Analyzing Fraud in Synthetic Banking Data: Initial Code and Results

Julien Michieli

March 2020

1 Introduction

Using the data set that I chose, my focused on finding cases of fraud within simulated bank data. Thus far I have done work in exploring the data and am now applying a series of ML algorithms that will later be compared against one another using different methods. The code referenced in this document can be found in the projects Github repository where it is further explained.

2 Data Exploration and Cleaning

This is an overview of all of the data cleaning and initial inquiries that I made into the data, as well as the cleaning. This involved looking at the different rates of fraud among different categories in the data set. Such examples include dividing the data among age, gender, the type of service being provided and the size of the transaction. Some examples of the types of discrepancies among these divisions are reflected in the plots shown in the the literature review.

Cleaning the data was not a difficult task as there were almost no irregularities and the data did not contain any missing values. Only one area of the data I decided to omit was when the age was when the gender was unknown. I took this action because since the attribute gender included a term for enterprises and I wanted to always make sure that I knew if the customer was a person or an enterprise. These cases represented a very small percentage of the overall data and when investigated tended to have lower rates of fraud than the other categories that were left in the data. The other case of unknown data was inn age but upon inspection I found out that all cases when the age was unknown where for the enterprises in the data.

It was now that I would decide which variables were important enough to pass on to the ML algorithms. These decisions were made using different queries in the sqldf package in R as well as some other visual packages like ggplot2 and others. The variables that I decided to keep were step, age, gender, category and amount. The variables to be omitted were customer and merchant IDs and zipcodes. This was because through different queries and correlation tests performed on the data that these variables seemed to have very little correlation with anything else present. Although if I would like to look into individual merchant cases if possible as there were far fewer merchants than customers in the data. Due to the nature of of fraud among financial data it is only a small percentage of transactions are actually fraudulent. In my data set it is less than one percent that are actually fraudulent. Machine learning algorithms tend to do better with more balanced data sets. Two method of combating this are under-sampling and over-sampling, both of which involve balancing the data set in some way. Over-sampling balances the data by replicating cases from the smaller class until the classes are balanced, while under-sampling eliminates cases from the larger class until the classes are balanced. In the data that I am using, since there are so many cases overall some combination of eliminating non fraudulent cases and replicating fraudulent cases may be optimal for training, instead of just one method or the other. I was able to do this in R and will be testing the Oversampled, undersampled and unbalanced original data using the different ML algorithms.

3 Machine Learning and Analysis

Part of what I am doing is applying different machine learning algorithms whose results will then be applied to a voting system to see if any of the results can be improved or if any other phenomenon occur. The results of these different algorithms will later be applied to voting systems in an attempt to improve upon the results through consensus.

3.1 Decision Tree

The decision tree algorithm creates and then applies a flowchart to the the test data. this flow chart will create rules and cut offs for the test data and the tree will ultimately filter down the tree in an attempt to separate the class attributes based on the other attributes present. I am using the decision tree algorithm from the party library in R for this portion.

3.2 Random Forest

The random forest is an extension of the decision tree algorithm but in this case many different trees are used instead of one master tree, which can help with the problem of over fitting.I am using the ronadom forest algorithm from the randomForest library in R for this portion.

3.3 Naive Bayes

The naive Bayes model is a simple algorithm that is based on the idea of conditional probabilities and determines the class of the case through the highest product value of the conditional probabilities.I am using the naive Bayes algorithm from the naivebayes library in R for this portion.

3.4 Bayesian Network

A Bayesian network is a type of ML algorithm of which the naive Bayes is one of the simplest forms. Bayesian networks are represented by directed acyclic graphs where the nodes are represented with variables and edges represent conditional dependencies. Nodes in the network that are connected represent conditional dependencies, with each node being associated with a probability function. I am using the Bayesian network algorithm from the bnlearn library in R for this portion.

3.5 Support Vector Machines

Support vector machines are a machine learning algorithm that attempt to split the data by class by creating a vector that has one less dimension than the data points wherein the data on one side of the vector will represent one class and the data on the other side represents the other class.I am using SVM algorithm from the e1071 library in R for this portion.

3.6 K Nearest Neighbours

K nearest neighbours is an algorithm that maps the the sample data and followed by mapping the test data against that and assigning the test data class attributes based on the majority of the closest k neighbours.I am using the kNN algorithm from the stats library in R for this portion.

4 Future Work

My code thus far represents a work in progress and this section outlines what the next steps I will be taking will include.

4.1 Results

While I have a plan for the ML algorithms that I will be using and what I am going to do with their results I will attempt to tweak some of their inputs to see if there is any effect that this may have on results and if I can explain why this happens. The main obstacle that I am facing right now is that given the volume of data that I am working with, when the ML algorithms are being run R will crash. I am currently working on a solution to this by extracting a representative sample from the data sets so that the algorithms will run smoothly on my device, as lab computers that may be able to handle this are unavailable. Although I currently have no proven results the code for each of the ML models can be found on my repository.

4.2 Additional Machine Learning Algorithm

While I currently have six ML algorithms that I will be using to apply to the data due to the nature of the voting systems shown below it greatly simplifies the majority rule voting when there are an odd number of algorithms being considered. It also give more leeway in the other voting systems for what is considered the cut off point for accepting or declining the different collective outcomes. One that I am currently considering is logistic regressing.

4.3 Voting Systems

I will be applying the three listed voting systems to further process the results of the machine learning algorithms to see if the results can be improved upon. I will be applying an odd number of ML algorithms to simplify some of the voting systems. In the descriptions for the voting systems below it will be assumes that an odd number of algorithms will be used.

- Optimistic System: This system of voting makes assumptions that have the most potential risk. This voting system will be assuming that if at least one or two of the ML algorithms shows that a case is fraudulent then it will be assumed that it is fraudulent.

- Majority System: This system of voting is analogous to a two option first past the post system. If a case is shown to be fraudulent in a majority of the algorithms then it will be assumed to be fraudulent.

- Pessimistic system: In this type of system the risk is minimized as much as possible. In this system only cases that have an overwhelming majority of the algorithms showing that the case is fraudulent will be deemed fraudulent.

4.4 Other balancing techniques

While I have used two balancing techniques already in over and under sampling there are others that may be worth exploring. One of these involves the creating of new synthetic data that can be used to train the algorithm. As this sounds interesting and may provide differing results to the methods that I have already tried it may be worth exploring.