Stephen Recupero

I. Dataset Background

The dataset being analyzed in this experiment, dataset 20, consists of log-transformed gene expression data for patients who have had breast cancer. Included in the dataset is a total of 212 individual patients along with log-transformed gene expression values for 22,215 genes across each patient's genome. Every patient is labeled with either "Relapse" or "No Relapse," for a given time interval of 6.5 years.

II. First Steps & Preprocessing

Data was preprocessed and class descriptors were converted into data. A value of 1 was given to "Relapse" and 0 for "No Relapse." It was first observed that the "Relapse" class had far fewer samples than the "No Relapse" class, with 60 observations for the former and 152 observations for the latter. It is clear that with regards to calculating predictor classifiers this distortion must be accounted for such that sensitivity and specificity are both high.

III. Correlation Matrix and Establishing non-Independence

Having preprocessed, I calculated the correlation matrix between each gene and between each gene and phenotype expression class. I next sorted and acquired the top ten positively correlated genes with phenotype expression class and top ten negatively correlated genes with phenotype expression class. By convention of formatting, the positively correlated genes were associated with increasing the potential of breast cancer relapse, while the negatively correlated genes were associated with decreasing the potential of breast cancer relapse. I next researched the top ten for the negatively and positively correlated genes, and found validating information about 2 positively correlated genes and 2 negatively correlated genes.

CCNB1: experimental drugs have been approved for targeting in gene therapy, strong association with early primary breast cancer, association with breast cancer recurrence

RACGAP1: experimental drugs have been approved for targeting in gene therapy, strong association with early primary breast cancer, association with breast cancer recurrence

ZFP36L2: has been found to inhibit cell proliferation including MDA-MB-231 breast cancer cells, but has not been found to cause apoptosis

NOP14: has been found to suppress breast cancer progression by inhibiting NRIP1/WNT/(beta)-catenin pathway

I next sought to establish non-independence of these highly correlated genes with the phenotype expression class to demonstrate the relevance of future predictor classifiers. I performed chi-squared tests for the top ten most positively correlated and top ten most negatively correlated genes. The p value generated for the top ten most positively correlated genes averaged to 0.99999, while the p value generated for the top ten most negatively correlated genes averaged to 0.97553, which is still approximately 1 but with a higher variance. Additionally, one negatively correlated gene, had a particularly relatively low p value of 0.76998. In detecting Mutual Information to demonstrate the difference between the entropy of each respective relatively highly correlated gene and the entropy of the same gene conditioned by the information of the phenotype expression class, we observe an average of 0.09535 mutual information for the positively correlated genes and 0.06932 mutual information for the negatively correlated genes. Both averages are non-zero and demonstrate further the non-independence of certain genes and the phenotype expression class. Again we notice a difference between the positively correlated genes and negatively correlated genes. This difference is most likely representative of the imbalance in phenotype class distribution, with the top negatively correlated genes being related to a distribution that has mainly No Relapses or “0” cases and thus intuitively being less strongly related to the phenotype expression class, while the positively correlated genes have a stronger more identifiable relationship with the phenotype expression class. This is further indication, that the class imbalance must be accounted for in predictor classifiers.

IV. Hypothesis Testing

Due to the assumption of normality being unable to be applied to this data set, we had to resort to non-parametric Hypothesis testing. Wilcoxon rank-sum test

was an easy choice as our data set has 212 patients and can thus be separated into different samples with the assumption of independence between the samples. 212 patients, in genome mining is a rather not insignificant amount of samples. In the Wilcoxon rank-sum test utilized on this dataset, the assumption or null hypothesis is that data extracted from differential classes or phenotype expressions in this case, will produce the same rank statistics, or in essence, that the data extracted from differential classes comes from the same distribution. Thus, for a particular gene, if the null hypothesis is rejected, with a specific threshold or significance level, then the gene comes from a differential probability distribution dependent on the class it is coming from. I used the standard 5% significance level or alpha, and thus if a p value for a particular gene is less than 0.05, the null hypothesis is rejected. Applying the hypothesis test on my dataset, I found 4977 genes out of 22215 genes to be differentially expressed with significance level 5%. However, type 1 error is very high for just a general Wilcoxon rank sum test, as a hypothesis test was performed on each gene. To accommodate for this error, we used the Bonferroni Correction, which heavily decreases the amount of genes which are determined to be differentially expressed. With the correction applied, only 41 genes were determined to be differentially expressed. Now this preliminary hypothesis testing was performed on the entire dataset for exploratory purposes. With regards to the genes or features selected for the predictor classifier, hypothesis testing was performed only on the Train data, and the features selected differed depending on the split between train and validation data sets. A test set was set aside prior to any accuracy calculations and NO hypothesis testing was performed on the test set.

V. Oversampling

Now while hypothesis testing can better feature selection for a predictor classifier, which is very high for this dataset, the large class imbalance must still be accommodated for. While undersampling is often used to improve class imbalance by decreasing the samples of the overrepresented class, it is not a feasible option for this dataset as 212 samples is too few amount to have any disregarded as it would lead to improper classifiers. On the other hand, Oversampling seems to be a clear choice to improve this imbalance.

Oversampling generates samples of the underrepresented class. For this dataset, I chose to generate samples from the underrepresented class until there was a class balance. The level of generation can be chosen differently, but this seemed the straightest shot. Oversampling can be performed through the generation of new patients or by randomly duplicating old patients of the underrepresented class. I chose to use Synthetic Minority Oversampling Technique (SMOTE) and pursue the novel generation of samples. In practice, SMOTE finds the nearest neighbors of a sample from the underrepresented class and then by choosing a random nearest neighbor, it generates a novel sample between these two points. For the purpose of building predictor classifiers, SMOTE was used only on the Train dataset and executed after feature selection. This could have been implemented differently, with SMOTE being applied prior to feature selection, and that could have led to different samples being generated as the features determine a given neighborhood of samples.

VI. Classification

Learning Framework

We will now perform classification using different predictor classifiers on the given Breast Cancer Relapse dataset. We will use two predictor classifiers: Logistic Regression and Random Forest. Both classifiers are discriminative methods rather than generative. From an initial standpoint, Random Forest was chosen because the tree depth hyperparameter could be varied such as to facilitate the optimal hyperparameter selection in training and validation, and then implemented on the test set that was set aside. Logistic Regression was utilized as a base model to compare the accuracies of Random Forest at varying levels of the hyperparameter. We detail the exact process for building the classifiers and the exact dataset splits.

Train/Validation - Test Split:

We first made a static split of 2/3 of the dataset for training and validation purposes and 1/3 of the dataset set aside for testing purposes for once the classifier with the optimal hyperparameter is chosen.

KFold Cross Validation:

For this dataset, a 10-fold train-validation split was chosen. These splits were performed on the 2/3 Train/Validation dataset. These splits were random so as to generate the most accurate representation of accuracy for a given predictor classifier. Stratified splits were used so that the proportion of classes was preserved in each split. Stratified splits were necessary as the class imbalance could distort accuracy.

Pre-processing:

Prior to testing accuracy on the validation set of any given split, pre-processing was performed on each training set. Feature selection was performed by means of the Wilcoxon Rank-Sum test with the Bonferroni correction on each training set. The amount of features thus differed depending on the given split. SMOTE was also applied on each training set, PRIOR to testing accuracy on the given validation set. SMOTE was NOT applied on the validation set as well.

Evaluation:

Evaluation as discussed in this section is only in terms of mean accuracy of the following metrics, calculated from the performance of the given predictor classifier trained on the given train set and then applied on the given validation set. Mean accuracy is calculated from the 10-fold splits.

Our choice of training metrics include Balanced Accuracy, Precision, Recall, and F1 Score.

Balanced Accuracy was chosen because of the extreme class-imbalance. An accuracy score would not be able to accurately evaluate as a classifier that classifies every patient as “No Relapse” or 0 would achieve a high accuracy while the amount of false negatives would actually be very high. In practice, Balanced Accuracy equals the (sensitivity + specificity) / 2. In essence, it evaluates the correct classifications for each class at the same weight.

Precision was chosen as to measure the accuracy of positive predictions, or in the case of our dataset, the accuracy of predicting “Relapse.” This measure is especially relevant so as to limit the unnecessary cancer treatment if the

classification model was used in practice. By definition, Precision is the (True Positives/ (True Positives + False Positives).

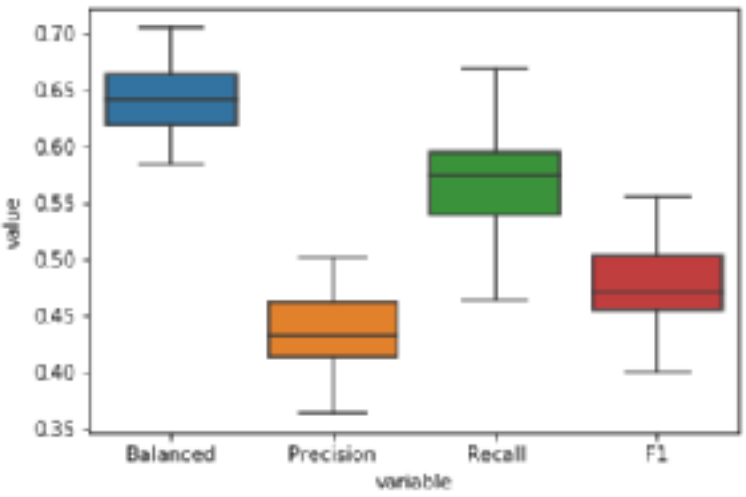
Recall was chosen so as to measure the accuracy of negative predictions, or in the case of our dataset, the accuracy of predicting “Relapse.” This measure is especially relevant to our dataset as it captures the efficacy of how relevant a given classifier is in actually predicting if a given patient is not relapsing. A low Recall in our case can indicate a classifier that is highly skewed to classifying “No Relapse,” and is especially significant as our class imbalance is driven by the large amount of No Relapses. By definition, Recall is True Positives/(False Negatives + True Positives).

F1 Score is by definition (2*Precision*Recall)/(Precision + Recall). It is an especially relevant metric as it accounts for the presence of false negatives and false positives in such a way that is missing from balanced accuracy. These serious false classifications lead to aforementioned issues and so this metric measures the classifier relative to these false classifications.

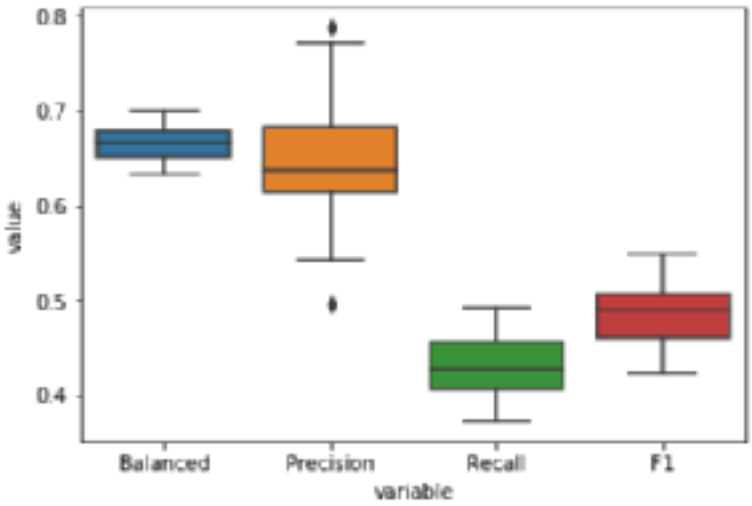
Logistic Regression

For Logistic Regression, as there was no hyperparameter to tune, we acquired different metrics for when there was feature selection and for when there was not feature selection. Below, we have box plots shown, which were calculated from 32 runs of the classifier model without feature selection and then with feature selection. As one can see, with feature selection leads to overall higher precision and less variable balanced accuracy, but the balanced accuracy largely stays the same. Also, recall decreases rather significantly. We can attribute the less variability of the balanced accuracy to the feature selection providing a more standardized and selective input into the classifier. However, the decrease in recall is highly problematic and as the classifier cannot be tuned by hyperparameters, it leads us to consider Logistic Regression as not the optimal classifier to apply to the test set.

Without Feature Selection



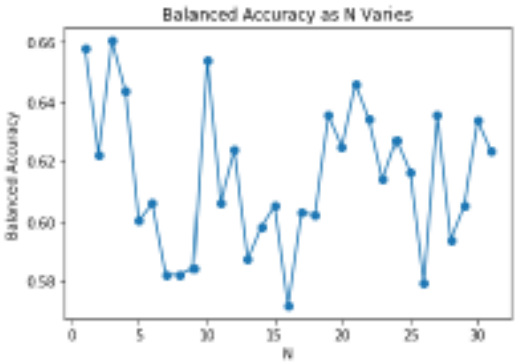
With Feature Selection

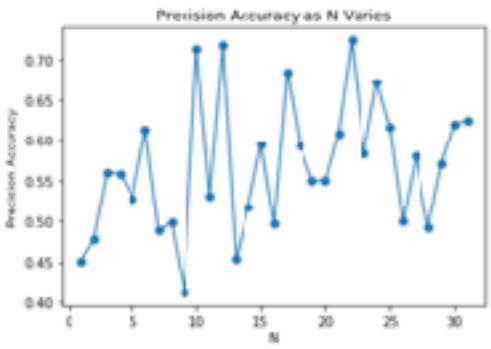


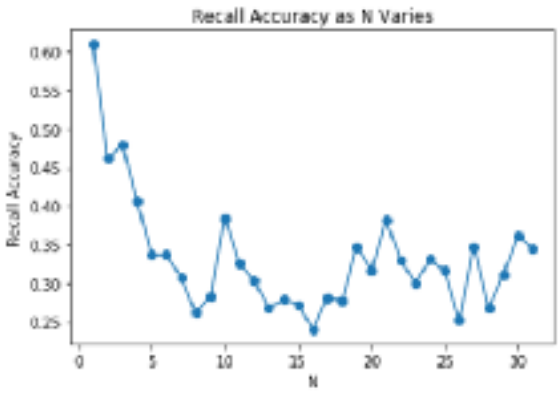
Random Forest

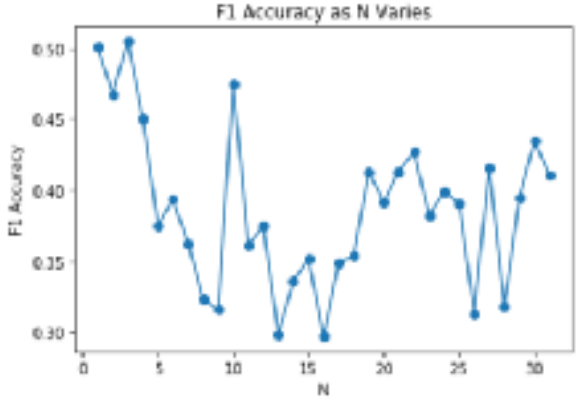
For random forest classification, we had a hyperparameter to tune, the depth of the tree. We thus calculated accuracy for each metric at each value of the hyperparameter, which spanned from 1 to 32. We also performed this variation of hyperparameter on feature-selected training sets as well as on training sets without feature selection. Additionally, we performed this variation with and without SMOTE applied as well as without feature selection in both cases.

<u>Without Feature Selection</u>

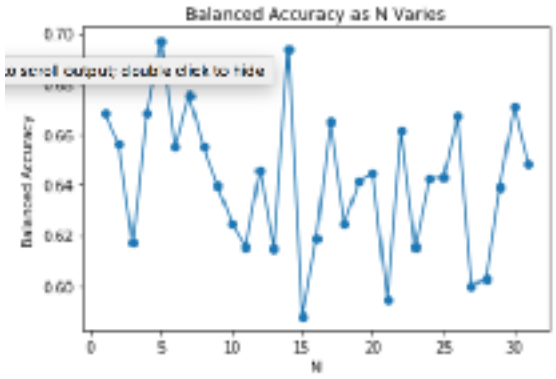


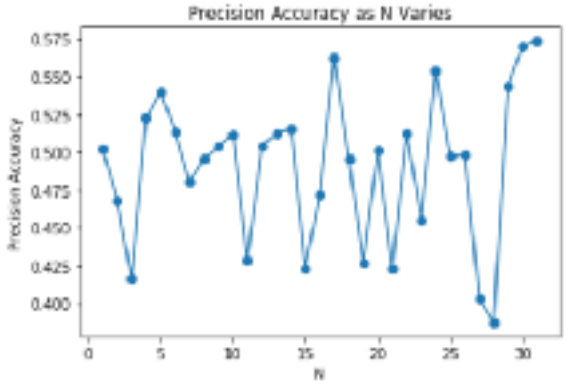


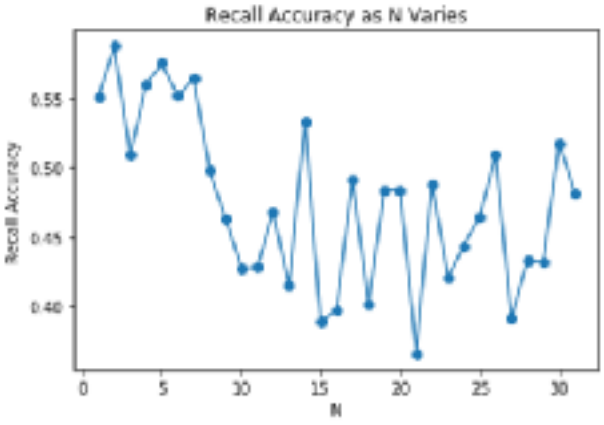


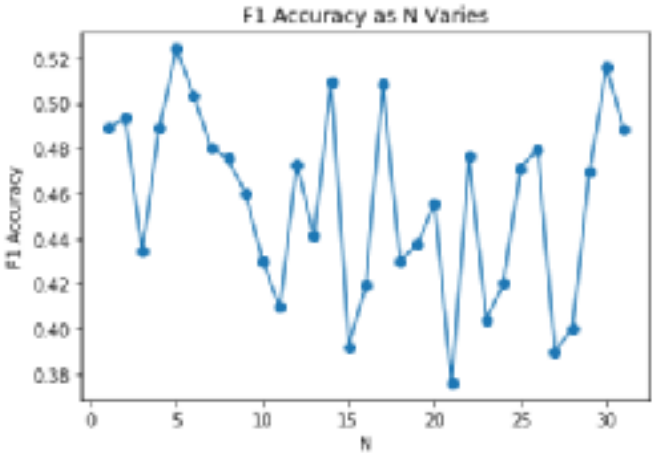


With Feature Selection

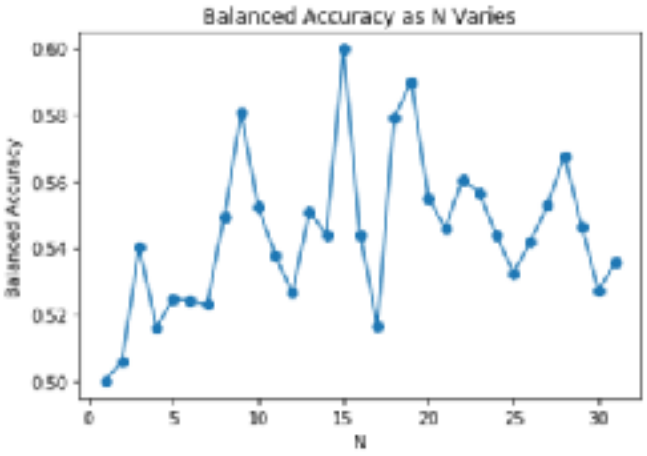


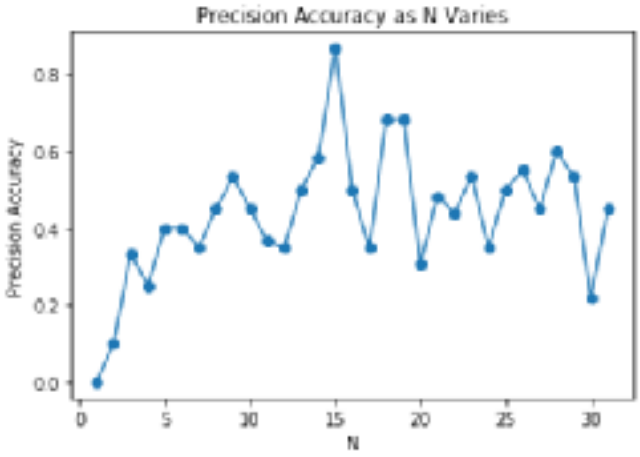


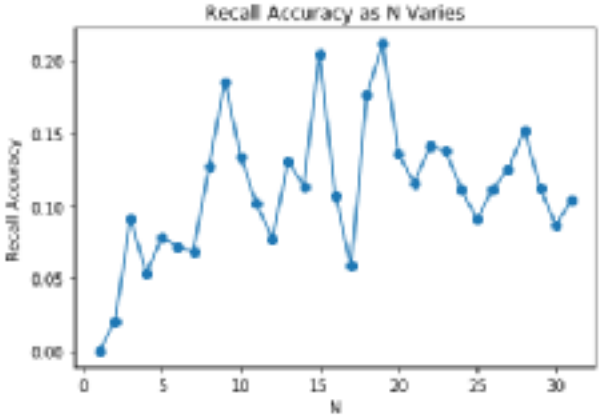


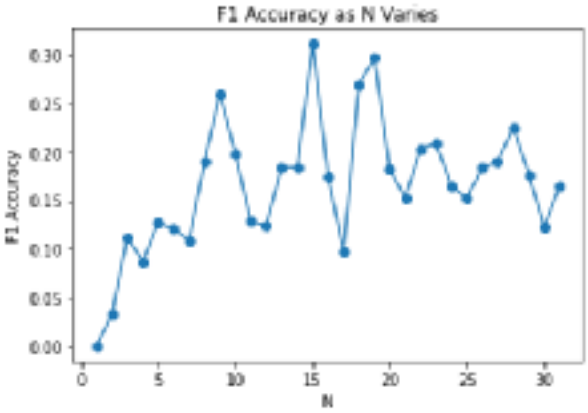


Without Feature Selection or SMOTE









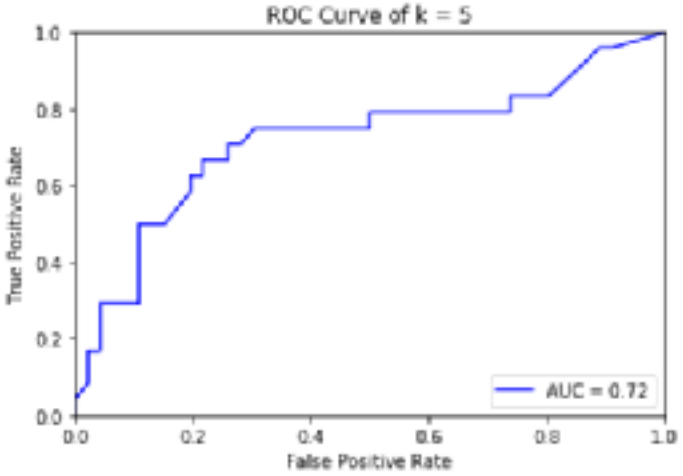
As we can see from the above plots, with feature selection we see improvement in balanced accuracy, Recall and F1. Precision actually decreases with more feature selection. This is slightly different from the results seen from feature selection on Logistic Regression, but they are different classifiers. Without feature selection or SMOTE we see a far lower balanced accuracy and a seriously decreased recall and F1. This result can be explained by the fact that SMOTE allows for more accurate classification of the underrepresented class and thus allows for a decreased amount of False Negatives. Thus the absence of SMOTE leads to a classifier that almost solely classifies all patients as “No Relapse” and leads to very low recall.

With feature selection and SMOTE applied, we observe the highest Balanced Accuracy (.7) and F1 Accuracy(.53), and rather high Precision and Recall at k = 5 for Random Forest classification. It is clear that results are very variable and this rather exceptional accuracy of tree depth = 5 may be a random result. But for the ultimate test of a classifier with a tuned hyperparameter on the test set, we intend to use the Random Forest k =5 classifier.

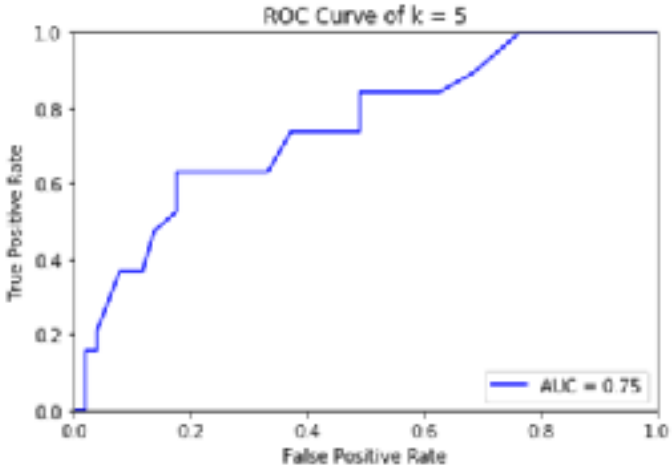
Classification on the Test Set

Now having selected the hyperparameter k = 5 for Random Forest Classifier, we test accuracy on the set aside test set. Performing preprocessing in the same process as before, we receive a balanced accuracy of 69.4% with feature selection and SMOTE. Without Feature Selection or SMOTE the accuracy is 56%. This is honestly not a great improvement, but it is still significant. Additionally, the ROC curves do differ to a certain degree.

Feature Selection & SMOTE



Without Feature Selection & SMOTE



As we observe in the ROC Curve for the application of the classifier with feature selection and SMOTE, the curve is more bowed out, thus allowing for a slightly more distinguishable discriminatory threshold.

VII. Conclusion

As we have observed, feature selection, through Wilcoxon Rank-Sum test and Bonferroni Correction, and oversampling, through SMOTE, overall have led to slightly improved performance of both predictor classifiers. The final application of the tuned hyperparameter for the Random Forest Classifier was also quite hopeful, with 69% balanced accuracy, which was one of the highest accuracies I observed. It is clear that SMOTE dramatically increases Balanced Accuracy and Recall for this particular dataset due to the serious class imbalance. Feature Selection enhancements are less clear in their positive effects and the effects of its application tends to be more variable. Overall, accuracy metrics were not very enthusiastic even with the added improvements. This may be resulted from other necessary techniques of preprocessing that were lacking. Additionally, the dataset itself was highly imbalanced, had a very small sample size, and contained a very large amount of features. Future additional changes or enhancements could include applying SMOTE prior to feature selection, as well as applying deep learning techniques.