

**MMA 861**

**Analytical Decision Making**

**Professor Sumit Kunnumkal**

**Team Project**

**August 19th, 2018**

## Order of files:

|  |  |  |
| --- | --- | --- |
| **Filename** | **Pages** | **Comments and/or Instructions** |
| MMA 861 2019 Assignment 2\_Ossington.docx | 11 |  |

**Additional Comments:**

**Introduction:**

Daily Fantasy Sports (DFS) is a subset of fantasy sports which has been popular for many years. In DFS, DFS team manager would select athletes/players (decision variable) to create a team, the athletes’ actual statistical performance for the day would be used to compute the points/value score by each DFS team.

Every team is assigned a budget and must draft a complete team, while meeting all position requirements. The cost of each player is set by the DFS operator and this cannot be controlled by DFS team manager, team managers’ decisions are only related to which player to select for the team.

DFS is not considered gambling but a “games of skill”, as such one can argue that statistical analysis and modelling could facilitate in the DFS team manager’s decision-making process of selecting players while utilizing a model that would provide a relative advantage against other DFS team managers.

Team Ossington has decided to develop two decision models that would assist DFS team managers on selecting players to form a team. The first one is an optimization model while the second is a simulation model. Both models are created to facilitate the decision-making process for DFS team mangers.

For DFS there are two main processes. The first is estimating the expected value (expected points) of each athlete available for selection. The second is to create a team that will score the most points based on the projected value of each player while meeting the team constraints. Both tasks can operate more efficiently by employing the usage decision models.

For the first analysis, team Ossington has created a model for selecting the best team using an optimization model. The second model is an athlete-valuation simulation model, the model uses DFS manager’s expectation and quantifies the expected output by using statistical analysis. For the analysis we are using Major League Baseball (MLB) DFS competition, details about the models are as follows:

**Optimization Model**:

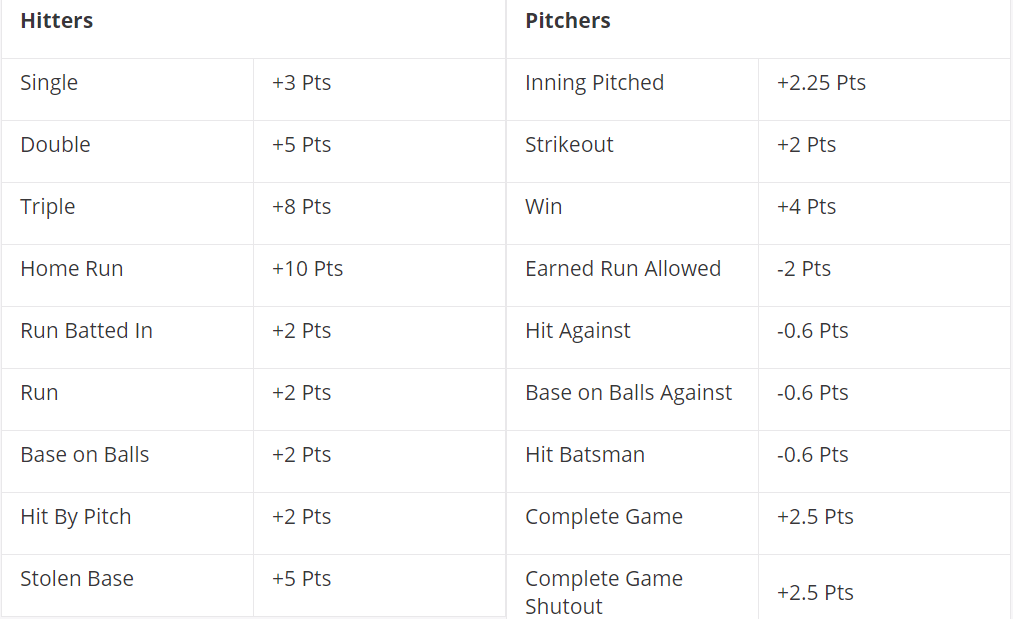
The main purpose of this model is to facilitate the formation of a team (10 players) while maximizing points earned. The decision variable for this model is which athletes to draft. The model does not forecast the athletes’ performance; however, it is needed as an input for the model to work.

There are certain restrictions for this model. First, the total players drafted must not exceed the allocated budget, for this model we used a budget of $50,000. Second, a complete team must be drafted while meeting the DFS league’s position requirement. The model assumes the following position requirement:

|  |  |
| --- | --- |
| **Position** | **# Required** |
| Pitchers | 2 |
| Back Catcher | 1 |
| First Base | 1 |
| Second Base | 1 |
| Third Bases | 1 |
| Short Stop | 1 |
| Out fielders | 3 |

Real data from Draft King, a DFS operator, was used to perform this optimization model. The analysis requires a list of all available players from the draft, in addition to three other pieces of information such as position, cost, and value.

When we talk about value, we are referring to the expected points an Athlete could generate. The actual value they generate could differ, as the actual points earned is dependent of their performance during the game. The points for each athlete is calculated based according to the table below:



Athlete’s value is the key input for the objective function to work. For this model, we used the value data provided by Draft King; however, value should be projected by the DFS team manager. The technique used to project athlete value could be qualitive or quantitative. The accuracy of the projected point value is not an important consideration for this model. The purpose of this model is to optimize the DFS team construction process assuming the players projected values are forecasted correctly.

Team Ossington assumes that the optimization model will do a better job in drafting an optimal team compared to the DFS team manager. This is because the amount of draft combination available is simply too overwhelming for a human mind to process. Assuming there are 100 players to choose from and with a roster of 10 slots, the possible draft combination is in the excess of 17,000 billion.

However, there is only a subset of combination that meets all the constraints. If the DFS team manager were to select a roster manually, most likely, he would select the most valuable player in their mind first and move on to the next best player that is within budget reach. This process would go on until the budget has been completely used. The outcome of a manual team selection process described would be budget maximization.

The objective of DFS is to maximize points scored and not maximizing budget utilization, as all DFS team managers have the same budget. The optimization model Team Ossington constructed would help solve this. It would select players that would maximize points rather than budget utilization for the team.

In addition, this process removes bias from roster selection. However, if we assumed that DFS team managers projected player points correctly without bias, there would still be a chance where DFS team mangers’ bias could impact the player selection process. The reason being is, they might not like a player because they have a weird pitch, they are too short, or simply there is a feeling that they are not just good enough. The DFS team manager would unconsciously shy away from selecting those players even if they are the best available player. Using an optimization model, players would get selected based on the statistical analysis of their performance and would not be impacted by unconscious bias.

This method is like the one HR departments uses. When HR department provides hiring managers with resumes with no name and only qualifications.

**Optimization Model: Limitation and Implementation challenges**

One limitation of the model is the decision variables in excel which are limited to 200. The dataset from Draft King has athletes list well beyond the 200 mark, as a result the model failed to operate correctly. To address this issue, DFS team manager can create a custom player list of 200 players. In actual practice, this is an exercise a DFS manager will have to go through, as there would be players that should not be on the draft list. This can occur when the player cannot play due to injuries, or simply because it is his turn to be in the bench. Moreover, it is also highly unlikely that one would go beyond the 20th player in your ranking for each position when creating your team. We assume the model will still operate effectively within this limitation.

Using the average point expectation from Draft King, the results showed that majority of our budget should be spent on pitchers. This analysis will change from sport to sport, the reason we saw this outcome was primarily driven by the scoring system in MLB DFS. This does highlight that in MLB DFS, DFS team managers should spent more time in estimating the correct value for pitchers.

**Athlete Performance Simulation:**

To properly utilized an optimization model, DFS team managers must project points for each Athlete. The way most managers do this is based on historical averages. Different managers would differ slightly in how they calculate averages; however, most managers rely on this methodology. The assumption is that athletes’ past performance would be a good indicator of future performance. If we approach this from a statistical point of view, there is a significant amount of adjustments to be made for more accurate prediction. For those of us that watch sports, we know that athlete performance is not always consistent. Some athletes would be more consistent than others, those players are more likely to have a lower standard deviation in their points scored. This information is useful to us, because it allows us to enhance our player point projection model.

At the end of the day, most DFS team managers feel there is still an art to this game. They feel that their experience should help predict if a player will have a good game or not. We can assume that DFS team managers can accurately predict which athletes will have either a good game or bad one. If they think a player will have a bad game, how much should they adjust down their points by?

If assume athlete’s performance is a random sample, under central limit theorem an athlete’s point should center around his mean. We can use their average points and standard deviation to calculate the upper bound and lower bound of their expected points. If we believe a player is going to have a bad game, we can give them their 5% distribution of the points earned, if we feel that they will have a great game we can assign them the 95% distribution of the points earned, and if we are unsure how they will perform, we can just give them their historical average.

The model assumes that an athlete’s point earned is correlated with winning and losing a match. This means that the distribution of an athlete’s performance can be impacted by the outcome of the match. Just say our typical DFS player is smarter than your average player, and he determines the likelihood of winning using historical average or Vegas odds. With this assumption, winning is not a binary event but a probability distribution. How would a DFS team manager quantify all those stats together? Let’s use Simulation then!

The above process describes one of the reasons why we have developed the simulation model. It allows DFS team managers to quantify their predictions. If we assume MLB Team A will win and athlete 1 will have a good game, we can now quantify the expected performance of the athlete. The user will assign a winning percentage to the player’s team, and their prediction if the player will have a good/bad/average game. Using that information, the model will quantify the expected points earn by the athlete. DFS team managers might feel that 95% distribution is too high, but this is does not matter as it is something they can adjust.

To build up the simulation, we used a binomial model to determine if a team will win or lose. The athlete’s performance is model with two set of distributions, which is dependent on the outcome of the binomial model. The mean and standard deviation input for winning/losing are also control by the team manager. They can adjust the distribution to factor other considerations. They might feel that recent games would be a better indicator, as such they could give a higher weight to more recent games.

The assumption here is that an athlete’s performance is related to the outcome of the match, and individual athlete’s performance are uncorrelated with each other. However, if this simulation were to be used for other sports we might reconsider correlation differently.

One outcome that was unexpected is that players points distribution is no longer normally distributed after factoring the likelihood of winning (with the assumption that winning and losing has a correlation with the player’s performance). Some distributions were hugely skewed, this might indicate other application for this model. It can be used to identify players to avoid, and players that are undervalued.

At first glance, one would assume that the skew is always negative when the player’s team has low projected winning percentage. However, this is not always the case, because there could be players who tend to perform better when the team is going to lose. It could be for many reasons, such as more playing time, or just better match up.

The probabilistic model might or might not work, but it is irrelevant given that the purpose of this model is to build a framework that gives DFS team managers a statistical method to quantify their view on an athletes’ performance.

The model is built in such a way that can allow for user to change their views. If a manager believes an athlete’s most recent performance is a better indicator of the points he will achieve, this will be reflected in the average they input into the average point. If they feel that the athlete’s performance is more based on team match up, they can calculate the average using only games against the current opponents. The binomial component can be switched to a prediction of whether an athlete will have a good game or bad game. We can use the historical point scored against the next opponent and see if it is above or below their mean, the total games that the athlete scored above their mean divided by total games played against this opponent will give a probability of having a good or bad game. We can then create a bimodal distribution of having a good or bad game using the calculation above. The real-life application of this model is endless.

**Athlete Performance Simulation - Limitation and implementation challenges:**

Using basketball as an example, you would assume that the better starter plays; the less minutes bench player gets, which would make their performance inversely correlated. For baseball this relationship does not exist for the most part, except for pitchers who is on the same team.

To handle this part, it is assumed that DFS user will adjust average points for pitchers to account for this or just ignore substitute pitcher from the draft list. The data used for the model were simulated, as there was a cost associated with obtaining real data. Team manager would have their own method of calculating average and would significantly change the results.

The results of the model were not tested against actual performance, as the output is largely dependent on the user’s view. Another limitation of this model is the usefulness of the model is highly dependent on how good a user’s input is. If DFS team manager are terrible at prediction an athletes’ performance this model will likely come out with garbage results. The model is only meant to provide a framework to assist DFS team manager in valuing the prediction.

In actual practice users who employ this model will likely take expected points from different part of the distribution. This can be done by using a predictive model to account for several different factors.

**Conclusion and future development**:

The optimization model has shown some advantage; however, the players list requires some analysis and scrutiny before being applied to the model. Even with the work required to build a custom draft list, this model will still be a life changer for most players. It is hard to imagine that a DFS team manger would be able to outperform the optimization model by using a manual approach.

The outcome of the athlete performance simulation model was better than expected, due to its wide range of application. When the model was constructed, we did not realize that the binomial decision could be tailored for different scenarios. As a result, you can use the simulation to model a wide range of predictions and quantifying the impact of the prediction.

In the long run it would be nice to be able to combine the two models. The output from the athletes’ performance simulation can be used for the optimization model as well.

**Technical Appendix Optimization Model:**

**Notation:**

PPi = Position Pitcher, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

PBCi = Position Back Catcher, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

PFBi = Position First Base, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

PSBi = Position Second Base, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

PTBi = Position Third Base, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

PSSi = Position Short Stop, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

POFi = Position Outfield, binary variable where 1 = true and 0 = false. i=1 to n where i = number of Athletes available for selection

PCosti = Cost of drafting Athlete i, where i = number of Athletes available for selection

PValuei = Project Player point Value for Athlete i, where i = number of Athletes available for selection

**Decision Variable**:

ASi = Athelete Selection is binary variable where 1 = Player Select, and 0 = Player Not selected. i = 1 to n where n = the number of players available for selection

**Objective:**

Max(AS1\* PValue1 + … +ASn\* PValuen = Total Points)

**Constraint:**

AS1\*PCost1 + … + ASn\* PCostn <= 50,000

AS1 + … + ASn = 10

PP1 + … + PPn = 2

PBC1 + … + PBCn = 1

PFB1 + … + PFBn = 1

PSB1 + … + PSBn = 1

PTB1 + … + PTBn = 2

PSS1 + … + PSSn = 1

POF1 + … + POFn = 3

**Athlete Performance Simulation Model:**

Team Winning = Binomial (1, Winning Percent)

If Team Wins then Normal Distribution (Winning Mean, Winning Std)

If Team Loose then Normal Distribution (Losing Distribution, Losing Std)

If the Player has a good Game = 95% Percentile Point Score

If the Player has as average game = 50% Percentile Point Score

If the Player has a bad game = 5% Percentile Point Score

