CS 178 Final Project Report

Group Member: Zhengran Ji(87537895), Kaggle Group Name: John Jii, Public Position: 66, Private Position: 76

1. Introduction

In this project, I apply four kinds of different ML algorithms to the given dataset. There are 107 features for the x data and 1 feature for the y data. The size of the training dataset is 7432. Before training, I split the training data into training and validation data with the ratio of 4:1 and shuffle the training data. Finally, I rescale the x data into the same scale to avoid some feature has so large scale that affects the coefficient finding.

2. Model

2.1 Linear Classifier

I use the linear classifier in the mltools package and set the learning rate to be 0.01 and the iteration to be 100. The training set AUC score is 0.66 and the validation set AUC score is 0.67.

2.2 **Neural Network**

I build a neural network using the Keras package. My network has two hidden layers each with 100 neurons. The output layer of the network has 1 neuron with sigmoid activation since we are trying to predict the probability. The accuracy on the training set is 0.96 but for the validation is 0.59.

2.3 **Decision Tree**

I first try a decision tree with all the parameters being the default value, and the tree overfits the training set. Then, I start to do complexity control on each parameter to find the best parameters for the tree. After doing complexity control, I build a tree with max depth = 5, minaret = 64, and minLeaf = 8.

2.4 **Bagged Decision Tree**

I assembled 100 decision trees, for each tree, I set it to overlearn part of the data. Then, I take the average of the prediction from each tree and output as the predicted value. The AUC score for the test data for this model is about 0.74, which is the highest score I got.

2.5 **Random Forest**

For random forest, I used the sklearn package. I set the random state=42, n jobs=-1, max depth=50, n estimators=100, oob score=True. After the training, the training error is 0, and the validation error is 0.34.

3. Table for the models

Model	Score	Train accuracy	Validation accuracy
Linear Classifier	0.6926	0.6155	0.6067
Neural Network	0.6646	0.7535	0.6215
Decision Tree	0.6926	0.6544	0.6512
Bagged Tree	0.7398	1	0.6606
Random Forest	0.6581	1	0.6539

4. Conclusion

This problem is a classification problem, and there are 107 features in x data. In this case, according to the theories we learn, a decision tree might be a good model to fit the data. Then, I conducted several experiments using different models we learned. Base on the score and the accuracy, I can conclude that a Neural network with only dense layers is the worst model to fit the data. Linear classifiers and single decision trees can be used to fit the data but the accuracy is low because these two models are too simple. Bagged Decision Tree is the best model to use for fitting the data since it balanced well between complexity and not-overfitting.