­­­Applying Machine Learning Methods for Credit Card Fraud Detection

Dayanara Lebron-Aldea, Karthika Pai

STA 141C: Big Data and High-Performance Computing

**Introduction:**

Credit card fraud happens when a person’s identity is robbed, and their credit/debit cards are used as a fraudulent source of funds in a transaction. Frauds are limited to 0.1% of all card transactions and difficult to detect since they appear to be like any other transaction; so often times there is special attention to the costumers shopping behaviors and the area where transactions are normally made. Lately, credit card companies have recurred to implement machine learning methods to fight these illegal transactions and protect the costumer’s assets, by training models to detect these drifts from the customers normal shopping behavior.

This project focuses on exploring the logistics and implementation of machine learning methods through their application on Kaggle’s credit card fraud detection dataset. Our emphasis is in improving accuracy for each model and comparing them across different ML techniques to see if different methods also contribute in achieving a higher accuracy.

**Methods:**

*Data:*Our data contains 30 predictor variables composed of 28 Principal Components (PC), Time and the amount of the transaction. PC’s are centered at 0 and possess a standard deviation of 1, which is index of PC’s derived from standardized data. The data was made available by kaggle this way to protect the consumers information.

Only 0.17% of observations in our dataset are frauds, making our data imbalanced and posing a challenge in order to correctly build a predictive model without overfitting or under fitting. In order to tackle this problem, we applied oversampling and under sampling as techniques to adjust the class distribution of the dataset. Oversampling occurs when you replicate observations from the minority class to match the size of the majority, and under sampling occurs when observations from the majority class are eliminated to match the size of the minority class.

SMOTE ( Synthetic Minority Over-Sampling Technique) was used to oversample our dataset, by creating synthetic data based on the minority class data points. To undersample, we the used RandomUnderSampler method from the imblearn package which randomly selects a sample with replacement

*Model Selection:*

**Results:**

**­­­­**

**Discussion:**

**References:**