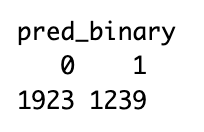
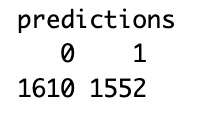
**Describe your baseline algorithm and provide necessary tables and/or figures to summarize its performance based on the training data**

The first algorithm that I used to do binary classification is the XGBoost algorithm. Using this algorithm, I got an accuracy of 0.892453 on the leaderboard. The prediction results are shown as below.

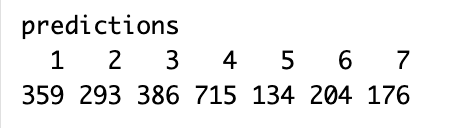


I then tried the neural network algorithm with two layers. The first layer has 10 nodes and the second layer has 5 nodes. The accuracy on the leaderboard is 0.923382, which is higher than my baseline algorithm. The prediction results are shown as below.

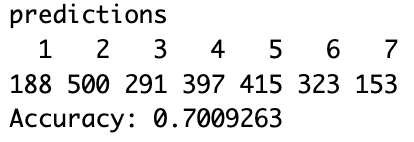


Next I tried the SVM algorithm, which turned out to be my final algorithm which I will discuss later.

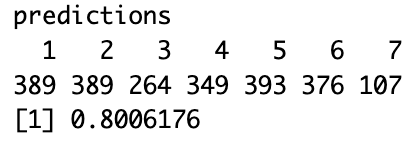
For the multiple classification task, my baseline algorithm is the naive bayes algorithm. I split the training data to train part and test part and calculate the accuracy. The accuracy is 0.62197. The results are shown as below.



The next algorithm I tried is the svm algorithm. The accuracy increased but is still not high enough. The results are shown below.



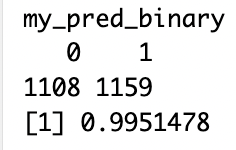
Then I switched to the knn algorithm with 5-fold cross validation. The accuracy increased significantly but is still not high enough. The results are shown below.



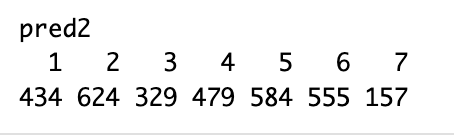
I finally tried the random forest algorithm and that algorithm gives me the highest algorithm, so I choose it to be my final algorithm which I will explain more in the next section.

**Describe your final algorithm and provide necessary tables and/or figures to summarize its performance based on the training data**

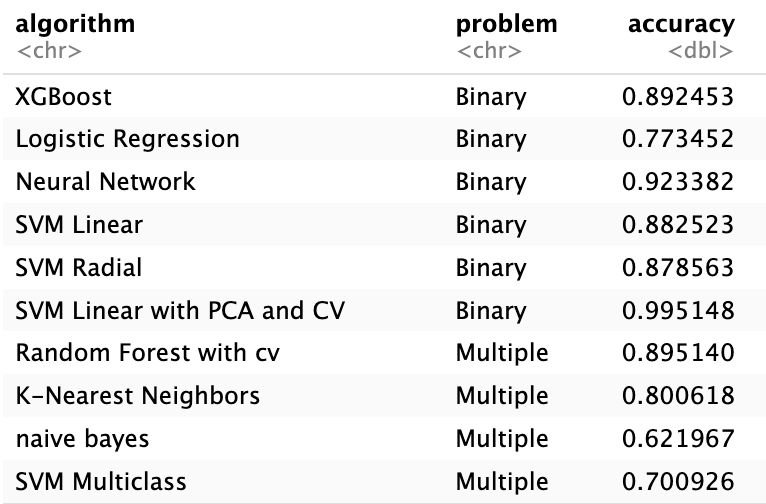
My final algorithm for the binary classification is the SVM algorithm combined with PCA and cross validation. Since we have high dimensional data, I first used Principal Component Analysis to perform dimension reduction. I selected the first 20 principal components and used the Support Vector Machine algorithm with a linear kernel to perform binary classification. I also used 5 fold cross validation to improve the accuracy. After adjusting the parameters in the svm function for many times, I finally get the results shown below. The accuracy is calculated by separating the training data with 70% for training and 30% for testing. Since the accuracy is 0.9951478, which is pretty close to 1, I decided to stick with this method.



For the multiclass classification problem, my final algorithm is the random forest algorithm with 10 fold cross validation. The accuracy on the leaderboard is 0.89514. I also tried the knn algorithm, the svm method but they all give lower accuracy. The results of classification are shown below.



**Use a figure or a table to show your leaderboard performance. Describe your efforts to improve the performance.**

****

My baseline algorithms have low accuracy, so I improved the accuracy by adjusting parameters in the functions, switching to another algorithm, using PCA to perform dimension reduction and adding cross validations.

**Comment on your final results and potential ways to further improve the classification accuracy.**

The results from my final algorithm still do not have very high accuracy. In order to further improve the classification accuracy, I may also try ensemble learning methods and other boosting algorithms such as Gradient boosting and adaptive boosting. I could also try using the kennel trick to reduce the bias and shrinkage methods to reduce variance of my algorithm.