Seper Abdi – Final report (Data Mining CS 530) 05/2022

1. Preprocessing

We start with preprocessing the data. First, we group the data based on the ID of the players by averaging their performance through multiple years that they play. Also, to only have numerical features, we remove the 'Year' and 'DraftYear', as well as 'Name of the player since they do not affect the prediction. To keep the ID of the players in the test data, we keep the row names in the test data, which corresponds to the player's ID.

Then, we consider the pick column to be the label of the data, and all the other columns as the features. Since it is only important to do prediction to see whether a player is selected in the first round or not, we change the pick value to 0 if the player is selected in the first round, and to 1 if the player is not selected in the first round or has not been selected at all. This way, we change the problem to a binary classification problem. To do this label change, we know that if a player is selected in the first round, its pick value is between 1 to 32, and if not, its pick value is larger than 32 or NaN. So, anything smaller than 32 is mapped to 0 and anything larger than 32 or equal to NaN is mapped to 1.

Since the data is heavily unbalanced toward label 1, we apply random over sampling to balance the data. Unbalance data in machine learning is one of the main challenges and it may result in the failure of a machine learning model. This way, we are making sure that the poor prediction of the models is not because of the unbalance data provided to the model. Then, as a preprocessing step, we standardized the data, using the distribution of the train data.

Since the test data does not have label, similar to the real-life scenario, to report the performance of the models, we split the data to train and validation datasets. To do so, we split the train data to two parts, with 80% train data and 20% validation data.

2. Classification

We consider six models. The first model is a neural network with six dense layers, with 1024, 512, 256, 128, 64, and 1 nodes, respectively. We used batch normalization between the layers.

The second model is SVM with RBF kernel. The third model is logistic regression with L2 regularization, the fourth model is naïve bayes, the fifth model is ensemble of these models with hard voting, and the last model is ensemble of these models with soft voting. We used the balanced accuracy as the metric. The best performance is achieved using deep neural network. So, we applied this model on the test data to report the predictions. Below is the accuracy of different models with NN as neural network, Log as logistic regression, SVM as support vector machine with RBF kernel, NB as naïve bayes, En1 as the ensemble model with hard voting, and En2 as the ensemble of models with weighted voting. The predictions on the test data has been saved in an .npz file.