Quinn Samms
SCU ID#: 1590472

Program 2 Report

Rank & RMSE (At time of writing)
Rank: 4
RMSE: 3.8451

My Approach
My approach for the regression problem consisted of preprocessing the data frame by transforming non-numerical data into numbers, and feature selection methods on top of dimensionality reduction to create data that can more accurately be used to predict donation amounts.

For data preprocessing, I created a function called convertToNumerical, which takes a pandas data frame as an argument. This function loops through each column in the data frame, and creates a dictionary mapping non numerical values to a number. Each time a new entry is recorded in the dictionary the value of the next item is incremented by one. This creates a data frame that has all numerical entries which can now be used for our regression model. Additionally, some of the entries in the data frame has 'NaN' values which had to be dealt with. For all these entries I replaced NaN with 0.0 with a simple call to the 'replace' method via pandas.

Next, for feature selection and dimensionality reduction, I used another panda's data frame method called corr, this function returns a 2D representation how correlated the column's entries are to each other in the data frame, using this function we can find how correlated each column is to the target variable. I filtered out a majority of the irrelevant columns using a minimum correlation threshold and then preformed PCA dimensionality reduction on the remaining most relevant columns.

Finally, after data preprocessing and feature selection/dimensionality reduction, I used a simple Linear Regression model from sklearn libraries to predict the potential donation amounts from the given data.

Methodology
My plan to approach this problem was to start simple and use dimensionality reduction methods that I have used before in lab along with simple regression techniques to get a baseline score and improve upon that using different feature selection and reduction techniques, along with different data representation methods, etc. to build the best prediction model.

To start, I needed to represent all my data numerically that wasn't originally numerical. The most simple and intuitive solution that I thought of was to assign each unique entry in a column with its own unique number mapping. I researched online and found similar solutions implemented by others so I knew it would be a valid approach to my problem so I implemented it in my solution.

Next, I first started using merely PCA reduction along with a linear regression model to get the ball rolling, this yielded me decent results around the middle of the pack, and after trying a couple

different dimensionality reduction techniques my results were not getting any better so I knew I needed to experiment to get better results.

My next thought was to only include a subset of all the columns since there are so many that there are bound to be some columns that need not be included in the regression model. I researched online and found the corr method for a panda's data frame and I thought it would be perfect for my problem since theoretically the most correlated columns to the target should be the best at predicting a new target value.

After using the corr function to select a subset of columns I tried the regression model on my new data and it yielded worse results than any of my previous attempts. My problem was my threshold for correlation was too high and it was removing too many columns so that it could not accurately predict a target value, I played around with the threshold value and found that the best results came when the number of columns was reduced down to around 150.

My next ideas were to use PCA dimensionality reduction on top of the newly selected columns. I used PCA reduction with the number of components at 10 and I got the best results I had so far. I knew I was on the right track after this and I adjusted the n_compnents parameter to try and optimize the solution which yielded me my final submission and best result with n_components set to 5.