**Project Final Deliverable:**

1. **Project title and description**

NFL Combine Individual Forecasting; our project is a data analysis. Our main goal was to ease the analysis of past NFL Combine participants’ event test results and measurables by position to give the user insight on how they compare with their own results and measurables. We believe that our project would be essential to the thousands of high school and college football players who dream of going to the National Football League by giving them a visual and information that can not only show what the people who tend to get drafted are capable of but also where they fall in comparison.

1. **Team members and roles**

Humdan Ahmed: ***Tester*** *– Determines whether the code meets the requirements. Identifies what parts of the requirements are satisfied, and what parts are not yet satisfied. Makes sure that new code doesn’t break the old functionality – that the system still works.* ***(humdan.ja@gmail.com)***

Naaja Jacob: ***Project manager*** *– keeps track of individual task commitments, due dates and status (complete, in-progress, overdue, etc.). Reminds (and sometimes nudges) team members on their tasks. Reports weekly team update. Needs to be organized, a good communicator and willing to “push” people a bit.****/Analyst-****Created a description and/or diagram of the components (modules, functions, etc.), and how they fit together. Makes sure that all the code produced by the individual coder fits together****/(naajajacob@gmail.com)***

Dimeji Olayinka: ***Researcher*** *– Gathers the data or information needed. For example, gets sample data from clients, finds Python functions or modules that do specific functions needed by the team, etc.****/Analyst/*** *– Leads team members in analyzing the problem, breaking it down into sub-problems, identifying system components****/ (98olayinkadimeji@gmail.com)***

1. **Summary of accomplishments**

Our code allows the user to look at tendencies and patterns amongst the even test results by position. The user can see the any positions spread of event scores through a pie chart. User also can import their own event test results and measurables. The user can the see there measurables and results compared to other actual combine participants in the data set on a scatter plot.

1. **Summary of learning**

We learned that it is possible with effort to assign values based on user input to one variable. That variable can then be used as a parameter through several functions. This can be used for many different applications where a user might want to see more of something that they are interested in and to leave out data on information that they would not care about. This basic method of filtering is used today in many applications that people are selling. For example, in product research for people who are looking for the right product to sell on amazon, they will use an application that sorts through products that contain a key-word that the user has entered. The program will then display the different statistics on those products in an organized manner such as how much one unit costs, the fees they pay amazon for listing that product, how many units they have sold of that product, how many reviews that product has, the estimated monthly revenue/profit that item can generate etc. This simple functionality saves people loads of time from having to sort through many pages and removes all of the unnecessary information that is only relevant to a consumer. The people that create these applications sell them for around $30-$300 a month for different types of functionality.

Even the impossible is possible with time and collaboration; there’s a lot that we accomplished in this project that seemed too hard or complicated during brainstorming and even while trying to get all of the appropriate code to do what we wanted. A method that we implemented that worked out really well was similar to SCRUM agile software development in which we met face to face or through facetime to discuss where we were in our development and clearly defining our next steps and dividing up the work accordingly. We also realized that projects of large size cannot be done hastily and the proper way to complete is in baby steps. That means starting from an idea, and then creating the pseudo code, then cleaning and organizing data, then the rough drafts of the functions, and incrementally increasing detail until the entire scheme works together fluidly. This of course is the easy part before deciding how our application or webpage’s UI/UX will look.

**5. Computational Thinking and Software Engineering Techniques**

We initially started off using decomposition to determine what our plan would be to tackle this idea. We used this method of thinking to distinguish what functions or methods would help us do what, who would be working with which parts, and in what order we had to create certain parts of our code to move on to other functionalities. We also used pattern recognition and algorithms to develop the functions that would generate the output to the user based on different variations of input. Specifically, we were using pattern recognition in our development because we might create a function that takes a certain value or set of values from a data set to use in a functionality and we were able to use that functionality in different datasets and for other values, or in our case skill sets. We used algorithms in that same process when collecting input data from the user, we wanted to let that data determine what parts of the data set would be displayed and what functions would be used. For this, we did not want to write functions over and over for every specific instance so we tried to create universal functions that could work with whatever data is chosen.

For testing techniques, it was mostly trial and error. We began with fixed functions that used specific parts of the data set and did not require input so that we could make sure we were getting the output we desired. From there we would begin to test functions with different values or parts of the data we were using to make sure that the function was not data specific and would not create errors for other values. Once that was working, we began testing using our own inputs to make sure that the functions were displaying the correct information, requesting the correct inputs, following the correct order, and breaking when they were supposed to.

**6. Next steps**

* Allow the user to filter the data graphs by year and/or round the participant was selected
* Give the user options to which kind of graph would the want to display the data instead of making a default one
* Allow the user to focus the data analysis on a particular team so they could see the preferences a specific team has for their position players
* Use a GUI

**7. Extra credit (GUI, Plotting function, GitHub, Web service) Optional**

*Plotting Function:* The most difficult part was trying to figure out how to change the color of one dot because there are certain rules to plotting functions. Also, it was difficult to not only add the user input to the data frame but to also reference it later in its own data frame in order for it to be plotted appropriately.

*Attempted GUI*: GUI has a lot of moving parts. In order to get it to work efficiently in the way we wanted, it seemed as though that we would have to change the functions that we had already created. Which meant almost starting from scratch.

**8. Individual contributions**

Humdan Ahmed:

* Found sources that provided different data sets to choose from
* Combined data sets into one large data set to be used in the application
* Created filter that changes user input to be compatible with the data set’s values to be used in functions
* Created functions that created pie charts based on the data for vertical, broad jump, cone, and shuttle
* Created separate data sets for each position, some being cleaned of their NA values.
* Assisted in testing the functions to find errors or lack of functionality.

Naaja Jacob:

* Created Data Sets for each position
* Created reference lists to be used throughout other functions
* Handled the plotting functions
  + Height/Weight Comparison (Shows where the user falls in comparison to people of similar body characteristics that play the same position)
  + Comparison
* Handled all user input
  + Intaking user’s statistics and adding it to the data frame
  + Intaking user’s answers that decides which functions to use

Dimeji Olayinka:

* Found final Combine data set,
* Made function that created a pie chart.
* Made functions that turned the forty test, vertical test, bench test, height, and weight of the players into lists and dictionaries to pass as parameters for the pie chart function
* Contributed to the last half the graph\_prompt function where the pie charts of the specific tests are called based on user input.

**9. Include the final code, presentation materials, along with previous updates & code snapshots in the ZIP file. Anything that shows the progression of your project should be included in the ZIP file. You may include a link to a video that demonstrates functionality.**