DS-3000 Project | Phase 3

Algorithms/Models Being Explored

We have currently explored creating classification models using K Nearest Neighbors, Decision Trees, and Random Forest models for the purpose of predicting which team will win a given game. We are also planning to soon implement Naive Bayes and SVC as additional algorithms. All three of the currently implemented models boast an accuracy of about 60% at this moment - note that this is without any hyper-parameter tuning.

Training and Evaluation of Models

The preliminary training conducted thus far has only explored the aforementioned models, all of which are notably classification models. These models attempt to predict the winner of a given match granted the average performance of each player on each team. A future functionality we will pursue is using these models for regression to predict the number of rounds each team would win for a given matchup of two teams. Implementing these regression models would be straightforward and would offer insight as to whether our data is better suited for the task of regression or classification.

The present method of training and testing the data involves partitioning the data into a training and test set by a random selection of matches. Individual player data spans the entirety of the period captured, effectively measuring a player's historical performance and using that as a predictor of the match outcome. An alternative approach that was recently attempted allowed for the temporal partitioning of the dataset into smaller periods of time. If, for example, the dataset encapsulated a two-year period, the dataset could then be partitioned into four smaller datasets (each spanning a six-month period). The benefit of this partitioning is the ability to use more

focused and relevant player data that can then more accurately predict the actual level of performance we can expect from this player over this smaller time span.

Hyper-Parameter Tuning

Each model will be tuned for the greatest accuracy using sklearn's GridSearchCV across several of the aforementioned temporally partitioned datasets. For KNN, we will try various values of k from approximately 1-100. For Decision Tree and Random Forest, we will test various values of n_estimators from approximately 1-25. After aggregating results from the different time periods of datasets individually, we will be able to select parameters holistically such that their optimal value can be ascertained for any future predictions.

Expectations for Performance of Models

We expect to produce a prediction accuracy that is slightly better than 50% for match winner classification models. Although this might seem to be relatively conservative, predictions between any top ranked teams prove rather difficult as the margins between skill at such a level prove to be slim. That is to say that "upsets" happen rather frequently, and placement can oftentimes be volatile. That said, producing a model that can exceed 50% by *at least* a small margin may very likely still prove useful in a legitimate gambling setting. Granted that the predictions yielded thus far have exceeded 50% by roughly 10 points, we have definitely met our performance goals. Although, through further tuning optimization, we would like to see how far we can push our model without making too significant of changes.

As for the round wins prediction regression model we intend to create, a RSME of about 4 rounds is what we expect to be a realistic goal, although we would hope to get closer to 2-3 rounds. In a single match (best of 30 rounds), "strong" or "decisive" victories will still see the losing team pick up quite a fair number of rounds, usually at around 6-9 rounds. As such,

landslide victories are extremely rare. Therefore, being able to judge the expected intensity of victory or defeat could aid in properly gauging the respective monetary weight that a gambler might put on a given match. One final addition we are considering is creating a secondary regression model that is already told who won. This could be interesting in a gambling setting for those who are confident in a team winning, but would like to know how large of a round difference there will be (similar to guessing how many rounds before someone is knocked out in a combat sport). We are generally rather unsure about how accurate this model might be, so we will reserve a statement about accuracy until further testing is performed.