THIS PAGE CAN BE DELETED FROM YOUR FINAL REPORT GUIDANCE (text in blue can be deleted from your final submission)

Report Layout and Style

The following guidelines **must** be adhered to:

- All **text will be black** in the report unless in very exceptional circumstance. •

Main body text must be at least **11pt font** using either **Arial or Calibri** font. •

Main body text will have **1.5 line spacing**.

- **Margins** will be a minimum of **2 cm on each side**.
- All **pages** will be **numbered consecutively**.
- **Figures** must have **captions** and be **numbered** (e.g., Figure 1).
- **Tables** must have **captions** and be **numbered** (e.g., Table 1).
- **Figures** may be **black and white**, or **colour**.

Word Count

There is a **10,000-word limit** for the project. Do not see this as a target, but rather a limit to the number of words we expect a project to have. All words in the main body, excluding words in figures and tables, will count to your word count. If you think you will go over 10,000 words, you should consider what can be removed from the main body and placed in an appendix.

Copyright and Intellectual Property Rights (IPR)

Your report should be written considering that it will be within the public domain. Normally, you retain copyright over your written work and Intellectual Property Rights (IPR) over any technical work. There are situations where this might not be as simple, for example when working with a company or on a larger university project. There are strategies you can utilise:

- You can inform your supervisory team that the project cannot be made available to other students undertaking an BSc Project.
- You can provide a shortened report for sharing that does not contain the protected information.
- You can assign IPR to the external collaborator, although you should take great care when doing so. It is best to speak to your supervisor.
- Non-disclosure agreements can be made between the external collaborator and the supervision team.

12th Man - Final Year Project

Project Report

By

Matthew Lowrie

Submitted to
The University of Roehampton

In partial fulfilment of the requirements
for the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Abstract

Since 2010, data has become more prevalent in football and other sports. From pre-match statistics to in-depth player analysis, the use of data to create hypotheses and solve problems within sport has been steadily increasing over the past years. Some of the most noticeable clubs that use data to solve their issues are: Liverpool, Chelsea and Brentford all focus on data and have seen major success over the past decade. Liverpool and Chelsea have managed to win domestic and continental trophies whilst Brentford have managed to climb the leagues to the premier league and solidify their position

in the league. This report is aimed to solve the issue of misallocation of resources in major football clubs around the world. This is targeted at how clubs can overspend on players, or not give players enough training or match time. Alternatively they could not be willing to invest money in a top talent as they do not have the right background data on them.

The project aims to utilise a dataset to allow managers, players, scouts, owners and many more to look at the stats of their team or teammates to see the weaknesses and strengths within the team. This will be done by using different functions to sort the dataset to supply the user with the correct results allowing them to draw conclusions from the data. For example there is a linear regression model that allows users to predict a players performance based off of two other stats they have achieved in the previous season. These functions will contain features such as a player scouting system, statistical analysis tools, machine learning tools as well as straight-forward graphical visualisation.

Some of the techniques used include a player scouting system to allow scouts to find talents without having to watch them in person, saving time and money. As every club in the world always tries to be more efficient with their spending and time allocation, I believe that tackling this issue would have the most impact in a real world environment.

Declaration

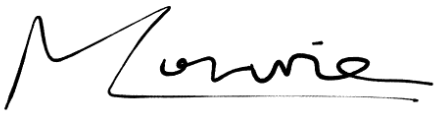
I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

Matthew Lowrie

Date: 13/04/2023

Signed:



Acknowledgements

Kimia Aksir: Supervisor for the project, provided feedback and advice throughout the project timeline.

Arturo Araujo: Supervisor for the project, provided feedback and advice throughout the project timeline.

Charles Clarke: Provided support and guidance throughout the final part of the project

Trevor Reese: Provided feedback on the code segment of the project

Mandev Sehra: Provided feedback on the code segment of the project

Sayeed Yahya: Provided feedback on the code segment of the project

Table of Contents

1. Introduction

Background and targets:

My Project is about how data science has revolutionised football into the modern era it is in now. This has been done by delving deeper into the stats behind each player so a team can make more effective decisions with their current players or transfer targets. Films such as 'Moneyball' [13] have highlighted the importance of stats in sports but my project will focus on football, however it will be easily adjustable to suit different datasets or sports. The film 'Moneyball' displayed the impact of this technique on the big screen, based on a true story where a new set of staff (a GM and an analyst) allowed a below average team to go on the biggest winning streak at the time in Baseball. Their technique was a new idea at the time as instead of signing a player on how they looked in game, they

signed players, for low prices, based on the stats they delivered. As a result of this they broke records and changed how every club ran themselves after their breakout season.

Some of the questions that produced my project were, “How could a club effectively find new players?”, “How can clubs know where to loom for their players” and “How can a manager predict a players performance based on their previous performances?”. In order to answer these questions different functions such as a player scouter, multi linear regression model and a function to compare player performances in different leagues, were created. A rebound effect of these would be an increase in a clubs ROI as there would be less players being signed that do not fit into a team.

Motivation:

My motivation behind this project is from my personal interest in football and the hectic transfer windows that happen twice a year. I wanted to create a prototype that can be easily adapted and transformed into a program that can be implemented into a website to help allow clubs to increase their performance in the transfer window. I also wanted to see whether I could make something that can solve the real world issue of clubs not allocating their resources effectively. From the perspective of a fan, there would be times where a club would sign a player that the fans did not agree with. This sparked my interest in the behind the scenes work that went into scouting and signing a new player as I wanted to understand why certain decisions were made.

Liverpool FC [14] and Arsenal FC are the most recent examples of how signing and starting players who were previously under the radar can have an exponential impact on the success of the club. For example, Mohammed Salah, from Liverpool, was a relatively average player who had glimpses of top quality skill but did not show enough for a top level move. However Liverpool came in to acquire the player and now he is one of the best players in the world within a few years of moving to his new club. The ability to look at a player's data and output allowed them to make the decision to sign him and place him in the starting 11 for their team. This analysis allowed them to sign a world class player for a price which was much cheaper than what he was worth as no other club did the proper analysis on him.

Other examples are: Saka, Martinelli, Saliba and Trent Alexander Arnold from Arsenal and Liverpool. The business the clubs were able to pull off all stemmed from their data scientists which directly ties in to my end goal of my project.

Goals:

My goal for my project is to deliver a program that will allow the user to perform in depth analysis of every player in the dataset (https://www.kaggle.com/datasets/vivovinco/20212022-football-player-stats), as a result the program can be used to train new data analysts for different sports by using real world data resulting in a higher output from professional clubs.

My work will be significant as I intend to make it so it is easy for the code to be adjusted to suit different sports or datasets making it very flexible for different environments. This will benefit consumers within the sport scouting and analysis sector as it will allow them to easily compare and discover players to find how to get the most out of them. From the business perspective the use of this technology will allow the club to gain a higher return on investment to present to their shareholders, thus boosting the value of the business and allowing more growth in the same timeframe.

Due to this work and discovery of a new method of scouting, clubs all over the world in different sports have altered the way they look at players. but I believe the way this is conducted can be changed if some of the data being analysed now is not as useful as it is made out to be, so my project will only be focusing on the data that is necessary to allow a quicker but just as effective analysis.

The main stakeholders of this project are: clubs, managers, players, scouts and sports fans. Professional clubs are a stakeholder as they can use the program to decide whether their manager or players are working well together as it allows them to look deeper into the stats. Managers and players will be able to analyse their performances and their teammates performances to gain an insight into how they are excelling or falling behind compared to other players/ teams. The scouts can use the program to find underrated stars and rising talents to help the clubs invest their money better to gain an increased ROI from their budgets. As a sports fan the project can let them use the code to understand why their club is making decisions, for example they can use the code to understand why their manager is playing certain people instead of others as they can see the stats the player is producing.

Legal:

The legal/ ethical issues associated with my project mainly concern the use of the dataset, however I have already checked and made sure the dataset is valid and authentic to use for my project. This is important as we do not want to be using fake or altered data in a project which can greatly affect a team's business and have major consequences. Another legal consideration is the GDPR act that was signed in may 2016, this monitors and limits the use of data and makes it necessary to alert the user on how data is being stored and the intended use for it.

Another legal consideration is the CCPA act which makes it mandatory to allow the user to request their data to be deleted, to give the user control of how their data is collected and distributed.

Making sure my program is compliant with these laws is important as if the project was publicised, it would have to comply in order for it to be publicly usable. Or, the users could take legal action against the company using it as it does not comply with the legislations.

Social:

A social consideration would be how environmentally impactful this project is. However, as the project can allow teams to scout players by the data collected instead of flying out scouts to constantly make trips back and forth to scout and report on the player, the project can reduce the amount of greenhouse gases released in this process.

Another social consideration is to make sure the program isn't exclusive to a certain category of people. It is important to be inclusive on who can use the program as it can also help improve the total output of the project if anyone can use it.

Ethical:

There are some ethical issues that concern my project. One of these is, how different clubs will have an advantage over their competitors if they have access to this tool and they do not. This is because

it will allow a team to outperform their competitors by acquiring better players through their data driven response which can yield a higher return in comparison to the other club.

Another ethical issue is how it will impact the labour market of the scouting industry within the football scope. Although the program may reduce the amount of people going to watch a player to scout them, it will greatly increase the amount of jobs within the tech portion of the company. This is an ethical issue, however due to the advancements in the world it is an adjustment that is bound to happen either way. So, I do not believe my program will have a great effect in accelerating this adjustment.

The issue of misleading data is also a concern as it is very important to make sure the data that is being used is very accurate, to reduce the amount of incorrect data that is being presented by the program. To make sure this project does not contribute to providing misleading data, it was important to make sure the dataset was collected from a reliable source.

Roadmap:

Within this report there will be many different sections that will describe and explain different components of the report. The roadmap of the report is as follows:

- Literature and technology review - This is where different reports and findings will be analysed on their findings regarding the user of data analysis tools in the world of sport. Also the different pieces of technology used within the code section of my project will be analysed on their utility and effect for the project's outcome.

- Design and methodology - This section explains the reasoning for using many tools such as the programming language and compiler. Also an explanation of the work that will be performed in the project.

- Implementation and the results - The strategy for how the work was implemented will be explained as well as how the results turned out. Whether they were what was expected or if they produced surprising results will be determined in this part.

- Conclusion - The entire project as a whole will be evaluated on the success, performance, results as well as the next steps or changes that should have been made from the start to achieve a different result.

2. Literature or Technology Review

Literature Review:

Within all sports there is a point where a new technology comes in and changes the sport completely, for football the change was the adjustment to a data driven response. This was mimicked from the Oakland Athletics breakthrough season in 2002 in which the head coach, Billy Beane, decided to put all of his faith into a computer generated form of analysis to guide their transfer strategies for the upcoming season which leads them to break records. The most recent event where a team has changed to stat driven tactics is the english premier league team Chelsea F.C, where the old owner, Roman Abramovich, sold his team to the new owner, Todd Boehly, who believes is allowing computer analysis to guide the team to success.

Why use Software Analysis?

As stated in [3] the approach of a computer software analysis approach allows for more accurate results. The technology that was used is the QSR*NVIVO 7, which is a piece of software that is designed to help with the management and analysis of qualitative data. According to the article the software allowed the users to perform in depth analysis with large data much quicker than if the user conducted their research without the software. Linking to my project, using software analysis has started to allow teams to scout players based on their stats which could be hard for someone to recognise if they were to watch the player normally.

Real World Example:

An example of this is Liverpool when they brought in Jurgen Klopp as their new manager in 2015. Before he managed the German team called Borussia Dortmund, like other managers, he was spending all of his time coaching his club on the field and on the training grounds. However when he came to liverpool, as stated in [4], liverpool's director of research, Ian Graham, approached Klopp to discuss his dortmund team and how he believes he could improve liverpool's results by incorporating statistical analysis more moving forward. To do this he would show Jurgen Klopp his previous games to highlight how the data analysis would help turn losses into wins by understanding his players strengths by looking at the data they produce. In my project I have features that allow the user to search for players that excel in certain areas and show the top performers in their fields. This is similar to the techniques Liverpool and Klopp used to turn an average Liverpool team into one of the best teams in football history.

Graham was given a large amount of responsibility, whereas other teams would rely on watching a player to decide whether they were good, he would rely on the teams data to find weak spots in their game and find players whose stats showed they were suited to help fill the gaps at the team. An example of this was a Brazilian forward called Roberto Firmino. Firmino used to be an attacking midfielder used to linking up play to a striker, however Liverpool's new idea was to have a fluid front three. This means there is not a normal striker which was a common practice at premier league clubs. Instead they used data analysis to find players who could link with Firmino to create a deadly attack, these players were Mané and Salah. The attack then went on to win multiple golden boots and titles, domestically and internationally, giving them the reputation of one of the best teams in the world. My technology allows teams to thus sign players for below market value.

Reasoning For Decisions:

In my project I decided to base my data analysis off of a whole season instead of a few games. The article [5] supports this choice by stating that in order for a successful analysis the user needs to have a large time frame when it comes to sports. The article gave an example of thirty games which is eight short of a premier league calendar. The reason for having a large sample size is to aid player performance, team performance and the prediction of future games or stats.

Having a larger sample set time frame means that the data collected will be a better representation of the general population. This makes the results be more accurate and will be less skewed, as a result of this the user can spot any anomalies when pre processing their data for analysis.

The online article [6] states how many different ways data science and analysis is used in football as well as how it helps improve a teams performance. The different ways in which data science has helped are: Team and match performance, Player performance, Player recruitment, Tracking match events and predictions.

Team and match performance allows the scientist to look at the stats a team produces as they will give a better representation of how a team performs rather than the scoreline. This is because, when relying on a scoreline a team can have a bad performance when it comes to scoring goals but could excel in other parts of the field that will be overlooked because it does not impact the scoreline.

Similar to my project, player performance is also used in real world sports. This is used for managers or scouts to see where a player needs to improve or what sort of style they suit the most. The reason I based my project around this was because of how it is revolutionising how teams conduct their business in the transfer window and how managers organise their teams throughout the season.

Another feature of my project is to allow teams to invest their money in players more wisely by finding hidden gems. This ideology is talked about in the article [7], the online article talks about how the investments in balkan countries are much lower as football is not as developed so the investments need to be placed more wisely to allow the best return of investment. Within the article a 28 year-old football data scout, Vlatko Jakimovski, talks about how the data scouts are much more reliable than the actual game scores when looking to find the best players. Furthermore he mentions how the teams can use the data to learn about any weaknesses of their opponents to organise their team to get the best chances of winning.

Technology Review:

Some key features of the technology I will be using will allow the user to gain an in depth analysis of the players within the dataset. This will be done on python as it is my language of choice when coding, furthermore, all the libraries I will be using will be suitable for my project such as; numpy, pandas, matplotlib, tabulate and many more.

Pandas:

One of the first libraries I use is the pandas library which is a commonly used data manipulation and analysis library. Subsequently, this will allow the user to easily access and use data structures and

various analysis tools. These will help when working with structured data such as tabular, time series and matrix data.

Within this library there are two main data structures that are available, these are; Series and DataFrame, which contribute to making data manipulation easier.

A series is a one-dimensional array which can hold many data types such as; integers, strings or python objects. A DataFrame is a two-dimensional array which is a structure similar to tables that have labelled rows and columns which can hold any data type. The pandas library also provides the user with a plethora of functions and methods used for data cleaning, transformation, preparation and analysis. An example of this is it lets you merge or concatenate datasets to perform statistical and mathematical operations.

In my project this is used to change, read and group the data from my chosen data set so I can then use other libraries to perform operations that can be used to create visualisations.

To summarise the pandas library it is a very powerful library for data manipulation and manipulation that is commonly used and found in data science.

Pros and Cons:

The pros of this library as referenced in [8] are it allows the user to do large amounts of work with less amount of writing to result in more efficient code, furthermore, there is excellent data representation to help present results. Another pro is that it is made for python meaning the compatibility is better compared to other libraries.

The cons are that there is a steep learning curve which means it's hard to just pick up and use, there is also difficult syntax.

Matplotlib Library:

The matplotlib library is another library that will be extensively used in my program. This library is used for data visualisation , plotting various types of graphs, charts and other visual aids, it is also used for scientific computing. As a result of this users can produce a high quality visualisation to help understand the data they are working with. Linking back to NumPy, matplotlib is built on top of this array as it needs to work with large datasets as well as, perform complex mathematical operations similar to NumPy matplotlib i integrated with other libraries such as seaborn, pandas and SciPy.

Matplotlib gives the user access to a big range of customisable graphs that can be plotted such as scatter plots, line graphs, bar charts, histograms, pie charts and many more which can be customised with colours, styles and themes.

The library is very user friendly with an easy to use interface with many tools for visualisation as well as tools for labelling and formatting plots to help with their visual clarity and readability.

Pros and Cons:

Some of the advantages referenced in [9] and [10] are; it allows the user to access large amounts of data in a simple way, it is flexible and supports multiple forms of data representation, it is easy to navigate, it is open source, and it's very customizable. However it is unsuitable for dataframes as the library needs external libraries to allow them to be compatible.

NumPy Library:

The numpy library is a powerful tool in python which is used for scientific computations and analysis. It is built on the python language and provides tools which are much more efficient at handling large data compared to the standard list type in python.

Within the library there are powerful multidimensional arrays which allow the user to handle data effectively such as csv files or databases as the numpy array provides the user with the ability to handle them with efficiency. There are also many mathematical operations that are supplied by the numpy array which allow the user to perform operations such as; addition, subtraction, multiplication and many more. There is also the ability to perform linear algebra on the data which can allow the user to do matrix multiplication,inversion and other common linear operations.

There is also the ability to perform random number generation which is an extremely useful tool for simulations,analysis and machine learning. Furthermore there is an ability to manipulate the data which functions such as slicing, indexing, reshaping and transposing. Which makes it easy to select certain parts of the data and extract them as a new dataset in a different shape if needed.

Pros and Cons:

Some of the pros and cons mentioned in [11] are that the library allows the user to complete challenges faster due to the built in functions as well as the ability to multiply matrices. Another advantage is the library allows the user to filter through the data to draw conclusions of work towards goals.

Tabulate Library:

Another library that I use in my project is the tabuate library which supplies the user with the ability to display data in a clear readable way. I have selected this library as it can work well with the other libraries mentioned earlier. The library can accept a wide range of inputs, including the numpy arrays spoken about earlier, which can then be customised based on the columns, indexes, adding certain headers or footers and formatting the data in multiple ways.

The library also supports the ability to extract the data in multiple ways. This is useful for pipelining the data to external places to provide further analysis. One of the formats the data can be extracted in is the HTML format allowing the data to be presented on a website from the python file.

To summarise the library it allows the user to display data in a readable format to the user whilst working with the other libraries to generate tables or other visualisations.

Pros and Cons:

The main advantage of this library in [12] is that it can transform messy data into neat readable tables to be more user friendly.

Seaborn Library:

Seaborn is another python library that is used within my project, unlike other libraries the seaborn library is not built on the numpy library but it can work with the library to produce a range of customisable visualisations to the user.

The library can produce a range of tables and graphs such as: scatter plots, histograms, bar graphs, line graphs and heat maps. These can be used to explore data and identify if there is any relationship between data points or any trends within the data.

developing on the customisable point of the library, the visualisations are fully customisable from the labels and colours to the size and shapes or the plots and graphs.

As stated earlier, although the library is not related to pandas it is designed to work seamlessly with the library helping it to produce visualisations from the dataframes found within the numpy library.

The ability to do more advanced statistical analysis is supported within the library as it allows them to do regression analysis and hypothesis testing. The hypothesis testing can be helped by the ability to produce interactive plots in the library, meaning the visualisations can be affected by interactions with the mouse and keyboard.

Pros and Cons:

As stated in [15] the advantage of seaborn is it is really easy for the user to visualise their data however it is not built in so on first use it will need to be installed.

Time Library:

The time library is used within my project to add immersion, as well as to increase the quality of the user experience. For example the time library is used to show a loading message when entering a new function. This breaks down the program to allow the user to read anything on the screen to stop overwhelming the user with text or information.

Pros and Cons:

The library is extremely easy to use and is completely compatible with all versions of python. Furthermore the library is very accurate with its time outputs and built-in functionalities.

However it has some drawbacks such as limited functionality and resolution, which means that the library only provides basic solutions to time related tasks. Also the library only has the ability to measure time to the second and cannot go to a greater resolution. The last drawback is that it can only work within a certain time range which is 1970-2038.

As a whole, the library is extremely useful for basic time related tasks but may not be as useful for more complex time related tasks.

Scikit-Learn Library:

Another library that is used within the code is the scikit-learn library which is used within the linear regression model. It provides the user with an easy to use library with a broad functionality, as well as that it is fully customisable and is integrated with other popular python libraries.

Pros and Cons:

Although the library has limited deep learning functionality and scalability. This means it is not suited for large scale datasets. Another drawback is it has limited support for time-series data. But despite that, it is a powerful tool for machine learning tasks.

Table Summarising Technology Review:

Library Name	Pros	Cons
Pandas	-Less writing and more work done -Excellent data representation -Made for python -Extensive set of features	-Steep learning curve -Difficult syntax -Poor compatibility for 3D matrices -Bad documentation
Matplotlib	-Provides a simple way to access large amounts of data -It is flexible and supports various forms of data representation -It is easy to navigate -Has numerous application -Creates advanced visualisation -Saves money as it is open source -Can run on multiple platforms -makes data analysis easier	-unsuitable for dataframes -lacks custom themes
NumPy	-Speed -Matrix multiplication -Filtering	-Lack of cross platform support
Tabulate	-Improves user friendliness	-cannot work with certain data grouping methods
Seaborn	-easy to represent data	-it is not built in
Time	-Easy to use -Compatibility -Accuracy	-limited functionality -limited resolution -limited time range
Scikit-learn	-Ease of use -broad functionality -customizability -integration with other libraries -Robustness	-limited deep learning functionality -limited scalability -limited support for time-series data

3.Design or Methodology

BSc Project Template Spring 2022_v02

The first idea for the project consisted of machine learning models for the selection of players from the dataset, however, this decision was slightly altered as if machine learning was used to select certain players it would become outdated each year. This is because the model would be trained on out of date data each season due to how different the data is after each year of football. For example Chelsea FC have been a top performing team every season for the past decade, however this year they have fallen behind their peers. So if data from a few years ago was used to select players for the team this season, inaccurate data would be outputted to the user.

So, in replacement for this, machine learning was only used as a sub-feature of the whole project. Instead the use of data science and analysis features have been used to allow the project to easily be updated by just replacing the dataset each year. Elongating the longevity of the project.

Coding Language:

For my project I have focused on the coding language called ‘python’. To be more precise I am using python version 3.11, the reason for me using this language derives from the scope of my project. As the project is heavily data driven, python stood out as the perfect language to use due to its ability to perform operations on data sets with built in libraries and functionalities to visualise and edit these datasets. As 3.11 is the latest version of python it also provides enhancements on previous versions of the language.

These improvements consist of enhanced performance. This means that the code execution is more optimised, as a result of this, big data projects are easily managed within the python language due to this enhancement. Due to the fact that I am working with data that involves players names, a language with improved security features improves the reliability of my code and the project as a whole. This is needed as there are many legal and ethical issues that can stop projects being publicised due to their lack of legal awareness. An example of this is the GDPR (May 2016) act that monitors and regulates the use of personal data and information. So python reduces the amount of security issues that need to be resolved.

Lastly as mentioned earlier in the report, python 3.11 has many built in powerful libraries dedicated to data handling, thus allowing users to effortlessly manipulate, clean, process and load their data with less issues.

Compiler Choice:

The compilers used for this project are: Visual Studio Code (Version 1.76) and google colab. The reasoning behind using these compilers are due to their standout features. Google collab has a neat format allowing users to format their code with text cells to help describe the operations or steps

being performed in the cell of code. In addition google colab works on the cloud so if the PC crashes the work will be safe.

Visual Studio code was used as there are a lot of text editing features which allows me to edit multiple instances of a word to change them all simultaneously. As the project is a data driven project it involves variable names or dataframes being repeatedly mentioned. So, when I need to edit all of the names, the compiler developed by Microsoft allows me to do so with ease due to its built-in features.

I decided to use multiple compilers to test compatibility so developers on different compilers can both have access to my project without having to change the environment of their workspace.

The design of my project consists of many functions that provide the operations needed to create the outputs requested by the user.

Project Management:

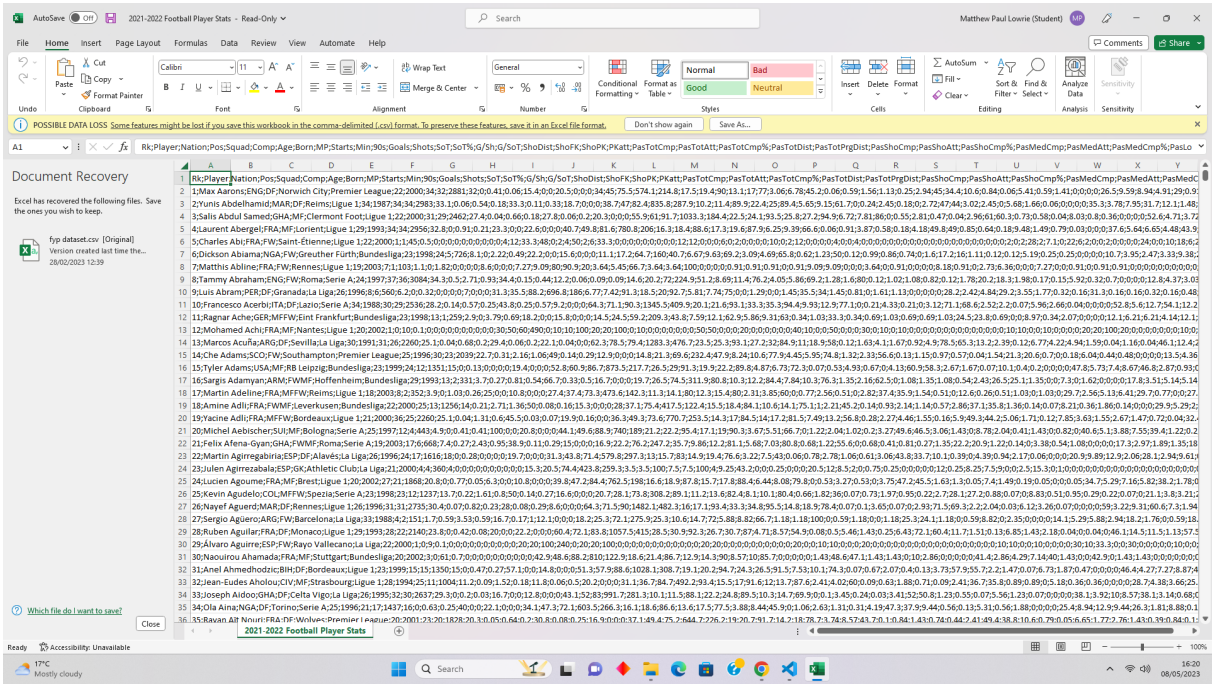
To manage my project, the initial course of action was a gantt chart and a kanban board to review progress. However, recently the public website called 'teamwork.com' has taken over as the main source of project management as it has many built in features such as a kanban board as well as a gantt chart to allow for anyone to easily keep track of the progress.

I decided to make this decision as it also allowed me to share my progress with any supervisors/ people monitoring the progress. Instead of sending multiple files any time some progress has been made.

Pre-Processing:

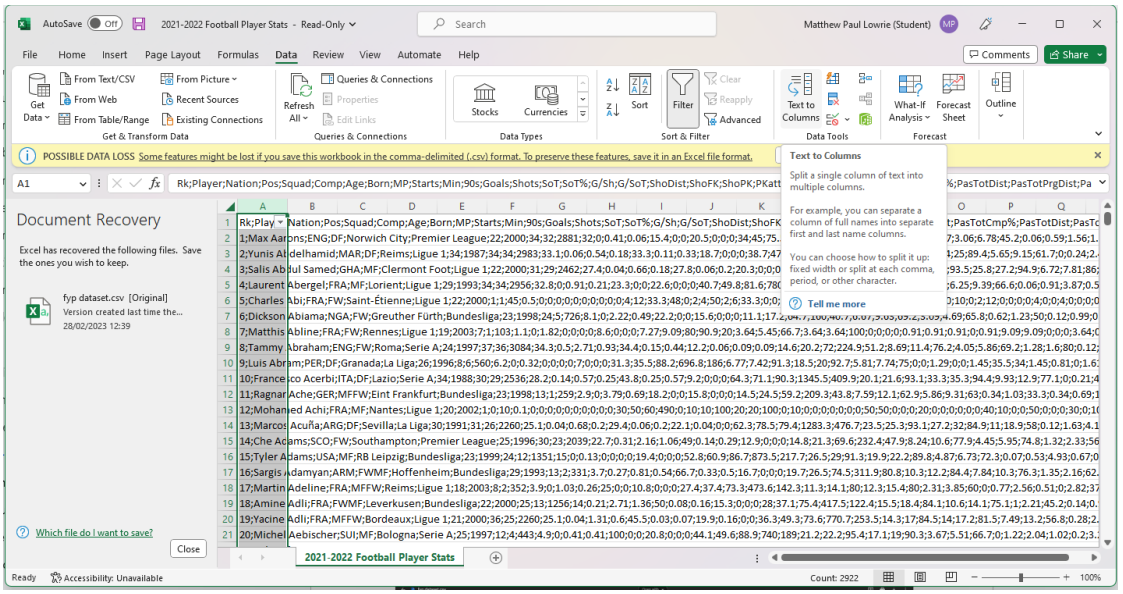
Before any operations take place the data needs to be pre processed to make sure all of the data is consistent and usable. Before the data was loaded into the compilers the first step was adding a delimiter to the excel file as the data was entered with semicolons separating the data. So to change this I added a delimiter of a semicolon to split the data in separate cells to make the data look like the image below.

Raw Data:

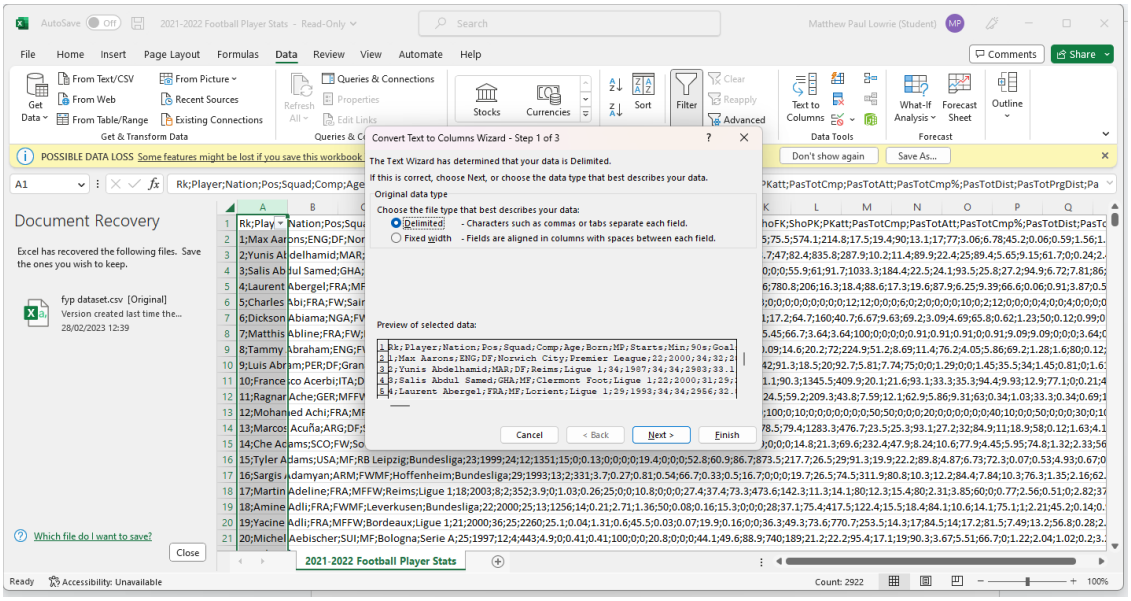


Adding a Delimiter:

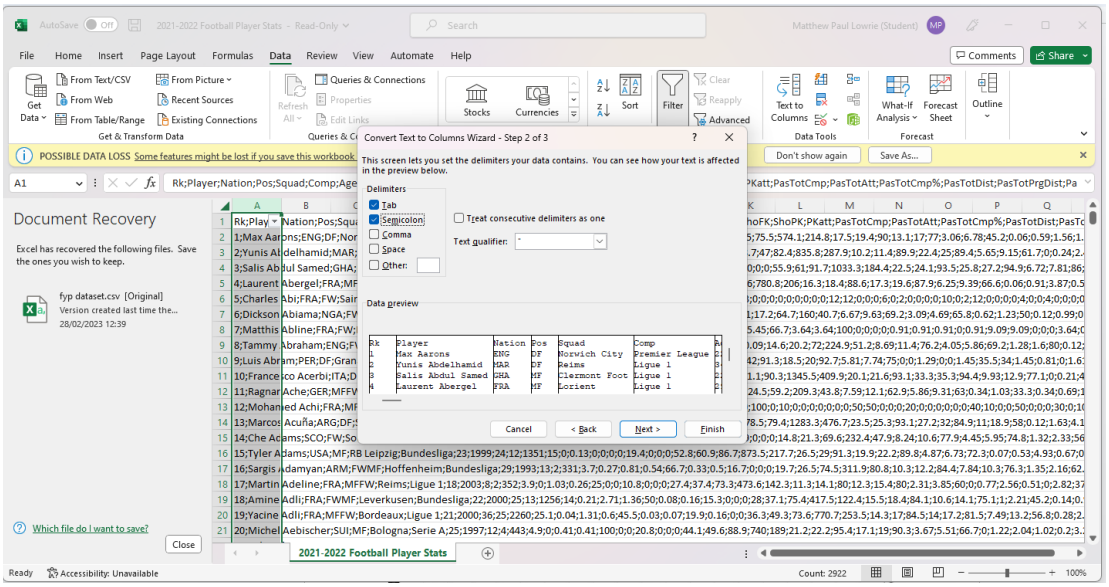
1. Select the first column then go to the ‘data’ tab to select the ‘Text To Columns’ button.



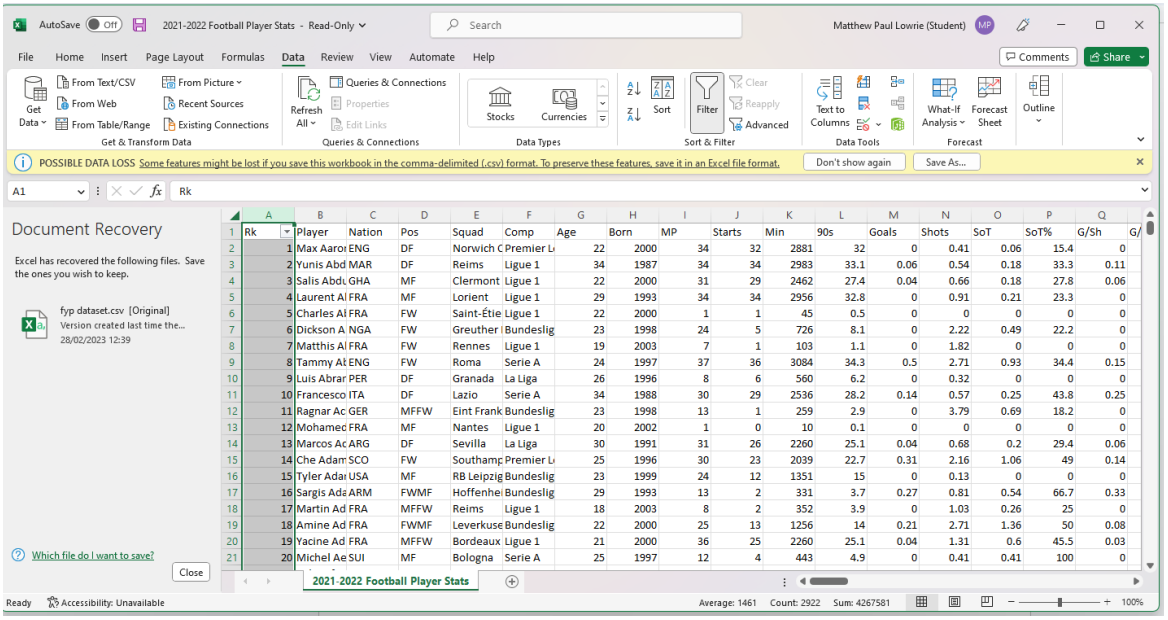
2. Select the Delimiter option:



3. Select the semicolon to separate the data then click finish:



4. The data will now be separated and displayed like the image below.



The data had to be separated as when trying to work with the data in python it would not allow libraries like pandas to work with the csv file, as all of the data was in one column which made it difficult to perform operations.

Once this was done there was still more pre-processing that was necessary for my project. As ,in my opinion, there were some pieces of data that were not necessarily needed I dropped them from the dataset. Contrary to the previous step I performed this in python as if someone was to recreate my project they could adjust some of the lines to keep extra pieces of data in the dataframe, allowing the code to be easily adjusted by other programmers to suit their project.

The way this was done was with the .drop() function.

```
#drop every bit of data we will not be using
data = Data.drop(['ShoFK'],axis = 1)
data = data.drop(['PKatt'],axis = 1)
data = data.drop(['PasTotCmp'],axis = 1)
data = data.drop(['PasTotDist'],axis = 1)
data = data.drop(['PasTotPrgDist'],axis = 1)
data = data.drop(['PasShoCmp'],axis = 1)
data = data.drop(['PasMedCmp'],axis = 1)
data = data.drop(['PasLonCmp'],axis = 1)
data = data.drop(['PPA'],axis = 1)
data = data.drop(['PasProg'],axis = 1)
data = data.drop(['PasLive'],axis = 1)
```

The reason I performed the operation on multiple lines instead of one was to make it more readable and easier to find the piece of data that the user wants to add.

The tables below will represent what is done by this code:

The first table is before ‘dropping’ any of the columns. The second table is after dropping the non useful columns from the dataset to make there be less wasted data.

Table 1:

Useful Column	Not needed column	Useful Column	Useful Column	Not needed column
data	data	data	data	data
data	data	data	data	data

Table 2:

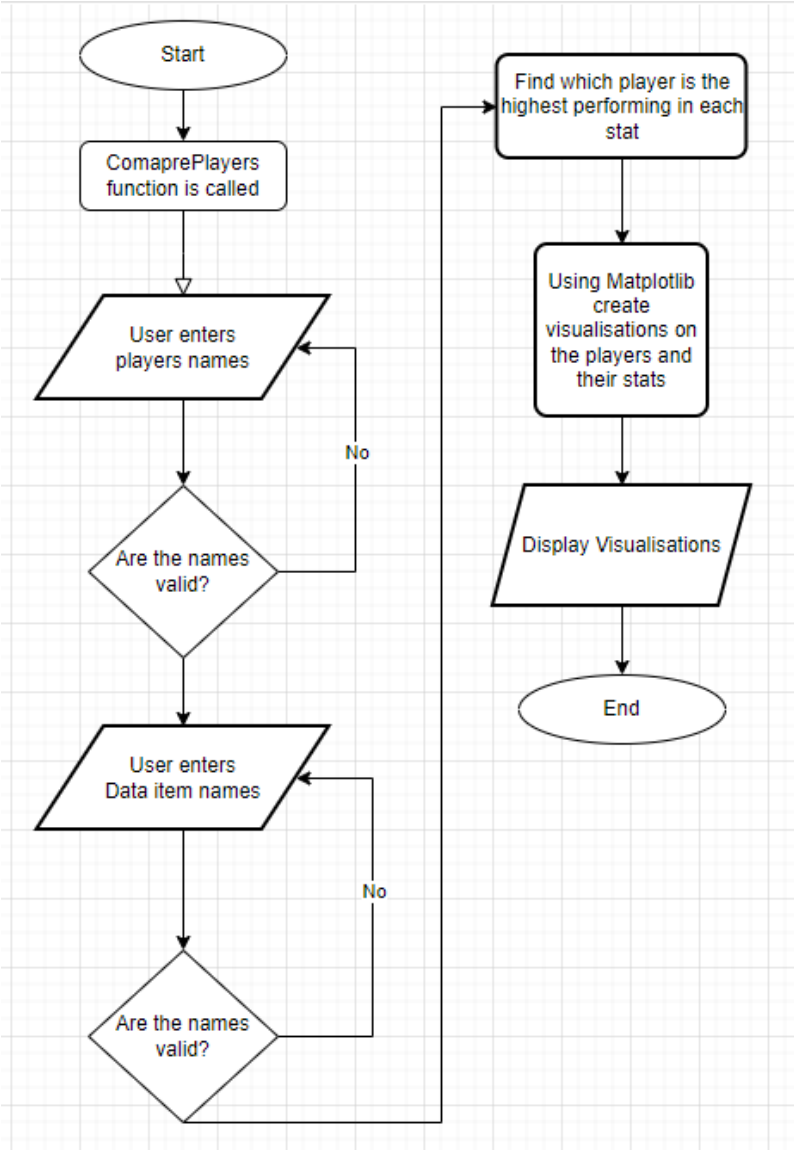
Useful Column	Useful Column	Useful Column
data	data	data
data	data	data

Visualisation:

In football, data analysis requires the data to be visualised so more conclusions can be drawn from the data. For example when watching a football match a graph of different players' football stats can be presented to the viewer to allow them to get a deeper understanding of the players.

One way this can be done is from the ComparePlayers function within the program which will allow users such as football scouts to compare different players to see who is the best.

The flowchart below will represent how this operation is performed then some images of the results will be shown to display the results.

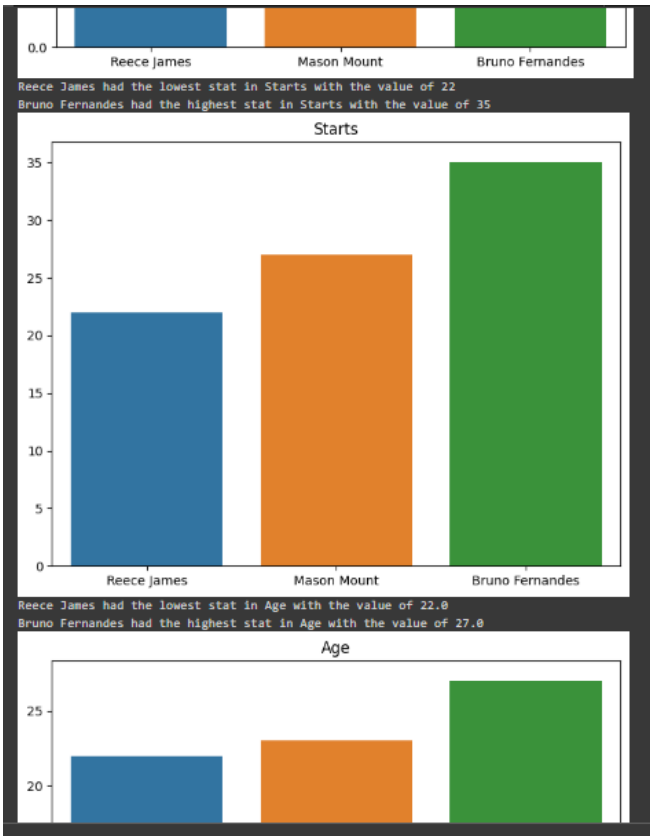


Now the following diagrams will show how it looks on the users end:

First the user enters the players and the data they want to compare:

```
Enter player name(s) separated by commas: Reece James,Mason Mount,Bruno Fernandes
Enter the stats you would like to compare (e.g SoT, Goals, Age) separated by commas: Goals,Assists,Starts,Age
Reece James had the lowest stat in Goals with the value of 0.24
Mason Mount had the highest stat in Goals with the value of 0.42
```

Once the user's inputs are validated the visualisations of each stat are displayed to show the user how they compare to each other.



Goals:

For my project my main goal was to add more features compared to other scouting systems for sports. A similar piece of work is the scouting system on “scoutingsystem.com”, which only allows a user to scout players and shortlist them, whereas compared to mine the user can perform multiple comparison features as well as analysis features, scout players based on performance as well as shortlist players that will be saved for their next uses. In order for me to do this, a bit of research to find out what scouts require for their jobs as well as managers and owners when it comes to analysing the team and the players. When this was done the results showed us that the comparison of players to their competitors is used by scouts as well as managers to see whether their current squads need any improvementsThe way this will change the way clubs operate is by allowing them to operate at a higher efficiency compared to other teams and previous years. This is because a rebound effect of the project will be allowing teams to invest better in young players to get a greater output from them with a lower risk due to the research taking place before any signings.

4. Implementation or Results

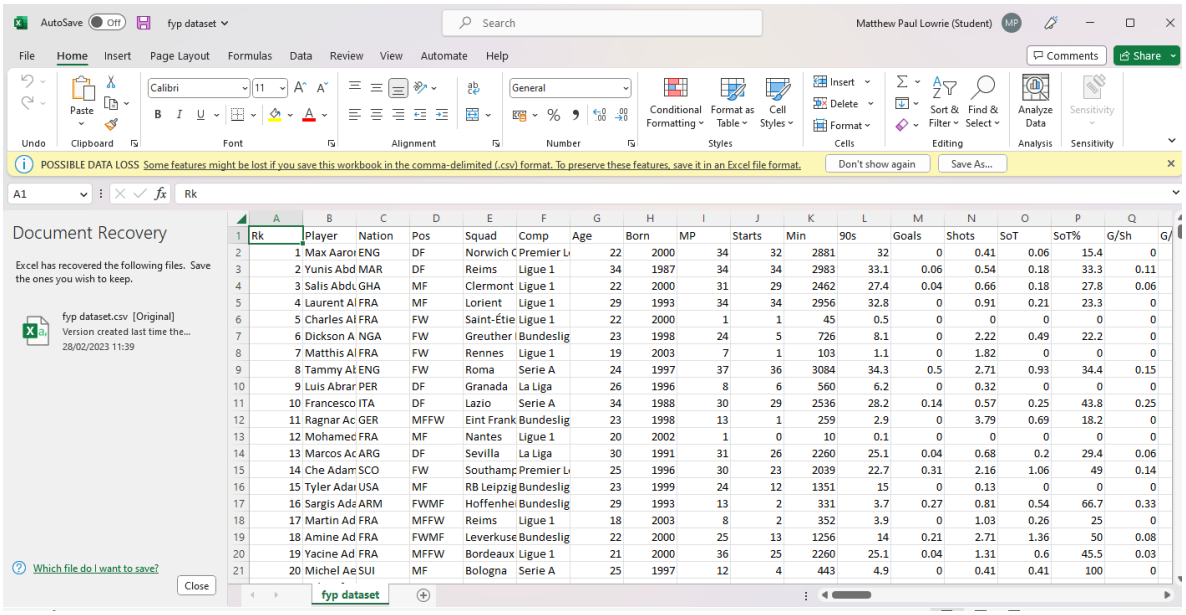
Overview:

As this project is mainly intended for consumers within the sports field, the results of the project had to deliver clear, precise and accurate data to the user as the results can affect people's livelihood if the project is a factor for a sporting club making a decision about a player or the team. In order for me to do this, the correct methods and libraries needed to be used in order to make sure the outputs from the code are clear and precise. Some examples are: numpy, pandas and tabulate, as together they deliver the ability to manipulate the data then clearly display the results to the user.

In order for my project to be implemented properly, the data has to be pre processed to make sure no resources are wasted processing unnecessary data within the dataset. As previously discussed there are two ways to complete this. One of the ways is to use excel or google sheets to manually adjust the database, which would be fine, however if someone was to use the dataset but wanted to recover some of the deleted data they would not be able to do so. To prevent this from happening the only part of the dataset that is adjusted in excel was the delimiters, this is just to separate the data into separate columns to make sure it can be processed in python.

This was easily done within the data tab in google sheets then applying the semicolon as the delimiter as that was what separated my pieces of data in the file. A delimiter can be added with code, but, to reduce unnecessary code I stuck with my decision to use google sheets built in features.

The results of this are shown below:

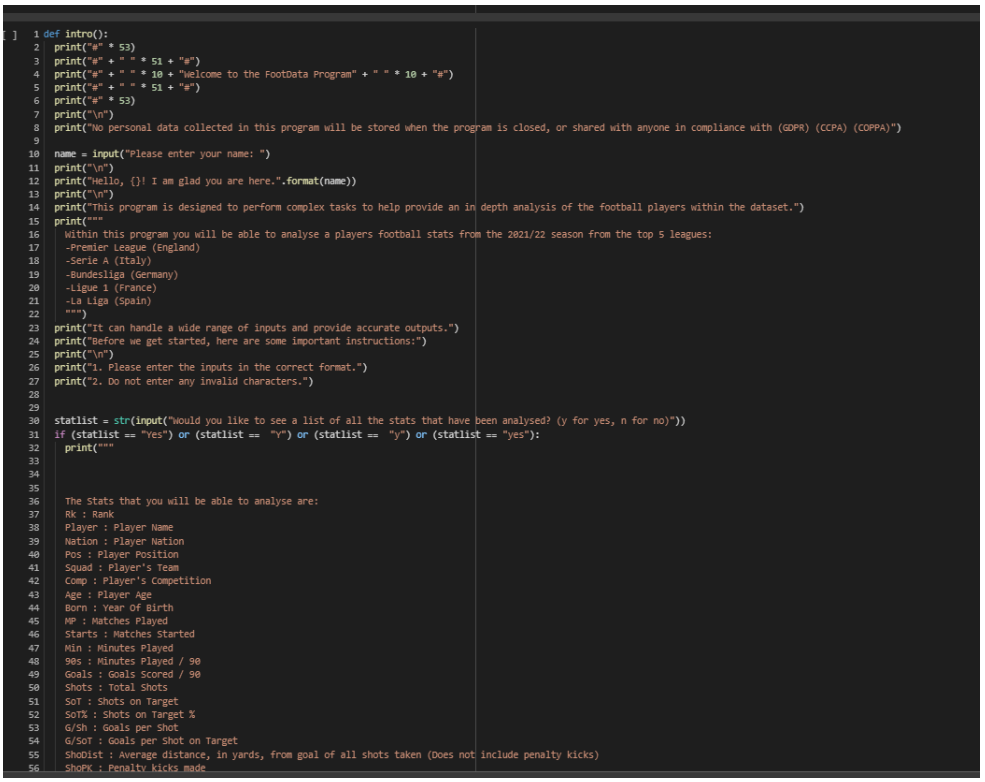


As shown, all the pieces of data are neatly arranged into their separate columns, which increases the readability and accessibility within a python program.

The next step was to remove all of the useless data from my program. The way I implemented this in my code was with the df.drop() function which removes any columns which have the header that matches the input of the user. The drop feature can be used to remove many pieces of a dataframe in one line to reduce the amount of lines within the code. Although I mentioned earlier that a main goal of my project was to complete my goal efficiently and effectively, to make the code more user friendly and readable the 'drop' function was split up into multiple lines. Additionally this allows the dataset to be easily edited by simply commenting out a line instead of having to remove a part of a line of code.

One of the most important parts of any coding project is the opening message to the user to explain exactly what the program does. For this project the implementation of this was to create a large print function to display the message and supply the user with further information if they requested it. One of these pieces of information was a list of all of the stats that are within the dataset. As the list of stats requires a lot of screen space the user has the choice whether they wish to see the list. This was done as if the user has already used the program multiple times it allows them to use the program with increased ease of use by reducing redundant information.

The following image provides a snippet of the information show in the introductory message:

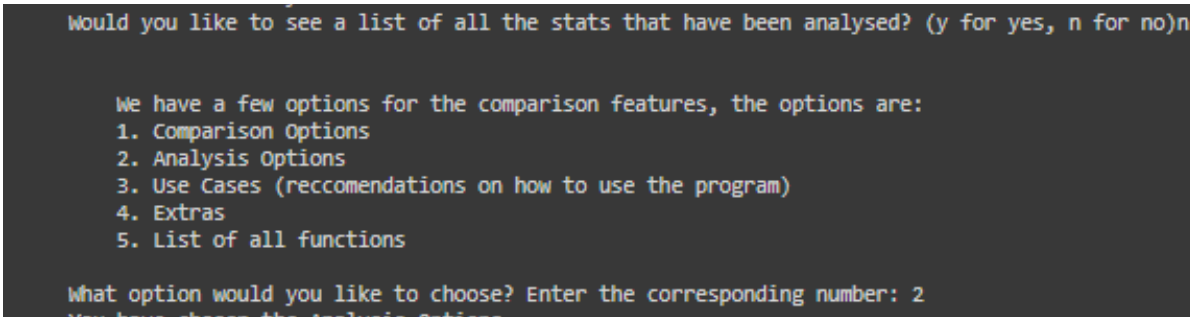


```
[ ]  1 def intro():
     2   print("#" * 53)
     3   print("#" + " " * 51 + "#")
     4   print("#" + " " * 10 + "Welcome to the FootData Program" + " " * 10 + "#")
     5   print("#" + " " * 51 + "#")
     6   print("#" * 53)
     7   print("\n")
     8   print("No personal data collected in this program will be stored when the program is closed, or shared with anyone in compliance with (GDPR) (CCPA) (COPPA)")
     9
    10   name = input("Please enter your name: ")
    11   print("\n")
    12   print("Hello, {}! I am glad you are here.".format(name))
    13   print("\n")
    14   print("This program is designed to perform complex tasks to help provide an in depth analysis of the football players within the dataset.")
    15   print("""
    16     Within this program you will be able to analyse a players football stats from the 2021/22 season from the top 5 leagues:
    17     -Premier League (England)
    18     -Serie A (Italy)
    19     -Bundesliga (Germany)
    20     -Ligue 1 (France)
    21     -La Liga (Spain)
    22     """)
    23   print("It can handle a wide range of inputs and provide accurate outputs.")
    24   print("Before we get started, here are some important instructions:")
    25   print("\n")
    26   print("1. Please enter the inputs in the correct format.")
    27   print("2. Do not enter any invalid characters.")
    28
    29
    30   statlist = str(input("Would you like to see a list of all the stats that have been analysed? (y for yes, n for no)"))
    31   if (statlist == "Yes") or (statlist ==  "Y") or (statlist ==  "y") or (statlist == "yes"):
    32     print("""
    33
    34
    35
    36     The Stats that you will be able to analyse are:
    37     Rk : Rank
    38     Player : Player Name
    39     Nation : Player Nation
    40     Pos : Player Position
    41     Squad : Player's Team
    42     Comp : Player's Competition
    43     Age : Player Age
    44     Born : Year Of Birth
    45     MP : Matches Played
    46     Starts : Matches Started
    47     Min : Minutes Played
    48     90s : Minutes Played / 90
    49     Goals : Goals Scored / 90
    50     Shots : Total Shots
    51     SoT : Shots on Target
    52     SoT% : Shots on Target %
    53     G/Sh : Goals per Shot
    54     G/SoT : Goals per Shot on Target
    55     ShoDist : Average distance, in yards, from goal of all shots taken (Does not include penalty kicks)
    56     ShoPK : Penalty kicks made
```

Another important feature of the start of the program is providing options for the user to allow them to make a choice for what path they will take when using the prototype. The main functionalities my program has can fall under two main groups. These groups are; comparison groups, which allow the user to compare how the data is affected by different variables. Or allowing the user to compare different players within the dataset to see how they differ in their stats. The other option is all of the analysis options. An example of some of the analysis options I have is a dependent heat graph which shows how each stat affects the others. This helps my project as it can help managers or scouts predict how a player will perform based on their previous stats or current

stats.

Similar to the main menu, the implementation is done via print statements. The reason print statements were choses was because they are very clear and easy to transfer over when transferring the code over to a front-end user interface, as python has languages especially for this.



The creation of the menus went exactly as planned due to the simplicity of them as overcomplicated menus can reduce the ease of use and confuse the user.

Comparison Options:

Firstly we will talk about the comparison options and the implementation of each choice.

The first option is comparing a player's performance to the rest of the players in their position. My goal was to implement this by using various libraries such as pandas and numpy to search through the dataframe to retrieve the data from a single player then use it to compare between the other players in their position. As this feature is a recurring theme throughout the project, the code needed to be easily adjustable to suit the other functionalities. Upon completion of the functionalities, they were successfully implemented so they can be adjusted and used for the different parts of the code.

Code:

```
#USED

def Compare_Player_To_Average():

 import pandas as pd

 from tabulate import tabulate



 # user enters their name

 name = str(input("Enter your players name: "))



 CORRECT = False

 # check if in dataframe
```

```
while not CORRECT:
    if name in data.values:
        CORRECT = True
    else:
        name = str(input("Name entered incorrectly! Enter your name: "))


# create a separate dataframe for the player
condition = data['Player'] == name
Player1 = data[condition]


# create a separate dataframe for the other players
pos = Player1.iloc[0][3]
condition = data['Pos'] == pos                #Pos = Position
Rest_Of_Players = data[condition]


# find average of the other players
Rest_Of_Players = Rest_Of_Players.mean()
Player1 = Player1.mean()


# calculate the difference between the second and third columns
diff = Player1 - Rest_Of_Players


# concatenate the dataframes and add the difference column
df_combined = pd.concat([Player1, Rest_Of_Players, diff], axis=1)
headers = ['Stats', name, 'Rest Of The World', 'Difference']
```

```
# print the concatenated dataframe using tabulate

print("\n The averages of each stat in their respective age ranges \n")

print(tabulate(df_combined, headers=headers, tablefmt='psql'))
```

As shown in the code, there are alot of built in features used. This is to increase the amount of people that will be able to use the code without installing any additional features or libraries. The main library that is used is the pandas library to concatenate the libraries together. Upon the first attempt of this the data was not being displayed how it was intended as when creating the 'difference' column, the first time the headers were different . This was an issue as when printing the results after concatenating the results, the tables were messed up as the headers were different, meaning the different columns could not be joined into the same table due to this issue. To fix the issue the headers were declared after the data frames were joined together, this would nullify any issues with clashing headers. In addition to getting the average of the other players in the same position, the difference between the two stats are also shown to help the user analyse the performance of the player. For example if a manager was using this to decide whether a player is over performing or underperforming they can enter the players name to see how they compare to the average of the rest of the players in their position.

This is heavily used in the real world of sports as many teams are transitioning to a data driven approach. Another library that was used to implement this feature is the tabulate library which is a very useful library when displaying information in a neat format.

There are, however, some limitations. These limitations are that the user can only compare one player at a time instead of multiple players. Additional columns could be added for the new players but there would also have to be new different columns for each player compared to each other as well as the average for the rest of the world. As a result, the decision to only allow one player was to make sure users can use the program easier. This is due to the fact the additional columns would make the outputs more complicated and harder to read and understand.

Another limitation for the program is that it is a command line user interface, meaning the only way for the user to interact with the program is via text. Due to this, the project would not be as efficient on mobile phones. To combat this, the code could be altered using the languages Flask or Django that are made to work effortlessly with python to implement the code into a web format. The result of this would allow the project to have a graphical user interface which can have buttons and other interactive features to get feedback from the user.

As most of the comparison options all use the same libraries and have similar features, testing was a vital part of the project to make sure it all worked well together as well as displaying the correct outputs. In order to test my code, the dataset had to be personally analysed to find out what the correct outputs for certain searches were. Once there was a set of players that would be displayed from a certain search, the functionality with this feature had to be run to see whether to correct outputs would be displayed.

All of the features displayed the correct outputs, except for the feature to compare a team to the rest of the league. However this was due to a simple coding error where the team was being compared to the rest of the world instead of the rest of the league. The simple fix to this was changing the filtering parameters on the dataset when searching for the rest of the league. Once this was completed the correct output was displayed.

Analysis Features:

The next set of features within the project were the analysis features. The main goal of these features is to allow the user to gain a deeper understanding of the dataset including how each piece of data is related to each other.

For the analysis options, a lot of it consisted of graphical visualisation to depict the data to the user so they can gain a visual understanding of the data. A difficulty or an issue that was encountered during the creation of this part was allowing the user to enter multiple players or various stats , which would then all be turned into graphical visualisations for the user to see. The first issue that popped up was with the graphs they had pre-defined parameters for the inputs. For an example in the code the matplotlib library was used to create various graphs however there was a set amount of input needed each time the user ran the code. As a result of this if the user wanted to enter a different amount of stats to be analysed they would be met with an error telling them that they cannot create the graphs due to the requirements not being met.

To solve this issue the parameters had to be turned into arrays before being sent into the graph.

```
4  def Graphs():
5      # Prompt user for input
6      user_input = input("Enter player name(s) separated by commas: ")
7      # Split input string into a list of names
8      names = [name.strip() for name in user_input.split(",")]
9
10     # Prompt user for input
11     user_input = input("Enter stat name(s) separated by commas: ")
12     # Split input string into a list of stats
13     stats = [name.strip() for name in user_input.split(",")]
14
15     # Define a colormap based on the number of players
16     num_players = len(names)
17     cmap = plt.get_cmap('tab20')
18     colors = cmap(np.linspace(0, 1, num_players))
19
20     for stat in stats:
21         values = []
22         for name in names:
23             search_value = name
24             search_column = "Player"
25             # find the index of the row where the search value is located
26             index = data.index[data[search_column] == search_value].tolist()[0]
27             column_name = stat
28             column_index = data.columns.get_loc(column_name)
29             values.append(data.iloc[index][column_index])
30         # Plot bar chart with appropriate colors
31         plt.bar(names, values, color=colors)
32         plt.title(stat)
33         plt.xticks(rotation=90)
34         plt.show()
35
```

On lines 6-13, the inputs are being collected from the user and turned into usable data for the program. This is so the data is in the correct format for the libraries to use. Next, on line 31, we feed the arrays into the line of code that accepts the arrays as an input for the creation of the graph.

This then allowed the user to enter various names and stats while still being able to change the amount of names they entered each time.

However, once this was changed there was still an issue with validation meaning the inputs entered by the user were not being checked to make sure if they are inside of the dataset.

```
# Prompt user for input
while True:
    user_input = input("Enter stat name(s) separated by commas: ")
    # Split input string into a list of stats
    stats = [name.strip() for name in user_input.split(",")]
    # Check if stats are present in the DataFrame
    invalid_stats = [stat for stat in stats if stat not in data.columns.values]
    if not invalid_stats:
        break
    print("Invalid stat(s):", ", ".join(invalid_stats))
    print("One or more of the stats entered is not present in the DataFrame. Please enter valid stats.")
```

The image above shows the ‘while’ loop implemented to make sure the inputs are valid and does not allow the user to exit the loop until the inputs are checked. After testing the new loop it has added an extra layer of error handling to make sure the experience when using the program is much smoother.

Another feature that had some issues with the implementation was the player scouting system, which is one of the standout features of the project. The feature allows managers, scouts and owners to search for players that are above average in certain aspects which their team needs strengthening in. As well as this the program also shows if there are any players that perform 20% above the average as well as the top performer(s).

One of the issues that was encountered during the testing process was getting the higher tier players from the dataset. The reason for this was the dataset was not given a proper condition to search the dataset with, thus, resulting in the wrong players being submitted to be displayed back to the user.

A small snippet of code will show how the condition is declared then passed into another argument to search through the dataset to gather the information and players that fit within the users search.

```
c = a.loc[a[STAT] == (a[STAT].max())]
```

The code snippet above shows how the dataset is sorted to only show players who have a max stat (the highest in a selected region). By putting the parameter outside of the original parameter, it has provided more accurate and consistent results.

The only other part of the analysis section that proved difficult to implement was the linear regression model due to the size of the dataset. Originally the model would not function due to there not being enough pieces of data to test and train the dataset. However at the time, the built in linear regression library was not being used. After some research and testing, the linear research library was implemented into my project resulting in the linear regression model to work as well as be visualised on a graph.

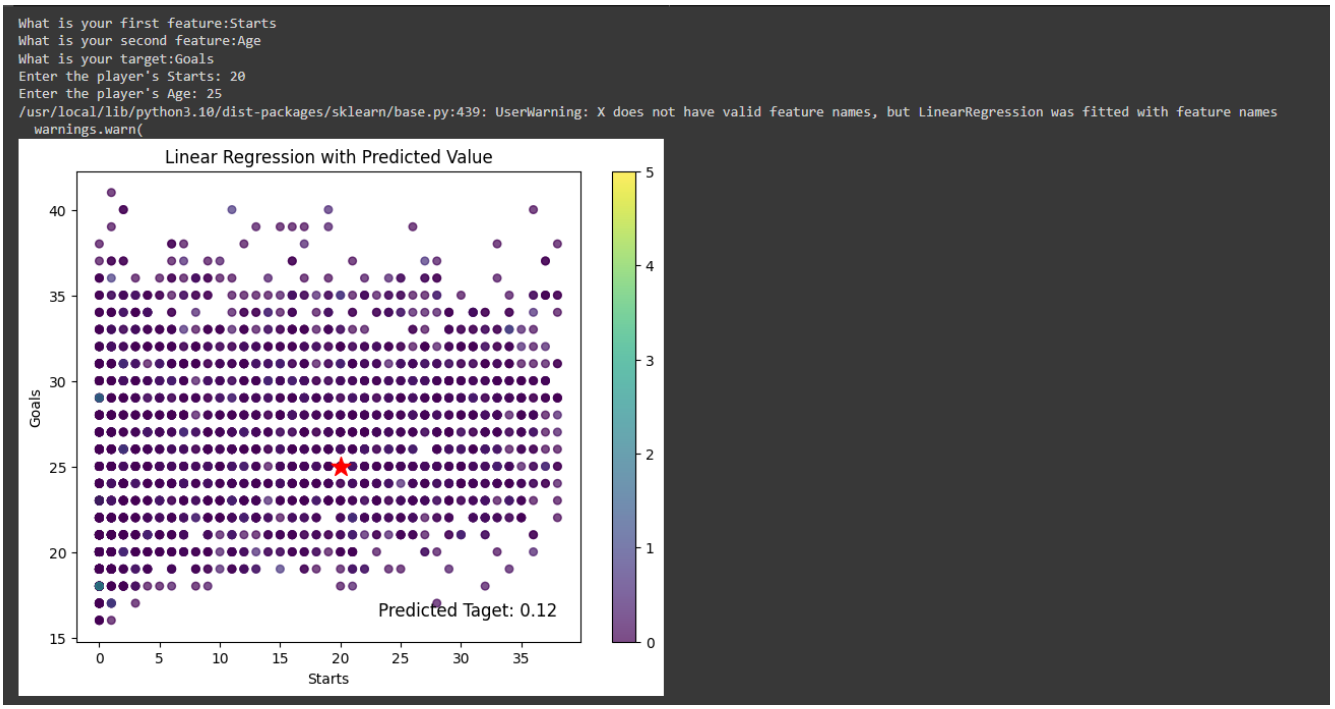
The model allows the user to decide what parts of the dataset they would like to use as features and targets so they have full customizability over the model and outputs. This is because the user needs to be able to predict whatever they need from the dataset, they can also use the dependency feature to use stats that have direct correlation with each other.

The reason adding a predictive multi-linear model was important is to allow teams that are not in

the top 5 leagues to still be able to use parts of the project to help them gain an edge over their competitors.

For example a team can use a player's stats from a previous season to predict whether their predicted goals per game or their predicted assists will be enough to warrant them a place in the team. The result of this would be a more effective team for the upcoming season resulting in more success.

The following screenshots show the outcome of the multi-linear regression model.



The user is prompted to enter two features that are used to predict the target attribute.

In the screenshot we say we want to use ‘Age’ and ‘Starts’ as features to predict the ‘Goals’.

For this run through of the program we imagine we have a player who has started 20 games this season, also, the player is 25 years old.

The model then predicts the player to score 0.12 goals per season and displays the outcome as a visualisation so the user can see how the player compares to the rest of the players in the dataset.

This will allow them to decide whether the player they are predicting for is worth the time and resources of the club to either sign or develop.

Summary:

To conclude the implementation, it was clear there would have to be some self taught skills that are required for me to complete the project and the sub-tasks within it. I explored the different options or alternatives to each problem to decide which route would provide the best outcome for the project to ensure the highest quality work is produced. Overall, when compared to other products similar to mine, they do not supply or offer the various amounts of features that this project does, thus increasing the usability for the majority of users.

It was important to me to evaluate the user needs for the project to deduce what functionalities would need to be present within the code and prototype. The future of this project would be to implement the project into a website to allow for online use wherever the user is, so they would not need to have the program downloaded onto every device. This would also make the project accessible for more people.

My final thoughts on the outcome of the project are very positive as the result has achieved my targets for my prototype and has set up the solid foundations for me to create an app or website for this project. Furthermore, the code is set up so it is easily readable and understandable for any future users so they can use, adjust and edit the code to give them the results they want.

5. Conclusion

Data Collection:

When starting my project a main concern was finding a dataset that provided correct information for my project, as wrong information could lead to legal and ethical issues if the project went public. The dataset had to also be available for me to use and not a private dataset so there would be no issues when using the code throughout the project timeline. The dataset I found was on kaggle which is a highly credited dataset site so it would be a valid and credible dataset to avoid any issues within my project. The dataset was also fully compatible with python also, there would be no risk of any legal issues from me having to change the data.

Overall, the data collection did not create many issues in the long term as the one found at the beginning was almost perfect, apart from the delimiter issues spoken about earlier in the report. To make sure the dataset was valid, many peers and the supervisors were asked about the dataset to get their opinion and advice on it.

Time Management:

When monitoring my project, time management can be a big issue and a limiting factor for how high quality the piece of work produced is. For this project, there were some time management issues that almost had major drawbacks on my project. For example, there was a lot of work that needed to be done outside of the project workspace which took time away from the code and report. The way this was combated was through the advice of all of my supervisors who provided me with many ways to allocate my time better.

Another way that my time management was improved was by swapping how I went about actually completing the work. Instead of trying to code for hours without a break, I instead changed to working in 50 minute intervals with a 10 minute break and did this until my designated task was completed. Due to this change I noticed a vast improvement with the quantity as well as the quality of work that I was producing in the same time frame.

Planning:

My project planning had to undergo some major changes from the start of the project to now due to the resources that were being used as well as the different ways that the work was being shared with the supervisors.

To begin my main form of project planning was through an excel version of a gantt chart and a kanban board. But, this was causing issues as it was time consuming to edit compared to other platforms that provided this content for free.

The website I changed to was teamwork.com that allowed me to create multiple task lists for each milestone so I could easily monitor and manage the tasks I had left as well as the time constraints that each task had to be completed by. Another useful feature is that different team members can be added so they can see the progress. This helped me with showing my work to my supervisors .

This was needed as google drive kept cousin issues when it came to sharing the work with my supervisors as it did not give them access to my files. This leads to communication errors and faults during the work timeline that could have easily been avoided.

Motivation:

My motivation for my project was mostly high, although there were some issues that caused my motivation to dip. Some of these issues were related to my personal life which affected my ability to complete the work, which made my motivation plummet as the stress was increasing drastically. When communicating this with my supervisor, I was given useful information and support thus, helping me successfully work around these issues and gain all my motivation back to complete the project.

A lack of motivation can have serious effects on a project as it can reduce the quality of work resulting in a greater workload in the future when going back to correct the issues created. Furthermore, it can cause the person doing the work to make poor decisions, reducing the useability of the project.

My main motivation which made me want to complete the project was to understand the background behind how a club operates during a transfer window. Also this would help me understand what types of analysis are used in the sport of football.

Future Work:

For the future of this project my main goal is to create a website that holds this prototype to increase the amount of people that are able to use it. Aswell as that it will reduce any ethical issues revolving around unfair advantages for teams using this software to gain a competitive advantage over their peers.

To do this, built in languages for web integrated python such as django or flask would need to be learned to make sure the code can be fully implemented into the webspace.

The reason I would like to do this as well is to challenge myself even more. The reason this was not incorporated into this project is because it would be a huge workload that would reduce the quality of the rest of the project within the timeframe given.

6. References

GUIDANCE (text in blue can be deleted from your final submission)

In this section, you **must** reference any sources used in your work. Typically, these sources will have come up during the investigation and related work sections. Your referencing must use the IEEE referencing style IEEE Citation Guidelines2.doc (ieee-dataport.org) .

It is **highly** recommended that you use reference management software such as Mendeley or Zotero.

Many students ask how many references are required. That is like asking how long a piece of string is. Your project should have as many references as is required for it. However, having few references indicates that no thorough investigation has occurred.

[1] García-Aliaga, A., Marquina, M., Coterón, J., Rodríguez-González, A. and Luengo-Sánchez, S., 2021. In-game behaviour analysis of football players using machine learning techniques based on player statistics. *International Journal of Sports Science & Coaching*, *16*(1), pp.148-157.

[2] Rajesh, P., Alam, M. and Tahernezhadi, M., 2020, July. A data science approach to football team player selection. In *2020 IEEE international conference on electro information technology (EIT)* (pp. 175-183). IEEE.

[3]Davis, N.W. and Meyer, B.B., 2009. Qualitative data analysis: A procedural comparison. *Journal of Applied Sport Psychology*, *21*(1), pp.116-124.

[4]Schoenfeld B. 2019 How Data (and Some Breathtaking Soccer) Brought Liverpool to the Cusp of Glory - The New York Times Last Accessed at 30th March 2023

[5] P. S. Harsha Vardhan Goud, Y. Mohana Roopa and B. Padmaja, "Player Performance Analysis in Sports: with Fusion of Machine Learning and Wearable Technology," *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2019, pp. 600-603, doi: 10.1109/ICCMC.2019.8819815.

[6] Cotton, R. 2022 How Data Science is Changing Soccer | DataCamp Last Accessed 30th March 2023

[7] Stojkovski B, 2021 How Data Analytics Is Transforming Football Last Accessed 2nd April 2023

[8] No author 6 Essential Advantages of Pandas Library - Why Python Pandas are Popular? - DataFlair Last Accessed 5th April 2023

[9]No author Top 10 advantages of Matplotlib in Python - TG Last Accessed 10th April 2023

[10] No Author Introduction to Matplotlib Library in Python Last Accessed 10th April

[11]

Wickramarachchi A.2020 Advantages of using NumPy for numerical operations. Speed gains and additional features offered by NumPy. | The Startup Last Accessed 13th April 2023

[12] Neuralis 2023 From Messy Data to Polished Tables: How Tabulate Can Help You in Your Python Projects - AITechTrend. Accessed 13th April 2023

[13]Di Gravio. W, 2021 The Real Story Behind 'Moneyball' Accessed 15th April 2023

[14] Gorst. P, Liverpool's data-driven transfer model at Melwood hailed as a Premier League leader Accessed 15th April 2023

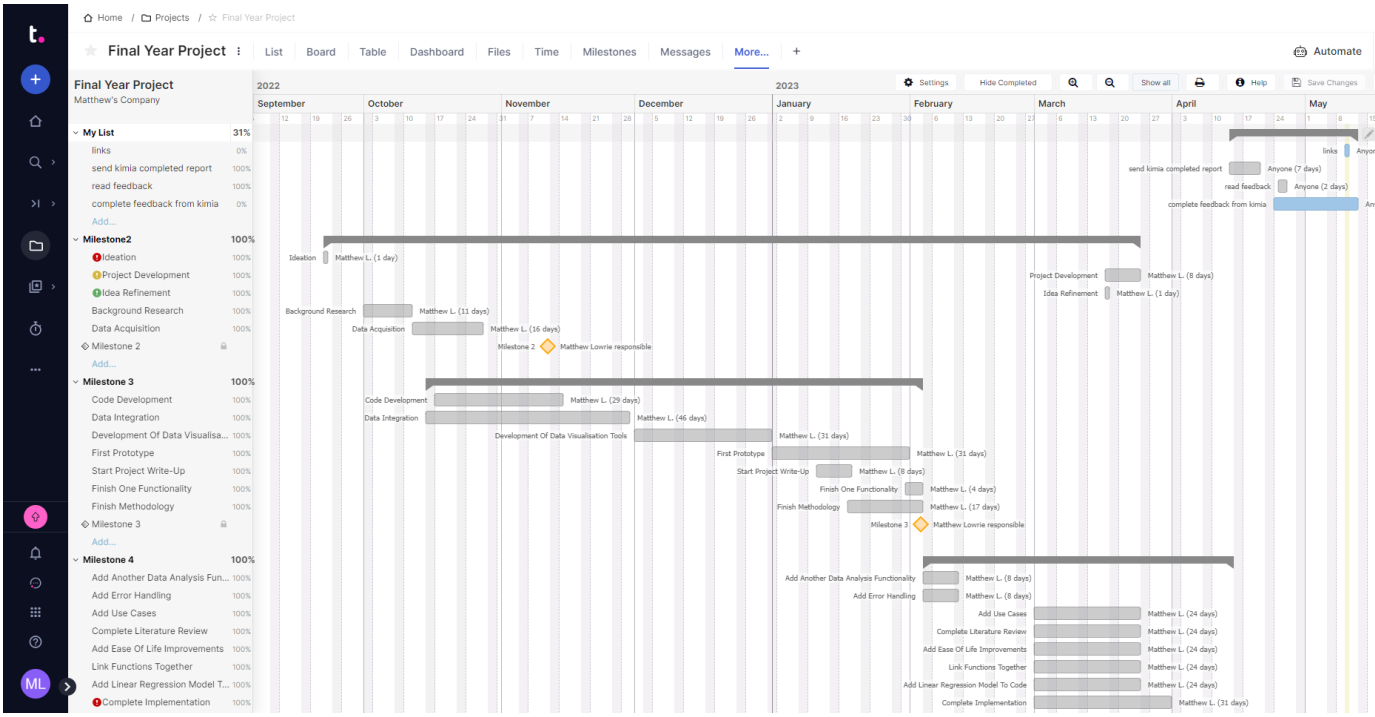
[15] No author Why Do We Need Seaborn? | Advantages and Disadvantages Accessed 10th March 2023

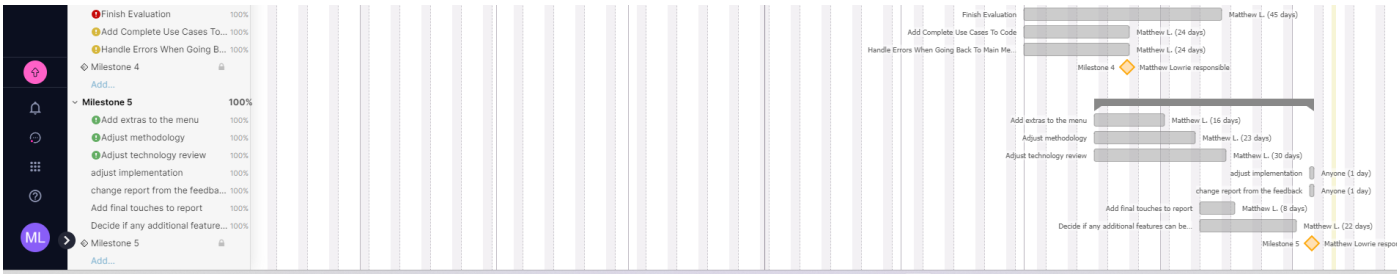
DATASET: https://www.kaggle.com/datasets/vivovinco/20212022-football-player-stats

7. Appendices

GitHub FYP repository - contains all of the project files including project code and datset
Project Initiation
Project Proposal
Mid-Point Review Form
Project Management - Gannt Chart, Kanban Board, Task Lists

Gantt Chart:




Finish Evaluation 100%
Add Complete Use Cases To... 100%
Handle Errors When Going B... 100%
Milestone 4
Add...
Milestone 5 100%
Add extras to the menu 100%
Adjust methodology 100%
Adjust technology review 100%
adjust implementation 100%
change report from the feedba... 100%
Add final touches to report 100%
Decide if any additional feature... 100%
Milestone 5
Add...
ML
Finish Evaluation
Matthew L. (45 days)
Add Complete Use Cases To Code
Matthew L. (24 days)
Handle Errors When Going Back To Main Me...
Matthew L. (24 days)
Milestone 4
Matthew Lowrie responsible
Add extras to the menu
Matthew L. (16 days)
Adjust methodology
Matthew L. (23 days)
Adjust technology review
Matthew L. (30 days)
adjust implementation
Anyone (1 day)
change report from the feedback
Anyone (1 day)
Add final touches to report
Matthew L. (8 days)
Decide if any additional features can be...
Matthew L. (22 days)
Milestone 5
Matthew Lowrie respon