Group 24 Progress Report: My Group's Project Name

Omar Abdelhamid, Khalid Farag, Omar El-Aref
{abdelo8,faragk1,elarefo}@mcmaster.ca

1 Introduction

Predicting the outcomes of football competitions and generating the full season league standings represents a challenging task that intersects statistical analysis and machine learning. The English Premier League (EPL), with its twenty teams and intense, unpredictable competition, providing a compelling context for this kind of task. In this project, we build a machine learning model that takes historical match data from the EPL including betting odds as the features and predicts the full season league standings. In addition to the predicted standings, we also predict the match outcomes for each match in the 2024-25 season, so the wins, draws, and losses for each team. This project is motivated by the growing interest in using data-driven approaches to understand and predict football standings and match outcomes. This project is valuable to many different stakeholders, including fans seeking deeper insights into their favorite teams/leagues, bettors looking to make informed decisions, media organizations seeking to provide insightful analysis, and clubs looking to gain a competitive edge in the league. In this report, we will describe the related work we encountered for our approach, the dataset we used, the features we used for our model, our implementation, our results and evaluation, and our feedback and plans for the remainder of the project.

2 Related Work

There have been a number of prior studies that have attempted to predict the football match outcomes and the full season league standings. One of the most notable studies is the work of Arman Tadjrishi, who proposed multiple models to predict the football match outcomes and the full season league standings. The models that he proposed are decision trees, random forests, and neural networks, where his best model was able to predict the full season league standings with an accuracy of 55% using a feedforward neural network. (Tadjrishi, 2025)

Another significant contribution to this problem were the work of Nyquist and Pettersson, who investigated the use of deep learning concepts for football match prediction. Their study employed recurrent neural networks (RNNs) to capture sequential dependencies in match outcomes, and they concluded that deep learning models can outperform traditional machine learning methods when sufficient historical data is available (Nyquist and Pettersson, 2017). Similarly, Constantinou *et al.* developed *pi-football*, a Bayesian network model that combines both objective and subjective factors, such as team strength, fatigue, and form, to improve prediction accuracy. (Constantinou et al., 2012).

Other studies have focused on improving model interpretability and feature representation. Hubáček *et al.* applied gradient boosted trees to relational match data, achieving high predictive performance and demonstrating that feature-rich, tree-based methods can rival deep learning in sports analytics (Hubáček et al., 2019). Furthermore, the foundational work of Dixon and Coles introduced a Poisson regression framework for football score prediction, which remains influential as a baseline for modern probabilistic forecasting models (Dixon and Coles, 1997).

3 Dataset

Our dataset consists of historical English Premier League match data covering seasons 2010–11 through 2024–25, obtained from the public repository (Football-Data.co.uk, 2025). Each season's CSV file contains match-level statistics such as team names, dates, full-time and half-time results, goal counts, shot counts, disciplinary records, and betting odds. The same loading functions can be

reused for La Liga, Serie A, and Bundesliga, which were used only for cross-testing and pipeline validation.

The files are imported and standardized in `importing_files.py`, which keeps a consistent subset of numeric features across all seasons:

Category	Example Columns
Match Info	Date, HomeTeam, AwayTeam, Season, Div
Full-time Outcome	FTHG, FTAG, FTR
Half-time Stats	HTHG, HTAG, HTR
Offensive Metrics	HS, AS, HST, AST, HC, AC
Discipline	HF, AF, HY, AY, HR, AR
Betting Odds	B365H, B365D, B365A

Table 1: Example columns included in the Premier League dataset.

To make the data team-centric, the script `data_imports.py` converts every match into home and away records using helper functions (`create_home_row`, `create_away_row`) and aggregates them with `build_team_year_stats`. For each team in each season, we compute:

- Wins, Draws, and Losses

- Goals For / Against / Difference

- Average Shots, Shots on Target, and Corners

- Average Fouls, Yellow and Red Cards

- Derived features such as *Big Chances Created* and total *Points*

These per-team summaries are concatenated by `build_season_team` into a unified training table of shape (n teams × n seasons). The target variable is total Points earned in a season, and all remaining numerical columns serve as features.

Preprocessing includes:

- Removing incomplete or inconsistent rows

- Converting categorical results (FTR) from {H, D, A} to numeric labels

- Normalizing features with `StandardScaler`

- Chronological split: all seasons ≤ 2023–24 for training, 2024–25 for validation

This design allows the model to learn season-to-season performance patterns and generate realistic predictions for the current Premier League season while minimizing temporal leakage.

4 Features

Our model uses a combination of match-level and season-level statistical features derived from the Premier League dataset. Each match record includes pre-match betting odds (B365H, B365D, B365A) and performance indicators such as goals, shots, corners, fouls, and disciplinary actions (yellow and red cards). These attributes provide both contextual and statistical information about each game.

Initially, we experimented with a feedforward neural network (`FootballModel`) that operated on season-aggregated data. In this setup, each team-season pair was represented as a fixed-length numeric vector containing features like average goals, shots, cards, corners, and derived statistics such as goal difference and total points. All numeric inputs were standardized using a `StandardScaler` to normalize their ranges before training.

To improve realism and capture team momentum, we later transitioned to a recurrent neural network (`SeasonRNN`). Instead of treating each season as an independent instance, the RNN processes matches sequentially in temporal order. Each input vector to the model consists of:

- Encoded quantitative match features (B365H, B365D, B365A)

- A binary role signal indicating whether the team is playing at home (+1) or away (−1)

- The current hidden states of both the home and away teams

No explicit feature selection was performed; rather, all relevant numeric features were retained. Feature engineering was minimal, limited to the computation of derived metrics such as goal difference and "big chances created." The neural network itself is responsible for learning internal feature importance and higher-level representations through backpropagation. The hidden states of the RNN function as learned embeddings that evolve over time, capturing form, fatigue, and consistency across the season.

Overall, all features and learned representations are integrated into a single neural network framework, allowing the model to jointly learn statistical patterns and temporal dynamics for improved match and season prediction performance.

5 Implementation

Our current model is a recurrent neural network (RNN) designed to predict football match outcomes (Home, Draw, or Away) using pre-match Bet365 odds as its core input features. Each team in the dataset maintains an evolving hidden state that represents its form, strength, and momentum as the season progresses. These hidden states are updated after every match, capturing long-term dependencies such as winning streaks or poor form. The architecture includes two role-specific GRUCells, one for home teams and one for away teams, along with a lightweight feature encoder and a prediction head. The encoder processes match-level input features, while the prediction head combines both teams' hidden states with the encoded features to output probabilities over the three possible results. This design allows the model to simulate how a team's past performance influences future games and provides a realistic temporal modeling approach to football outcomes.

The model is trained chronologically so that it only learns from information available before each match, preventing data leakage. We use Cross-Entropy Loss to measure prediction accuracy and optimize it with the Adam optimizer at a learning rate of 1e-3. Team states are preserved between matches and even across seasons, ensuring that training carries historical context into future fixtures. During validation, these same hidden states are continued rather than reset, allowing the model to make realistic predictions for the 2024–25 season based on prior knowledge from earlier years. This mirrors how real-world forecasts work, leveraging form, head-to-head momentum, and team trajectories. Each prediction step updates team states again, creating a rolling, causally consistent evaluation loop.

Our model currently reaches around 54% accuracy, showing that it has learned meaningful relationships between odds, team performance, and match outcomes. However, the model currently struggles with class imbalance, rarely predicting draws, an expected challenge in football modeling since draws are less frequent and less predictable. This imbalance triggers sklearn's UndefinedMetricWarning, which we silence for clarity. Despite this limitation, the model's performance and behavior confirm that it is learning temporal structure rather than memorizing outcomes, as it maintains competitive accuracy even when tested on an unseen season.

Moving forward, several improvements are planned. We will incorporate class weighting or focal loss to better capture draw outcomes and address imbalance. We also plan to extend the feature set beyond betting odds to include pre-match statistics such as recent form, home advantage metrics, and player availability. Additionally, truncated backpropagation through time (BPTT) can be introduced to strengthen the RNN's ability to retain useful information across longer sequences. With these refinements, we expect the model to produce more balanced, realistic predictions and to generalize more effectively across future seasons.

6 Results and Evaluation

6.1 Evaluation Strategy

For our evaluation strategy, we use train/validation split strategy to evalulate our model. Sepecifcally, we use all the EPL seasons from 2010-11 through 2023-24 as the training set, and reserve the 2024-25 season as the validation set. This approach simulates a realistic prediction scenario where we use historical data to predict future outcomes, avoiding data leakage that could occur with random splits. We do not use cross-validation, as the temporal nature of the data and the need to maintain season-level integrity make it more appropriate to use a single held-out season for validation.

6.2 Evaluation Metrics

We evaluate our model using several classification metrics appropriate for multi-class prediction:

- **Accuracy**: The overall proportion of correctly predicted match outcomes

- **Precision**: The proportion of predicted positive cases that are actually positive, computed per class

- **Recall**: The proportion of actual positive cases that are correctly identified, computed per class

- **F1-score**: The harmonic mean of precision and recall, providing a balanced metric

These metrics are computed for each of the three outcome classes (Home win, Draw, Away win), as well as macro-averaged and weighted-averaged across all classes.

6.3 Results

Table 2 presents the classification performance metrics for our model on the 2024-25 validation season. The model achieves an overall accuracy of 55.3% on 380 matches. The precision, recall, and F1-scores reveal a significant class imbalance issue: the model performs well on Home wins (F1-score of 0.660) and moderately on Away wins (F1-score of 0.581), but completely fails to predict Draw outcomes (F1-score of 0.000). This suggests that the model is biased toward predicting wins rather than draws, which is a common challenge in football prediction tasks.

Class	Precision	Recall	F1-score	Support
Home	0.531	0.871	0.660	155
Draw	0.000	0.000	0.000	93
Away	0.595	0.568	0.581	132
Accuracy		0.553 (380)		
Macro Avg	0.376	0.480	0.414	380
Weighted Avg	0.424	0.553	0.471	380

Table 2: Classification metrics for match outcome prediction on the 2024-25 validation season.

Figure 1 shows the confusion matrix for our predictions, which visualizes the model's performance across all three outcome classes. The matrix confirms the model's strong bias toward predicting Home wins, with 135 out of 155 actual Home wins correctly predicted, but only 0 out of 93 Draw outcomes correctly identified.
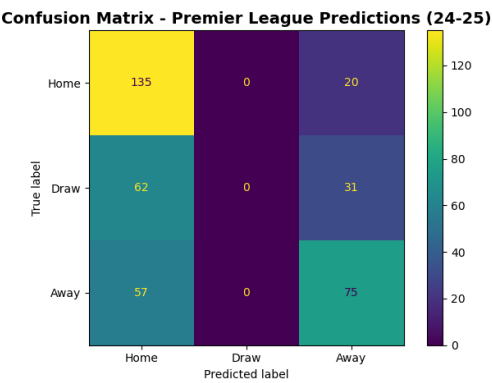

Figure 1: Confusion matrix for match outcome predictions on the 2024-25 Premier League season. Rows represent true labels, columns represent predicted labels.

Table 3 presents the predicted league standings for the 2024-25 season, aggregated from individual match predictions. Notably, the predicted standings show no draws for any team, which is a direct consequence of the model's inability to predict draw outcomes. The top three teams (Arsenal, Liverpool, and Manchester City) are all predicted to finish with identical records of 35 wins and 3 losses, which reflects the model's tendency to favor strong teams in its predictions.

Team	W	D	L
Arsenal	35	0	3
Liverpool	35	0	3
Man City	35	0	3
Chelsea	31	0	7
Newcastle	28	0	10
Aston Villa	23	0	15
Brighton	22	0	16
Bournemouth	20	0	18
Man United	20	0	18
Fulham	19	0	19
Nott'm Forest	19	0	19
Brentford	18	0	20
Tottenham	18	0	20
Crystal Palace	16	0	22
West Ham	11	0	27
Wolves	9	0	29
Everton	8	0	30
Leicester	5	0	33
Ipswich	4	0	34
Southampton	4	0	34

Table 3: Predicted 2024-25 Premier League standings aggregated from match-level predictions. W = Wins, D = Draws, L = Losses.

6.4 Baseline Comparison

As a baseline, we can compare our model's accuracy of 55.3% to the work of Tadjrishi (Tadjrishi, 2025), who achieved 55% accuracy using a feedforward neural network. Our model performs similarly, though the class imbalance issue with draws represents a significant limitation that requires further investigation and model refinement.

7 Feedback and Plans

During our progress review, our TA provided feedback emphasizing the importance of performing feature selection and considering the availability of data at prediction time. Specifically, the feedback noted that not all columns from the CSV files should be used for training, since many of them (such as goals, shots, and cards) are only known after a match has concluded. Using these variables would cause data leakage when attempting to predict future outcomes.

In response, we analyzed which features can be realistically known before a match. We concluded that the only truly predictive and pre-match features

are the betting odds (B365H, B365D, and B365A), as they reflect the expectations of bookmakers and the public before kickoff. Our next step is to refine the dataset so that only forward-available information is used to train the model. This ensures that our predictions remain realistic and generalizable.

Moving forward, our plan is to extend our current recurrent neural network (RNN) to explicitly model the temporal progression of each team throughout the season. In this design, each layer of the network corresponds to the sequential flow of matches: the first layer processes the betting odds of the opening game, and each subsequent layer incorporates both the betting odds for the next game and the previous game's result to predict the next outcome. This approach allows the network to learn how form, confidence, and performance evolve across a season, providing a more dynamic and realistic representation of team behavior.

By combining the TA's feedback on feature realism with this temporal RNN framework, our model will shift from static predictions toward sequence-aware forecasting. This advancement will improve predictive power while maintaining interpretability and real-world applicability.

Team Contributions

The project was an equal collaboration between us, with each member contributing approximately one-third of the overall work. All three team members worked together closely through regular calls, discussing ideas, debugging issues, and refining the model implementation in real time. Omar El-Aref primarily focused on structuring the RNN workflow, ensuring chronological data handling, and integrating the training and validation loops. Omar Abdelhamid contributed to model tuning, performance evaluation, and implementing feature scaling and state management across matches and seasons. Khalid worked extensively on the data preprocessing and feature construction, helping ensure the match tables were clean, and consistent. Overall, the development process was highly collaborative.

References

Anthony C. Constantinou, Norman E. Fenton, and Martin Neil. 2012. pi-football: A bayesian network model for forecasting association football match outcomes. *Knowledge-Based Systems*, 36:322–339.

Mark J. Dixon and Stuart G. Coles. 1997. Modelling association football scores and inefficiencies in the football betting market. *Applied Statistics*, 46(2):265–280.

Football-Data.co.uk. 2025. Football-data.co.uk: Historical football results and betting odds data. Accessed: November 9, 2025.

Ondřej Hubáček, Gustav Šourek, and Filip Železný. 2019. Learning to predict soccer results from relational data with gradient boosted trees. *Machine Learning*, 108(1):29–47.

Rasmus Nyquist and Daniel Pettersson. 2017. Football match prediction using deep learning. Master's thesis, Chalmers University of Technology, Gothenburg, Sweden.

Arman Tadjrishi. 2025. Predicting premier league match results based on historic data. https://armantee.github.io/predicting/. Accessed: 2025-11-11.