GB 656 Final Project: Participate in a Data Science Competition!

Xiao (Freeman) Zhang Team members: Benyuan Xie

Business Framing & Problem statement:

It's a data competition we need to use existing passenger data and transported results to predict the remaining passengers' transport results. It is 2912, and we, White Star Space Lines, launched the first flight of the cosmic titanic. On our way to the first destination—the torrid 55 Cancri E, the spaceship collided with a spacetime anomaly hidden within a dust cloud. Lots of people were accidentally transported. We have found the data and transmission results of some passengers, but the whereabouts of some passengers are still unknown. We need to analyze the existing data to predict the results of the passengers. This is very important to the future of our company, because shareholders have invested much money and we cannot make it wasted. If we can find all the transported passengers, the company's reputation will be redeemed, and we can save the company and business. At the same time, the analysis of passenger data and its transmitted results can help us modify the spacecraft and provide protection services and reinforcement of supporting facilities for passengers who may be at higher risk. This can provide better security protection for future travel while improving our market competitiveness, leading to higher profits.

It is a classification problem, which we need to determine the binary results for all the passengers. From the data file, we have train data and test data. In the train data, we have each passenger’s information, such as age, room on the ship, and use of ship services. Also, we have the results of whether these passengers were sent away, represented by true or false. For the test data, we have all the similar information of each passenger but not the transport result. Our task is to analyze the information of passengers in the training data and its results and build a model. Then, we need to apply the built model to the passengers in the test data and simulate their transport results. For both training data and the test data, the X-axis is the features of each passenger, which includes age, room, various consumption, Etc. The Y-axis is the result of whether the passenger was transported or not, either true or false. We need to analyze the X features and Y outcome of training data to predict the Y of test data.

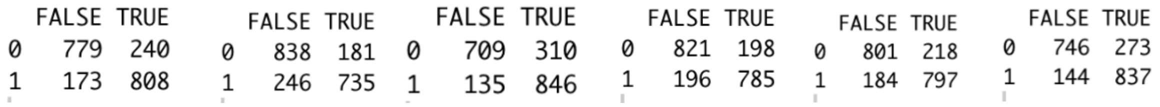
Approach:

Our overall idea is to perform data processing on the training data and divide it roughly according to the ratio of 25% of the test data and 75% of the training data. For 75% of the training data, we integrated the variables and built six different models based on passenger information to predict transported outcomes. We then apply the model to 25% of the test data for outcome prediction. The 25% test data here are not the passengers we need to predict but are separated from the training data because we want to compare the gap between the model-predicted results and the observed results. When we find the model with the lowest overall error and the largest AUC, we apply it to the actual test model, which is the data we need to predict. First, we need to organize and integrate the training data. All features can be roughly divided into two categories, numerical and categorical. For digital variables, we need to make up a lot of unknown data, or NA. Since many of the models we use and even the actual test data have unknown data, we hope to fill in these data reasonably to make the model more effective. Through analysis, we use the median to fill in the missing data, which uses the overall median of the data column. For categorical variables, we first turned it into a dummy variable; then, for the missing values with a few possible categories, we filled it with the most frequent value of their column. The exception is the room number in the data. The method we use is to randomize the existing room numbers because there are many categories of rooms.

After data filling and integration, we began to use various features and results of 75% of the training data for model construction. For all the features, we did not use every passenger's name and passenger ID. We used R and python to build 6 models for passenger delivery outcomes: logistic regression, classification tree, classification cross-validation, boosting, random forest, and Neural Net. After the construction is completed, we apply the six models to 25% of the test data respectively. The results of all models will be presented in decimal form, and we all use a threshold of 0.5. For passengers with a probability greater than 0.5, we mark it as 1, which means they have been transmitted, and for passengers with a probability less than 0.5, we mark it as 0, which means they have not. We generated a confusion table and calculated its prediction accuracy, specificity, and sensitivity for the predicted and actual observed results. Also, we graph the Roc curve and calculate the corresponding AUC score. Finally, we compared these criteria of the six models, selected the model with the highest comprehensive performance of accuracy, TPR, TNR, and AUC scores, then applied it to the actual test data, and found the remaining passengers who were transported away.

Comparison of Model results:

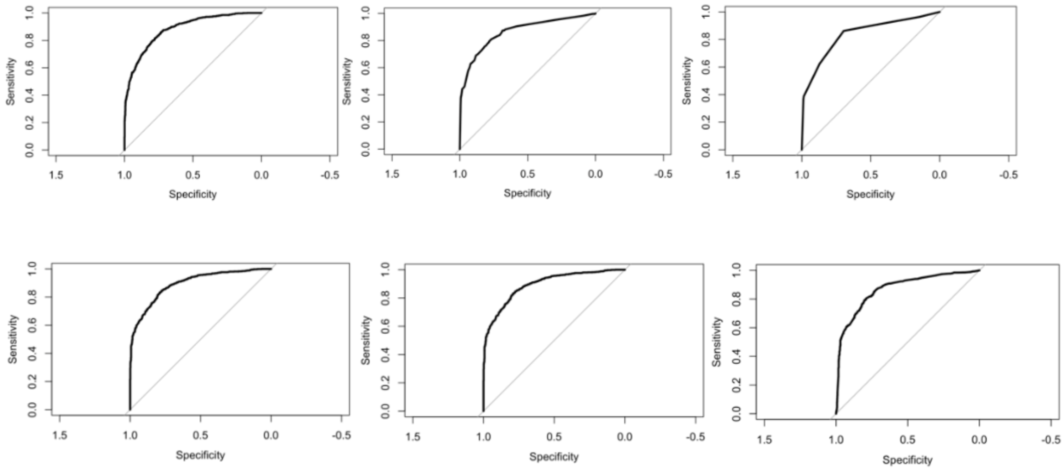
The following are the confusion tables. From left to right, the order is logistic regression, classification tree, classification cross-validation, boosting, random forest, and Neural Net.



```
   FALSE TRUE      FALSE TRUE      FALSE TRUE      FALSE TRUE      FALSE TRUE      FALSE TRUE
0    779  240    0   838  181    0   709  310    0   821  198    0   801  218    0   746  273
1    173  808    1   246  735    1   135  846    1   196  785    1   184  797    1   144  837
```

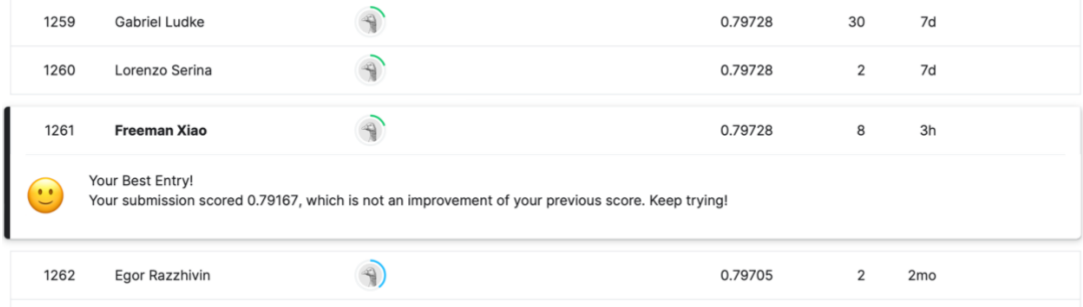
We have an accuracy of 79.35%, 76.45% specificity, and 82.36% sensitivity for the logistic model. For the classification tree with prune, we have an accuracy of 78.65%, 82.24%

specificity, and 74.92% sensitivity. After cross-validation, we have an accuracy of 77.75%, 69.58% specificity, and 86.24% sensitivity. We have an accuracy of 80.3%, 80.57% specificity, and 80.02% sensitivity for the boosting model. For random forest, we have an accuracy of 79.9%, 78.61% specificity, and 81.24% sensitivity. Lastly, the single neural net model has an accuracy of 79.2%, 73.21% specificity, and 85.32% sensitivity. From the confusion table, we can see that the cross-validation model has the best sensitivity or the true positive rate, but it has the worse true negative rate. By contrast, the classification tree model has the best specificity; however, it has the worst sensitivity. In summary, the boosting has the best accuracy; although it does not have the best sensitivity or specificity, it may have the best ability to distinguish them. Following, we have the Roc curve for each model and the AUC score; the order is the same.



The logistics model has an 88.2% AUC score which means that there is an 88.2% chance that the logistics model can distinguish between the positive and negative classes. The AUC score for the rest of the five models is 86.36% for the classification tree, 83.56% for cross-validation, 89.41% for boosting, 89.36% for the random forest, and 86.91% for the neural net. As we can see, the boosting model has the best AUC score, and its curve also appears to cover more area.

Clearly, all the models have great performance; almost all of them have a correct rate of about 80% and have an AUC score of more than 85%. However, obviously, boosting is the best among the six models. Although it could have better sensitivity and specificity performance, he has a relatively higher correct rate and AUC score. We applied the Boosting model to formal test data and submitted the results. Before I submitted it, I predicted that he would give us about 80% of the score because the confusion table had a similar result. The result is pretty much the same!



1259	Gabriel Ludke	0.79728	30	7d
1260	Lorenzo Serina	0.79728	2	7d
1261	**Freeman Xiao**	0.79728	8	3h

Your Best Entry!
Your submission scored 0.79167, which is not an improvement of your previous score. Keep trying!

1262	Egor Razzhivin	0.79705	2	2mo

We ended up with a score of about 0.79728 and a rank of about 45 percent, which is not bad for the first data science competition!

Summary:

The entire data competition is a relatively basic classification topic, but it was an excellent exercise and got me out of the classroom for the first time with real applied data. Compared with applying the model learned in class, it is very challenging to organize and analyze the overall data before building the model. It is equally essential to deeply understand the data to make reasonable adjustments and choose the appropriate model. At the same time, the actual data model construction always has no completely correct answer. We can always find better methods in model construction and data adjustment, and the adjustment of one variable can sometimes bring about a one percent improvement, which is really interesting.