Project Status Report

EECS 349

Tim Herrmann, Mikhail Todes, Nathan Corwin
trh701
mbt810
nca318

Task: Predict a Major League Baseball player's batting average. The inputs are previous season statistics and the output is a numeric value predicting the next season's batting average. This task is important as Baseball is a multi-billion dollar industry in the US and Canada. Being able to predict player performance is a useful tool for general managers when making decisions on player contracts.

Data Set and Attributes: We are using the following 9 statistical attributes for our model. All statistics are numerical and are for the year prior to the one we are predicting:

- Year
- Games Played (G)
- At-Bats (AB)
- Runs Batted In (RBI)
- Walks (BB)
- Strike-outs (SO)
- Salary
- Age
- Batting Average

We have gathered ~14,000 data points where we have the aforementioned statistics as well as a known batting average for the following year. Our current method has been to use 60% of the data for training, 30% for validation, and are holding out 10% for final testing.

Preliminary Results: As of this point, we have begun our initial trials with model generation. We are using WEKA as our machine learning software package. To evaluate our model accuracy, we first needed to establish a baseline. To do this, we calculated the average absolute error based on a simple prediction that the batter would have the same batting average as the previous year. This returned an average absolute error of .052 for the training set.

The first model we built was a simple Linear Regression Model. We were able to achieve an average absolute error of .0426, showing an improvement of ~18.1% over the baseline. Next, we built a model using the M5P algorithm. This model combines decision trees and linear regression to create a more complex model with different linear regressions at each leaf node. With this model, we have been able to achieve an average absolute error of .0408, a 21.6% improvement over our baseline.

Future Plans: The next model we plan to try is a multilayer perceptron model. After that, the remainder of the quarter will be spent refining and improving our existing models. We will do this by tuning the various parameters applicable to each model. We will also introduce our validation set and perform operations such as cross-validation and decision tree pruning. After implementing and testing these refined algorithms we will compare all the results on our held-out data and select the algorithm

that has predictions with the smallest absolute average error. Finally, we will compile all the results and present them on a website along with our final algorithm.