that it is workable, we need to proceed and train and test our classification algorithms. Before doing so we first perform an 80/20 train-test split on the data which splits it into two parts. I used the k-fold cross-validation method for resampling in order to avoid overfitting for our somewhat limited sample. This method involves shuffling the data set, splitting it into *k* groups, working with each individual group and then summarizing the model ability using the sample of model evaluation scores.

Next, we can look at the different classification algorithms that are available to us, and how they might perform for our current data. By comparing several algorithms and visualizing their performance at a glance, we can make an educated decision about which one to use for our model. I compared a handful of commonly used classification algorithms side-by-side: Logistic Regression, Linear Discriminant Analysis, K Nearest Neighbors (KNN), Classification Trees, Support Vector, and Random Forest. Figure 4 shows the results of this snapshot.

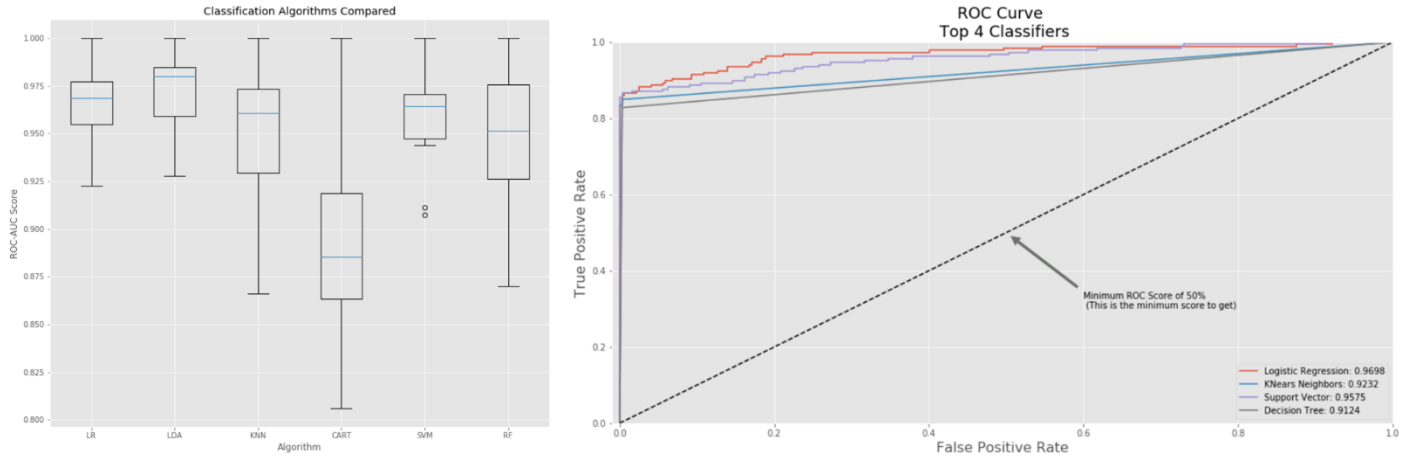

Figure 4: Classification algorithms compared

We can see that several of the compared algorithms show similar results, but a few outperform the rest. Namely, the Logistic Regression algorithm looks to be more accurate than the alternative

classifiers for our data. After training our model we then make predictions using our trained model. The results using Logistic Regression are more than acceptable. The 0 class (transactions without fraud) is predicted with 94% precision and 99% recall whereas the 1 class (transactions which are fraudulent) has 97% precision. This means that only 3% of the transactions which are fraudulent remain undetected by the system. This can be further improved by providing more training data.

By exploring, cleaning and preparing our data, we were able to use logistic regression to produce an accurate model for predicting fraudulent credit card transactions.

Appendix

Main libraries used:

- Matplotlib: Visualization with Python
- Scipy: Python-based ecosystem of probability distributions and statistical functions.
- Numpy: Core library for computing with Python.
- Pandas: Open-source data analysis and manipulation tool.
- Seaborn: Python data visualization library based on Matplotlib.

References

1. Albon, C. (2018). Machine learning with Python cookbook: practical solutions from preprocessing to deep learning. Sebastopol, CA: OReilly Media.
    a. Alternatively, this textbook presents algorithmic method for data analysis and deep learning using the Python language.
2. Boyle, T. (2019, February 4). Methods for Dealing with Imbalanced Data. Retrieved from https://towardsdatascience.com/methods-for-dealing-with-imbalanced-data-5b761be45a18
    a. A helpful article on types of data imbalance and how to deal with such imbalances.
3. Couronne, R., Probst, P., & Boulesteix, A.-L. (2018, July 17). Random forest versus logistic regression: a large-scale benchmark experiment. Retrieved from https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2264-5
    a. This research article discusses the differences, benefits and disadvantages to using random forest versus logistic regression algorithms for regression and classification.

Detecting Credit Card Fraud

4. DataFlair Team. (2019, October 11). 11 Top Machine Learning Algorithms used by Data Scientists. Retrieved from https://data-flair.training/blogs/machine-learning-algorithms/
    a. This article lays out some of the most commonly used machine learning algorithms for data analysis and model development. The article begins by discussing supervised learning algorithms, and moves into unsupervised algorithms, to cover a breadth of available options.
5. DataFlair Team. (2020, February 19). Project in R - Uber Data Analysis Project. Retrieved from https://data-flair.training/blogs/r-data-science-project-uber-data-analysis/
    a. This is an unrelated project which analyzes Uber pickup data for New York City. The assessment of the data and general work through are helpful as they provide a sort of vague wireframe for how to address a data set with the intent to create predictive models.
6. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statistical learning: with applications in R. New York: Springer.
    a. This textbook helps to review many statistical methods for data analysis using R, including logistic regression and random forests.
7. Knaflic, C. N. (2015). Storytelling with data: A data visualization guide for business professionals. Hoboken, NJ: Wiley.
    a. A textbook which offers a guide to data visualization, from the exploratory step through the project presentation step.
8. Machine Learning Group. (2018, March 23). Credit Card Fraud Detection. Retrieved from https://www.kaggle.com/mlg-ulb/creditcardfraud/home
    a. This is my data source and general description of our data set, housed within Kaggle.

Detecting Credit Card Fraud

9. Machine Learning Group. (n.d.). DEFEATFRAUD: Assessment and validation of deep feature engineering and learning solutions for fraud detection. Retrieved from https://mlg.ulb.ac.be/wordpress/portfolio_page/defeatfraud-assessment-and-validation-of-deep-feature-engineering-and-learning-solutions-for-fraud-detection/
    a. This project description outlines the overall goals of the Machine Learning Group as they attempt to develop new and improve upon existing mechanisms for detecting credit card fraud transactions using machine learning algorithms.
10. Nadim, A. H., Sayem, I. M., Mutsuddy, A., & Chowdhury, M. S. (2020, February 13). Retrieved from https://ieeexplore.ieee.org/document/8995753
    a. A secondary article discussed the use of machine learning algorithms to address credit card fraud, investigating the use of various regression algorithms in the process.
11. Narkhede, S. (2019, May 26). Understanding AUC - ROC Curve. Retrieved from https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
    a. This article from Towards Data Science explains what the Area Under the Curve and Receiver Operating Characteristics curve are in terms of machine learning.
12. Puh, M., & Brkić, L. (2019, July 11). Detecting Credit Card Fraud Using Selected Machine Learning Algorithms. Retrieved from https://ieeexplore.ieee.org/document/8757212
    a. This article discusses the growth in interest for applying machine learning techniques to the mission of detection fraudulent credit card transactions and points out the challenges that can arise with these attempts.
13. Shift Credit Card Processing. (2020, February). Credit Card Fraud Statistics. Retrieved from https://shiftprocessing.com/credit-card-fraud-statistics/
    a. This page offers general credit card fraud data and information about the topic of credit card fraud.

Detecting Credit Card Fraud