Ensemble Methods: Adboost and Majority Voting

Christopher Griffith - *@cdgphysics*

I. INTRODUCTION & MODEL PARAMETERS

This project focused on the performance metrics of two ensemble methods: Adboost and Majority Voting. An email messages data set of 4601 records with 57 attributes was standardized before being used with any model. Therefore, each feature category of the data set was transformed to have a mean of 0 and a standard deviation of 1. It was worth investigating how the performance of the models would change due to the standardization of the data set. Also, the class label distribution of the training/testing split datasets maintained the class label distribution of the original dataset using stratified random sampling. Regarding parameters for the classification models, the Adaboost classifier used a decision tree classifier as the base estimator while the Majority Voting Rule classifier used the following three estimators and parameters: (1) k-nearest neighbors classifier with the default 5 neighbors, (2) random forest classifier with the default 100 decision trees and (3) logistic regression with 10,000 maximum iterations. After investigating the mean positive class (spam label) predicted probabilities of each estimator within the fusion classifier, it was determined that each estimator provided reasonable estimates of the class probabilities to warrant a soft voting classification approach in addition to a hard voting classifier. As illustrated in Figure 1, the estimators produced relative reliable probability densities for the 50/50 training/test data split and were great candidates for a soft voting classifier which leverages the argmax of the sums of the predicted probabilities of the three estimators independently.

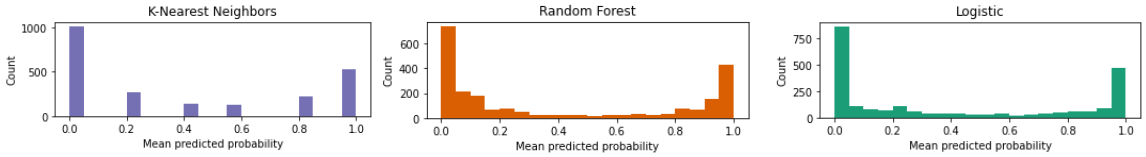


Figure 1: Positive Class (Spam Label) Predicted Probabilities for the three estimators of the Soft Voting Classifier: 50/50 Training/Test Split

II. SUMMARY OF MEAN PERFORMANCE

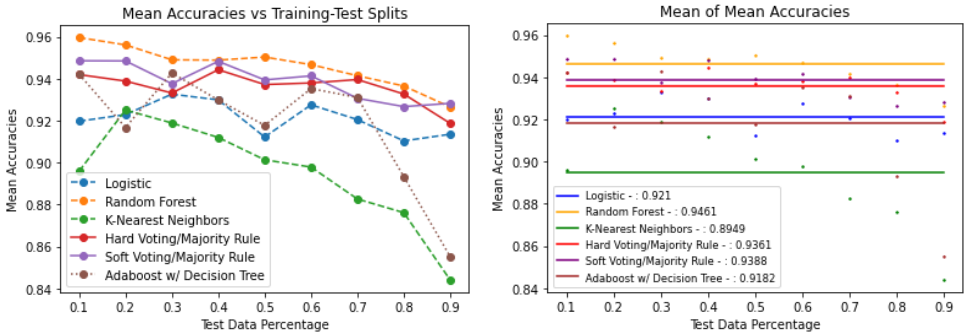
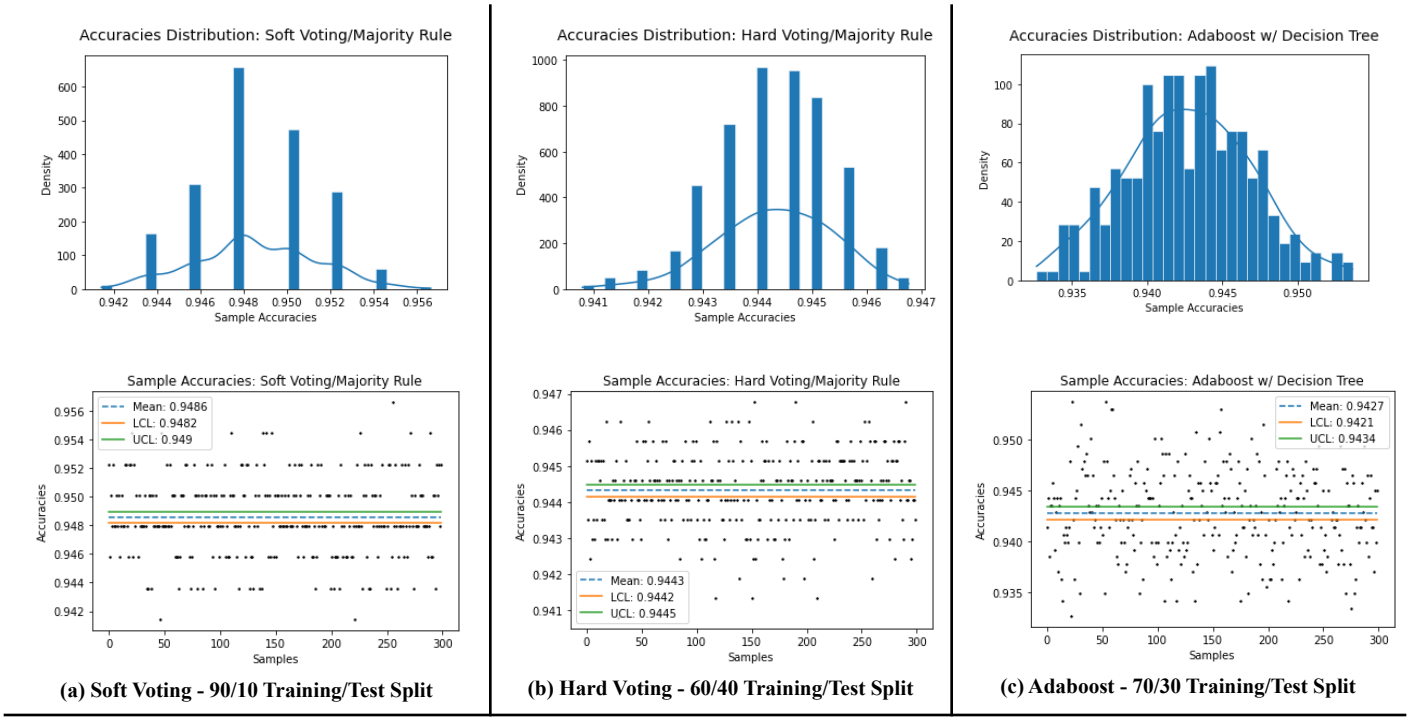


Figure 2: Mean Accuracies of Classifiers Based on Training/Test Splits

For every classification scheme in this project (Logistic, Random Forest, K-Nearest Neighbors, Hard Voting/Majority Rule, Soft Voting/Majority Rule, and Adaboost with Decision Trees), 300 models were curated and the mean of the performance metrics, including the classification accuracies, confusion matrices, ROC and PR curves, were evaluated and compared to all six classification schemas. For example, the Random Forest classifier was applied to the data set 300 times for a given split percentage and the mean accuracy of the 300 samples was recorded. This process ensured that distribution accuracies of the 300 models closely resembled a Gaussian Distribution. This process was also duplicated for each test data split (i.e. 0.1 means 10% of the data records was used to validate the model while 90% was used to train the model) and overall 2700 models for each schema were produced. Figure 2 shows the mean accuracies (300 samples per schema per split size) compared to each schema across nine different test splits (i.e. 10%, 20%, 30%, … , 90%). The best performing classifier overall for this project was the Random Forest classifier with a test split percentage of 10%. From the ensemble models, Soft Voting at a 10% split ranks the highest followed by Hard Voting at a 40% split. The best Adaboost performance occurred during a 30% testing split. Despite the results, practically the Hard Voting classifier would be most appropriate considering it only needed 60% of the data to train in order to reach similar results as the top two performers. As expected, once the models are trained with less data, the performance metrics decrease slightly or significantly based on the classifier. K-Nearest Neighbors performed the worst when only 10% of the data was used for training.

III. ENSEMBLE METHODS: MEAN ACCURACY DISTRIBUTION AND ANALYSIS OF BEST PERFORMANCE METRICS



(a) Soft Voting - 90/10 Training/Test Split **(b) Hard Voting - 60/40 Training/Test Split** **(c) Adaboost - 70/30 Training/Test Split**

Figure 3: Accuracy Distribution (300 Samples) and Mean Accuracy 99% Confidence Interval for Best Performing Classifiers

One of trends was that as the test split percentage increased (i.e. the ratio of records used to validate the models' performance), the distribution of the accuracies became more gaussian. Since the sample size of our study was large enough (n = 300) to produce a Guassian Distribution of accuracies, a mean confidence interval of 99% (i.e. $\bar{x} \pm 2.576\frac{s}{\sqrt{300}}$) was produced for the classification techniques of this project. Figure 3 provides the accuracy distribution along with the upper confidence limit and lower confidence limit of the mean accuracy for all three top performing ensemble methods. The Confusion Matrix, ROC and PR curves for the Soft Voting - 90/10 Training/Test Split classifier with the best mean performance metrics are illustrated in Figure 4.

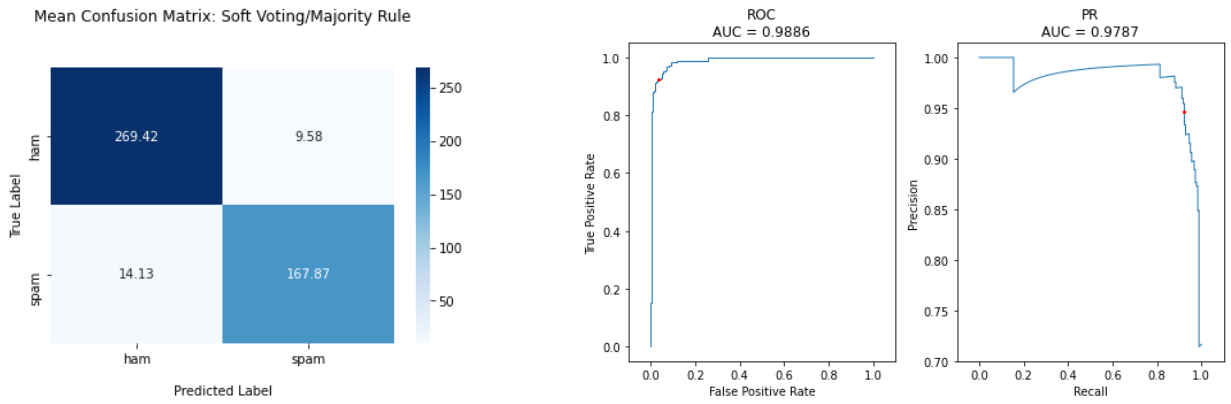


Figure 4: Confusion Matrix, ROC and PR Curves for Highest Mean Accuracy - Soft Voting 90/10 Training/Test Split

IV. CONCLUSION

	Highest Performance	Lowest Performance

Classifier	Test Split	Mean Accuracy	Test Split	Mean Accuracy
Logistic Regression	30%	93.2657%	80%	91.0350%
Random Forest	10%	95.9653%	90%	92.6515%
K-Nearest Neighbors	20%	92.5081%	90%	84.3999%
Hard Voting/Majority Rule	40%	94.4313%	90%	91.8715%
Soft Voting/Majority Rule	10%	94.8576%	80%	92.6714%
AdaBoost Ensemble w/ Decision Tree Classifier	30%	94.2732%	90%	85.5111%

Table 1: Best Mean Accuracies with Performance Metrics per Classifier

Based on the results in Table 1, the Random Forest classifier has the highest mean accuracy of the list of classifiers used in the project. The Soft Voting Ensemble Model using Logistic Regression, Random Forest and K-Nearest Neighbors barely outperformed the Adaboost Ensemble and Hard Voting Ensemble Models. If the training/test split percentage was a factor in evaluating which model is most suitable for the data set, the Hard Voting Ensemble Model performed well with only 60% of the data required for training.