Deliverable 3: Report

Vision

Our current vision for the system we are building is to create a sports betting package for the National Basketball Association (NBA). The goal focuses on implementing different statistical analysis tools that may be useful in making decisions for betting. Specifically, we want to create models that can predict future stats based on various features. While our original plan was to build a sports betting algorithm that was ideally applicable for sports betting outside of strictly the NBA, such as football and soccer, we decided to narrow our vision to perfect the precision of the player over under calculations for the NBA. Through specifying our goals for the algorithm, we are able to divert our main focus to maximize performance of one form of sports betting. In terms of evolutions within the system, we originally wanted to construct our own file reader through creating tables based on CSV files from self-coded functions. However, after implementing columns and data table functions, it was very difficult to debug, so we decided to use OCaml CSV to read and square the data. After extensive research, we also decided on using OCaml Owl and OCaml Torch for our numerical libraries to assist in the statistical analysis. Overall, we have adjusted our goals through trial and error to improve our feature extraction and statistical analysis for NBA sports betting.

Summary of Progress

Between MS2 and MS3, Eric worked on data manipulation and feature extraction for choosing and filtering select data from the NBA dataset for easy applicability to the algorithms. Tim focused on adding test cases for the moving average, column, and datatable functionality as well as working on viewability of table printing (things like to_string for datatable that allow for

all the columns to line up). Tanish worked on building a linear regression model and a neural network deep learning model for predicting scores and over/unders. Anthony continued working on the column and datatable functionality, adding extends and capacity, and changing add function as well as testing column and datatable functionality (with Tim). Lastly, Selena focused on making a TUI (text user interface) that allows the user to pick what data and what algorithm to apply.

Activity Breakdown

For each team member, give a bulleted list of the responsibilities that team member had, the activities in which they participated, the features they delivered, and the number of hours they spent working.

Eric

- Found sqlite database from Kaggle with NBA game data, used python to create a CSV with only the relevant data
- Created python code to web scrape advanced box score data from official NBA statistics website with functionality to combine data from multiple seasons
- Worked on data extraction, data manipulation, and feature extraction.
- Hours Worked: 26

Tim

- Implemented simple average and weighted average for column data structure (essentially mutable arrays, using fold left function)
- Implemented simple moving average and weighted moving average for both column data structures (arrays) and float lists (most likely going to be used due to CSV parsing)

- Hours Worked: 25

Tanish

- Developed OCaml functions to take in file path for a CSV and convert to string list list and float list list – now we have the ability to plug these structures into models for results
- Researched using linear regression to regress upon player box score stats with points scored as the output feature – found Owl OCaml library which will be helpful
- Hours Worked: 25

Anthony

- Made major progress in creating the data table structure and managing how we process CSVs
- Worked on pipeline from an entire CSV data file to filtering down to the relevant data for each model (particular player, a certain time period, playing a particular team, etc.)
- Implemented test cases for the data table and columns modules.
- Hours Worked: 24

Selena

- Researched sports betting algorithms to consider best approaches to evaluate player over under bets
- Read OCaml CSV documentation to create the data into “square” to apply operations on each data point
- Made a terminal interface that allows users to select models (Linear, Neural Network, EMA, etc) and data to apply the model algorithms to
- Hours Worked: 24

Productivity Analysis

We were a relatively productive group, however we all did have our own individual tasks, so were able to meet slightly less than between MS1 and MS2. However, we still had clear-set goals for each individual, which were relatively disjoint from each other. Much planning was done beforehand to make sure that integration of everything was easy, however we did run into some small barriers while doing this.

We accomplished pretty much everything planned during our sprints, and our estimates were pretty accurate. Some functionalities did not perform as well as we hoped, but for the most part, most functionalities were implemented as planned. In the future, though, we will definitely take more into account homework and exams coming up when deciding what work that we should get done.