

---

# SMARTSCORE

---

Machine Learning Powered Fantasy Football Application



APRIL 28, 2025

AARON DOYLE  
C00272515

## Table of Contents

<i>Introduction .....</i>	<i>2</i>
<i>Description of Project.....</i>	<i>3</i>
<i>Description of Conformance to Specification and Design .....</i>	<i>8</i>
<i>Description of Learning .....</i>	<i>10</i>
<i>Review of Project.....</i>	<i>12</i>
<i>Acknowledgements .....</i>	<i>14</i>

## ***Introduction***

This report will be outlining the overall result and process in the making of the machine learning powered web application SmartScore. This was made for my final year project, I wanted to do this as my project because I have been playing fantasy football for many years and there's a lot of time and effort that goes into researching things such as form, injuries and fixtures every week. This project aimed to bridge the gap between casual and avid fantasy football players by providing machine learning powered recommendations to help aid managers in selecting their teams for the upcoming game weeks and compete against their friends in these leagues.

This is a web application that is written in Laravel PHP, Tailwind CSS, JavaScript and the machine learning was all written through Python and Jupyter notebooks. I chose this tech stack as I had used Laravel on my internship during third year and I really like how it was able to handle the full stack of a web application from using blade files on the front end to controllers in php on the backend. It was something I really wanted to further my knowledge on and doing this project has really allowed me to do so.

## *Description of Project*

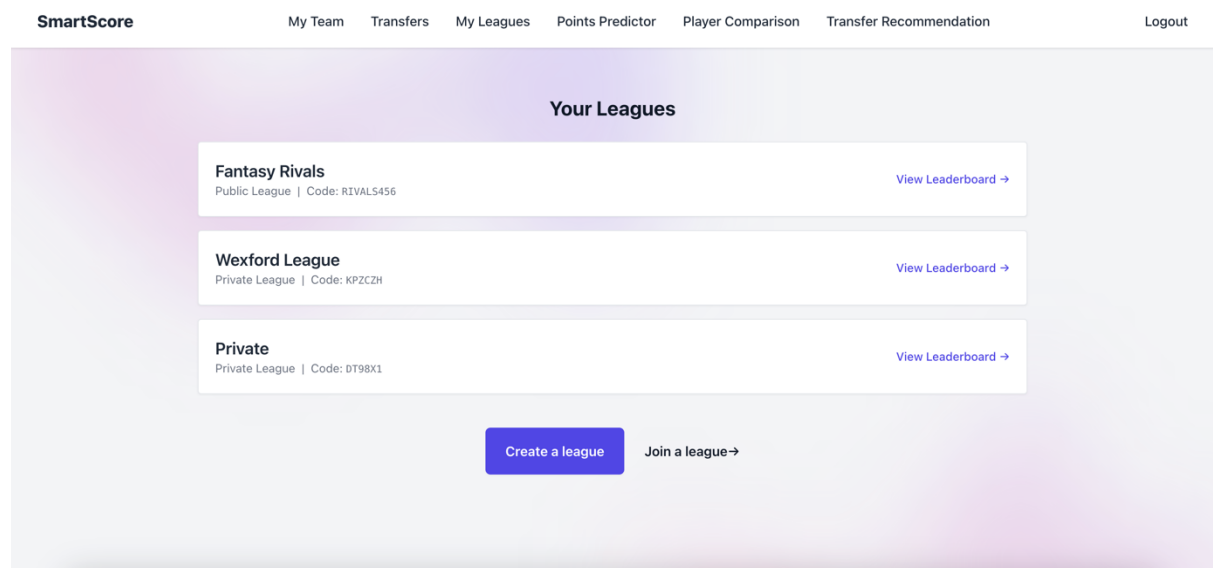
SmartScore is a machine learning powered fantasy football web application that aims to help users in their decision making on a week-by-week basis with predictions and recommendations for their team.

Some of the core features and functionality in this application include:

- **League Creation** – Users can create leagues with their friends
- **Team Selection** – Users can pick their starting eleven and subs every week
- **Player Transfers** – Users can transfer 3 players a week in their team
- **Points Per Week Predictions** – Machine learning predictions for every game week
- **Player Comparison Model** – Users can ask AI who is the better choice of two players
- **Transfer Recommendations** – Users can ask this model who the best replacement is for a player in their team
- **User Authentication** – Users can create an account which has their team info and leagues saved

These were key as I wanted to keep the traditional fantasy football style of playing for familiar users and integrate this seamlessly with the machine learning to help them along the way.

### *League Creation:*



As can be seen in the screenshot above this is the leagues page for the user, I have a few leagues setup whilst I was creating this feature. This works by creating a league by giving it a name and then the app generates a random six-character string which is the league code which can be then given to friends and they can click join league, enter that code and they're now in the league too. This format was inspired by the way Fantasy Premier League do their league system.

## Team Selection:

The screenshot displays the 'Team Selection' interface. It features a central column of player selection options, each with a 'Pick' button and a dropdown menu showing the player's name, team, and points. The players are categorized by position:

- Goalkeepers:** Pick Goalkeeper 1: Kepa Arrizabalaga (Bournemouth) - 5 pts
- Defenders:** Pick Defender 1: Rayan Ait-Nouri (Wolves) - 1 pts; Pick Defender 2: Kristoffer Ajer (Brentford) - 0 pts; Pick Defender 3: Manuel Akanji (Man City) - 1 pts; Pick Defender 4: Nathan Aké (Man City) - 0 pts
- Midfielders:** Pick Midfielder 1: Elliot Anderson (Nott'ham Forest) - 5 pts; Pick Midfielder 2: Joe Aribo (Southampton) - 0 pts; Pick Midfielder 3: Asher Aghionne (Crystal Palace) - 0 pts; Pick Midfielder 4: Marco Asensio (Aston Villa) - 0 pts
- Forwards:** Pick Forward 1: Harvey Barnes (Newcastle) - 3 pts; Pick Forward 2: Joshua Zirkzee (Man United) - 0 pts
- Substitutes (Any Position):** Sub 1: Gabriel Jesus (Arsenal) - 0 pts; Sub 2: Danny Ward (Leicester) - 0 pts; Sub 3: Oleksandr Zinchenko (Arsenal) - 4 pts; Sub 4: Martin Ødegaard (Arsenal) - 4 pts

As can be seen in the screenshot above this is the team selection screen. This is where users can select their starting eleven who will gain them points for the upcoming game week. This screen is initially empty, and users are given a 100m budget to build their team, every player has a price and the better the player is the higher the price is. This is a standard method of team building for fantasy football and I wanted to keep it like this because I want this application to still feel familiar to users who have been playing fantasy football for a long time but still easy to understand for a new user.

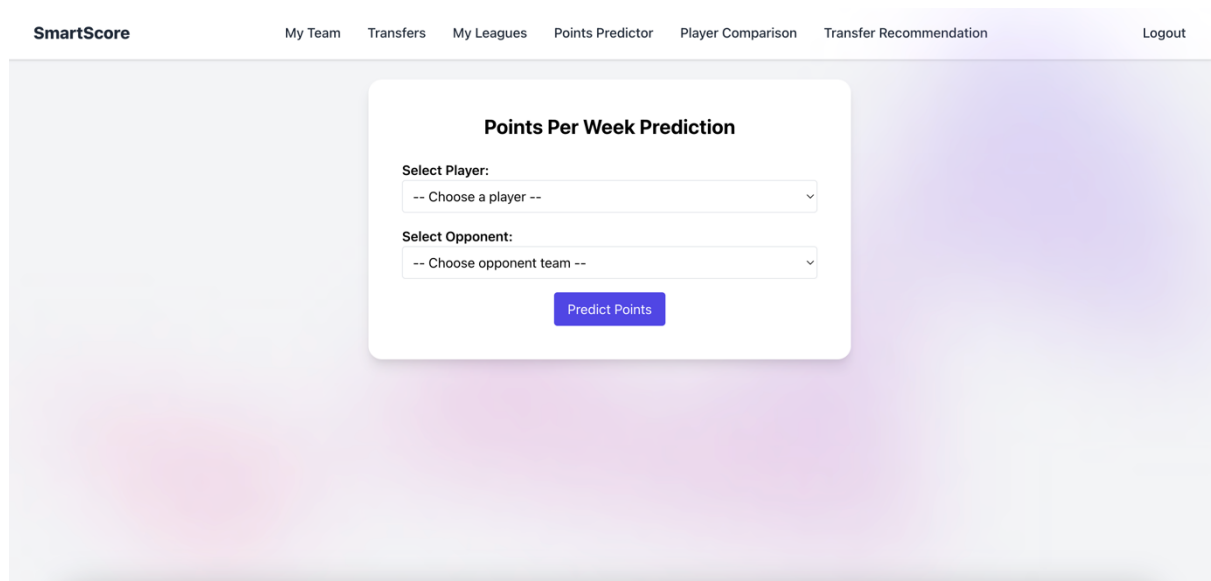
## Player Transfers:

The screenshot displays the 'Make Your Transfers' interface. It features a central column of player selection options, each with a 'Pick' button and a dropdown menu showing the player's name, team, and price. The players are categorized by position:

- Goalkeepers:** Pick Goalkeeper 1: Kepa Arrizabalaga (Bournemouth) - £4.5; Pick Goalkeeper 2: Danny Ward (Leicester) - £4.0
- Defenders:** Pick Defender 1: Rayan Ait-Nouri (Wolves) - £4.5; Pick Defender 2: Kristoffer Ajer (Brentford) - £4.5; Pick Defender 3: Manuel Akanji (Man City) - £5.5; Pick Defender 4: Nathan Aké (Man City) - £5.5; Pick Defender 5: Oleksandr Zinchenko (Arsenal) - £5.0
- Midfielders:** Pick Midfielder 1: Asher Aghionne (Crystal Palace) - £4.0; Pick Midfielder 2: Elliot Anderson (Nott'ham Forest) - £5.0; Pick Midfielder 3: Joe Aribo (Southampton) - £5.0; Pick Midfielder 4: Marco Asensio (Aston Villa) - £6.5; Pick Midfielder 5: Martin Ødegaard (Arsenal) - £8.5
- Forwards:** Pick Forward 1: Harvey Barnes (Newcastle) - £6.5; Pick Forward 2: Gabriel Jesus (Arsenal) - £6.0; Pick Forward 3: Joshua Zirkzee (Man United) - £7.0

This screen style and design wise is virtually the exact same as the team selection screen except it has a different functionality. The transfers screen allows the user to transfer out up to three players a week for players that aren't currently in their team. This is where the user can implement the changes that the AI has recommended for them.

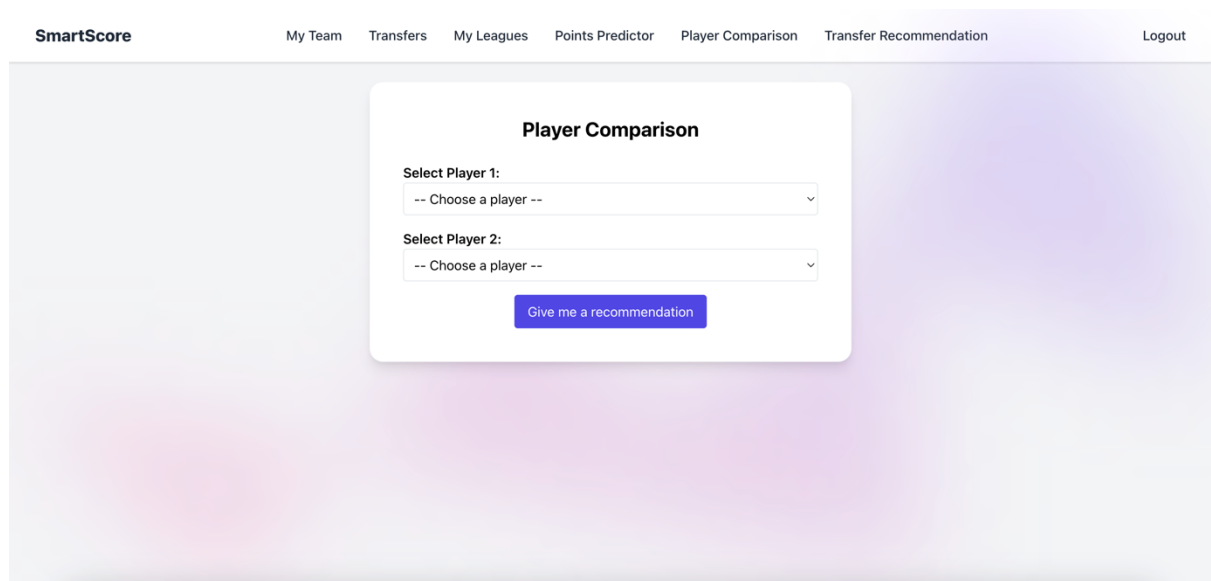
### ***Points Per Week Predictions:***



The screenshot shows the 'Points Per Week Prediction' interface on the SmartScore website. The navigation bar at the top includes 'SmartScore', 'My Team', 'Transfers', 'My Leagues', 'Points Predictor', 'Player Comparison', 'Transfer Recommendation', and 'Logout'. The main content area features a white card with the title 'Points Per Week Prediction'. Inside the card, there are two dropdown menus: 'Select Player:' with the placeholder '-- Choose a player --' and 'Select Opponent:' with the placeholder '-- Choose opponent team --'. Below these menus is a blue button labeled 'Predict Points'.

As can be seen in the screenshot above this is the first of my models and the most complex of them the Points Per Week predictor. This model works by giving it a player and the team they are coming up against in the upcoming game week and it can accurately predict how many points they should get in that matchup. This is based off a multitude of factors including stuff such as the players form, the oppositions form, the players teams form and fixture difficulty for example. There are 4 different random forest models in use here as I needed a different model for each position as there are different key features for each of the 4 main positions on the pitch and they can't be judged in the same way.

### ***Player Comparison:***

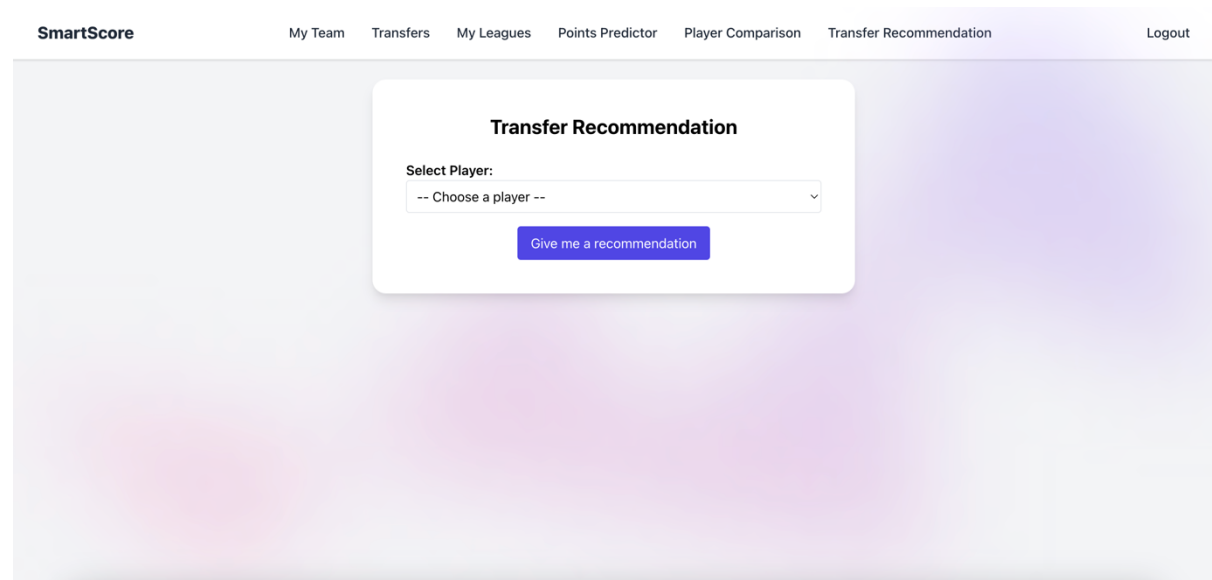


The screenshot shows the 'Player Comparison' interface on the SmartScore website. The navigation bar at the top includes 'SmartScore', 'My Team', 'Transfers', 'My Leagues', 'Points Predictor', 'Player Comparison', 'Transfer Recommendation', and 'Logout'. The main content area features a white card with the title 'Player Comparison'. Inside the card, there are two dropdown menus: 'Select Player 1:' with the placeholder '-- Choose a player --' and 'Select Player 2:' with the placeholder '-- Choose a player --'. Below these menus is a blue button labeled 'Give me a recommendation'.

This is a collection of machine learning models that aim to help the user decide between two players to bring into their team. Deciding between players can be a tough decision to make each week so this model gives the statistically better choice for the manager. Each position is

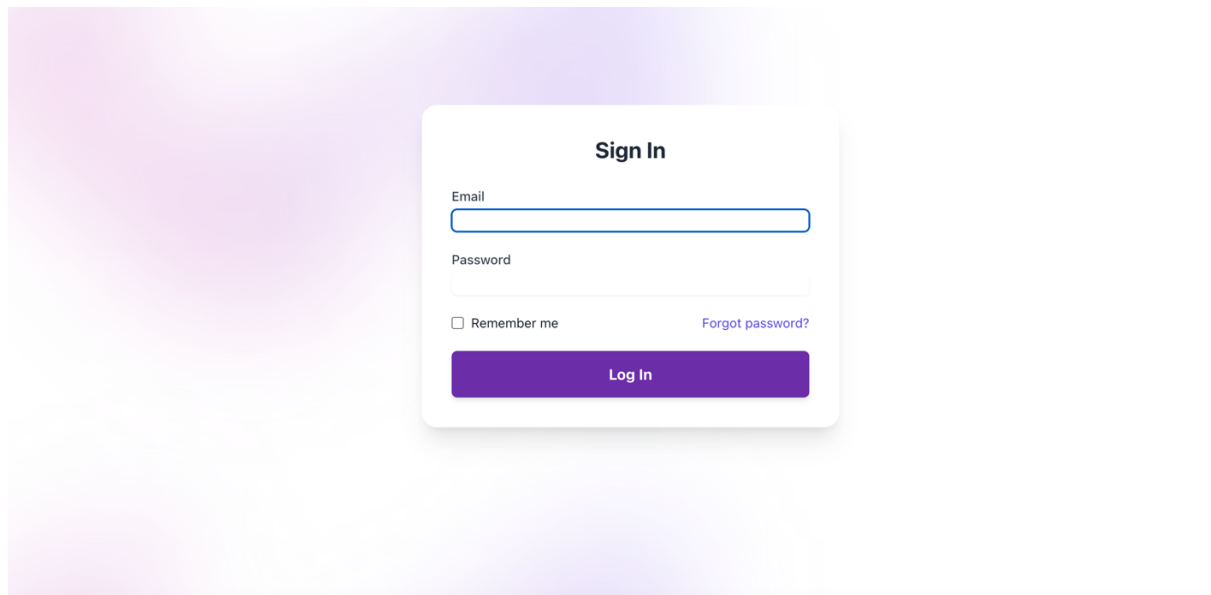
based off of key statistics that real coaches and scouts look at when judging a player. I created this by combining and weighting different statistics per position to create a target variable which the model can then calculate and predict the two players on using a random forest regressor.

### ***Transfer Recommendation Model:***

The screenshot shows a web application interface for 'SmartScore'. At the top, there is a navigation bar with links: 'My Team', 'Transfers', 'My Leagues', 'Points Predictor', 'Player Comparison', 'Transfer Recommendation', and 'Logout'. The 'Transfer Recommendation' link is highlighted. Below the navigation bar, there is a central white card with a purple border. The card is titled 'Transfer Recommendation'. Inside the card, there is a label 'Select Player:' followed by a dropdown menu. The dropdown menu currently displays '-- Choose a player --'. Below the dropdown menu, there is a purple button with the text 'Give me a recommendation'.

This is a collection of K-Nearest-Neighbour models that aims to be able to help the user with their transfers. The aim of this is if for example a player in your team got injured and you aren't sure who will be able to replace them to the same level, this model will give you the most similar statistical replacement for that player. This is very helpful as you need to take budget into account, so it isn't always going to give you the most expensive player to replace you injured player with. This model works by getting the second most similar player to the inputted players statistics as the most similar will just be the inputted player. This once again is a collection of four models as different statistics will matter in different positions and I wanted it to be as accurate as possible.

### ***User Authentication:***



I have implemented the usual login and register logic so that users can create accounts, and everything is saved to their account including teams and transfers.



# ***Description of Conformance to Specification and Design***

In the original specification for this project the intention was to implement a machine learning powered fantasy football web application that would be able to help users make more informed decisions each week for their fantasy teams.

Some of the core features that I aimed to include were as follows:

- ***AI-Based Transfer Recommendations***
- ***Custom League Rules***
- ***AI Team Selections***
- ***Points Per Week Predictions***
- ***Player Comparisons***

## **AI-Based Transfer Recommendations**

This feature ended up working extremely well and very close to how was originally planned. My idea for this model was for example if a user wanted to transfer a player out from their team, I wanted to have a machine learning model that would be able to recommend the most similar player to replace that player in your team. This is exactly what my transfer recommendation model does. I used a different K-Nearest-Neighbours (KNN) model for goalkeepers, defenders, midfielders, and attackers. KNN ended up being a great choice for what I was looking to do as with a KNN model it can map all of the players and give me a list of the most similar ones based off of the inputs to which I just chose to display the most similar. This allows me to focus in on the most important stats per area of the pitch based on research from scouts and coaches online as to what they were looking for when looking at a player. My model now allows you to select the desired player from the dropdown box and then it will query the model and get the best replacement for the desired player based off both of their statistics.

## **Custom League Rules**

Custom league rules were something that I did want to implement at the start as no fantasy football app was utilising this and I thought that it would be a fun way to freshen up the traditional method of scoring in fantasy football and give some new fun and interesting strategies to team selection. Unfortunately, I couldn't implement this for one main reason, this being that for my machine learning models I had to use fantasy points as a target value, so I had to use a specific scoring metric before I trained them. It would have been impossible to have a custom point scoring system as it would require training and building a different model every time a user wanted a prediction which isn't feasible for an application like this. It is something I would have liked to implement however if I had more time to try and think of a solution to this issue as I do think there is some real value in a custom points system.

### **AI Team Selections**

My initial idea for this was to be able to recommend players to put into the users starting eleven from their bench. I wanted to try and develop this idea more as I went along but I was unable to figure out a full feature for this, so aspects and the ideas from this idea ended up being covered in other features that I made such as the player comparison and transfer recommendations. There wasn't enough in this idea to make a singular feature out of. One of the ideas I did think about doing was recommending a starting eleven to the user using transfers, but this would have resulted in the same team being recommended for every user which undermines the point of fantasy football and picking a team in the first place.

### **Points Per Week Predictions**

My initial idea for this was to have a machine learning model that would be able to take in a player and then display the predicted number of points that they are expected to get in their upcoming game week. This feature ended up being a huge success, I deviated slightly from the original plan of having it on the team screen and it being for the next game week. I wanted to have it as a separate screen where you select your player and then who they are coming up against and then it returns the predicted points for that feature. I wanted to go with this approach instead of the original idea of just having it for the next game week because it allows users to plan further ahead and check future fixtures for their players therefore giving more depth and use to the feature. I used a random forest regressor for this and trained it using many different statistics such as the players recent form, the home teams form, the away teams form and the league position of both teams just to name a few. I wanted to make sure fixtures and form were considered as football depends so heavily on these metrics when trying to predict the outcome of a game.

### **Player Comparisons**

The initial plan and idea for this was to have a model that would be able to take in two players and then give a recommendation of who the better of the two would be for the upcoming game week based off their statistics. This ended up being exactly what I implemented, I carried out research into the key statistics per position and then I weighted these by importance which gave me a formula which I could normalise and turn into a target variable. I used a Random Forest Regressor for this once again. I ended up choosing this model a few times throughout the course of creating this project as they are very good at not overfitting to a dataset with them being a collection of decision trees. The final model now works exactly as expected in the original specification.

## ***Description of Learning***

This section can be broken down into two separate types of learning, technical and personal.

### ***Technical Learnings***

Throughout the course of this project my technical understanding of web development and machine learning has improved immensely. I worked with Laravel for my internship in third year where I got some good experience working with the framework but building a project with it from start to finish was unexpectedly challenging. I had to learn about how controllers work and handle most of the logic that's needed on the front end and how to interact between the two. The toughest part about working with Laravel was being able to host the machine learning algorithms. I used Laravel Cloud to host and deploy my project which was very easy and good to use however it came with one major issue and that's that it didn't offer Python support.

This was a huge roadblock for me as I didn't know how to get around this issue. However, after carrying out some research I found Render.com. Render can host Python projects so what I had to do was create a new repo with all of my machine learning models in it and deploy that separately to my project and then call that whenever I wanted to use one of my models. This ended up working very well in the end and there is minimal waiting time when you call a model.

The biggest challenge in this project however was the machine learning and obtaining the data for the models. Obtaining data was a huge problem that I ran into early on as essentially all of the good football statistic APIs are behind paywalls. This is when I came up with the idea to build a web scraper and obtain the data myself. I got all my data from FBref who are one of the leading data providers for football data in the world. I had used their site in the past when looking at in depth statistics and I knew how good they would be for what I was looking to do. The main issue with obtaining data from a web scraper is that there is a lot of pre-processing involved in it as the data is still extremely raw. However, I learned a lot about pre-processing in my Data Science module throughout fourth year and this knowledge turned out to be invaluable for me. This is what all of my models were powered from as I was able to source and obtain every bit of data I needed from the one source.

The other big challenge working with machine learning was trying to figure out what models were the best choice for what I was looking to do. This ended up being quite time consuming and was a process of trial and error at times but eventually after conducting research and building out multiple types of models I got the right models for my needs and tweaked them as needed to get the highest levels of accuracy possible for each of the features that I implemented.

### ***Personal Learnings***

There were a lot of personal learnings that I had throughout this process too which weren't just technical.

Time management was an issue which I had to learn to manage and tackle very early on and then throughout the process of building this project. I found at times I would get tunnel vision

on trying to implement a certain piece of functionality no matter how long it took which in hindsight probably wasn't the best idea. I did improve on this as I went on and towards the end, I had planned out my days ahead so I knew what I needed to get done and I wouldn't stray away from what I needed to get done.

One thing I had to improve on for this project which I didn't anticipate was my ability to research. I ended up having to do a lot of research and reading of interviews and articles especially as I was trying to find out key statistics to look for when building my model so it could be as accurate as possible.

Another personal aspect that I had to improve on throughout the duration of creating this project is how to deal with issues that weren't expected. When building a project like this, things will always go wrong but I learned and got way better at being able to step back, think about the problem and overcome it. A big one for me was deployment, it was vital to me that my web app was deployed and accessible but having to get around it by hosting my models and python code separately was a big challenge but I did manage to do it.

## *Review of Project*

In conclusion, I am extremely happy with how this project turned out as I managed to create all the core functionalities of the application and build what I aimed to do in a fully functioning fantasy football application.

Some of the major successes that came out of this project include the machine learning models. I am extremely happy with how well the models turned out as they were the primary focus of this project. The points predictor, player comparison and transfer recommendation models exceeded my expectations in their accuracy ratings. The decision to create a different model for each of the four main areas of the pitch was key, it was a lot more effort as I had to create a lot more models and filter a lot more data, but it was crucial in the accuracy of the models overall as football is such a broad sport, different statistics are key in different areas of the pitch.

Another major success for me was the deployment of this application, it was a tough process and I had to get help and research along the way but getting the app fully deployed across 3 different locations as the database, Laravel app and my python models are all deployed in different places so getting all of these to interact with each other as expected was a real success for this project.

Some of the things that I would have liked to have completed include the custom points system. I discussed this earlier in the document, but I really do think there is great value in this as it gives a new breath of life to the traditional method of playing fantasy football but I just couldn't figure out a way to do it with the machine learning models.

Another thing I would have liked to have done is automate the points totalling per week. I have a method of doing this and it doesn't take too long I have a scraper which I have to run for each of the ten games I just couldn't figure out a way to automate the process which would have been ideal for the application.

This automation is still an outstanding thing I would have liked to have got done alongside my fixture analyser. I have a formula built to calculate the difficulty of a matchup in the league based off team position and both teams form in their last five games. This was implemented in the machine learning models as an aspect to base their decisions off of but I would have liked to have had this as a solo feature to implement on the application too but unfortunately I ran out of time.

If I was starting this project from the start again there isn't a massive amount that I would change however one thing that I absolutely would make sure that I adhere to is more efficient time management. Especially at the start of this project I was prone to procrastination and poor time management, being on top of this from the start of the project would have left me in a fantastic position coming into the busy seasons of college where I am juggling my other modules and my final project. Another thing that I would have done much earlier is look at the deployment and hosting options for the project as this ended up being quite a headache as I tried three or four different hosting platforms before coming across Laravel Cloud which ended up being the best solution for my issue.

I would advise students in the future who are at the start of their fourth-year project is do as much research as you can before jumping straight into coding. It can seem like the best idea at the time and feel like you are making a lot of progress however unless you are sure of what structure your project needs and what you are aiming to build sometimes it can end up being pointless code as in a few weeks or months' time you might have to go in a different direction with the project.

Overall, I am very happy with the tech stack that I chose for this project. I had a decent knowledge of Laravel PHP coming into this and in completing this project my knowledge is now ten times better of the framework. Building a full application from beginning to end was extremely beneficial in my learning of the framework as on my internship I would be working on most parts but not all of it so seeing how everything interacts was extremely beneficial for my learning.

Using Jupyter Notebooks and Python was a great choice for me in making the machine learning models due to their great selection of libraries. Libraries such as Pandas and Sci-Kit Learn were absolutely crucial in filtering and sorting my data and then building and training my machine learning models. I learned about these libraries in my Data Science module throughout this year which was a massive asset to me as I was completing this project.

## *Acknowledgements*

I would like to take this part of the report to firstly thank my project supervisor Paul Barry. Paul was a massive help to me throughout this project from planning, to code and feature ideas. His insights and knowledge of the area of development were crucial and invaluable to me and the progress of this process.

I would also like to thank my friends for helping me test aspects of this project and bounce ideas off them when I was struggling with Scope Creep.

I would like to thank my lecturers, especially Greg Doyle my Data Science lecturer as without the knowledge that I gained from this module this project would not have been as successful as it turned out to be.

I would also like to acknowledge the use of ChatGPT in parts during this project which was used to help me solve issues in the code and create templates for the front-end alongside Tailwind CSS templates which I then tailored to my own needs.