MCGILL UNIVERSITY

FINAL REPORT - GROUP 23

ECSE456 - DESIGN PROJECT 1

Picking Winners for Fantasy Basketball with Machine Learning

Group Members
Ege AYDEDE - 260576559
Stephen POOLE - 260508650
Florence REGOL - 260639261
Asher WRIGHT - 260559393

Supervisor
Prof. Ioannis PSAROMILIGKOS

December 1, 2017

Abstract

The initial motivation for this project came from a paper, "Picking Winners", written by people at MIT. In this paper, the objective was for them to try to predict winning teams in fantasy hockey using integer linear programming. For us, we strived to use machine learning for a similar purpose - to predict winning teams in fantasy basketball. Seeing as our project pertained to machine learning, a field of study which none of us were very familiar with, we wanted to put emphasis on learning the necessary material properly before implementing a solution. In other words, learning the material itself was one of our main goals for this project. Applying this knowledge outside a theoretical framework, in order to actually build something tangible, was also an important objective for us. In order to achieve these goals, we broke our design process into three steps: Research, data collection, and implementation. This semester we touched on all three of these steps. Specifically, we have completed a significant portion of the research necessary to understand the theory, have collected all the data required for the project, and have built a preliminary working neural network. Beyond this, we also completed the high-level architecture of the entire system. To accomplish these steps, we took traditional approaches of reading, watching lectures, and following tutorials on topics relevant to our project, all of which gave us the knowledge we needed to complete the work done thus far.

Acknowledgements

Here we would like to acknowledge everyone who helped make this project the amazing success that it is.

Contents

List of Figures

1 Abbreviations

NN: Neural Network
ML: Machine Learning
NBA: National Basketball Association

2 Introduction

The primary objective of this project was to build a system to predict and output winning line-ups of basketball players for daily fantasy basketball competitions. A winning line-up is any line-up that, upon being entered into the competition, yields more money than the entry fee (i.e. gains a profit). The goal of the project was to accomplish this using machine learning, which we believed would yield better results than have been achieved with optimization techniques. The project was separated into separate milestones that were intended to collectively allow us to build the final result. These milestones were:

1. Learning the necessary material regarding topics in machine learning
2. Collecting sufficient amounts of data to allow machine learning to actually take place
3. Finding relevant tools and technology that will allow us to actually implement our final system.

The importance of this project has two components, one personal and one societal. The personal importance of this project has to do with the team creating it. This project has been, and will continue to be, a valuable learning opportunity. It allows the creators to learn more about data collection, machine learning, software projects, best practices, and engineering design. Machine learning is a field of significant interest to all of us. There is also a societal impact, tied to both the sports and academic communities. If we find that we are able to accurately and consistently predict winning line-ups for fantasy basketball, it may impact the way in which fantasy sports are played by everyone. Rather than rely on intuition, or optimization techniques, there may be more movement to use these neural networks. This could not only change how fantasy competitions are played, but could be adapted to select which players should actually be played, in real life. Theoretically, a coach could use the same technique, although with a different metric (winning a game instead of maximizing fantasy points) to figure out who to play. There may be a transition from using personal knowledge of the sport and optimization techniques to using machine learning or other techniques. Machine learning could help find patterns that we did not know exist. Finally, on the academic side, it could lead to more research on the applications of machine learning to sports betting.

3 Background

3.1 Inspiration

The inspiration for this project came directly from a paper written by David Hunter and his colleagues, all from MIT. The paper targets "top heavy" daily fantasy hockey competitions and uses an integer programming (IP) optimization technique to try to "beat" them. The optimization technique involved the researchers coming up with ideas about what *could* be beneficial for a line-up, building a model around the idea, and then optimizing and testing it. For example, one of their ideas was related to choosing defencemen on the same team. Upon reading this paper, we felt that the same goal could be achieved while allocating more to the computer. The neural net aimed to solve this.

3.2 Neural Networks

4 Requirements

4.1 Design Problem

The problem we are trying to solve is... In order to fully understand the problem that we are trying to solve, it is important to detail daily, top-heavy, fantasy basketball competitions.

4.1.1 Competition Choice

When most people think of fantasy sports, they think of creating a team at the beginning of the sport's season, and following this team through the season. The types of competitions and specific competition rules actually vary greatly, and rather than try to create a system that can handle all of these, we decided to target a specific subset. The competitions we will target must meet the following criteria:

- Daily competitions
- Low entry fee (< $10)
- High, top-heavy, prize pool
- Multiple entries allowed

Daily competitions are different from the traditionally thought of fantasy season. They occur over a single day, and involve only the games being played that day. These competitions occur only on nights where real basketball games are being played. A header for one of these daily competitions on FanDuel.com that meets our other criteria can be seen in figure 1. This header shows the low entry fee of $7, the high prize pool of $100,000, and that multiple entries are allowed. It also shows a list of the real games that are occurring that night.

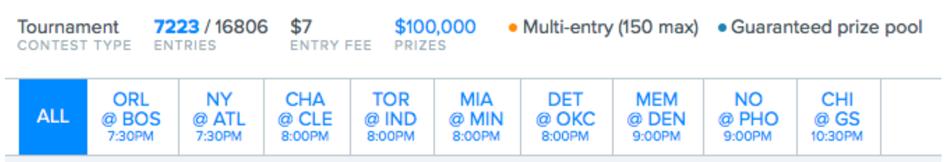

Figure 1: Daily NBA fantasy competition header

The third criterion in the list specifies a "top-heavy" prize pool. This was one of the conditions of the Picking Winners paper, and is one of ours, as well. The meaning of top-heavy is that most of the prize-pool winnings go to the top contenders. This is different from a 50-50 contest, where the top 50% of competitors are given double their entry fee. With top-heavy competitions, "contests give a disproportionately high fraction of the entry fees to the top scoring lineups." [2]. The entry fees for these competitions are usually less than $10, and the top placing competitors usually win thousands of dollars. We aim for these types of competitions because we aim to submit many different lineups, each with a high probability of success. The payout structure of a competition with a $4.44 entry fee and $400,000 prize-pool can be seen in Figure 2. One can see that the difference between first and tenth place is 200 times, whereas the difference between tenth and one-hundredth is only 10 times.

Using this information, we aim for these top-heavy payouts because we believe they will have a higher expected value for our system. We will submit many lineups to the same competition,

1st:	$100,000	**13th - 15th:**	$300	**446th - 620th:**	$18
2nd:	$40,000	**16th - 19th:**	$250	**621st - 870th:**	$16
3rd:	$20,000	**20th - 25th:**	$200	**871st - 1270th:**	$15
4th:	$10,000	**26th - 35th:**	$150	**1271st - 1770th:**	$14
5th:	$5,000	**36th - 50th:**	$100	**1771st - 2370th:**	$13
6th:	$3,000	**51st - 70th:**	$75	**2371st - 3170th:**	$12
7th:	$2,000	**71st - 110th:**	$50	**3171st - 4170th:**	$11
8th:	$1,000	**111th - 160th:**	$40	**4171st - 6165th:**	$10
9th:	$750	**161st - 230th:**	$30	**6166th - 9165th:**	$9
10th:	$500	**231st - 320th:**	$25	**9166th - 13164th:**	$8
11th - 12th:	$400	**321st - 445th:**	$20	**13165th - 23308th:**	$7

Figure 2: FanDuel daily fantasy competition payout

and expect each to be much above average (or else we would aim for 50-50). The goal is for one or two of our entries to be on the top-heavy side of a competition (not every competition), offsetting the losses of the other entries. Another reason we feel that obtaining a high ranking entry is feasible is because we believe that our system, if successful, will return uncommon lineups that other methods (intuition or optimization) will not return.

4.1.2 Entries

After selecting the competition to enter, the participant must create an entry, or lineup. This lineup must be comprised of players that are playing in the real games on that day. Additionally, the lineup must have a certain amount of specific basketball positions. A common requirement is that the lineup has to have two point guards (PG), two shooting guards (SG), two power forwards (PF), two small forwards (SF), and one center (C). Note that this may not be representative of a real basketball lineup. Figure 3 shows the lineup selection interface for FanDuel.com. On the right side, one can see the required positions for a given competition, along with the lineup's budget, in the top right corner. This budget is the amount of money each contender is allowed to spend on their lineup. On the left side, one can see the price of each player, along with their position, and their average fantasy performance for the season, on which the price is dependent.

4.1.3 Scoring

The participant's goal is to have the lineup with the highest score. The score is based on how well their selected players perform in real life, and is based on the amount of free throws, three pointers, field goals, blocks, steals, assists, and turnovers each player gets. A sample scoring system for competitions on FanDuel can be seen in Figure 4.

4.2 Requirements and Constraints

In order to solve the described problem, we had to comply with everything that was mentioned above. However, there are other requirements and constraints which are given below.

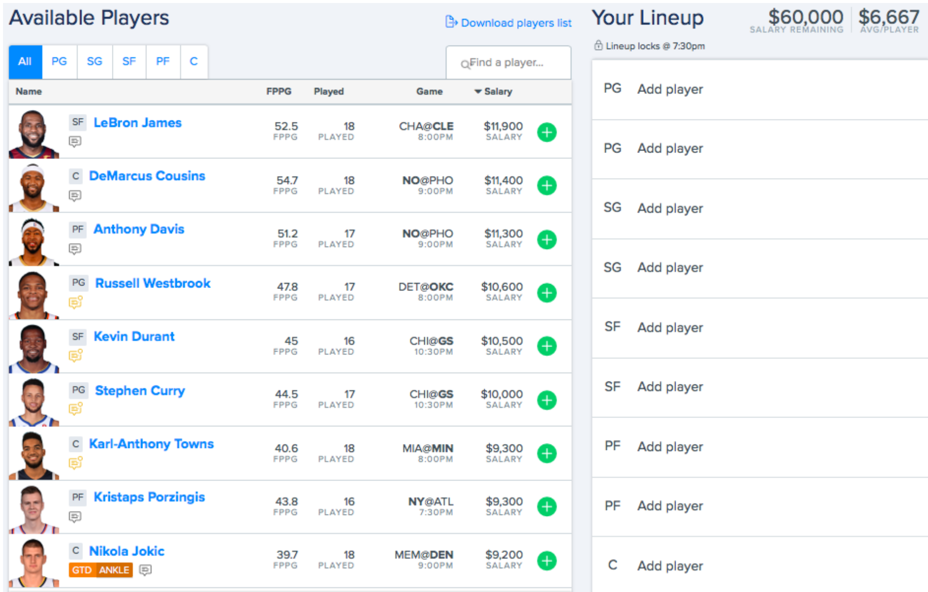

Figure 3: Fantasy basketball competition lineup

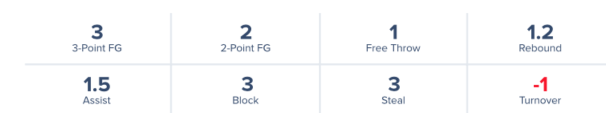

Figure 4: Fantasy basketball competition scoring

4.2.1 Functional Requirements

- The system should be able to use all previous NBA data
- The system should be able to output lineups for daily fantasy competitions based on which teams are playing, how much each player costs, and which positions are needed
- The system should output a projected score for each lineup

4.2.2 Usability Requirements

- The system should use a remote MySQL database
- The database should be secure and protected against obvious attacks
- The system source code should be version controlled

4.2.3 Technical Requirements

- The system should be able to train on the order of magnitude of minutes
- The system should be modular, allowing for quick and easy changes
- The system should allow NN parameters and other parameters (data amount) to be changed quickly for testing

4.2.4 Constraints

- The system must not require any expensive services
- The system must be trainable on our personal computers
- The system must be completed within eight months (two university terms)

5 Design and Results

A general timeline for our design solution can be seen in Figure X. Here, one can see that there are three sections. Ignoring the research section, which has been touched on in the Background section, we have two major sections into which the design solution can be divided. These two section are data and implementation.

5.0.1 Data

The data section of our design solution includes anything relevant to data, including collecting, storing, and accessing the data.

Collecting Data
We first needed to decide on a source...

Storing Data
As mentioned in the requirements...

Accessing Data
To access the data, we wrote...

5.0.2 Implementation

We began designing the software architecture before finishing the aforementioned database design. Our initial design was a single neural network that took in, as input, all of the stats of every active NBA player. Additionally, it had an extra input for every player which was set to 1 if the player was playing tonight, and 0 if the player was not. The goal of this neural net was to output the best lineup for the competition. A block diagram of this system can be seen in Figure **??**.

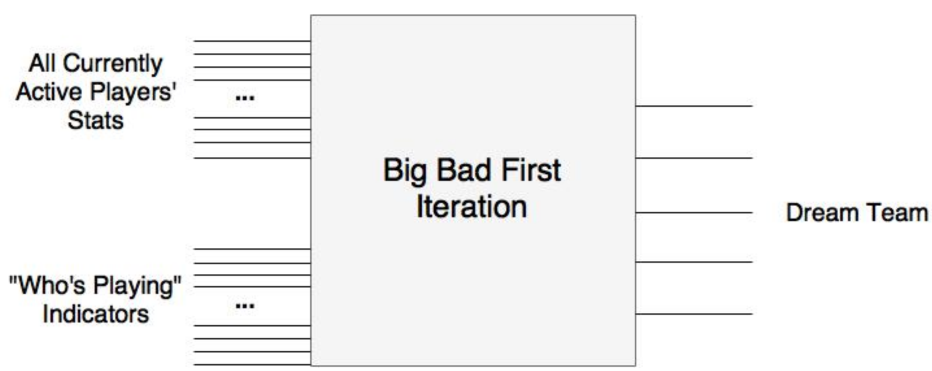

Figure 5: First design iteration

LIMITATIONS Feature characteristic : Should be informative, discriminating and independent

After discussing this design within our group, and with our supervisor, we came up with a few limitations. First, this system would only use data from active players. This was not taking advantage of the many decades of basketball data that we had access to. Second, it would also likely require a massive amount of data to be able to learn properly. This is because, generally, the amount of data required to learn adequately is correlated with the amount of inputs and complexity of inputs. Third, there would be many unused inputs each time the net is used. Since there are usually only about four NBA games each night, there are only about 80 players active on a single night. There are about 500 total active players, which means that there would be about 420 "unused" inputs. Although this is not explicitly a problem, it makes it seem like there would be a better way to structure the design. The final limitation of the initial neural network that we thought of was that it would be prone to discover "bad" correlations. Since the net only knows which players are playing, and not on which team, it would be bound to find correlations between players that are not playing against each other. To explain, say that NYK is playing against BOS in one game, and GSW is playing against CLE in another. The neural network may find a relationship between someone on BOS and on GSW, even though they are not playing against each other, and games are independent. Taking these limitations into account, we decided to come up with an improved architecture that was more flexible and required less data to train. We decided on a three step process, of which only one of the steps would involve machine learning, to start. The three steps are to calculate the player scores, use them in a NN to find good players, and then use a third system to choose lineups. An overview of this software architecture can be seen in Figure **??**.

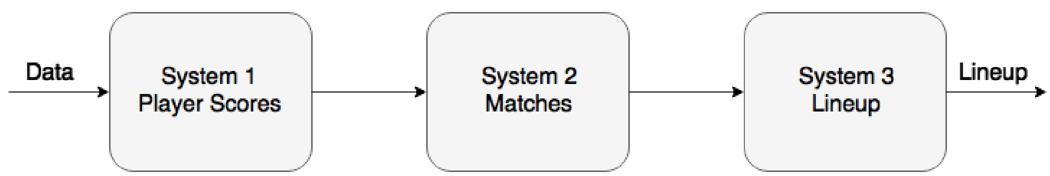

Figure 6: Redesigned System Architecture

These three systems will now be looked at in detail.

1. Calculating Player Scores
blah blah blah

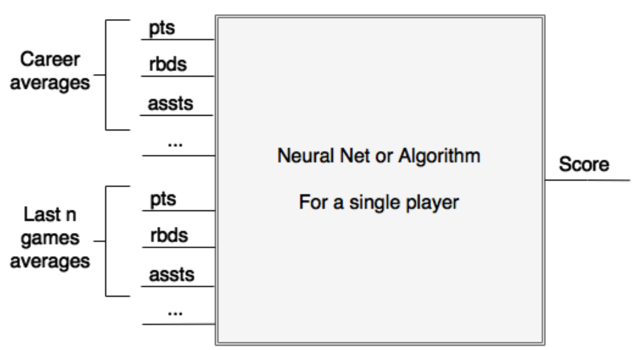

Figure 7: Player scores system

2. Matches Neural Network
blah blah blah

3. Choosing Lineups
blah blah blah

3.Designed simplified version of the problem for our first prototype
4.Result of this prototype

6 Future Plans

The design phase and data collection phase are mostly complete, but there is still a significant amount of building and testing that need to be completed before we have a full solution. What still needs to be done can be seen below

6.1 Improved Neural Net

As mentioned in our Design and Results section, our current neural network takes in as input the scores of the players who are playing in a match. It then outputs which team it thinks will win the match, with a probability attached to the output. Although this was the final solution of this term, after iterating on using score averages and unordered scores, it is still not sufficient to solve our problem. We need to be able to output "game scores" for each player, instead of which team wins. The final neural network should take in the player scores, and output the player game scores. The distinction is that the game scores are related to how well the net thinks that the player will actually play in that game, which is ultimately all that matters for the fantasy competitions. Once we have these game scores, we will then pass these to the third system, our lineup chooser. Another improvement we can make with respect to the neural network has to do with the NN parameters. These parameters include the learning rate, the amount of hidden layers, the amount of nodes in each hidden layer, and the amount of data used. There is a technique called "n-fold cross-validation" which we plan to use to optimize these parameters. It involves testing different sets of values of these parameters to determine the values that yield the highest success rate.

6.2 System 3: Lineup Chooser

Currently, we do not have a system that chooses lineups out of the players playing. After modifying the neural network to output the game scores, we will have a list of players ordered by how well we think they are going to perform tonight. With no constraints, we could simply take the highest scoring players. However, as mentioned in the Requirements section, fantasy basketball imposes constraints on the lineup. The first constraint is the budget of the lineup. Each player will have a cost, and our system will have to be able to generate a high-scoring lineup while keeping the total cost of the players under the total budget. The next constraint is that we have to select players with specific basketball positions. Thus, the system will also have to take in these positions and be able to generate lineups with it. Finally, as there will likely be many possible lineups with similar estimated scores, the system will need to have a random aspect to it that allows it to generate different lineups. A block diagram of this system can be seen in Figure X. This system will likely initially be implemented as a simple algorithm, but may have to be adapted, and could even end up being implemented as another neural network.

6.3 Player Playtime

One important statistic that we are missing from our system is the player playtime, i.e. how much the player plays each game. This is clearly an important factor in determining his score, since even if a player has a high per-unit-time score (e.g. 5 free throws per minute), if that player only plays for a minute, his score will be low. Although we have this data in the database, we do not use it. The reason for not using it is that it cannot (currently) be used as an input for the neural network, since we do not have a way to predict how long a player will play for in a game in the future. That is to say, we would not be able to use our neural network for an upcoming game, as we would have missing data. There are two approaches to remedy this. The first is to create a system that estimates the playtime for a player for a future game. This system could either be a simple algorithm, like "average the playtime that player has received over the past n games". Alternatively, an argument could be made that this system could be implemented with some sort of predictive machine learning. The second approach would be to use the playtimes in calculating the player's career score. Instead of adding playtime as an input to the neural net on a per-match basis, we would change the players' scores that are used as inputs for the neural net to be multiplied by how much they played last season, or during the last n matches, on average. This is perhaps too gross a simplification, since the playtime may change on a more match-per-match basis, so both of these approaches will have to be explored.

6.4 Testing

Once the above tasks have been completed, we will be at the testing stage of our neural network. We will first test it for games in the past, and see how our lineups would have done in contrived fantasy competitions. The testing process is detailed as follows:

If, after performing our tests, we see that there is an adequate (i.e. profitable) success rate, we will move to test it on real fantasy competitions. However, if we see that there is not an adequate success rate, or if we want to improve the already adequate success rate, we will have to make adjustments. We already have many ideas for adjustments that we can make, including the following:

- **Improve player scores system (system 1):** Currently, our first system takes a player's statistics and outputs a score representing how good that player is, "in a vacuum" (i.e. not against any particular team). The current equation we use for this is just the fantasy scoring equation. Although there is certainly a correlation between fantasy score, and how good a player is, it is likely not the most accurate relationship. For one, the fantasy score does not take advantage of all of the information we have, such as the player's age, shot accuracy, the player's win-loss record, and the player's plus-minus, which is an indicator of how well a team does as a whole, when the player is on the court. The new system could either take all of these factors into account, or could outsource to use someone else's equation/algorithm. An argument for making the latter decision is that we do not know more about basketball than the experts do, and they can likely come up with better numerical representations of how good a player is than we can. One argument against this decision is that the information is hard to obtain, and in order to train the neural network on past games, we would need to know the details of the algorithm to be able to calculate the score for these older games (as opposed to just getting the output).

- **Improve player playtime:** As mentioned in the Player Playtime subsection above, our initial solution to playtime will likely be to make the assumption that a player's playtime during the previous year is a good estimation of their future playtime. We also mention that a better solution could be designed that takes advantage of the playtime per game data. Moving to this second solution could help improve results.

- **Obtain more accurate positional data:** Currently, our source, NBA stats, gives us three posible positions: guard (G), center (C), and forward (F). The positions required for fantasy competitions are more specific, such as shooting guard (SG), or small forward (SF). Collecting this more precise positional data could lead to a higher success rate.

6.5 Timeline

7 Impact on Society and Environment

Whut. Only software product, nothing in the real world. Can be used to predict which player are best to add your team if system successful (If system can predict performance, it will mainly be because it was able to determine the correlation between the players). Need high processing power as system gets more complex.

7.1 Use of Non-Renewable Resources

Seeing as our entire project is made up of software components, the environmental impact is fairly minimal. Having said this, as the neural network grows in complexity, the processing power required to train and run it will grow as well. Consequently, the energy required to run the neural network may eventually become an environmental concern, depending on how much use it is getting. The use of the system - either within our team or by consumers should we decide to distribute the product - will be directly proportional to the energy use of the system. Seeing as most of the planet's energy is produced via non-renewable resources, this may be something to consider as the growth and use of the product continues to increase [1].

7.2 Environmental Benefits

If widespread use and adoption of our technology for picking fantasy teams occurs by the fantasy sports community, there is a potential to actually reduce the amount of energy being consumed by the fantasy sports community. Fantasy sports participants spend significant amounts of time online trying to figure out how to improve their fantasy teams; some spend several hours a day doing such research [3]. If our system gets adopted by a large enough portion of the people playing fantasy sports, this will remove all decision making from the hands of these participants, meaning that less time will be spent by them on their computers managing their fantasy teams. If our system is designed such as to minimize its energy usage, it is possible that the overall energy usage by the fantasy sports industry will go down.

7.3 Safety and Risk

The most significant risk associated with our project will be posed to the fantasy sports industry itself. By using machine learning to predict the outcomes of fantasy sports, fans may argue that it will remove the fun of the activity itself. If the project turns out to be rather successful, participants may also argue that machine learning is cheating the system in a way. Both of these things may cause participants to feel less compelled to play unless they also have access to machine learning techniques. Consequently, this may lead to a reduction in the number of people who participate in fantasy sports, thus negatively impacting the fantasy sports industry as a whole since it will not be doing as much business.

7.4 Benefits to Society

From an academic and research standpoint, the potential benefits of our project include expanding the list of domains to which machine learning techniques are applied. If our system is successful at predicting winners of fantasy sports, it may directly lead to being able to predict the outcomes of situations where the outcome is uncertain. Horse racing, for example, is something that is not far off from fantasy sports in many ways, and is certainly an area that could be explored following the completion of this project.

8 Teamwork Report

8.1 How We Worked as a Team

In order to facilitate and improve the efficiency of teamwork within our group, we made use of several tools and strategies throughout the semester. For communication within the group, we used the Slack messaging app. For collaboration on code, we made use of git for version control, and github for code reviews. Weekly in-person meetings were had amongst all team members in order to come with fresh project ideas as a group, and to ensure that all of us were on the same page. Weekly in-person meetings with our supervisor were also had for similar reasons, and to ensure that we stayed on track throughout the semester. Without the use of these tools and strategies, the development process for this project would have been much more difficult.

8.2 Individual Contributions

Below we show the individual contributions of each person. Note, however, that much work was done as a group, so there are many areas that were worked on collectively by everyone.

8.2.1 Ege

- Worked on designing the database schema for the database that would house all the data being collected for the project.
- Implemented the scrapers that would retrieve the player and match data from stats.nba.com.

8.2.2 Stephen

- Worked on algorithm that computes player scores, both historical and for a given game, and inserting given data into database.
- Performed data integrity analysis to ensure that all data received made sense, and if something was wrong, sought to find out what was wrong with is and what was missing.

8.2.3 Florence

- Built preliminary neural network that predicts which team will win a given basketball match.
- Worked on overall architecture design of three-part prediction system.

8.2.4 Asher

- Did everything, Asher pls help...

8.3 Evaluation of Teamwork and Difficulties

The four of us worked very well together as a team and have yet to have any problems thus far in the project. The number of difficulties faced this semester were minimal since we worked so well as a group. We hope to continue this same good work through next semester.

9 Conclusion

The first half of our design project has been a success. Although we have not yet created a full solution to the problem of picking winning lineups for daily fantasy basketball competitions with machine learning, we are on track to do so with enough time to test and reiterate on our design. We were able to gather a sufficient amount of data to train and test a first prototype that was able to predict the outcome of a basketball game. We also found that we were able to increase the success rate of this prototype by adjusting ML parameters, by increasing the amount of data used for training, and by altering the fixtures to be more meaningful. Our best version was able to successfully predict the outcome of NBA games with 71% accuracy. As the ultimate goal of our project is to be able to generate lineups, and not to predict the outcome of games, we still have work to do. We will have to change the NN to output the players' game scores, and then subsequently create a system to take in these game scores, the cost of each player, and the lineup budget, and output optimal lineups. After doing this, we will have to test our system, and improve certain aspects of it, such as how we obtain player scores. The end result will hopefully be a working and well-tested system that is able to generate profitable lineups for fantasy basketball competitions.

10 References

References

[1] "How much of U.S. energy consumption and electricity generation comes from renewable energy sources?", U.S. Energy Information Administration, 2017. [Online]. Available: https://www.eia.gov/tools/faqs/faq.php?id=92.

[2] Hunter, David Scott, et al. "Picking Winners Using Integer Programming." *ArXiv:1604.01455 [Stat.OT]*, 6 July 2016.

[3] YouGov. "How Much Time Do You Spend on Managing Your Fantasy Football Team?*." Statista - The Statistics Portal, Statista, www.statista.com/statistics/284718/amount-of-time-spent-on-managing-fantasy-football-team/, Accessed 30 Nov 2017

11 Appendix A

Put tf.py code