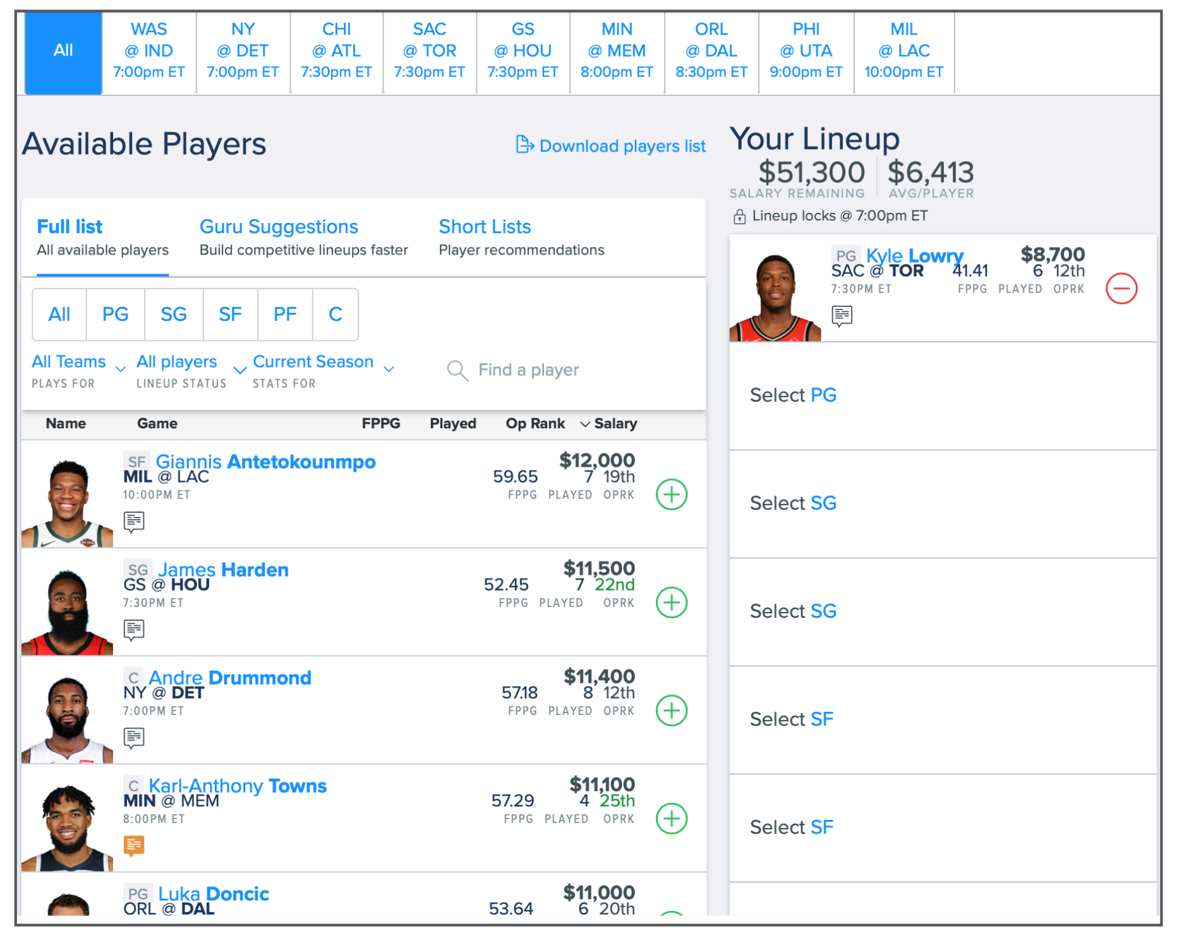
Optimizing Fantasy Basketball Lineups

Anis Ben Said – 911617154 Asher Wright – 913725516

Introduction & Problem Description

Online fantasy sports are played by around 60 million North Americans each year. It is a field ripe with opportunity for analytics, as there is an abundance of clean data, and more data is created every day as real athletes compete. This project targeted a subset of fantasy competitions: top-heavy daily fantasy NBA (basketball) competitions. An example competition can be seen in Figure 1.



All	WAS @ IND 7:00pm ET	NY @ DET 7:00pm ET	CHI @ ATL 7:30pm ET	SAC @ TOR 7:30pm ET	GS @ HOU 7:30pm ET	MIN @ MEM 8:00pm ET	ORL @ DAL 8:30pm ET	PHI @ UTA 9:00pm ET	MIL @ LAC 10:00pm ET

Available Players

Download players list

Full list — All available players
Guru Suggestions — Build competitive lineups faster
Short Lists — Player recommendations

All PG SG SF PF C

All Teams (PLAYS FOR) · All players (LINEUP STATUS) · Current Season (STATS FOR) · Find a player

Name	Game	FPPG	Played	Op Rank	Salary
SF Giannis Antetokounmpo	MIL @ LAC 10:00PM ET	59.65	7	19th	$12,000
SG James Harden	GS @ HOU 7:30PM ET	52.45	7	22nd	$11,500
C Andre Drummond	NY @ DET 7:00PM ET	57.18	8	12th	$11,400
C Karl-Anthony Towns	MIN @ MEM 8:00PM ET	57.29	4	25th	$11,100
PG Luka Doncic	ORL @ DAL	53.64	6	20th	$11,000

Your Lineup

$51,300 SALARY REMAINING $6,413 AVG/PLAYER

Lineup locks @ 7:00pm ET

PG Kyle Lowry SAC @ TOR 7:30PM ET 41.41 FPPG 6 PLAYED 12th OPRK $8,700

Select PG

Select SG

Select SG

Select SF

Select SF

Figure 1: Sample fantasy basketball competition

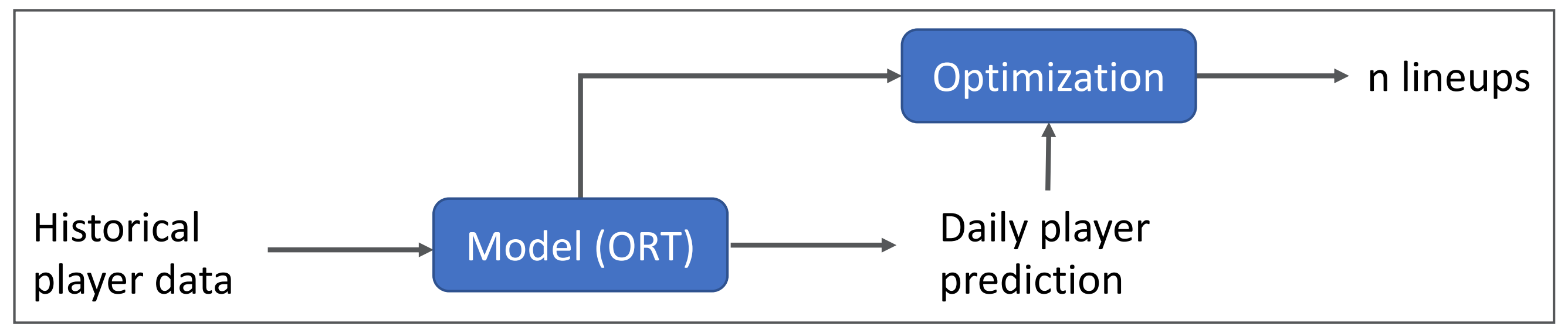
The top row shows the real NBA games happening on that night. The left column shows the real players in the games, along with a fictitious salary based on how many fantasy points they have scored in the past. Then, it is up to each competitor to select 8 of these players (adhering to positional constraints) such that the total summed salary is less than the total budget. Lineups are scored based on the players' performances in real life, and the competitors with the highest scoring lineups win. With these "top-heavy" competitions, most of the prize money is awarded to the top 10 competitors. When submitting many lineups, this means that only one needs to perform well in order to earn a profit. Thus, we aimed to produce many lineups, of which at least one was high-performing.

Data

We collected data from basketball-reference.com using the Python scraper basketball_reference_web_scraper[1]. We collected data for every player for every match since 2010 (178000 rows), and for every season (5000 rows). These data were used to train a model to predict the players' daily performances.

System Overview

This project was approached as a prediction and optimization problem. However, we aimed to use the prediction model while performing the optimization, different from "predict-then-optimize".



Historical player data

Model (ORT)

Daily player prediction

Optimization

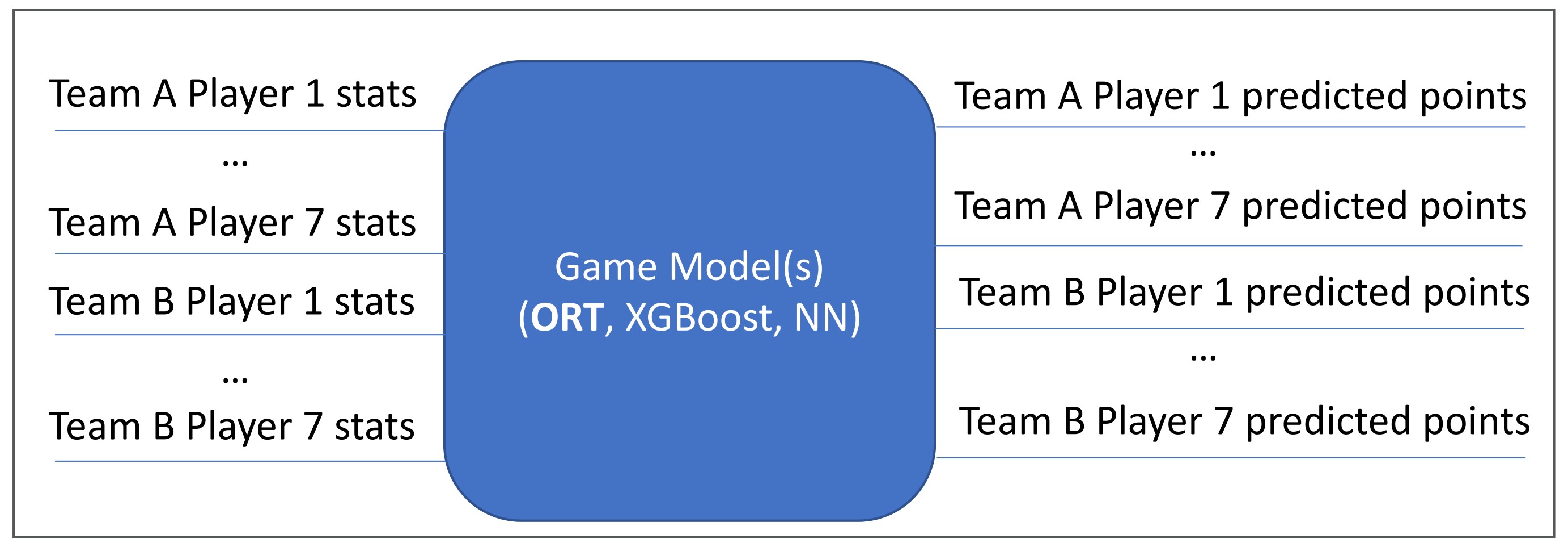
n lineups

Figure 2: High level system overview

Thus, we will explain both parts of the system: prediction, and optimization.

1. Prediction

The prediction model takes as input all of the information for a given NBA game. It has a list of players for both teams. For each of these players, it takes as input the player's recent stats (average of last 5 games), and their last season's stats. These stats include blocks, rebounds, steals, field goals made, etc., as all of these are important in fantasy points. Figure 2 shows the inputs and outputs for the models.



Team A Player 1 stats

…

Team A Player 7 stats

Team B Player 1 stats

…

Team B Player 7 stats

Game Model(s) (ORT, XGBoost, NN)

Team A Player 1 predicted points

…

Team A Player 7 predicted points

Team B Player 1 predicted points

…

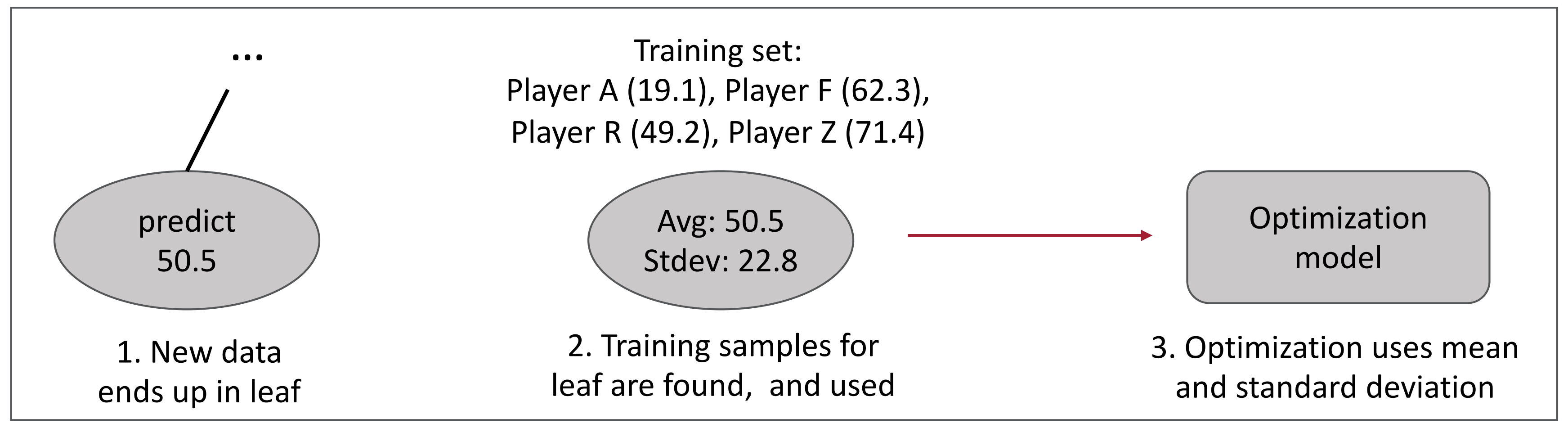
Team B Player 7 predicted points

Figure 3: Player fantasy points prediction model

An optimal regression tree (ORT) was chosen as it is a model that allows us to "decompose" predictions into which training data created the prediction (similarly to prescriptive trees)

2. Optimization

Rather than only using the ORT's point-predictions, we added pseudo "variance" to the predictions. To do this, for each new test sample, we examined the training data that was in the same leaf as the new sample. We then looked at the variance of the training data in that same leaf, and added this variance as an input to the optimization model.



…

predict 50.5

1. New data ends up in leaf

Training set: Player A (19.1), Player F (62.3), Player R (49.2), Player Z (71.4)

Avg: 50.5 Stdev: 22.8

2. Training samples for leaf are found, and used

Optimization model

3. Optimization uses mean and standard deviation

Figure 4: Using ORT model to compute pseudo "standard deviations"

Once the standard deviations and point predictions for each player are computed, the optimization model aims to choose lineups such that the *best* score across all of the lineups is the highest. To do this, we aim to choose high variance and high mean lineups. We also aim to reduce similarity across lineups. The idea is that having highly different high-variance lineups should maximize the chance that there is at least one exceptional lineup (which is all we need to be profitable). The base version of the optimization model is as follows:

Let B := the total budget
Let Pos := the set of positions = $\{PG, SG, SF, PF, C\}$
Let M_j := the number of players required in position j, $j \in Pos$
Let n := the number of players
Let m := the number of lineups desired
Let s_i := salary of player i, $i \in 1,\dots,n$
Let $p_{i,j}$:= 1 if player i has the position $j \in Pos$
Let μ_i := the predicted number of fantasy points of player i
Let σ_i := the predicted fantasy point variance of player i
Let γ := the maximum size of the intersection between two lineups
Decision variable z_i := 1 if player i is selected in the current lineup

We solve the following optimization problem *m* times

$$\max_z (1-\rho)\sum_{i=1}^{n}\mu_i z_i + \rho\sum_{i=1}^{n}\sigma_i z_i$$

Subject to

$$\sum_{i=1}^{n} p_{i,j} z_i = M_j, \forall j \in \{PG, SG, SF, PF, C\}$$

$$\sum_{i=1}^{n} s_i z_i \le B$$

Each time we solve, we store the lineups in matrix **x**. Then we add constraints to ensure lineups are different (k = current lineup to find).

$$\sum_{i=1}^{n} z_i x_{il} \le \gamma, \forall 1 \le l < k$$

We then resolve the model to get the k^{th} lineup

In addition to the system defined above (maximize score and variance), we tested a baseline version using point-predictions only, as well as one that used the variances to adjust the point-predictions, without using it directly in the optimization model.

Results

There are two parts of the system to test. First, the prediction model can be tested, and second, the overall lineups can be tested. Table 1 shows the mean absolute error (MAE) for each of the models used. With typical scores of 30, an MAE of 8 corresponds to being off by 25%, which is significant, and higher than predictions given by professional services. However, we believed that this tradeoff was worth it in order to have more than just point predictions.

Table 1: Performances of different models (predictions)

Model	MAE
XGBoost	8.00
Neural Network	8.35
Random Forest	8.18
ORT*	7.66

**The performance boost for ORTs is partially due to extra work on it vs. others*

Second, to score the lineups, we could not directly calculate the profitability, since we did not have historical FanDuel competition data. Instead, we defined a performance metric of how good our best lineup is relative to the best possible lineup. We empirically found that, when submitting 50 lineups, having at least one lineup's score 90% of the maximum leads to profitability. Table 2 shows the different systems we used and their metric scores.

Table 2: Performances of different systems (lineups)

System	Avg performance	Prob of > 90%	Prob of > 95%
Point predictions (baseline)	85 %	25 %	6 %
Maximize score and variance	84 %	14 %	7 %
Maximize adjusted score	85 %	28 %	7 %

One can see that on **average** we are unable to reach the 90% profitability mark in any of our three designs. However, some of our lineups hit 95% with a non negligible probability.

Conclusion

Using techniques similar to those in prescriptive analysis (using the underlying model when optimization) may be an effective way to gain an edge in fantasy sports, but the model and assumptions need to be accurate. Our prediction model wasn't accurate enough to yield a profitable system, and the heuristic for variance was not as helpful as we hoped. However, many improvements can be made, such as adding professional predictions as input to the model, and redefining the optimization model to better use the prediction model. Some improvements will be tested before the completion of the project, and will be included in the report. Finally, there are other ways to approach these competitions that have been shown to work in certain sports[2], and these could be adapted for the NBA.

References

[1] https://github.com/jaebradley/basketball_reference_web_scraper
[2] Hunter et. Al. *Picking Winners in Daily Fantasy Sports Using Integer Programming, 2016.*
[3] Bertsimas & Dunn. *Machine Learning Under a Modern Optimization Lens*, 2019. Dynamic Ideas LLC.