IE332 Project Phase 2 Report

Due: December 8th, 11:59pm EST

We certify that the submitted work does not violate any academic misconduct rules, and that it is solely our own work. By listing our names we acknowledge that any misconduct will result in appropriate consequences. Moreover, we have **read and understood the assignment instructions**.

"As a Boilermaker pursuing academic excellence, I pledge to be honest and true in all that I do. Accountable together - we are Purdue."

FULL name of each group member, as indicated in Brightspace:

1. Jack Demmy
2. Lauren Lamb
3. Ana Laura Sanchez
4. Jake Lim
5. Kabir Gupta
6. Valerie Baessa
7. Khush Garg
8. Ybyray Nagmanov

Passwords & Logins:

Database:
Login: g1117488
Password: team7

Group www Account:
Login: g1117488
Password: S7WsjX67

Website Login Examples(2):
UserOne
Login: ie332@gmail.com
Password:AB123456

UserTwo
Login: usertwo@gmail.com
Password:AAAA1111

URL To Website:
`https://web.ics.purdue.edu/~g1117488/signup.php`

Bonus Points:

1. When you receive or send a message there is an automated sound that goes through notifying the user they have received a message or their message has been sent successfully.

Contents

1 Introduction & Background

The project team was tasked with developing models capable of predicting National Hockey League player statistics. These models rank players at each hockey position and given and NHL season to predict rankings for, the model outputs a list of rankings. The team was also tasked with creating three value-based scores to indicate how valuable the player is with respect to the next player at that position, the average player at that position amongst all those ranked and how valuable the player is relative to the average player at each other position. The next major part of the assignment included evaluating the quality of a draft. This tool will generate a grade indicating how well each general manager performed. A report will be provided to the users that indicates the analysis that led to the grade, and which picks ended up being poor, good or neutral. This tool will also indicate the player at each draft position the general manager should have selected. In order to communicate these tools with the users, a web based utility will be designed that will allow the user to easily navigate through the tools described in order to meet their fantasy analysis needs. The website will allow the user to upload multiple drafts per season and get a detailed analysis back.

The main purpose of these tools is to provide our stakeholders with the best user experience of their fantasy leagues. With the large variety in skill set of our team, we were able to produce a creative final project while meeting all constraints presented.

2 Functionality

The website provides a user with an enjoyable fantasy experience. Upon arrival on the website the user is prompted to sign up or log in. Once their profile is created, they will not need to input information again. A user will only be prompted to login each time they access their profile. The login features were created in order to ensure a safe and private experience for our users. Team Fantasyland understands the value general managers place on ensuring all their information is kept private. Because of this privacy measures were taken extremely seriously. Pictured below in table 1 is the signup display and then the table 2 includes the login display.

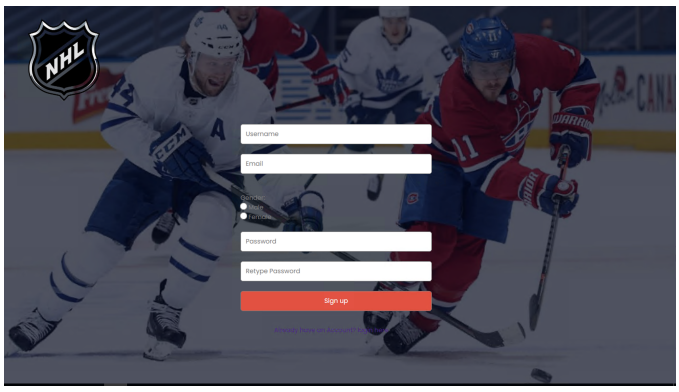


Table 1: User Signup Display

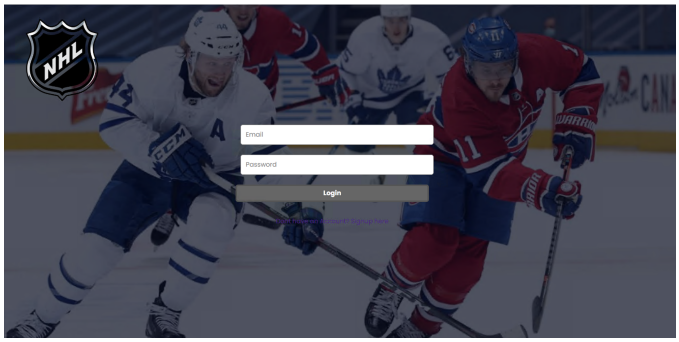


Table 2: User Login Display

Another possible issue the team took into consideration was the possibility of changing information. In response to teh constant changes in many people lives, if necessary, a user is able to edit their profile in settings. This will give them the opportunity to change their username, password, or even their name. An additional feature, the profile image was added in order to make the user experience more personal. A general manager has the opportunity to add/edit a profile picture if desired along with editing the rest of their profile as well. In table 3 the user settings display is pictured.

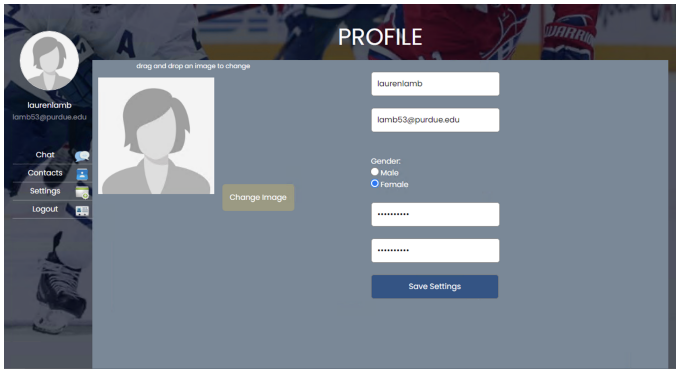


Table 3: User Settings Display

Also included in the users profile section is an interactive chat system. This system allows the general manager to connect with all general managers who are signed up for the website. This tool is great in connecting all general managers who share a common interest: fantasy hockey. On the chat interface, GMs are able to communicate with anyone, near or far. The system even includes automated audio depending on if you are receiving or sending a message. To ensure that a user is notified of any possible message, a bubble shows up on the profile image of the GM who has sent a message. In the case that a GM may also want to send a picture or share their draft scores, the chat system includes the option to attach any file desired. The goal of this system is to allow all GMs to communicate in a safe, but also visually appealing environment. Navigating through the side tab of the chat system, picture below in table 4, all of the GMs contacts are displayed under the contact tab(table 5). A GM can click on the name of the desired person they wish to communicate with. Once the name is clicked on they are redirected to a direct chat system. Where the GM can utilize all the listed features above. Below in table 4 is an example of the display of the chat system, changing depending on the user logged in. Table 5 shows the display of the contacts tab, all contacts are listed including their profile picture if they made one.

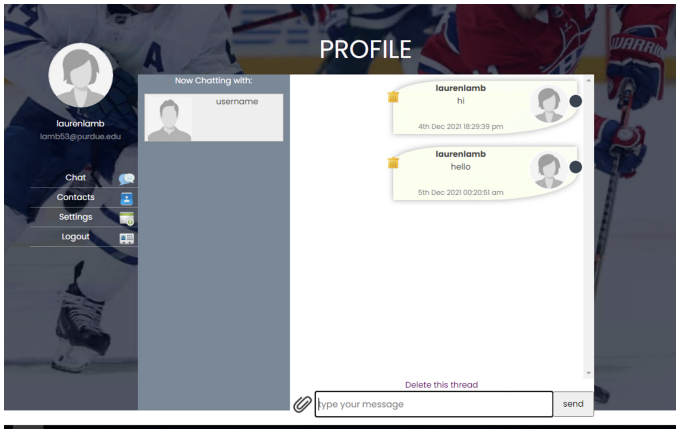


Table 4: User Chat System Display

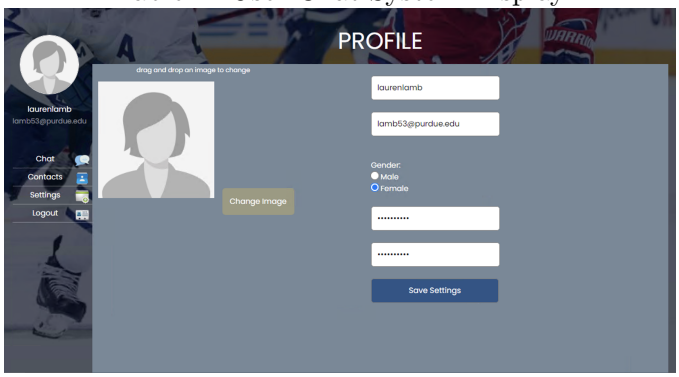


Table 5: User Contact System Display

Once access to the website is granted the user can navigate to multiple other features included in FantasyLand. The most important feature of the website is returning the user with their draft analysis. The user must first navigate to the upload tab seen on the top navigation bar. Under the upload, a user is prompted to upload their draft and asked how many teams are in the fantasy league, what pick they are, the date, and the season start and end. From there the users upload is taken into our system and run through our algorithm. Once ran through the heavily tested and accurate algorithm a score can then be output to the general manager.

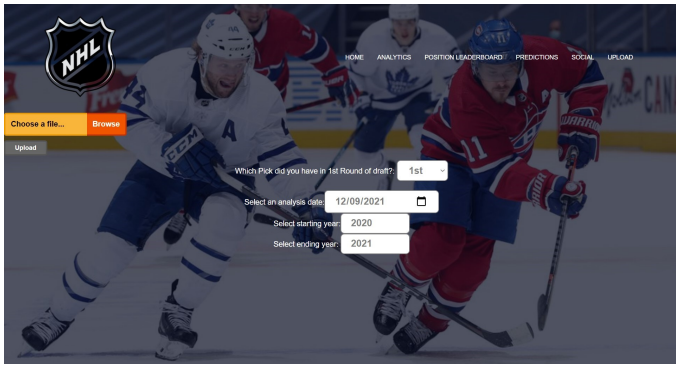


Table 6: Upload Display

When the algorithm is run, the user should navigate to the analytic page found on the top navigation bar. Under this page the GM is able to see why they did good, or bad or neither. This page includes three sub tabs. The navigation is pictured below in table 6. The first tab shows a user all their good picks.

They are able to see the name and a score for how well they performed. This same idea goes with what is presented under the neutral tab. Here, a user is able to see which of their pics performed neutral compared to the rest of the draft pics. Lastly, this slide outputs a table of the users bad pics. Included is (1) the player the general manager should have picked instead in order to perform better in the fantasy league. (2) The score of the player the general manager picked as well and (3) the score of the player they should have picked instead is also displayed so the general manager can perform an easy visual comparison. The GM is able to navigate easily through all of these tabs in order to view whatever analysis he wishes. There is no cap on how many season the GM uploads, the upload tab is for use however many times the GM desires. Each time a new report will be generated to giving a description to the GM.

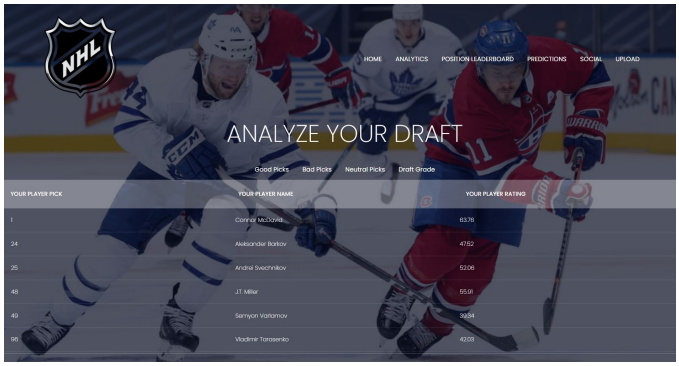


Table 7: Analytic Tab Display

Another feature included in Fantasyland is a report of all the picks in the draft, distinguished by position. This allows a user to separate the analysis by position and view a more overview display of the players in the draft. A user must navigate to the position leaderboard tab. Under this tab a user is presented with a sub navigation bar divided into the 5 positions: defense, rightwing, leftwing, center and goalie. The navigation is pictured below in table 7. Clicking on whichever position the GM desires, they can then see an detailed analysis of the position. An example for the left wings scoring is included in table 7. Included in this analysis are three value based scores calculated through Fantasylands algorithm. These scores include the players score compared to the next player at that position, their score compared to the average player at that position, and their score compared to players in all positions.

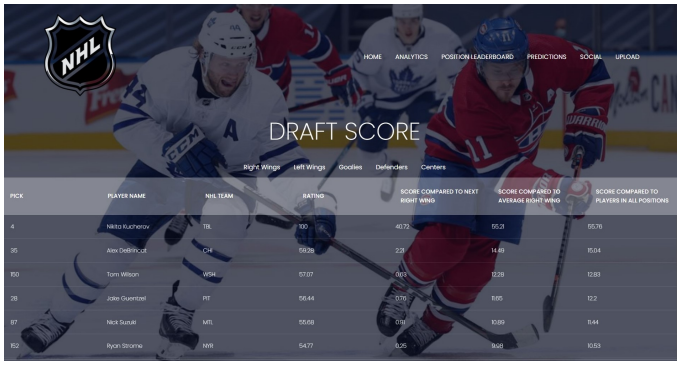


Table 8: Position Leaderboard Display

The last feature included in FantasyLand are the player statistics predictions for the full season. Based off of the uploaded draft, a prediction list is outputted of the players stats. This lists statistics included goals, assists, points per game, shots, hits, etc. Many more statistics are also included to inform the user with the best prediction found from Fantasylands signature prediction algorithm. Pictured below in table

9 is an example of the output the user would see whenn navigating through the page.

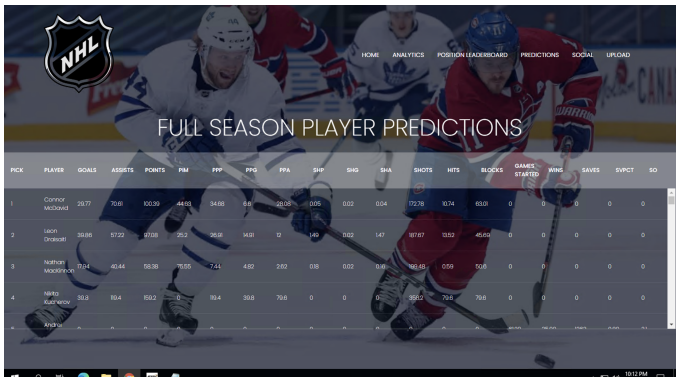


Table 9: Full Season Player Statistic Display

The goal of FantasyLand was to not only be an informational experience for the user, but to also grant the user with a social and personal experience. Through the tools of Fantasylands website a general manager is able to create an account a become a member of the fantasy league community.

3 Project Management

In order to successfully and accurately complete this project, the team divided the requirements into groups and assigned them to the group members based on their respective strengths and weakness. Throughout the course of this project the team faced many challenges, most notably communication difficulties. These difficulties ultimately resorted in complete restructuring of the group roles during the final two weeks of the project. Although faced with these challenges the team came together in order to complete the requirements to the best of their ability. Listed below are the team members respective roles in completion of this project.

Lauren Lamb	Web Interface Design
Jack Demmy	Database and Back end Design
Ana	Web Interface Design
Valerie	Database and Back end Design
Jake	Database and Back end Design
Khush	Back end Design
Kabir	Back end and Web Interface Design
Ray	No contributions

4 References

[1] Admin. (2021, November 15). Why is website security important?: Importance of web security. cWatch Blog. Retrieved December 8, 2021, from https://cwatch.comodo.com/blog/website-security/why-is-website-security-important/.

[2] Bailey, J. (2021, August 25). 11 website color schemes to help you find the perfect palette. 99designs. Retrieved December 8, 2021, from https://99designs.com/blog/tips/website-color-schemes/: :text=Blue3A20the20most20versatile20and,demands20attention20and20stands20out.

[3] Chiappa. (2017, January 12). How does red color in web design attract your visitors. Freepik Blog. Retrieved December 8, 2021, from https://www.freepik.com/blog/red-color-web-design-attract-visitors/: :text=It20highlights20the20page20while,widely20used20by20ecommerce20sites.

[4] Kerem Cigizoglu, H., and Özgür Kisi. "Methods to Improve the Neural Network Performance in Suspended Sediment Estimation." Journal of Hydrology, vol. 317, no. 3-4, 2006, pp. 221–238., https://doi.org/10.1016/j.jhydrol.2005.05.019.

[5] LightBoxLightBox 12155 bronze badges, Toni LeighToni Leigh 7, amp; Gaina Roman-Gaina Roman 2111 bronze badge. (1962, August 1). Is there an advantage to using the gray background web pattern? User Experience Stack Exchange. Retrieved December 8, 2021, from https://ux.stackexchange.com/questions/60815/is-there-an-advantage-to-using-the-gray-background-web-pattern: :text=Light20grey20is20therefore20a,Feint20textures20also20work20well.

[6] McNulty, K. (2020, February 20). Tidy web scraping in R-tutorial and resources. Medium. Retrieved September 22, 2021, from https://towardsdatascience.com/tidy-web-scraping-in-r-tutorial-and-resources-ac9f72b4fe47.

[7] Omgitsdomi. (2016, July 13). Measuring single game productivity: An introduction to game score. Hockey Graphs. Retrieved September 22, 2021, from https://hockey-graphs.com/2016/07/13/measuring-single-game-productivity-an-introduction-to-game-score/.

[8] Regular Expressions. Regular expressions. (n.d.). Retrieved December 8, 2021, from https://cran.r-project.org/web/packages/stringr/vignettes/regular-expressions.html.

[9] R-Session 11-Statistical Learning-Neural Networks (2015, April 27) - Youtube from https://www.youtube.com/watch?v=lTMqXSSjCvk.

[10] Tutorial: Web scraping in R with rvest. Dataquest. (2020, April 13). Retrieved September 22, 2021, from https://www.dataquest.io/blog/web-scraping-in-r-rvest/.

[11] What is user interface design? The Interaction Design Foundation. (n.d.). Retrieved December 8, 2021, from https://www.interaction-design.org/literature/topics/ui-design.

[12] World Leaders in Research-Based User Experience. (n.d.). Customization vs. personalization in the user experience. Nielsen Norman Group. Retrieved December 8, 2021, from https://www.nngroup.com/articles/customization-personalization/.

5 Appendices

5.1 Interface Design

Fantasyland interface designers aimed to create an interface which users find easy to use and pleasurable. From interactiondesign.com a variety of important points were learned about successfully designing a user interface (What is user interface design). Some of these points included that users value usability more than design, it is necessary to keep interfaces simple with resources that only serve a purpose to the user, and respect the users eye attention regarding layout; use proper alignment and draw attention to key features. It is important to pay attention to the task flow the user will be following in order to complete what they came to the website to do.

When brainstorming the website map Fantasyland's interface design the team made a list of what needed to be included. These main tasks included allowing a user to upload a draft and get an output of how well they did; what picks were good, bad, or neutral. The website also needed to include a way for general managers to communicate with each other. Using these main points the flow of the website was designed around using these as easily and user friendly as possible.

In interface design it is also important to gain a user's trust. Security was a very important factor in designing. Website security is important to protect our business, brand and reputation. In order to operate long term, security would have to play an important role in all aspects of the website (Admin) Fantasyland offers a secure way to navigate through the website. When first arriving at the website a user is prompted to sign up. This information is securely saved to the database. A user must then log in from there to gain access to all the tools provided on the rest of the website. This security protocol adds an extra step to the users navigation, but its payoff allows the user a personalized website. Once logged in, a user immediately has access to their own personalized chat system. They may chat with any member of the website. Only the user can view their chats, no other member can gain access to these chats. A feature of the chat system used to enhance experience is the ability to add a profile picture. This aspect plays a role in user customization. This can enhance user experience because it allows users to control their interaction(World Leaders in user based experience).

A common goal throughout Fantasyland's website was conveying our brand through our designs and flow ability between different pages. Branding is essential in recognizing our company Fantasyland. This also played a role in the point mentioned above that you can draw attention to important features through design. It was mentioned not to use colors or brightness excessively. For our website we made great use of grey. This is an unobtrusive, neutral color which does not glare, isn't too contrasting and doesn't clash with colors used in the design of the website(LightBox). As seen in the table 2 the background of our page features a gray picture. Most of the data displayed throughout the website is presented on top of the picture with slight transparency. The transparency is used to form contrast, and not have such a harsh block of color throughout the page. Without transparency the background does not showcase the text as well, the transparency helps the text pop of the background (Codrops). This also causes the feel of the website to be softer. Another important design decision of our website included the colors chosen for accent colors. Red and blue were what was chosen after researching how certain colors came off to the users. Blue is the most versatile and universally liked. It has been shown to inspire feelings of trust making it a favorite for website design (Bailey). Red was used to bring energy and power to the website. It is very successful in drawing attention to the different resources our website has to offer (Chiappa).

Flow of the website is another point that can be mentioned. A user navigates through the entire

website using a navigation bar at the top of the website. There are 6 possible tabs a user can explore to. These tabs allow clear and concise navigation to the different tools of the website. The division of these tabs is also logical. It was important to make sure the tabs included the correct data and it flowed together.

5.2 Interface Design Challenges

The top technical issue faced was taking our successful code and website design and formatting it in a clear, user friendly way. The goal was to make the website safe, easy to use, as well as display the data in the best way possible. Figuring out how to tweak the CSS code in order to display the data and information was challenging. In order to solve this problem the team watched many YouTube videos that targeted different design goals we wanted to achieve. Some of these goals included how to display and navigate through a table, how to format the message system, as well as how to navigate the user through the log in process. From these YouTube videos, as well as much trial and error over the past couple months our team was able to successfully display the website with the help of CSS how we thought would provide the best user experience. From this experience the main thing learned was how challenging styling a website can be. As we worked through these styling issues it was clear that the smallest accidental tweaks could alter the whole look of a website. The biggest takeaway from designing the css documents was making sure to clearly label

Another technical issue faced was connecting the database to the HTML website. This issue presented itself while developing the login process. At first we thought the database had connected, but we then realized only one user had been added and the website wasn't allowing anymore users to create themselves. Our first step in solving this problem was diving into the PHP code. We ran it through a PHP calculator and it communicated to us that our PHP code was showing no issues. After this conclusion, we looked more into the design of the database table. After running through all the attributes, it became clear that we had made the user ID incorrectly. We had forgotten to have the attribute auto increment. This caused there to only add one user to the system, because the ID was not incriminating and there was no "room" for more users. After changing this error in the database design we were able to successfully connect the database throughout the rest of the website design process.

5.3 Database Design

Point form explanation of our ERD:

- Users log into and utilize the website. Each user has a unique ID, a username, email, gender, password, date joined and can upload a profile image.
- A user can message another user using the chat system that has a id, mdgid, sender, receiver, message, files, date, seen, received, deleted_sender and deleted_receiver.
- A fantasy league has a league ID, an NHL season (year) the league occurs in, a password, the number of teams participating in the league, and a league name.
- A user can join zero or more fantasy leagues in a given NHL season and a fantasy league must contain multiple users (fantasy general managers)
- A fantasy hockey draft corresponds to one fantasy league and takes place during one particular NHL season. A draft can be organized by many fantasy leagues, but a fantasy league can only perform one fantasy draft in one season.
- Users participate in a draft for one or more of the fantasy leagues they play in and a particular fantasy draft hosts one or more participating users.
- Users own/manage one or more fantasy teams while a fantasy team is owned by one and only one general manager. A fantasy team competes in an NHL season and has a team name. A fantasy team belongs to only one fantasy league while fantasy leagues consist of one or more fantasy teams.
- An NHL hockey player has a unique player ID, a player name and a unique URL associated with their player web page on the website Hockeyreference.com
- A fantasy team selects one to many NHL players in their fantasy draft and players can be selected by zero to many fantasy teams.
- NHL players belong to one of two categories of players: skaters and goalies. Skaters and goalies each have differing statistics in a given NHL season.

Pictured Below is a Final ERD of our database design

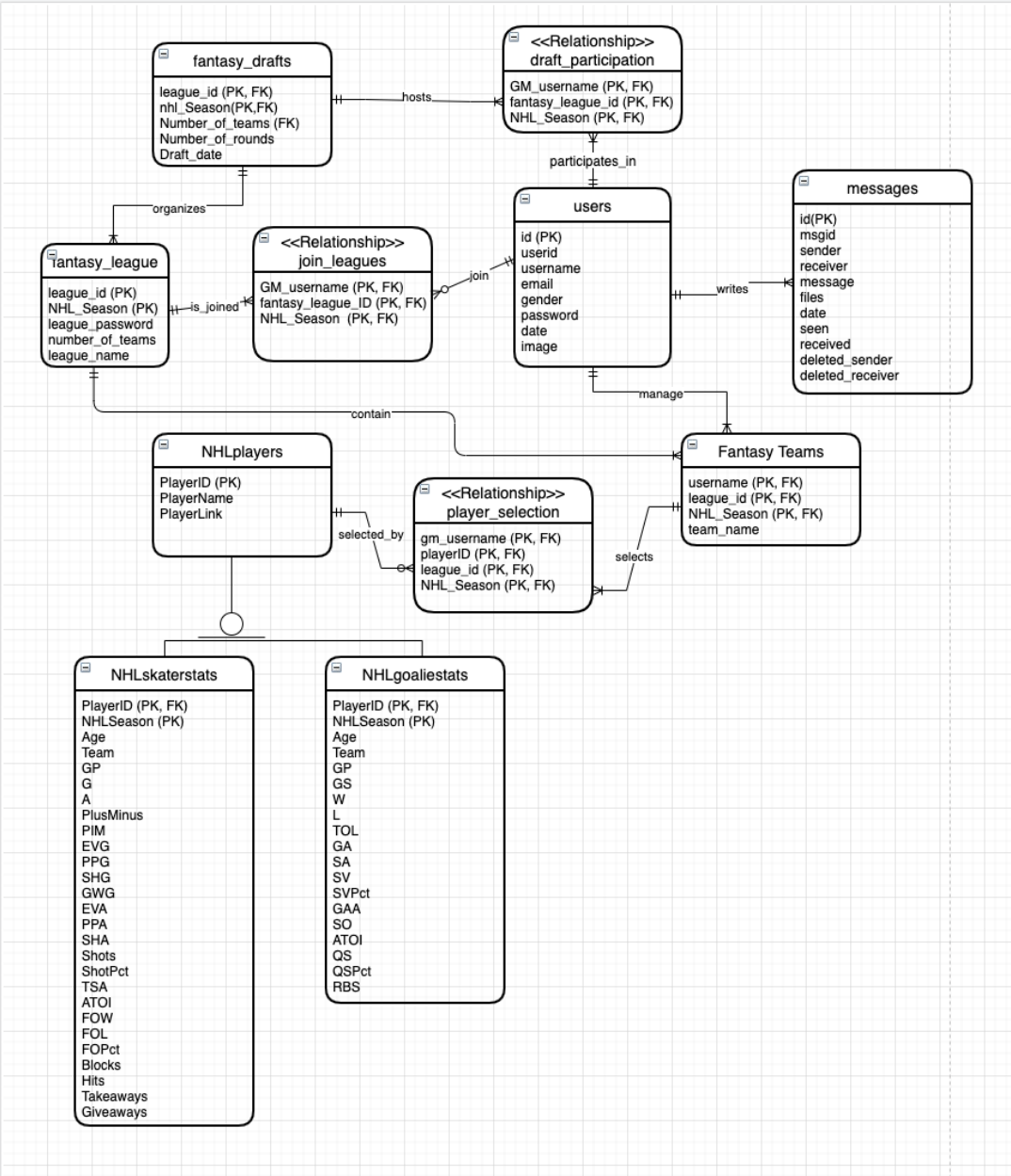


Table 9: ER Diagram

5.4 Database Design Challenges

1. A challenge encountered in the database design was how to store player data. In order to properly apply our player prediction and draft evaluation models, an NHL player's game by game statistics as well as their season statistics needed to be easily accessible. Since the group used player data scraped from Hockeyreference.com, it needed to find an efficient method for storing player data in our database so that the machine learning algorithms could quickly and properly pull the correct statistics for each NHL player in a given mock draft.

   Starting with storing game by game data for players, the group's approach to this problem was to scrape the data for each NHL season individually dating from the 2020-2021 to the 2007-2008 NHL seasons. Each row contained a player's statistics for an individual game in that season. Once the game by game data was properly stored for all NHL players in the 2007-2008 through the 2020-2021 seasons, there is no need to update these tables for skaters and goalies, respectively, in the database since these seasons have been completed and the data will never change.

   Storing game by game statistics for the current season (the 2021-2022 NHL season) was a more complicated challenge because new NHL games are being played each and every day meaning player statistics need to be constantly updated to ensure the information in the database was up to date. Updating game by game statistics for the current season was a challenge because manually re-running the R code used to scrape the data from Hockeyreference.com to our database would be inefficient and time consuming. Therefore, we created a cron job where the R code could be executed without manually being run in RStudio. Unfortunately, we were unable to run this cron job via our group account, so one group member agreed to add the cron job to his personal computer and resolve the issue.

   The need to update player statistics for the current season daily is why in the database there are four different tables for storing data. Since skaters and goalies play drastically different positions and therefore have differing scoring categories, the tables for skaters and goalies were separated. Within both skaters and goalies the tables for game by game data were split into two more tables, one for the current season (2021-2022 NHL season) and one for previous seasons. Once the 2021-2022 NHL regular season is concluded, the game by game statistics in this season can be pushed into the tables used for storing previous seasons, and the table used for the current season can be re-designated for the upcoming 2022-2023 season. This process can be repeated each year to allow our system to be applicable in the future.

   Storing player data season by season presented its own problems. The first attempt in storing this data was to store it season by season. Using this approach, season statistics for each player was scraped and stored for the current season, 2021-2022, then for the 2020-2021 season and so on. The group quickly realized the issue with this approach was it was difficult to match a player's data across different seasons. Just a player's name could not be used to find their season by season data because players can have the exact same name so it would be very plausible to extract the wrong data for a player if another player had the same name. To fix this issue, the approach to storing season by season stats was corrected by scraping the data player by player. This approach allowed for the gathering of all of an individual's historical data at once. Also, for each player, they are assigned a player ID to help better identify which player is which and which data belongs to which player. This method of storing season by season data by each player listed on the Hockeyreference.com optimized the ability to find the data of players in a mock draft.

2. Another challenge encountered in the database design was connecting the portions of the ERD associated with NHL players and the portions of the ERD associated with the fantasy general managers

(the users of the website). To the group, these appeared to be two different systems and as a result its first inclination was to create two separate ERDs, one modeling NHL players and the other modeling fantasy general managers and their drafts. After acquiring more knowledge on how ERDs work and database design in general, the group quickly realized creating separate ERDs makes no sense because having only one ERD for your system is optimal and helps in implementing your ERD into a single database.

In order to create one ERD for the database, the group had to find a method of connecting what was identified as two separate systems, that being the NHL players and fantasy general managers systems. It is sensible to do this because in an NHL fantasy hockey draft, it is the fantasy teams who select the NHL players to their teams. While this may appear to be a trivial or simple solution to this problem, given the team's lack of practice and experience, this was a hurdle we needed to clear.

3. Another challenge encountered in the database design was ensuring the primary keys in each table were both correct and enough to uniquely identify each row in the table. This is a common issue with database design and implementation, but ensuring proper primary key selection is critical for an efficient and usable database. The number of primary keys needed in a table differs depending on the entity and its attributes.

   In order to ensure the primary key selection was correct, each table needs to be

5.5 Predicting Player Statistics

There was a need for choosing the appropriate regression-based machine learning algorithm that could be used to predict numerical dependent variables. It was eventually decided to use neural networks for this problem and model the dependencies between the statistics for next season and statistics for the previous season. Neural networks was chosen for the following reasons:

1. It gave the team the opportunity to explore recurrent neural networks for time-series data analysis. There was a need for an algorithm which can handle sequential data. Recurrent Neural Networks allows for the output of particular layer to be saved and fed back into the input data.
2. The nature of the data makes it suitable for neural networks to handle it efficiently. The hidden layers in neural networks make it an ideal choice for accurately predicting statistics for next season.

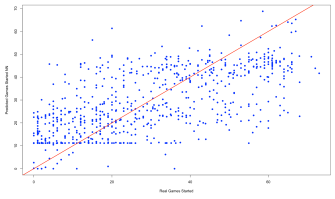
The model predicts GamesStarted, Wins, Saves, Save Percentage, and Shutouts for all goalies. Similarly, the model predicts Goals, Assists, TotalPoints, PenaltiesinMinutes, PowerplayGoals, PowerplayAssists, PowerplayPoints, ShorthandedPoints, ShorthandedGoals, ShorthandedAssists, ShotsonGoal, Hits, and Blocks for skaters. The prediction statistics were calculated in two different codes for goalies and skaters.

The prediction for goalies statistics was done in R under the file named 'Predictions_Goalies.R'. The data for each season were obtained from the database, and a new dataframe was created using the appropriate SQLQuery.Player ID was chosen as the variable that made it easy to join tables during data preparation . This was done to make sure that all the players were correctly matched with their stats when using the neural networks. As result, the data frame generates columns with the stats of the previous season followed by the stats of the next season for each player. After inner-joining seasons from 2011-2012 to 2021-2022, all the data frames were combined to creating a data frame that contained all seasons from 2011-2012 to 2021-2022. This was done to be able to input the entire data frame into the neural network function. All the data from the stats that are being predicted, except for save percentage for goalies, from season 2019-2020 and 2020-2021 were multiplied by 82/70 and 82/56 respectively. This was done since there were fewer games played in those two seasons due to Covid-19. In 2019-2020 only 70 games out of

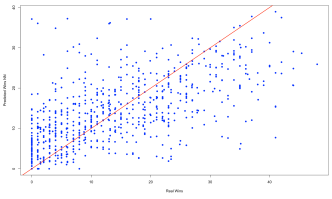
82 were played, and in 2020-2021, only 56 were played. Since there were fewer games played in the season, the overall statistics were less for each player. To account for those games that did not occur, the data was scaled. Since the Neural Network algorithm uses the previous season to predict the next, it would use 2020-2021 seasons data, which had fewer games played, to predict 2021-2022. Thus, it would create inaccuracy in the model. More about this will be discussed in the challenges section. Then all columns are converted to a numeric type and any NA value in the data is replaced by the median value of the column. For the Neural Network model to work the data has to be normalized with values between 0 and 1. This was done with the scale function in R with the center as the minimum value and the scale as the maximum value minus the minimum value, which normalizes the data. This is done from lines 187-195 in the code. Then from lines 198 to 200, the data is randomly separated into training and training data where 60 percent is training and 40 percent for testing. Then the neuralnet function in R from the neuralnet package to perform the Neural Network algorithm. In the first neuralnet function in line 203, games started column from the future season (starting with 2012-2013) is being predicted using age, games started, wins, save, save percentage, and shoutouts from the previous seasons (starting with 2011-2012) as the predicting variables. Then the second, third, fourth, and fifth neural networks use wins, saves, save percentage, and shoutout respectively as the response variable, using all the same predicting variables for all in lines 204-207 of the code. Then each neural network is entered into the predict function to find the prediction for each of the five variables from lines 211-215. Then the code from lines 219 to 223 converts the results for the normalized predicted statistics to the original form. Then a data frame is created with all the predicted variables, Player ID, and season as columns. The model predicted a few small negative values close to zero. These negative values were predicted for players that had an actual value zero or close to zero. The model predicted it would decrease but it is not possible to have a negative amount of games started, wins, or any other variable so the negative values were converted to zero in line 229.

The Root Mean Square Error was performed to understand the relationship between the predicting variables and response variables. The RMSE value can tell how well the model fits the dataset, as it gives the average distance between the actual and predicted values. The RMSE is obtained in lines 255-259 using the rmse function from the Metrics package. Then from lines 261-265 the RMSE value is normalized, to be easier to interpret, by dividing the RMSE value by the maximum minus minimum value. The normalized RMSE value for games started is .2063816, for wins, it is .1909623, for saves it is .2061579, for save percentage it is .1308724, and for shoutouts, it is .1694125. Values closer to zero represent better fitting models. All of the RMSE is below or close .2 which means the model fits the data well.

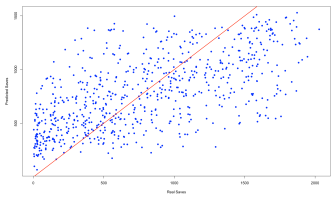
The predicted values for each variable were plotted against the actual dataset to visualize the behavior in lines 244 to 253.



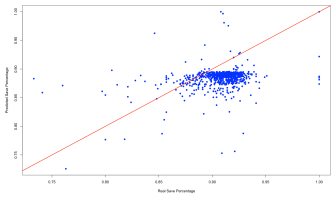
Goalies Games Started



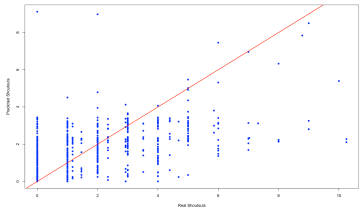
Goalies Wins



Goalies Saved



Goalies Save Percentage



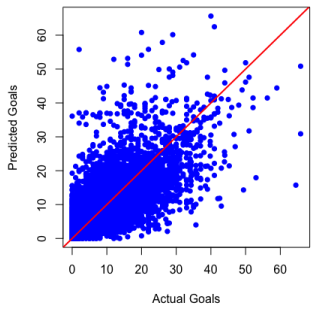
Goalies Shout-out

It is seen in the scatter plots that there is a positive correlation between predicted values for the variable and actual values. It forms a diagonal shape following the regressions line although the points do not lie extremely close to the regression line the plot does show a diagonal trend. None of the plots show a heteroscedastic shape which means that the variance error is constant across levels of the dependent variable. This means the model is just as good as predicting the best, average, and worst players. This in turn means that statistical significance is not overstated.

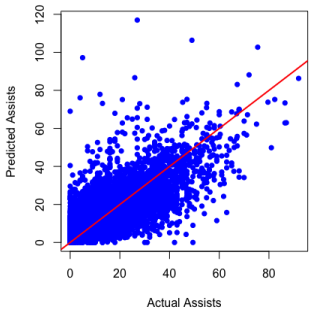
Prediction for skaters was done in R under the file named 'Predictions_Skaters.R'. For data import, which was done from lines 22 to 140, the same approach was taken as it was for goalies. First, the statistics from 2011 to 2021 were imported to R, then inner-joined with two successive seasons so that could be inputted in the neural network model. Since the dimension of skater statistics data frames for seasons from 2011 to 2021 was too big to handle multiple times and took many lines of code, a for loop was used in the code from line 219 to 276. To effectively use the for loop, the data frames for all seasons were combined in a list using the list function in R so that they could be extracted from the list with corresponding integers from one to ten. Inside the for loop, in lines 223 to 224 apply function was used to obtain the minimum and maximum value as similar to the approach for goalies. Then, in line 225, the data frame with skater statistics was normalized by scale function with the minimum and maximum value taken from previous lines. The normalized data frame was then divided into test and train sets in line 230 and 231. To be more specific, 60 percent of rows of normalized data frame was set as training data frame and the rest was set as testing data frame. With the training data set that was obtained in previous lines, a function called neuralnet was used for predicting. After the neuralnet was done, the function predict was used for actually predicting next year statistics for skaters given the current year's statistics. Although the output was generated from the task above, the predicted values were still needed to be denormalized in order for them to represent the real-life data. Thus, from line 259 to 269, the predicted data was denormalized column by column. Then, each predicted statistics column was combined into one data frame in line 271. After the for loop for the neural network process, all seasons' predicted statistics were combined into one big data frame in order to export them to the database more easily. Also, for consistency, data cleaning and column renaming were done from line 278 to 296. Then, the data was exported to the database server in order to integrate with other functions for fantasy drafts. After the exporting was successfully done, 305 to 324 in the code generated the data plot for comparison between actual and predicted values for skaters. Then, for accuracy testing, RMSE values for skater statistics were calculated from line 329 to 360 as they were in goalies statistical data. In order to successfully generate the RMSE values for all categories of statistics,

the code compared the relationship between predicting variables and response variables. Specifically, the RMSE values calculated are 0.0971980 for goals, 0.1002709 for assists, 0.0886482 for points, 0.0874529 for penalties in minutes, 0.0919754 for power play goals, 0.2954618 for short handed goals, 0.0795669 for power play assists, 0.3047277 for short handed assists, 0.1053022 for shots, 0.1035972 for blocks and 0.0990606 for hits. Although the RMSE values for short handed goals and short handed assists are a bit higher than RMSE values for other categories, they are still below 0.4, which means the actual data fit the prediction model decently well.

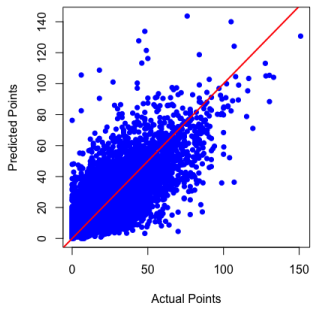
However, since the two successive data frames are combined using inner-joined by PlayerID, players who did not play for the previous season, so-called rookies, are not taken into account in the prediction model. However, the group was able to handle the rookies successfully in the evaluation model, using the actual statistics.



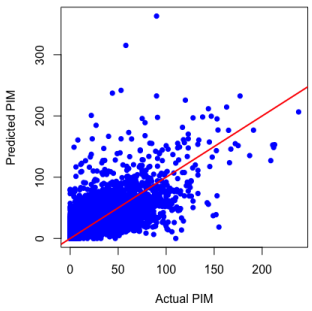
Skater Goals



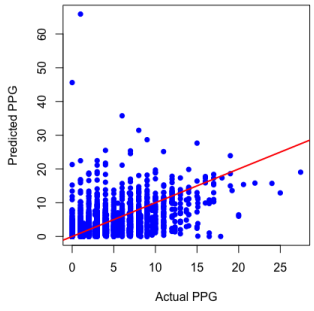
Skaters Assists



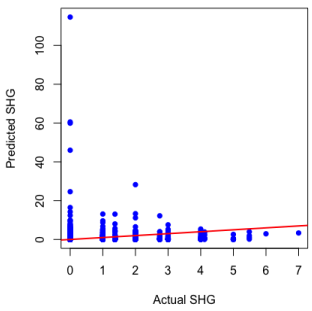
Skaters Points



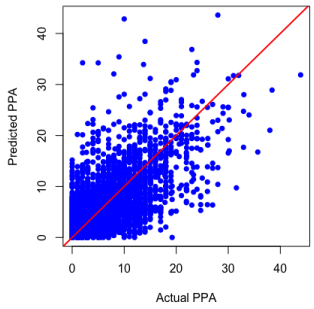
Skaters Penalties In Minutes



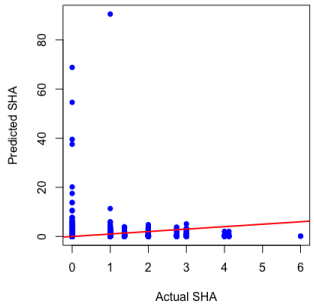
Skaters Power Play Goals



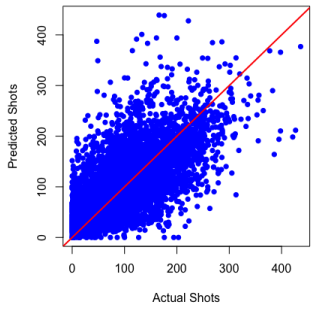
Skaters Short Handed Goals



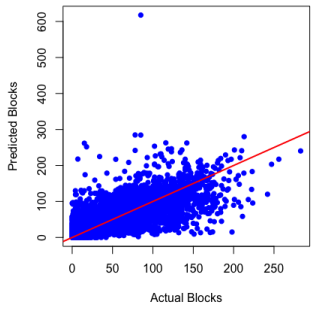
Skaters Power Play Assists



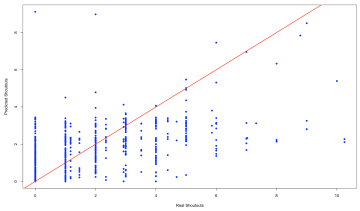
Skaters Short Handed Assists



Skaters Shots



Skaters Blocks



Goalies Shout-out

5.6 Predicting Player Statistics Challenges

1. One of the problems encountered while doing the prediction statistics was regarding the group member that was taking charge of it. The group member was Ybyray Nagmanov and he dropped out or disappeared with the prediction for the statistics and without giving any notice to the rest of the team members. He stopped answering before the Thanksgiving break and the group tried to contact him for more than a week without a response. He still has not made contact with the group. It was thought by the group that he had the entire prediction statistics completed but at the last minute, there was nothing for that part of the project since everything was on his computer. The group overcame this by assigning two people to work fully on the prediction statistics model for the remaining of the time(about a week and a half). The prediction was tackled by small subparts and was able to be combined and to work at the end. The group learned that big obstacles and surprises can happen at any time and the only way to resolve the issues is by working extremely hard and not giving up.

2. In regular NHL seasons, each NHL team plays 82 games, 41 home, and 41 away games. However, due to the pandemic, there were fewer games played in the 2019-2020 and 2020-2021 NHL seasons. Specifically,

for the 2019-2020 season, they played 70 games and for the 2020-2021 season, only 56 games were played in the entire season. Due to this special kind of situation, the predicted statistics for players were affected, and the accuracy of next season's predicted data was lower than it was for the regular season. Thus, the team came up with the idea of scaling, assuming that the players should have had as many opportunities to play games as they had in regular seasons. Thus, 82/70 and 82/56 were multiplied by statistical data in season 2019-2020 and 2020-2021, respectively. Line 152-157 in the 'Predictions_Goalies.R' file and lines 159-160 in the 'Predictions_ Skaters file.R'.

3. Another challenge was related to inputing the data to the neural network. To put the data into the neuralnet, the previous season's data had to be joined with the next season's data. To do the task, the team tried using the cbind function in r to combine the data frame. However, the number of rows in the previous season's statistics had to match with the number of rows in next year's statistics to use cbind. Also, the previous and next season's data that the team imported from the database were not in the same order. Thus, the team used inner-joined two seasons' data with a function called inner_join. Specifically, the team used PlayerID, which is one of the primary keys in the database. After implementing the inner_join function, the team could successfully generate the data frame with matching players for rows. Line 161-175 in the 'Predictions_Goalies.R' file and lines 167-176 in the 'Predictions_Skaters.R' file.

5.7 Player and Draft Evaluation

When evaluating an uploaded mock draft, the first thing needed is to match players in the mock draft with the players on the database. In order to achieve this, the following pseudo-code/process was used to match players in the draft with players in the database:

```
1. Grab a list of players from the database who have played in the NHL season the mock
draft is for.

2. Match the players in the draft with the queried players from the season of the
draft by matching the full name of the player and ensuring the NHL team names match
up. This helps to ensure that player's who are have identical names are not
improperly matched.

3. For players in the draft who do not find a match with the players in the
queried set, try instead to match players by just the last name. It is rather
common for the first name of a player to not exactly match between the mock
draft and the database of players. Last names, however, are much more unlikely
to differ, so matching the last name and NHL team of the player is a reliable manner
to match players who could not be matched in the previous step.

4. Find, if any, players in the mock draft who do not appear in the queried set of
players because they are either injured, suspended, etc.
```

The code which implements the pseudo-code above can be found in approximately lines 208-388 in the file DraftEvaluation.R'.

This process efficiently matches the players in a given mock draft with our players in the database. Once the players are found, their data then can be easily obtained from the MySQL database. Since the database contains game by game data in the regular season for every player dating back to the 2007-2008 NHL Season, the algorithm is used to determine which games in the current have been played up to the date to perform the analysis on, which is determined by the website user.

Once each player's current stats for the NHL season are calculated, the player's predicted statistics for the remainder of the season need to be calculated. In order to integrate current season and predicted season statistics, the following algorithm was utilized to determine how much of the current season stats and predicted season statistics for each player should be determined:

```
seasonStart = date regular season starts
seasonEnd = date regular season ends
daysInSeason = seasonStart - seasonEnd
predictedStatsWeight = seasonEnd - analysisDate / daysInSeason

playerSeasonStats = currentStats + predictedStats*predictedStatsWeight
```

The code for this algorithm can be found in approximately lines 376 - 447 in the file 'DraftEvaluation.R'. It makes sense to multiply the predicted stats by a certain proportion of the season because the analysis date occurs in at some point in the middle of the NHL regular season, so the predicted stats should only apply on the remainder of the NHL regular season. Now, the integration of current season stats and predicted stats

can be applied as inputs to our draft evaluation model. The approach to this model involved performing simulations of random mock drafts to determine player performance. To achieve this, randomly created drafts were made from any given mock draft. The goal of this was to see how players perform as they move across certain teams and to determine if they correlate with winning the draft. In the simulation, every team in the mock draft was matched up with every other team in the mock draft for all weeks in the regular season. For each head to head fantasy match-up in the simulation, the players on the team who won the match-up (meaning that team won more categories than the other team) would get a win added to their win tally. The categories used for the head to head match-ups are goals, assists, points, penalty in minutes, power-play points, power-play goals, power-play assists, short handed points, short handed goals, short handed assists, shots, hits and blocks for skaters. For goalies, the categories are games started, wins, saves, save percentage ans shutouts. Consequently, the players on the losing team would get a loss added to their loss tally. This computational algorithm relies on repeated random sampling on the same group of players over many weeks to obtain an accurate indication as to what players are valuable to general managers.

The codes used for the simulations can be found in approximately lines 488-598 in the file 'SimScript.R'. Using the win-loss ratios determined by the simulation, a player's value or score is calculated by dividing the amount of wins the player was a part of by the amount of losses the player was a part of. This simulation was performed twice, one for the 2019-2020 season and the 2020-2021 seasons. It made sense to perform multiple simulations not only just to acquire more simulated data but also to acquire data from two different seasons since player behavior and statistics can be different across different seasons and the model needs to be applicable across any season, not just the seasons used in the simulation.

The score obtained from these simulations was used as a response variable and the performance statistics of the two simulated seasons were used as the input variables for the neural network. The algorithm was trained on this data using the appropriate percentage as the training set and sixty percentage was selected as it gave the lowest RMSE Value. Lines 467-508 of the code in the file 'DraftEvaluation.R' represent the algorithm used.

The integrated current season and predicted remainder of season statistics was used as the input for the model to get a predicted score which was used to rank all the players. This score was used to sort players by positions and the mean was calculated for each position. Line 510-562 indicate the part of code in 'DraftEvaluation.R' which uses the predicted score to rank players across all positions.

Lines 564-586 indicate how we use the predicted score and the mean across all positions to create three new value-based scores that allow for comparison between a player and the average player at the same position and each other position. It also allows for comparison between a player and the next best player at the same position.

The main strength of the approach is using simulations allows for the direct comparison of skaters and goalies when determining their value in the perspective of a head-to-head category fantasy scoring system. Another strength is the algorithm will determine what percentage of the total season predicted stats will be used resulting in data that correctly represents the remainder of the season.

A weakness of the model is when using the predicted statistics, whether a specific player is currently on a hot streak or in a slump is not taken into account because the approach assumes players perform at the same productivity level throughout the entire season. This means our model is simplified and does not take into account in-season trends for the remainder of the season after the analysis date inputted by

the user. Another weakness of the approach is only two simulations were performed for two mock drafts, one for the 2019-2020 NHL season and the other for the 2020-2021 NHL season. The issue is not with performing only two simulations because the simulations were performed across two different seasons to see how well the model performed on different season. The issue, however, is with the selection of the seasons themselves. This was not realized by the group at the time of the simulation execution, but both the 2019-2020 and 2020-2021 NHL seasons were impacted by the ongoing coronavirus pandemic. While a normal NHL season contains 82 regular season games, the 2019-2020 season consisted of only 68-71 regular season games (depending on the particular NHL team) and the 2020-2021 season consisted of only 56 regular season games. As a result, since the model was developed using simulations performed on these two COVID-19 affected seasons, this could have created a model which will not perform as well on normal 82 game regular seasons like the ongoing 2021-2022 NHL season.

5.8 Player and Draft Evaluation Challenges

1. Easily the biggest challenge encountered in the draft evaluation model was comparing NHL skaters to NHL goalies in respect to their fantasy value. This was challenging because skaters and goalies play drastically different hockey positions. In general, a skater's job is to score goals while a goalie's job is to defend the goal. Therefore, the stats for these two categories of players is drastically different so comparing their value is inherently tricky. To workaround this issue, the simulations performed in the development of our machine learning model included both skaters and goalies, meaning the value of skaters and goalies in terms of head-to-head fantasy hockey is being performed simultaneously. In other words, the simulations allow for the accurate comparison of skaters to goalies.

2. Another challenge presented in the draft evaluation model was determining how to properly read in the draft file the user inputs to the website. The format of the file uploaded by the user is a ".txt" file so there needs to be a method for properly and efficiently reading in the draft file to allow the draft to be analyzed in the programming language R. Formatting the uploaded ".txt" file into a readable and usable data frame was more of a challenge than the group though it would be beforehand. The main issue is that the format of the file can be slightly different so there needs to be a way of reading any type of file in the same manner no matter what the ".txt" file is. To achieve this, regular expressions were used ('cran.r') to separate out the parts of the text file into a data frame. Using the regular expressions allows for the draft file to be represented as a data frame in R for all drafts uploaded. Properly reading the draft was the first stepping stone for the draft analysis.

3. Another challenge presented in the draft evaluation algorithm was how to determine what picks made by the fantasy general manager (the user of the website) were either good, neutral or poor. The group needed to come up with an algorithm/criteria for determining what categorized as good, neutral or poor. The following was what we decided to use in determining whether a pick was good, neutral or poor:

   Good: The player's score was greater than or equal to one standard deviation plus the mean of the scores of the group of players selected after him and before the next pick by the same team.

   Poor: The player's score was less than or equal to one standard deviation minus the mean of the scores of the group of players selected after him and before the next pick by the same team.

   Neutral: The player's score was above one standard deviation minus the mean and below one standard deviation plus the mean of the scores of the group of players selected after him and before the next pick by the same team.

   If a pick is deemed poor, the player the general manager should have drafted is found by finding the maximum score of the group of players selected after him and before the next pick by the same team.

   Another problem with evaluation picks comes with the nature of the snake draft process. With this drafting system, the team at the end of the current round gets to select first at the beginning of the next round. This presents a challenge with evaluating a pick when the next pick in the draft is made by the same general manager. To solve this issue, the group of players selected after the pick and before the next pick by the same team is the exact same for the two players, since they are both drafted by the same fantasy team. Each of the two players is essentially subjected to the same analysis and compared to the same group of players.

5.9 Testing

The most critical part of testing the system was to test the mechanics of the upload page named "form_2.php" in the WWW folder. When the user inputs a ".txt" draft file into the system, input validation occurs to

ensure the user inputs are not valid. When they are not valid, the user is prompted an error message and asked to re-try uploading. Once the user

Trial	File	User Team	NHL Season	Analysis Date	Pass/Fail
1	mock4.txt	1	2019-2021	12/09/2021	Pass
2	mock4.txt	1	2021-2020	12/09/2021	Pass
3	mock4.txt	1	2020-2021	07/17/2021	Pass
4	mock4.txt	4	2020-2021	12/09/2021	Pass
5	mock4.txt	5	2020-2021	03/10/2021	Pass
6	mock4.txt	7	2019-2020	02/04/2020	Pass
7	mock1.txt	1	2021-2022	12/02/2021	Pass
8	mock1.txt	1	2020-2021	12/02/2021	Pass
9	mock1.txt	1	2022-2022	12/09/2021	Pass
10	mock2.txt	3	2021-2022	06/14/2018	Pass
11	mock2.txt	4	2021-2022	10/17/2021	Pass
12	mock2.txt	1	2021-2022	10/11/2021	Pass
13	mock2.txt	1	2020-2021	01/21/2021	Pass
14	mock1.txt	14	2021-2022	01/09/2021	Fail

NOTE: The files "mock1.txt", "mock2.txt" and "mock4.txt" can all be found inside the WWW folder.

The following tests can be used to determine the functionality of the website. These tests were written with the intent of exploiting any possible bugs in our website to ensure when users type in inputs they cannot have the tool perform calculations it is not capable of. For the algorithm, the NHL Season must be in the format 'XXXX-YYYY' where 'XXXX' is one less than 'YYYY'. Trials 1,2 and 9 all test this and pass as an error message is prompted onto the screen informing the user they need to input a valid NHL season. It is also important to insure that the algorithm performs on the team the user wants to focus on. Trials 5, 6, 10 and 11 all test this and the algorithm performs the draft analysis for those teams in the draft. Testing the algorithm for analysis dates outside the scope of the given NHL season is also crucial. If the analysis date is before the date range of the given NHL season, the analysis should not be performed. Trial 10 checks for this and a message is displayed prompting the user to enter a valid season and date.

5.10 Testing Challenges

1. A main challenge concerned with testing was allowing for input validation. Having sophisticated input validation makes it less possible for errors to occur when the user works with your website. There was one input validation the group struggled with and that was allowing the user to input which team they want to focus on. The initial vision was for the user to first select the number of teams in their draft and then have a drop-down menu prompting them to select which spot in the first round they drafted to find their team name. The group was unable to fully implement this and instead there is one drop-down menu with the message prompt "Which Pick did you have in 1st Round of draft?". In this drop-down, you can select from the 1st to the 14th pick in the draft. The drawback with this is that not all drafts have the same number of teams. This drop-down menu would be ideal if all drafts inputted into our website had only fourteen teams but obviously this is not reality. A draft can have eight, ten, twelve or sixteen teams. This is why our 14th trial fails because the user says they picked in the fourteenth position in the 1st round but the number of teams in "mock1.txt" is only twelve. Since our system cannot detect this and report this back to the user, our tool fails for this test and given more time modifications can be made to fix this error.
2. Another challenge encountered with testing concerns working with the analysis date the user inputs. In order to make our system less prone to error, the group wanted to compare the analysis date with the NHL season given to make sure the analysis date was compatible with the NHL season uploaded to the website. To accomplish this, the format of the analysis date needed to be changed from string to time and allowed us to compare withe the NHL season the user inputs allowing us to check if the analysis date does fall within the season. Overcoming the challenge of working with the analysis date in PHP and adding this constraint to the system makes our tool more user-friendly and less error-prone.

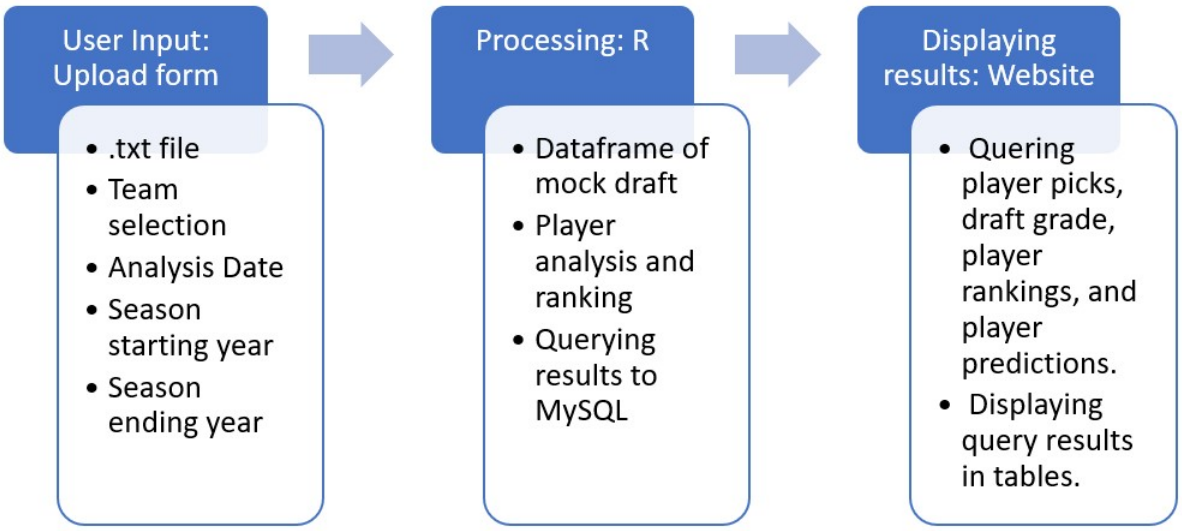
5.11 Queries and Visualization

Fantasyland includes many visualization tables that include trivial queries in order to display the data. Table 5 shows an example of the display of many of the tables. In order to output this data to the user, the query below is used.

```
$query = "SELECT * FROM centersRanked";
```

The table the information is being pulled from was changed based on which position or analysis was being displayed to the user. This trivial query was able to display the majority of our output for our algorithm to our users. This made the query very dependable because it was the same across many different php documents. The tables used throughout the website included a display of all the draft picks by their position: center, left wing, right wing, defense and goalie. Separated into their own individual tables, because they have their own evaluation based on position. Similar tables were also used to display and individuals managers draft evaluation. The analysis included three sub tabs divided into the managers good picks, bad picks and neutral picks. These tables use the draft uploaded from the manager which then runs through our teams algorithm. The results of the algorithm are saved to the database. The tables then query this analysis from the database and output the results to the general manager. This is the best way to display the results because it allows the manager to filter through all their individual picks as well as filter through the draft as a whole. It doesn't make the navigation challenging but instead only allows the manager to see exactly what needed based on the algorithm analysis.

The upload form was necessary for the general managers to submit and receive an analysis for their drafts.



The file for the upload was called "form_2.php" and contained an HTML form which calls "draftupload3.php" that took the inputs from the form and executed the R script "DraftEvaluation.R" for analysis. "form_2.php" line 39 took the file from the user. Lines 46 to 61 took in the user's choice for their pick in the first round of the draft. The type of the option was chosen to be a number as it was easier to pass to R as an argument relative to text such as "Team 1" which was not transmitting completely. In line 63, the line for taking the analysis date was written. This had to be a date type as getting the correct format for the date was important for passing into the R script. The date type is also more intuitive for the user and prevents errors. Line 64 and 65 took in the starting and ending years of the chosen season and were of number type as it was easiest way of passing it into the R script. The goal was to send clear inputs to the R script for processing. The R script "DraftEvaluation.R" was used to analyze the mock drafts and player performances. After the analysis was completed, tables
A more nontrivial use of the queries was implemented in the interactive user chat system.

This query below is used in the messaging system. It is used to display the messages from a certain user. It makes sure that the sender and the receiver is the person logged in to the system. This is important, because it makes sure that data and messages from other users are not being displayed to other general managers that are logged into the system.

```
$sql = "select * from messages where (sender = :userid || receiver = :userid) group by
msgid order by id desc limit 10";
```

This query below is used in the settings php. It assists in updated the database when a user changes their settings on their profile page. When a user changes their information on their page, the query sends the new data back to the database and changes it for the specific user. It is important in also making sure that it updates it for the correct user id. If this was done wrong it could begin updating different data for the whole website which could lead to users not being able to log in to their webpage.

```
$query = "update users set username = :username,gender = :gender,email = :email,
password = :password where userid = :userid limit 1";
```

5.12 Queries and Visualization Challenges

One of the main challenges came with getting comfortable with queries and their syntax. Many times in the beginning, the tables would be queried, but the table would fill with empty values. Something small was off with the connection that was causing the data to not display correctly. The first step taken to ensure our team was completing this portion correctly was checking to make sure the connection was working and secure. We understood that if the website was not connected to the database in the first place, data would not be visible. We were successful getting the database connected by making a single PHP file that the web pages referenced, this made it so it wasn't subject to change between documents. The next step came in making sure the actually queries were correct. One thing we learned was that the names have to be identical, capital letters and all.

5.13 Unsuccessful Attempts

1. There were many challenges involved with player evaluation. This was initially treated as a classification problem where all skaters and goalies were clustered into three categories: Poor, Average and Good. All NHL players were classified into these three categories for each position. It was assumed that k-means clustering would be helpful in determining which performance statistics were associated with being a good, average, and poor player for each position. Random Forest was used as the supervised machine learning algorithm to accurately predict the category for each player. Linear weightage was implemented between the variables that accounted for the maximum variance to derive a value for each player. A different approach for evaluation of the players was needed as this method had some serious issues regarding our assumptions. The file RandomForest.R displays this approach that was applied across all positions to get a value for each player

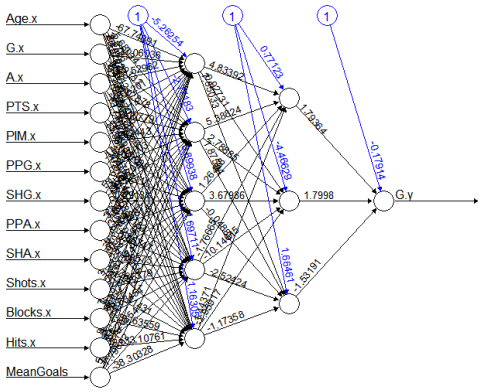
2. Another challenge encountered when creating the website was being able to have GM's request access to other GM's reports and scores. There were multiple reasons why this porting of the website was so challenging for the team. When adding a friend's system in PHP and MySQL the group was able to successfully create a user's and a relations table in the database and query the database to send request to other users, but this did not let other users access the data of other GMs just yet. The team proceeded to figure out a way to keep querying the database to be able to display other GM's reports but due to many failed attempts, poor team organization, there was not enough time to achieve this task in the end.

3. Attempts were made to input more variables into the neural network and create more accurate predictions for these players into the future(2021 onwards). A time series is a collection of data obtained by observing a response variable over time. The historical performance of a player can be treated as a time-series data where each year is treated as a state and the next year's prediction is based on all previous season data. It was recognized that there was a need to explore and analyze the data of each player over the various seasons he has played. There were challenges in finding the appropriate deep learning networks to achieve the following as it required the observations to be over regular and continuous intervals. Long Short Term Memory Network was considered but could not be implemented as irregularity was observed in the number of timestamps for each player in any particular season. Some players had played more seasons than others. One solution to this approach would have been treating each player in a particular season as a separate time series, but that would have been computationally difficult to achieve.

Simple forecasting methods like incorporating the mean of each statistic over the past seasons to get a better predicted statistic was considered and implemented. The drawback of using this method is that it makes use of every piece of information equally to make a forecast. This shall be problematic for accurate predictions as the performance of a player is not linearly correlated with time. Some players play at a better age and some don't. Another solution to this problem would involve using a metric which computes the average of the most recent observations to generate a more accurate prediction for the next season.

This algorithm was only implemented on players playing from 2015-2016 season till 2020-2021 season. We took two seasons for comparison at once, for example, 2018-2019 and 2019-2020 and extracted the common players playing in these two seasons. The historical statistics for each category for each player were extracted from the database by using the appropriate queries. The data was structured in such a way that performance statistics for a player for the previous season were combined to the data frame containing calculated mean for each statistic from all previous seasons. The dataset was aligned again to have the real statistics for the next season. The resultant dataset was used as the input data for the neural network. "PlayerID" was the variable chosen to sort the data conveniently and get the desired output. With the

help of a for loop, we could iterate the same code for the last five years we had chosen. Lines 14-305 in AttemptsUnsuccessful1.R indicate the restructuring of the dataset



Predicting statistics for 2019-2020 season using 2018-2019

The above neural network shows the neural network that has been used to predict the goals for season 2019-2020 using the performance statistics from the previous season 2018-2019 and the variable 'Mean-Goals' indicates the mean of all historical goals the player has scored.This was done similarly for all statistics where we took the mean of all statistics as an additional input to the original neural network.