Section 1)

Joe Hallal (Main focus: The algorithm) :

- Repurposed a knapsack solution algorithm to create statistically optimized lineups based on projections.
- Created a greedy algorithm solution to ensure that cut down on runtime and shrank the problem size.
- Calculated correlations and implemented them in the algorithmic process to ensure high upside lineups of players from the same team.
- Tested algorithm to make sure it satisfied DraftKings player constraints and lineup rules. Also tested to ensure correlations were implemented correctly.
- Took on a role in data visualization and learned and used D3js to model fantasy players. Shows high upside players on a positional basis. Ensured it processes data dynamically.

Mihir Mankad

- Developed main idea and components of overall project, starting from basics such as its main goal
- Researched, found, and calculated necessary statistics and components that would be necessary to use in the algorithm
- Created a correlation value that takes into account player statistics and calculates a value that represents the strength of that correlation
- Created Web Like data visualization using D3js that represents a lineup of players along with some alternatives, and their impact on overall lineup
- Build Tracker application, which allows users to input their results history and learn how they can do better and what areas they excel in using pandas and plotting.

Eric Wendt

- Handled data scraping of players with projections using python
- Wrote database code for handling players/logins
- Created front-end web design/Angular code
- Helped Joe with algorithm
- Wrote SpringBoot code for backend framework

Section 2)

Non-technical overview:

Take the Lead Fantasy focuses on generating superior daily fantasy football lineups in an easy to understand fashion. It uses some of the industries best projections to create optimized lineups that have a high likelihood of being successful. However, we are not simply just finding

the highest projected lineups. One of our key features is being able to use correlations in our calculations. Each lineup will have at least one correlated player that is paired with the Quarterback of the lineup which allows for a greater upside. Our lineups can be generated on the frontend with an easy to use button and it does not require advanced knowledge to use. Take the Lead is geared towards novices in daily fantasy who want an easy-to-use way to get high value lineups.

In addition to calculating valid lineups, Take the Lead provides various tools that make Daily Fantasy Football easier to understand. The home page allows for manual lineup construction and separates players by position. It allows the user to interact with the entire player pool and make modifications to lineups. There is also a dynamic data visualization that helps explain why certain players are higher value than others. It grades players on a raw point and value basis and separates them by position. The tracker application allows users to submit their entry histories, and take a look at how they've been doing in particular areas. Take the Lead fantasy will give players the option to generate winning lineups in daily fantasy football.

Section 3)

Project website:
http://takethelead2.ngrok.io/

Components of website: D3 visualization, Automatic Lineup Generation, Manual Lineup Generation, Available player pool tables, Bankroll tracker, Login interface.

Team page with bios about us:
https://joehallal.github.io/TeamPage/

Section 4)
Packages used:

- Algorithm: Python pandas, Python numpy.
- Visualization: D3js
- Matplotlib python
- Java Spring
- Javax.json-api

Section 5)

Take the Lead uses innovative algorithmic solutions to generate user friendly lineups that are valid and have a high potential for success. Our process involves calculating a “core” of highly correlated players that using a greedy approach. This shrinks the problem size and then we use a solution to the knapsack problem to find the highest possible projected players under the constraints after that. This allows us to get the correlated players, highest projections, and maximum efficiency when calculating automatically generated lineups. The available players are

modeled in a player pool using D3js and provided to the user in a digestible way to understand why they are receiving certain players in their automatically generated lineups.

The backend framework for the website uses Spring Boot. This is coded in Java and provides a better interface for web design than pure servlets. The H2 database was used in conjunction with Spring Boot. This is an sql database, but sql commands were not used to make this happen. The front-end was done using AngularJS. This framework allows for easy parsing of lists, which is necessary for the lineup generation. The rest of the design was done using basic CSS and Bootstrap templates.

Section 6)

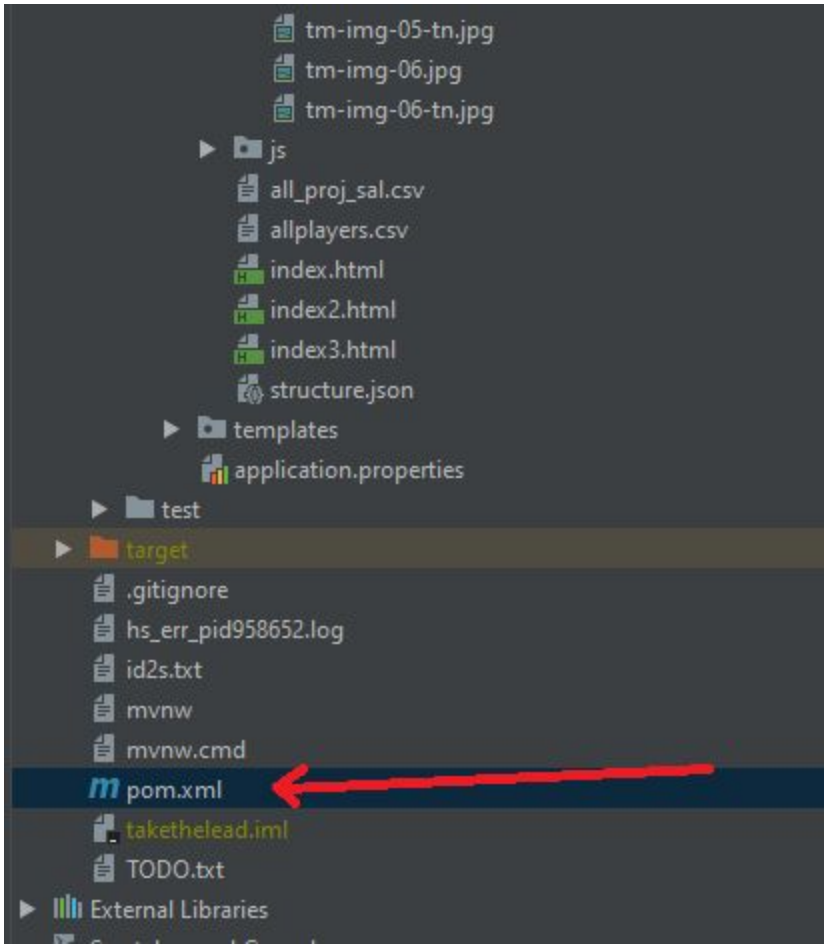
Joe Hallal:
I learned many things doing this project. Firstly, I had no experience with Python programming, pandas, or numpy and I learned a lot from using all of those. I learned many things about NP-complete problems and much of my time was spent trying and failing with various strategies. The process in designing the algorithm was mostly trial and error and in that procedure I learned of many other algorithms. The final solution I deployed ended up using a combination of strategies from different approaches and it was something that was both efficient and accurate. I also ended up learning about D3js to make the graphical representation of player projections. I started with a very basic visualization but progressed into something that looked professional and was eventually integrated into our final project. What I would do differently given different constraints would be to tackle the algorithm using an Integer programming solver. While pursuing a manual solution to the lineup building process was very educational and informative, if I truly wanted to create optimized lineups I would use Integer programming. This would provide me with a faster and more efficient way to test a larger portion of the solution sphere. There is also far more documentation available online about fantasy sports solutions using Integer programming. Luckily, if I ever wanted to explore this in my free time our algorithm is well abstracted and could be replaced with a different solver.

Mihir Mankad:
Throughout this project, I learned a lot about different packages and ways to do things. In the beginning, i had 0 experience with data visualizations, so i had to explore all of the different visualization packages that exist and try and figure out what would work best for our project. I also had to learn about NP-complete problems so that I could understand and work with our algorithm that generates the optimal lineups. There were essentially an infinite number of ways to approach these problems, so we had to use trial and error until we figured out an algorithm that worked well. Another thing I had to learn was how to apply an algorithm to a csv, as the tracker application needed to calculate information based on what was in the csv, and this csv would be different for every individual. For things that I would've done differently, I first off would’ve tried to find a better package for visualizations as D3Js is very complex and it took me way too much time to learn it and apply it to our project. Another thing i would do differently is try and brainstorm multiple ideas earlier, so i know exactly what i wanted to work on. The tracker

app was an idea that came very late in the process, so ideally in the future i'd like to have had the entire time to work on it. Lastly i would probably spend more time researching different APIs, instead of using the 1st one i found that worked decently well.

Eric Wendt:
The technical lesson that I learned while doing this project was definitely a painful one. I decided to use Spring Boot because I felt that it was a useful technology that showed up on many job applications. While this was helpful, I definitely wouldn't recommend it for any professional work. The framework is nicely laid out, but one of the main components to web-design, message passing, did not function nicely in the slightest. I spend an enormous amount of time trying to format json to be correctly sent from front-end to back-end and so forth. The explicit typing system of java makes this a genuine pain, and it impedes every aspect of the development cycle. I have done similar tasks in Python using Flask, and not having to make a builder object for every json object I want to parse is a godsend. There is no explicit typing, so message passing using Ajax was much easier.

Section 7)
Running the project
The first thing to do is to clone the repo. Then, open the pom.xml Maven file in intelliJ.



After this, the project files will automatically open in IntelliJ, and you can press the run button to start up the server. If you don't want to do this, this is all hosted on my server anyway, so you can go to http://takethelead2.ngrok.io/ to access the project.

Our project provides a fully functional way to generate valid and high upside Daily Fantasy Lineups. It will always generate a correlated lineup no matter the player information given, assuming there is a valid solution to begin with. One particular pitfall is there is no way to differentiate for the minor daily fantasy slates of players. The information imported is from fantasypros.com and will always import the full assortment of players. If you would like to play a contest of only the 8pm games and not the 4pm games then there is currently no way to eliminate players from the selection pool based on the imports.

As for next steps, one idea would be to branch out our application so that it works for every sport, and every daily fantasy sport contest. The algorithms would all be similar, but yet different because each sport acts much differently in DFS, and so a lot of work would need to be put in to generate lineups for all sports. Another group could try and make these new algorithms for every sport and then allow these lineups to be played anywhere. Another option would be to expand the website so that there are user interactions and discussions, where people can share and talk about lineups and strategies.