CS1951A Final Project Pre Proposal

Steffani Gomez (smg1) | Vilayvann Ky (vky) | Krishna Aluru (kaluru) | Jiajie Chen (jc124)

We're not doing the project as a capstone.

We want to make a fantasy nba lineup recommendation engine for fanduel and draftkings.

We're trying to give lineup recommendations that update each day for that day's contests. In addition we're making a real time visualization module that shows how players' (in the lineup) points are changing as the day's games progress.

Some questions we are asking include - Can we determine the optimal feature set to make good predictions? Can we determine how many contests should be entered each day to maximize wins? And more broadly, can we beat the system more than 50% of the time with our algorithm?

We are using data relevant for the scoring rubrics here and here from here and here. We have collected this data from the beginning of the course. The data we have obtained so far are: player names and teams, types of goals scored, and game dates.

We collect individual player performance per game played, scraped from the NBA website. Then we transform the data from a per-game format to a per-player format, to make it ready to process by our ML algorithms. We additionally obtain Swish Analytics predictions from their API for comparison purposes.

For the final project, we also wanted to include more sources of data. We first want to begin gathering historical player data from past seasons, to gain a larger data set to train on. We also want to obtain historical line-up performance, to analyze optimal lineup arrangements as a whole. Furthermore, we have a secondary source of player game performance at https://www.stattleship.com, which we would like to use to help clean our previously obtained NBA data.

The data is contained in the NBA stats that are linked above. We'll initially collect historical data from the beginning of this NBA season and then we will move on collecting data from the endpoints everyday and adding that to our database.

We are hoping to come up with 6-7 viable lineups that we think will do well in daily fantasy tournaments by applying machine learning algorithms to different variables in the data.

We want to have a visualization model for the lineups using D3 on a webpage and if we have time we'll also move onto using Bokeh to show how points and other data are changing in real time as the day's games progress.

Evaluation Module:

- Creation of module to check accuracy (5-6 hours)

Visualize best lineups

- Utilize d3, and possibly Bokeh, to visualize lineups and display that to the website, preferably using a different/prettier visualization than the one for our midterm report (6-7 hours)
- Display that to website and update the visualization everyday (3-4 hours)

Enhance projection framework

- Include historical data from the beginning of the NBA, and use clustering to use previous similar players to predict how current players will play. (8-9 hours)
- Create an evaluation module for these improvements (4-5 hours)
- See if there is any hidden data from our lineups (such as how many of the players are from different teams) and tweak our lineups to capitalize on this information (3-4 hours)

Ultimate projection framework

- In addition to all the other ML algorithms, we would love to use neural networks with a small number of layers to try to get our accuracy above 90% (8-10+ hours)
- Create an evaluation module for the neural networks (5-6 hours)
- Suggest high variance lineups (5-6 hours)

75%:

1. Create a framework for projecting fantasy scores.

        a. Use logistic regression with different combinations of uncorrelated variables and find the best combination. Use historical data from past 5 seasons for each player.
        b. Create an evaluation module for checking accuracy of our projections.
    2. Visualize best lineups based on projections.

100%:

    1. Create a more robust framework for projecting fantasy scores.
        a. Get historical data from past 30+ seasons and cluster players into groups so that you can use data of similar players to predict current players' performances. Try other ML algorithms to predict fantasy scores.
        b. Create an evaluation module for checking accuracy of our projections.
    2. Try to glean hidden information from lineups as well. Suggest multiple lineups based on our projections and the information we learn from looking at historical lineup data.

125%:

    1. Create the ultimate framework for projecting fantasy scores.
        a. Use neural networks with a small number of layers. Try everything possible to increase accuracy above 90%.
        b. Create an evaluation module for checking accuracy of our projections.
    2. Suggest high variance lineups - replace high salaried players in lineups with a low salaried player with high variance. This could beat a lot of usual best lineup systems because when the high variance player outperforms a high salaried player, the user gains two major advantages - 1) overall improved lineup because of increased salary space; 2) already beat most other lineups that resorted to the high salaried player. We'll try to suggest 10 such lineups for each day's contests so that the odds of winning at least one increase and you can bet on multiple low salaried, high variance players.

Once we create an evaluation module and make our visualization more robust, we'll have met all the requirements for our 75% goal. As a backup for our 100% goal, we'll document everything that prevented us from achieving it. This could include difficulties with collecting and cleaning the much

larger dataset of 30+ seasons, no improvement in accuracy of predictions, etc.