Michael Fosco

This report is a continuation of the last report on building a model to predict who will likely experience financial distress within the next two years. Unlike last time, the missing values for this dataset were filled in with the median for that variable. This decision was made because upon closer examination of the data it was revealed that much of the data was skewed and so using a median would given a better approximation to the majority of people than the mean. When testing the models, 5-fold cross validation was used. This means the data was split into five groups; four groups were used for training and one for testing until every group had been the test group once. For each metric, the mean across the five runs was used. In the attached data files one can also find the standard deviations for each metric for each run of a classifier.

A variety of metrics were calculated for each model. The area under the curve (AUC) is a measure of general performance of the model. When a statistic is calculated at a threshold, such as precision at .05, then the lower the threshold the more the model is predicting yes as opposed to no. Precision is a measure of when the model predicted financial distress, how many times was it right. Recall measures: out of all of the true data points where financial distress will happen, how many did the model identify correctly. I also calculated the accuracy of the model at a threshold of .5. For each of the above metrics, the closer to one the metric is, the better the model is by that metric. The training and testing times of the models were also calculated and were measured in seconds.

In determining the best classifier to determine who will experience financial distress in the next two years I tested the following models: logistic regression, k-nearest neighbors (KNN), random forest, the extra trees classifier, adaboost classifier, support vector machine, naïve bayes classifier, decision tree classifier, and a stochastic gradient descent classifier (SGD). Due to the number of classifiers, I unfortunately decreased the window for the tested parameters for each classifier in order to be able to create this report on time. Nonetheless, I was able to test across a few parameters and identify a classifier that appears to do better than most other classifiers. All reported statistics are for the best metric for that class of models across all combinations of parameters. The statistics are also included in a table at the end of the report.

For logistic regression (LR), I varied the window to calculate the difference between points and the level at which the algorithm converged. Logistic regression tended to be in the middle of the pack in most metric measures. Its best AUC was .827, accuracy was .9339, and precision at .05 was .4239. Its accuracy was a little better than just guessing on the data.

The k-nearest neighbor (KNN) models varied from having a uniform weight on all points to having distance determine the weight. I was able to vary the number of neighbors included in the calculation between 1,000 and 10,000. Models with the weight being based on distance did better in general than those with a uniform weight. KNN had a poor AUC of .6508, a slightly better than guessing accuracy of .9332, and a low precision at .05 of .1651. KNN also had the slowest test time. In summary, I would not recommend using KNN.

For random forest (RF) and extra trees (ET), I varied the number of estimators from 25 to 100, the depth of the tree from 1 to 50, and the minimum number of samples that must be present for a successful split from 20 to 50. RF had the highest AUC of .8646, the highest accuracy of .9375, and the highest precision at .05 of .4847. It was also in the middle of the pack as far as train time though it was on the lower end as far as test times. ET was in the upper portion of AUC with .8436, had the second highest accuracy of .9368, and the second highest precision at .05 of .4781. Types of classifiers that included trees tended to perform well.

Adaboost (AB) was another model within the top tier. Its best AUC was .8602, accuracy was .936, and its precision at .05 was the third best at .479. It did however have the second slowest train time (.82 seconds), and one of the slower test times. In contrast to AB, support vector machines (SVM) and the stochastic gradient descent (SGD) performed poorly. SVM varied calculating the area of splits using a linear function, polynomial function, or a sigmoid. SGD used a perceptron algorithm (single-layer neural network) to reach convergence for the model. SVM’s best AUC was .529, accuracy was .9319 (the worst one), and its precision at .05 was .1211. SGD had the worst AUC at .506 and the second worst accuracy of .933. SVM also had the slowest train time with its best train time taking 16 seconds. I would not recommend using SVM or SGD.

Decision trees (DT) performed well while the naïve bayes classifier (NB) did not perform particularly well. There were no variables to vary for NB though for DT the depth of the tree varied from 10 to 50 and the minimum number of samples that must be in the result of a split varied from 10 to 50 as well. DT had one of the higher AUCs with .8408, an accuracy of .9348, and a precision at .05 of .4471. In general DT performed in the upper half of the pack of models. On the other hand, NB tended to be in the bottom half of the pack with an AUC of .7057 and an accuracy of .932.

Overall, I would recommend utilizing a random forest classifier to determine who is most likely to experience financial duress within the next two years. RF had the best overall performance, highest accuracy, and high precision. Also, since predictions are for the next two years, the speed of training should not matter too much. Additionally, the testing time is not too long for RF. Note that this report only summarized statistics for the best results for any combination of parameters for all models. For a list of all of the models tested and their statistics, please see “resultsTable.csv.” There are also other attached csv files that include the top ten (or less if fewer than ten models were generated) models for each classifier for the metric in the name of the file.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | AUC | Accuracy | Precision at .05 | Recall at .05 | f1 at 0.05 | Train time (sec) | Test time (sec) |
| LR | 0.827 | 0.9339 | 0.4239 | 0.3174 | 0.363 | 0.0738 | 0.0013 |
| KNN | 0.6508 | 0.9332 | 0.1651 | 0.1277 | 0.1414 | 0.7359 | 2.6677 |
| RF | 0.8646 | 0.9375 | 0.4847 | 0.3628 | 0.415 | 0.3436 | 0.1195 |
| ET | 0.8436 | 0.9368 | 0.4781 | 0.4876 | 0.4093 | 0.2757 | 0.1204 |
| AB | 0.8602 | 0.936 | 0.479 | 0.3793 | 0.4101 | 0.8209 | 0.0167 |
| SVM | 0.529 | 0.9319 | 0.1211 | 1 | 0.1253 | 16.3747 | 0.1008 |
| NB | 0.7057 | 0.93249 | 0.26169 | 0.19588 | 0.224 | 0.038 | 0.0079 |
| DT | 0.8408 | 0.9348 | 0.4471 | 0.35 | 0.3904 | 0.291 | 0.0037 |
| SGD | 0.506 | 0.933 | 0.0668 | 1 | 0.1253 | 0.0825 | 0.0012 |