

SmartScore

Machine Learning Powered Fantasy Football Application



April 28, 2025

Aaron Doyle

C00272515

Table of Contents

[Introduction 3](#_Toc194486881)

[Description of Project 4](#_Toc194486882)

[Description of Conformance to Specification and Design 5](#_Toc194486883)

[Description of Learning 6](#_Toc194486884)

[Review of Project 7](#_Toc194486885)

[Acknowledgements 8](#_Toc194486886)

# 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:

A screenshot of a sports league

Description automatically generated

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:

A screenshot of a computer

Description automatically generated

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:

A screenshot of a phone

Description automatically generated

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.

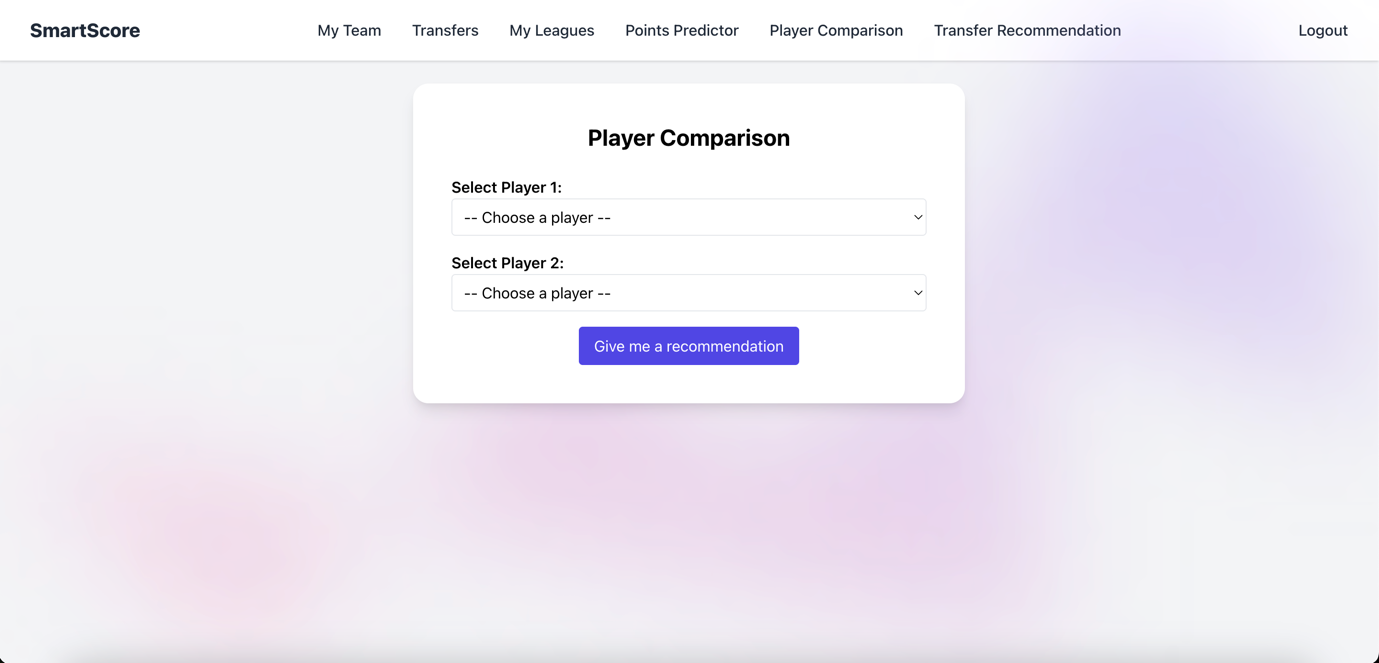
Points Per Week Predictions:

A screenshot of a computer

Description automatically generated

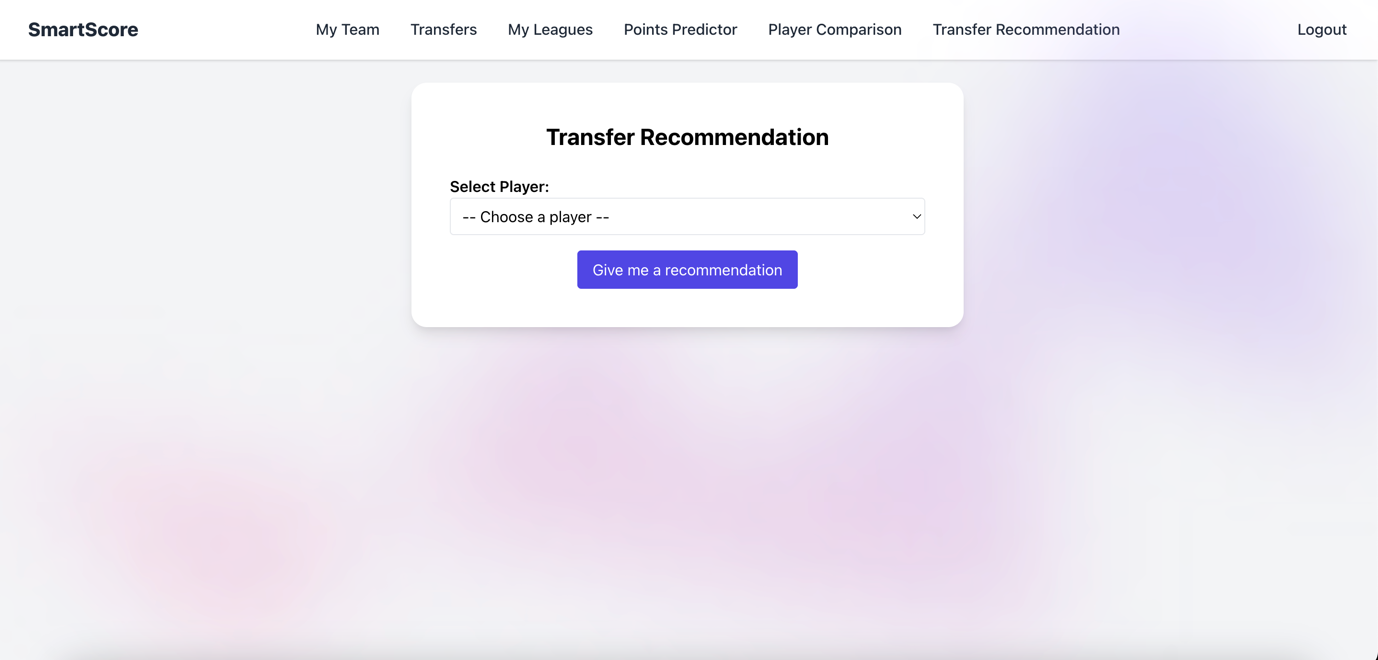
As can be seen in the screenshot above this is the first of my models and the most complex of them the Points Per Week predictor. This model works by giving it a player and the team they are coming up against in the upcoming game week and it can accurately predict how many points they should get in that 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:



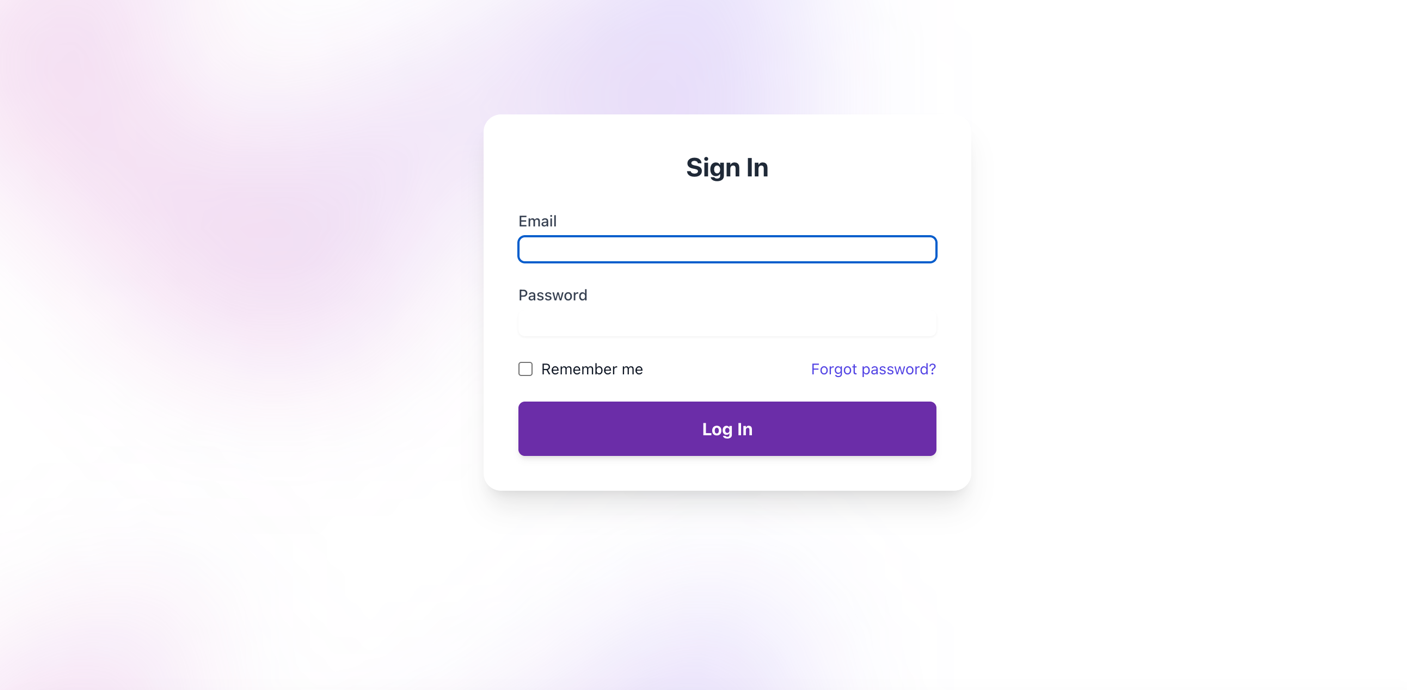
This is a collection of machine learning models that aim to help the user decide between two players to bring into their team. Deciding between players can be a tough decision to make each week so this model gives the statistically better choice for the manager. Each position is based off of key statistics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



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.

# Description of Learning

# Review of Project

# Acknowledgements