**Predicting Loan Default with SVM and KNN Models**

**Preparing Data for Modeling**

For creating these models, due to performance concerns we had to reduce the size and dimension of our cleaned/joined dataset. To accomplish this, I ran a logistic regression model on the data and used the p-values obtained from this model to help reduce the dimensionality of the data. After reducing the number of columns, I created categorical variables as needed.

At this point, I inspected the “TARGET” variable, and discovered that there were significantly more data points with the value of 0 than 1. I decided to sample an equal number of data points consisting of 5000 rows where the target variable equals 1, and 5000 rows where the target variable equals 0. Now our data has been reduced to a more reasonable size for training and testing the SVM model of 10 data points.

Next, I checked the continuous variables for any possible outliers using boxplots. I discovered outliers in the AMT\_ANNUITY, DAYS\_EMPLOYED, total\_amt\_credit\_sum, and total\_amt\_credit\_sum\_debt columns that reduced the accuracy and sensitivity of the final model. Once these data points were removed, we ended up with a reduced dataset of 8,563 data points.

For judging the viability of a model, I decided to focus on accuracy and sensitivity. In the response variable, customers that ended up defaulting on a loan were classified as “1”, so I believe a financial institution would be most concerned with this metric, along with the overall accuracy of the model.

At this point we’re ready to split our data into training and test datasets and create our models. Both SVM and KNN model functions in R have options to automatically scale our data for us, so we didn’t need to worry about performing that step manually.

**Create SVM Model**

To get the best performance, I used a simple linear kernel along with testing 9 different values for C (ranging from 0.00001 to 1000) to find the best way to tune the model for accuracy and sensitivity. I found the best performing SVM model to have a C value of 0.0001, an accuracy of 64%, and sensitivity of 86%. Out of all the tested models this model appeared to have the best overall accuracy and sensitivity.

Using this optimal C-value, I attempted to model the TARGET variable against all the predictor columns individually, but none of the individual columns appeared to be more accurate than creating the model utilizing all predictors.

**Create KNN Model**

Using the same training and test data, I trained a K nearest neighbor model. I tested different values for “k” (ranging from 1 to 100) and discovered that the optimal k-value for predicting loan default was at around 75 (based on the visualization I created, this appeared to be where the optimal cutoff for k would be without overfitting the model).

From this KNN model, I was able to derive a vector of probabilities, so I decided to take this k-value and test different cutoffs as well (ranging from 0.1 to 0.9). I discovered that the most accurate KNN model had a cutoff value of 0.4. With this cutoff value we have an accuracy of 64%, and sensitivity of 86%.

**Conclusions**

It appears that both models perform equally well, if you tune the model correctly depending on the metrics you wish to optimize. While we were able to tune both models to achieve a high sensitivity, overall accuracy was somewhat low. We were also unable to utilize the full dataset due to performance issues with the SVM model. For these reasons, I’d estimate there are better modeling options we can utilize that are able to handle the entire dataset, and therefore more accurately predict loan default.