

# **SMARTSCORE DESIGN DOCUMENT**

**AN AI POWERED FANTASY  
FOOTBALL APPLICATION**



**AUTHOR: AARON DOYLE, C00272515  
SUPERVISOR: PAUL BARRY**

## Table of Contents

<b>INTRODUCTION .....</b>	<b>3</b>
DOCUMENT PURPOSE .....	3
LARAVEL .....	4
<b>TECHNOLOGIES .....</b>	<b>4</b>
TAILWIND CSS.....	4
PYTHON & JUPYTER NOTEBOOKS .....	4
PANDAS .....	4
SCI-KIT LEARN .....	4
<b>MODELS TO BUILD .....</b>	<b>5</b>
TRANSFER RECOMMENDATIONS .....	5
PLAYER COMPARISON .....	5
FIXTURE ANALYSIS & TEAM PERFORMANCE PREDICTOR .....	5
PLAYER POINTS PREDICTOR & TEAM SELECTION RECOMMENDATIONS .....	5
<b>FIRST ALPHA RELEASE .....</b>	<b>6</b>
OVERVIEW .....	6
DATA SCRAPING .....	6
DATA PRE-PROCESSING & CLEANSING .....	6
MODEL CREATION .....	7

## *Introduction*

SmartScore is a web-based machine learning powered fantasy football application that aims to help fantasy football manager players with their team selection and management decisions throughout every game week. The app aims to do this by creating machine learning models that can take in the data from previous game weeks and analyse these to provide advice and recommendations for the managers so that they don't have to dedicate a lot of time to researching their team.

## *Document Purpose*

The purpose of this document is to develop and outline the internal structure of SmartScore. This will include descriptions of the internal architecture and structure of the site and how it's going to function as a finished product.

This will be outlined using details on each of the technologies I plan to use, details about the models I plan to build and information about the Alpha that I am making for my first iteration.

# *Technologies*

## *Laravel*

Laravel is a PHP web framework that focuses on building full stack web applications, I chose to use Laravel for the main bulk of my website as I have had experience using it in the past and with their blade components it makes it very easy to design front end aspects for the site alongside their routes to make linking the back end easy as well. Laravel is also great for scalability and integration which was another reason why I chose it as it links very well with other frameworks that I will mention further on.

## *Tailwind CSS*

I have chosen Tailwind to handle my CSS as it simplifies regular CSS massively allowing for easier and quicker development on the front end alongside their catalogue of open-source components to build upon, it also links with Laravel extremely well, so it was an easy choice.

## *Python & Jupyter Notebooks*

I will be using python to create my machine learning algorithms as it is the best language by far for building models due to its extensive number of libraries that can be used for the algorithms. Another reason why I am using Python is because it is fantastic for dealing with data when using these libraries. I will discuss some of these libraries in upcoming headings. Jupyter notebooks will also be used to build the models, they are python-based files that allow you to code in blocks and this helps as you can debug very quickly which is a massive advantage when you are making complex models like these.

## *Pandas*

Pandas is a Python framework that is used for reading in datasets to your project, it is great for pre-processing data in a dataset to ensure that everything runs smoothly by the time it reaches your database and model. It is also great at integrating with other libraries that I will discuss further down this list. These factors made it an easy choice for my project.

## *Sci-Kit Learn*

Sci-Kit Learn is another Python framework that provides different types of machine learning models such as decision trees and linear regression that can be used to create the models for this application. It also provides features such as being able to split your data into test and training sets to ensure the model gets the best training possible. All the features that I listed above made this an easy choice to include for my project as it makes the process of making the models way easier.

# *Models To Build*

## *Transfer Recommendations*

For this model my current plan is to use a Neural Network as the type of model. I am currently planning to use this as this will have to be a slight combination of a few models. It will have to consider a few different factors such as the difficulty of upcoming fixtures, current form and stats from the season so far. I think a Neural Network will work well for this model as they are very good at recognising small and niche patterns within the data which will work well for this as that prediction won't be an obvious one to make unlike some of the models.

## *Player Comparison*

For this model I am currently planning to use a Random Forest model as it is a binary result. The model will take in the information of two selected players, it will then be able to analyse their stats, output and fixtures and recommend which player is the better option for the upcoming weeks. A Random Forest model should also work well for this data as there isn't going to be a whole lot of data to work with and Random Forests work well with smaller datasets. One thing that I could potentially do to boost this dataset if necessary is get older season data that exists online and combine it with current season data so that I can train the model with more data.

## *Fixture Analysis & Team Performance Predictor*

For this model I plan to use a Neural Network. I am planning to use this type of model for this predictor because as I mentioned above, they are fantastic at recognising patterns and making predictions based off that. This will be helpful for this model as it will have to go through the current form of both teams and their stats to predict who they think will win an upcoming game. Once again if there isn't enough data for this, I plan to use previous seasons data to train the model as the training can be done based off of matchups from any time period it doesn't just have to be this season.

## *Player Points Predictor & Team Selection Recommendations*

For this model once again, I am planning to use a Neural Network to build it. This is for the same reasons as listed above, I believe that the pattern recognition that this model gives will massively benefit me here as it will be able to recognize the patterns of when a player is likely to gain points based off player form, team form, opposition ranking and their current form. This model will be one of the tougher ones to build I believe as it is going to be nuanced patterns that will have to be found in the data there won't be too many obvious ones. This same model can be used for the team selection recommendations as based off the points that it is predicting a player to get it can then recommend you make a substitute or transfer for example if a player is predicted to go on a poor run of form.

# *First Alpha Release*

## *Overview*

For my alpha I have decided to make a sample machine learning model. For this I wanted to focus on importing data using a web scraper that I have built using Python, then I cleansed it using pandas and finally I plan to build the model using a Decision Tree and this was implemented using SciKit Learns Decision Tree model. The model that I am looking to make is a predictor that will be able to tell if a player will score above 15 goals in a season based off their stats.

## *Data Scraping*

I have built a data scraper to obtain my data from the FBREF website, this website is one of the leading providers for in depth statistics for football and especially the Premier League which is the league that I will be focusing on for my application. This took a lot of time to get working as I had a lot of trouble extracting certain data that I needed to build my data. Eventually after searching through the HTML file I discovered that the table I need was commented out and this was why my code couldn't find the table when I initially was searching for it. I got around this by making a small script that will search the html file and delete the necessary line with the comment symbol and removing this allows the table to be picked up correctly. This took a large portion of my time as it was initially very awkward to scrape anything from the site due to them giving me 403 errors due to bot detection. I got around this issue with the use of headers which essentially work as a form of ID so that the scraper appears as a normal user instead of a bot which stopped this error.

## *Data Pre-Processing & Cleansing*

After I figured out the data scraper and had my data in a CSV file, my first step was to cleanse the data. The model that I wanted to build for this alpha was whether or not a player would score over 15 goals in a season or not so my first step was to focus on the attackers. I did this by searching for the players who were forwards in their position column and moving these to a new CSV just for attackers. I then realised I didn't have a predictive value for this as it is current season data so I divided the players goals by games played and multiplied this by 38 to get a prediction for the amount of goals every player would score for the season. This isn't a perfect solution but I cross checked with previous seasons data and the amount of players hitting the 15 goals mark was very similar to my prediction so I decided to proceed with it. There weren't really any major issues once the data was tidy however there was a double header on the data initially which was causing me a couple of issues but after doing some research I learned that you can skip over this quite easily. Another small issue that I initially ran into because of this is setting the parameters of the model because it couldn't detect the correct header names that I was looking for as it was just taking the first name as the header but this was solved once I figured out how to skip and delete the first header.

## *Model Creation*

For my chosen model I decided to use a decision tree as this model would be providing a binary result as the player either will or won't hit the 15 goal a season mark. This ended up working well as the data was numerical which made it easier to train the model on. The three statistics that I ended up using to train this model were Goals per game, Expected goals per game and non-penalty expected goals per game. I chose these as using per game statistics gives a better idea of how a player is performing overall and it gives a bit more information than just goals as a player could score 5 in one game which can skew predictive stats a bit. This model that I built has a 95% accuracy and can predict whether or not a player will score over or under 15 goals a season based off of the statistics that I listed above. There weren't really any major issues once the data was tidy however there was a double header on the data initially which was causing me a couple of issues but after doing some research I learned that you can skip over this quite easily.