FANTASY FOOTBALL FORECAST TOOL

by

JASWANT JAYACUMAAR

This report encompasses a commercial idea to predict the Captain of a Fantasy Premier League team using machine learning and it talks about accuracy offered by various models.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1. Background

Fantasy Premier League (FPL) is one of the most popular fantasy sports games globally, based on the real-life performances of players in the English Premier League (EPL). In FPL, each participant acts as a virtual manager and builds a squad of real-life EPL players to accumulate the highest number of points over the course of the football season. Points are awarded based on players' actual performance in real matches, including goals, assists, clean sheets, saves, and minutes played while points are deducted for negative contributions such as yellow/red cards or own goals.

Each FPL manager builds a squad consisting of 15 players:

- 2 Goalkeepers
- 5 Defenders
- 5 Midfielders
- 3 Forwards

Each gameweek, the manager must select a starting XI from their squad. One of these players is assigned the **captaincy**, and another can be designated as **vice-captain**. The captain earns **double points** for that gameweek, making the captain choice a pivotal factor in overall scoring and success in the game.

However, several constraints make team and captain selection more complex:

- The total squad must be assembled within a fixed budget (e.g., £100 million)
- A maximum of 3 players may be selected from any single Premier League club
- Formations must follow valid football lineups (e.g., 3-4-3, 4-4-2, etc.)

Given these constraints, strategic selection and forecasting become essential, especially for the captain, whose doubled points can significantly influence weekly and season-long performance.

1.2. Methodology and Objective

This project aims to assist FPL managers in making smarter decisions by developing a **points prediction system** for Fantasy Premier League players, with a special focus on **captain selection**. We aim to build a predictive engine that uses **historical performance data** from past matches to forecast the expected points of players in upcoming gameweeks.

Our methodology involves:

- **Data Collection**: Gathering historical data for EPL players, including their past match stats (goals, assists, minutes played, clean sheets, etc.), team performance, fixtures, and opponent difficulty.
- **Data Preprocessing & Feature Engineering**: Cleaning the data, handling missing values, and generating meaningful features such as form, fixture difficulty rating (FDR), and xG/xA (expected goals/assists).
- **Model Experimentation**: Applying machine learning models (e.g., regression, tree-based models, or neural networks) to predict individual player points.
- **Captain Recommendation**: Based on predicted points, the system identifies optimal captain candidates while accounting for team constraints, budget, and gameweek fixtures.

Ultimately, the goal of this project is to develop an intelligent, data-driven assistant that empowers FPL managers to make informed decisions, particularly when it comes to selecting their weekly captain, a decision that can significantly influence overall scores due to the points multiplier. By leveraging historical player data, fixture context, and predictive modeling, the system aims to offer actionable insights not only for captaincy but also for general squad optimization. This tool aspires to bridge the gap between raw football statistics and fantasy game strategy, helping users consistently gain a competitive edge throughout the season.

CHAPTER 2
THEORY

The foundation of this project lies in developing a predictive system for the FPL, focusing on two core objectives:

1. To determine whether a player is likely to be in the starting eleven for a given gameweek
2. To predict the number of points that the player is expected to earn in that match

To address these objectives, a combination of classification algorithms and regression models is employed. Classification models are used to assess the likelihood of a player starting, while regression models estimate the player's potential points based on historical performance and data.

2.1. Player Selection Prediction (Classification)

In FPL, selecting players who are guaranteed to start is crucial, as non-playing or substitute players usually score fewer or no points. Thus, the first step in our pipeline is to classify whether a player will start in a match based on various features such as historical appearance data, injury status, recent form, and opponent characteristics.

2.1.1. Naive Bayes Classifier

Naive Bayes is a probabilistic classification technique based on Bayes' Theorem. It assumes that all features used for prediction are conditionally independent given the class label. While this assumption rarely holds in real-world data, Naive Bayes often performs well in practice, especially for high-dimensional data.

This classifier calculates the posterior probability of each class and assigns the class with the highest probability. In the context of our project, it helps determine whether a player will start, based on features such as minutes played, team rotation patterns, and fixture congestion.

The key advantages of Naive Bayes include:

- Fast training and prediction time
- Insensitivity to irrelevant features
- Effectiveness even with limited training data

2.1.2. K-Nearest Neighbor (KNN)

K-Nearest Neighbor is a non-parametric, instance-based learning algorithm that classifies new data points based on the majority label of its 'K' nearest data points in the feature space. The concept of similarity plays a crucial role in this method, and similarity is typically measured using Euclidean distance.

In our case, KNN is used to compare a player's current attributes (form, fitness, minutes played) with those of historically similar players to predict their likelihood of starting in the next match. The model does not make assumptions about the underlying data distribution, making it useful when patterns in data are highly nonlinear.

2.1.3. Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees and merges them to improve prediction accuracy and control overfitting. Each tree in the forest makes an individual prediction, and the final output is determined by majority voting (for classification).

This algorithm is particularly well-suited for complex datasets with both numerical and categorical variables. In our project, Random Forest helps capture the multifactorial dependencies that affect whether a player is likely to start, such as fixture difficulty, recent substitutions, and injuries.

2.2. Points Prediction (Regression)

After identifying the players who are likely to be in the starting lineup, the next step is to estimate the number of points they might earn. This is treated as a regression problem, where the target variable is the FPL points scored in a match, and the independent variables include performance metrics, opponent strength, and situational factors.

2.2.1. Logistic Regression

Although typically used for binary classification, logistic regression can be adapted to classify performance thresholds (e.g., whether a player scores more than a certain number of points). It models the probability of a binary outcome based on a linear combination of input features.

In this project, logistic regression is utilized to estimate the likelihood of a player crossing a defined points threshold, thereby assisting in evaluating potential returns from captaincy decisions.

2.2.2. Support Vector Machine (SVM)

Support Vector Machine is a powerful supervised learning algorithm used for both classification and regression tasks. In the regression context (Support Vector Regression - SVR), SVM attempts to fit a hyperplane that predicts a continuous target value while maintaining maximum margin from the closest data points.

SVM is particularly effective in high-dimensional feature spaces and can handle non-linear relationships through kernel functions. This makes it a valuable tool for modeling the complex relationship between match variables and FPL points.

2.3. Summary

The combination of classification and regression models enables a layered prediction strategy. First, the classification models filter out players unlikely to start, minimizing the risk of selecting inactive players. Then, regression models predict the expected points of the shortlisted players, allowing the system to identify the optimal captain for the gameweek and support general squad planning.

This hybrid approach enhances the reliability and accuracy of FPL recommendations by integrating both participation probability and performance prediction into a unified decision-making framework.

CHAPTER 3
DATA ANALYSIS

In order to train any supervised machine learning model, the availability of a reliable and informative dataset is crucial. For this project, our primary objective was to predict the performance of players in the Fantasy Premier League (FPL) based on historical data. However, the initial dataset obtained posed several challenges that hindered direct model training.

3.1. Dataset Assessment

Upon loading and inspecting the dataset, we quickly realized that the raw features provided were insufficient for building a high-accuracy prediction model. A correlation heatmap (see Fig. 3.1) highlighted a significant lack of strong relationships between the existing input features and the target output (i.e., FPL points). This low correlation suggested that the data in its original form, would likely lead to underfitting or ineffective predictions.

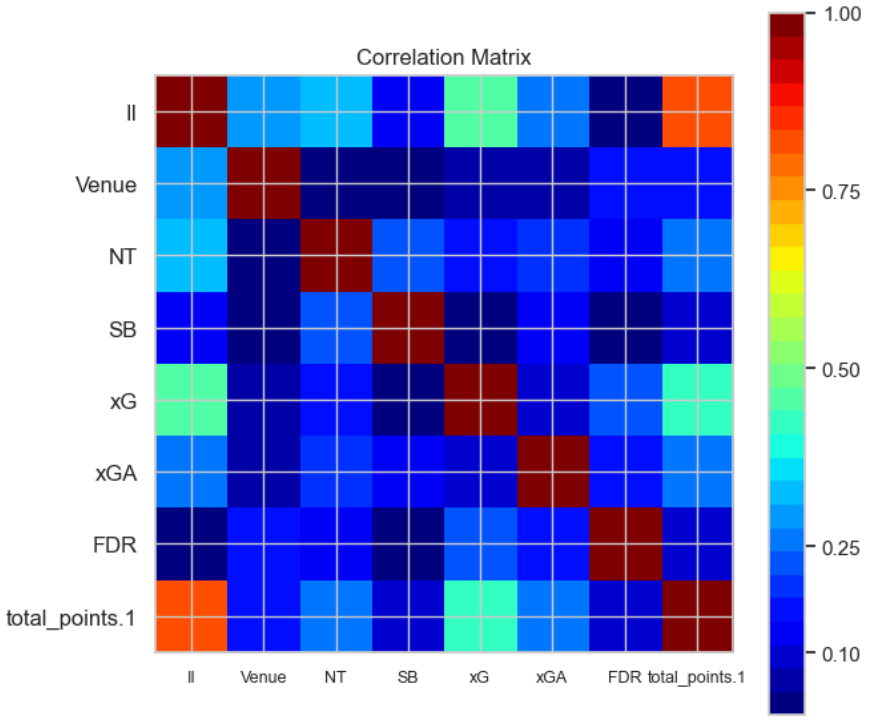

Fig 3.1. Correlation Matrix of Engineered Features Derived from Original Dataset

3.2. Feature Engineering

To overcome the limitations of the original dataset, we turned to feature engineering, a process of creating new, more meaningful variables based on existing ones. The aim was to enhance the predictive power of our model by incorporating derived metrics that better capture player performance trends, opponent difficulty, and community sentiment. These engineered features helped increase correlations and made the models more intelligent and context aware.

Below are some of the key features we introduced:

1. **Form Factor**: This feature was designed to capture a player's recent form. It was computed by taking the average of the player's performance-related stats (e.g., goals, assists, minutes played) over the last four gameweeks. The idea was to give more weight to recent matches, as they often reflect a player's current momentum or drop in performance.
2. **xG and xA (Expected Goals and Expected Assists)**: These are advanced football metrics widely used in analytics. xG estimates the probability of a shot resulting in a goal based on factors like distance from goal, angle, and type of assist. Similarly, xA measures the likelihood that a pass will become an assist. Including xG and xA helped us incorporate more nuanced performance indicators beyond simple goals and assists.
3. **Fixture Difficulty Rating (FDR)**: FDR is a composite metric that reflects how difficult a player's upcoming fixture is likely to be. This was calculated based on the historical performance of the player against the opponent, as well as the current form and defensive strength of the opposing team. This feature allowed the model to account for context, such as whether a player is facing a top-tier team or a struggling defense.
4. **Net Transfers (NT)**: Net Transfers represents the number of FPL users who transferred a particular player into or out of their squad during the current gameweek. This feature served as a proxy for community sentiment and could be used to detect trends, for e.g., a spike in transfers may signal confidence in a player or a reaction to recent performances or injury news.

3.3. Objective of Feature Engineering

The overarching goal of this process was to increase feature-target correlation, improve the model's generalization capabilities, and introduce domain-specific football intelligence into the machine learning pipeline. With these new features integrated into the dataset, the performance of the classifiers and regression models significantly improved in both training and validation phases.

Through this step, we successfully transitioned from a raw, underpowered dataset to a more enriched and insightful one that enabled the construction of smarter, context-aware predictive models.

CHAPTER 4
RESULTS AND DISCUSSIONS

To evaluate the effectiveness of different models in predicting player performance for FPL, we implemented a range of both classification and regression algorithms. Table 4.1 summarizes the accuracy percentages for each model across the training and testing datasets.

Table 4.1. Accuracy Comparison: Training v Testing Datasets

Model	Training Accuracy (%)	Testing Accuracy (%)
Naive Bayes	83	76
KNN	80	81
Logistic Regression	47	19
SVM	53	24
Random Forest	83	95

4.1. Model Performance Overview

From the table, it is evident that classification models, particularly Random Forest and KNN, performed significantly better than regression models. The Random Forest classifier achieved strong accuracy on both the training and test sets, making it the most reliable model for our task. K-Nearest Neighbour also showed strong generalization capability, indicating stable performance without overfitting. On the other hand, logistic regression and SVM underperformed, showing very low accuracy on the testing set—highlighting their limited ability to handle the complexity and variability in the dataset. Figure 4.1 displays the accuracy output for each classification model as shown in VS Code, with subfigure (a) comparing Logistic Regression and SVM, and subfigure (b) comparing KNN and Random Forest.

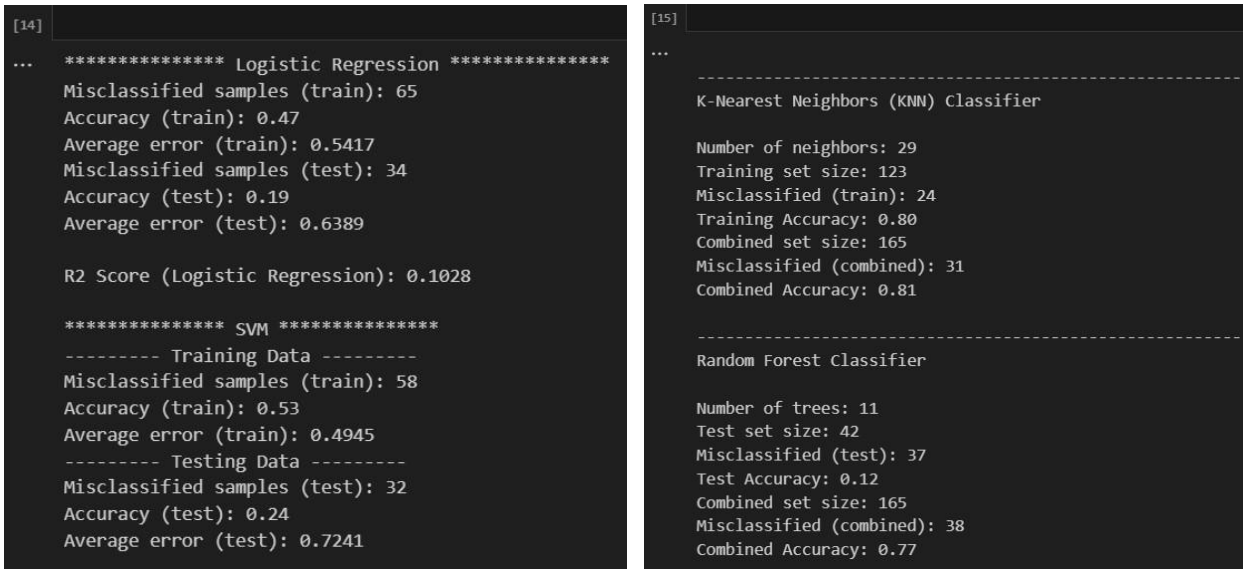

Fig 4.1. Model Accuracy Comparison Displayed in VS Code: (a) Logistic Regression and SVM, (b) KNN and Random Forest

Figure 4.2 shows the confusion matrices comparing the performance of the custom Naive Bayes classifier on the training data (left) and the Scikit-learn GaussianNB classifier on the test data (right). The output generated by the Naive Bayes implementation includes both textual accuracy metrics and visual confusion matrices for evaluating the performance of two Naive Bayes models: a custom implementation and scikit-learn's GaussianNB. The textual output reports a strong training performance for the Custom Naive Bayes model and moderate test accuracy for GaussianNB. These figures provide a quick overview of how well each model classifies the binary outcomes (0 and 1), but the confusion matrices give a much more detailed insight.

The left-hand plot in the figure represents the confusion matrix for the Custom Naive Bayes model on the training set. It confirms that the model performs reasonably well across both classes, with a good number of correct predictions for both positive and negative labels. In contrast, the right-hand plot depicts the confusion matrix for the GaussianNB model on the test set. Although the reported accuracy appears acceptable, the model completely fails to predict any class 1 labels correctly—all actual class 1 samples are misclassified as class 0. This lack of class balance highlights a significant limitation, reinforcing the idea that accuracy alone can be misleading. The image and text together validate the importance of confusion matrices and other diagnostic tools in evaluating classification model performance beyond a single metric.

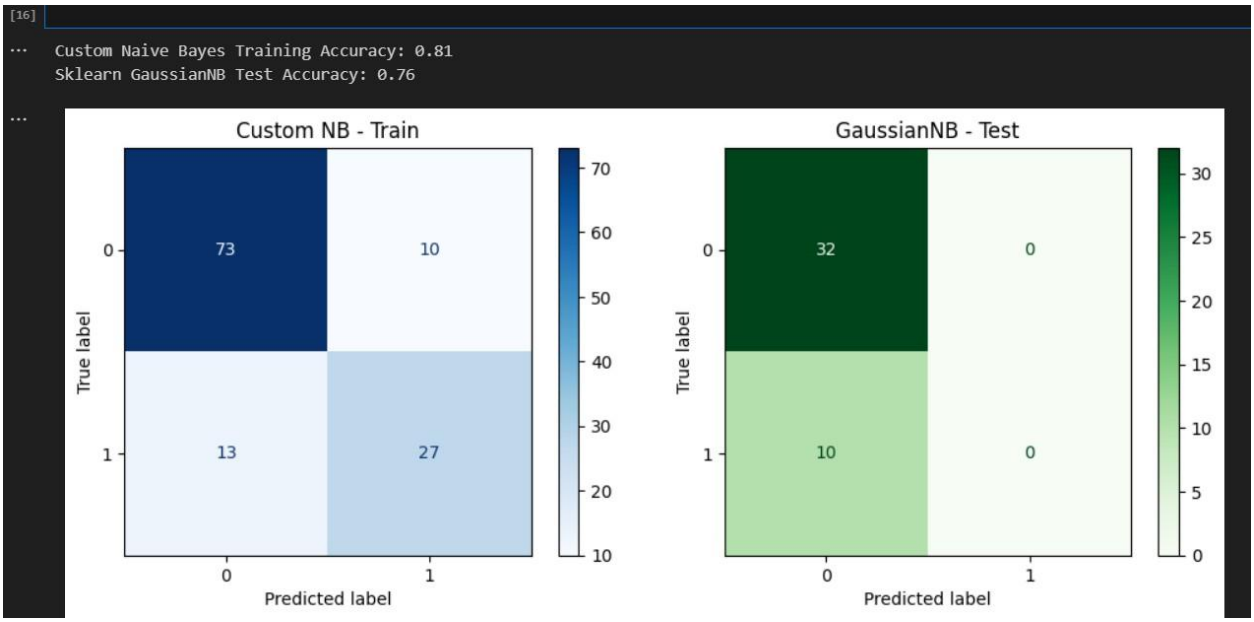

Fig 4.2. Confusion Matrices Comparing Custom and Gaussian Naive Bayes Classifiers

4.2. Regression vs. Classification for FPL Points Prediction

Our initial approach considered the use of regression models to predict the exact number of points a player might earn in a match. However, the results showed that regression techniques were not effective for this problem due to the high variability and randomness inherent in sports data. Events such as injuries, unexpected tactical changes, or player rotation introduce inconsistencies that are hard to capture with a regression model. For instance, a top-performing player might be benched or injured unexpectedly, leading to a sudden drop in points that cannot be reliably predicted using past data.

As a result, the regression models not only learned patterns that were noisy or anomalous but also failed to generalize to unseen data, as reflected by their poor test accuracy.

4.3. Classification as a Practical Alternative

To address this challenge, we reframed the problem from regression to classification by categorizing players into two performance tiers:

- High scoring
- Low scoring

This binary classification proved to be a more robust and practical approach. It eliminates the need to predict the exact number of points and instead focuses on identifying players who are likely to deliver strong performances.

This is especially important for captain selection in FPL, where the captain earns double the points. Choosing a high-scoring player as captain can significantly boost weekly totals. Thus, even a binary prediction (high vs. low performance) is sufficient for making strategic decisions and maximizing total points.

4.4. Future Improvements

While the current model offers a strong foundation for predicting FPL player performance, there is significant scope for enhancement. These improvements span both the algorithmic design and the data representation, and can help address the current limitations in accuracy, generalizability, and usability.

4.4.1. Algorithmic Enhancements

i. **Multi-Layer Neural Networks**: The current dataset exists in a non-linear feature space with limited directly correlated inputs. A multi-layer neural network can model these complex patterns more effectively through deeper architectures and hidden layers, potentially improving prediction accuracy.

ii. **Recurrent Neural Networks (RNNs) and LSTM**: Traditional models lack memory of past player form. RNNs inherently handle sequential data through feedback loops, allowing the model to consider recent performance trends. LSTM (Long Short-Term Memory) networks further enhance this by distinguishing between short-term and long-term player performance, which is especially useful for modeling form.

iii. **Segmented Modeling by Player Characteristics**: Instead of applying a one-size-fits-all approach, we can build position-specific or volatility-aware models. For example, defenders and forwards behave differently, and separating models based on team, role, or risk profile could yield better results.

iv. **Expanding and Diversifying Training Data**: Training the model on a larger and more diverse set of players will help reduce overfitting and improve model

robustness. This will enable the model to learn better generalizations from the variability in player behavior and match conditions.

v. **Validation with Diverse Player Types**: Future testing should include targeted validation across different categories of players such as consistent performers, differential picks, or budget options to ensure the model performs reliably across all user scenarios.

4.4.2. Roadmap and Feature Addition

i. **Predicting the Starting 11**: Extend the system to predict the full starting lineup for a given gameweek, helping users manage not just their top scorers but their bench and rotation risks as well.

ii. **Point Predictions with Suggested Transfers**: Build a complete pipeline that not only predicts expected points for each player but also suggests optimal transfer decisions based on performance, form, and budget constraints.

iii. **Alternative Player Recommendations**: Rather than suggesting only one best pick, the model can offer a set of high-potential players who are likely to score similarly. Final selection can then be based on user-defined preferences or constraints (e.g., favourite teams, differential picks), which may not be represented explicitly in the dataset.

Incorporating these improvements will not only enhance model accuracy but also make the tool more **intelligent, flexible, and user centric**. As Fantasy Premier League continues to evolve and attract millions of users, these advancements can transform the system into a fully functional **fantasy assistant**, guiding users' week by week through analytics-driven decisions, transfers, and captain picks.

CHAPTER 5
CONCLUSIONS

While our results demonstrate that classification-based models, particularly Random Forest and KNN perform well in predicting FPL outcomes, it is essential to recognize the broader context in which such models operate. In addition to this, we explored probabilistic modelling through both a custom-built Naive Bayes classifier and a standard implementation using scikit-learn. The custom model showed fairly strong performance during training, suggesting it was able to capture some of the underlying patterns in the data. However, when tested on unseen data, the scikit-learn version struggled to correctly identify one of the outcome classes, revealing a limitation of the Naive Bayes approach in handling imbalanced or nuanced feature relationships. This exercise offered a valuable perspective on different modelling paradigms and the importance of evaluating models from multiple angles. Nonetheless, this comparison illustrated how probabilistic models, while interpretable and computationally efficient, may not be the most suitable for the unpredictability and complexity inherent in FPL data.

Fantasy football, by nature, includes many unpredictable variables such as last-minute injuries, managerial decisions, or weather conditions that cannot be fully captured by historical data alone. These limitations make FPL an inherently dynamic and challenging problem for machine learning models. It is unrealistic to expect 100% accuracy, especially given the non-deterministic nature of real-world sports data. Unlike deterministic systems where the same input always yields the same output, FPL includes a high degree of randomness that models cannot fully account for.

That said, this unpredictability is precisely what makes FPL appealing, not just for fans, but for machine learning engineers as well. It introduces an element of strategy and uncertainty that keeps the game engaging. In such cases, having a human-in-the-loop becomes invaluable. A model may suggest a high-performing player as a captain based on historical trends, but it is up to the user to assess whether that recommendation makes sense in light of recent news, line-up changes, or tactical shifts.

As Fantasy Premier League continues to grow in popularity, with more users investing time and even money into it, the value of accurate prediction models will only increase. There is real potential here, not only for improving user experience but also for creating commercial

applications. A model that can consistently outperform average predictions could serve as the backbone for fantasy analytics platforms, tipster services, or even monetized tools for fantasy investors.

In summary, while our model shows promising results, it also highlights the need for cautious interpretation, continuous improvement, and human judgment. The fusion of data-driven insights and human intuition is key to making the most of what FPL and machine learning has to offer.