Project Title: Premier League Match Predictor

Team members: Karl Joonas Jõepere, Rene Dapon, Karl Markus Kiudma

GitHub: https://github.com/renedapon/eplpredictor

Business Understanding

Business Goals

The English Premier League is one of the most competitive football leagues in the world, with an enormous public audience and a strong interest in match predictions. The uncertainty of football outcomes drives both entertainment and financial activity, especially in betting markets where decisions are often based on emotion, team loyalty, or biased pundit opinions. However, modern football statistics offer another perspective: performance indicators such as expected goals, shot volume, possession rate, and defensive actions can be quantified and potentially used to anticipate match results. Our project grows out of curiosity to see whether statistical learning can outperform human intuition and become a reliable predictor of match outcomes. Rather than designing a commercial product, we aim to build a personal tool that supports informed decision-making, tests the practical predictive power of football analytics, and helps us gain experience in data science.

The business goal of our project is therefore twofold. First, we want to create a functioning prediction model that can forecast the outcome of upcoming EPL matches using publicly available historical data. Second, the project should contribute directly to our development as data analysts by giving us hands-on experience in data preparation, feature selection, model building, and model evaluation. Since the project targets only personal use, the "client" we design for is ourselves. For us, success means generating predictions that are sufficiently accurate to be interesting to analyze and potentially useful for small-scale betting. We do not expect perfect forecasts, but we expect our model to perform noticeably better than guessing at random or always choosing the stronger team historically.

Situation Assessment

To achieve this goal, we work with team-level statistics from previous and the ongoing Premier League seasons, collected from FBref.com. This data includes offensive, defensive, and possession metrics but excludes individual player performance. Our project is based on the assumption that aggregated team behaviour reflects future performance well enough to be used for forecasting. We also assume that publicly available statistics are sufficiently reliable and that historical performance remains meaningful when predicting upcoming matches. Certain aspects of football such as player injuries, refereeing decisions, short-term form shifts, or random events cannot be measured or predicted using our data. This means that no model can fully capture football uncertainty, and we recognise this as a limitation.

The project relies on student developers, Python-based tools such as Pandas and Scikit-learn, and standard personal computing hardware. There are no financial costs. The largest resource expenditure is time invested in collecting, cleaning, engineering, and evaluating data. Another notable risk arises from overfitting, where a model appears strong on historical data but fails on unseen matches. To reduce this risk, we commit to evaluating our results using independent test sets and ensuring that predictions are generated strictly using pre-match information. Since we intend to predict future games, the system must function without manual player-level updates and without dependency on post-match statistics.

Although our project has no commercial objective, it still produces measurable benefits. The primary benefit is educational: building and refining a real prediction tool provides practical experience in machine learning and evidence-based decision-making. A secondary benefit is entertainment value from testing whether our own statistical model performs better than intuition or expert opinion. Because the project requires no monetary investment, the benefits clearly outweigh the minimal time-related costs.

Terminology

Match Outcome (W/D/L) – the categorical result of a football match, classified as win, draw, or loss.
Expected Goals (xG) – a statistical measure estimating the probability that a shot will result in a goal, based on shot quality and characteristics.
Team Statistics – aggregated performance metrics that describe a team’s offensive, defensive and possession behaviour across matches (e.g., shots, xG, passes, tackles).
Web Scraping – an automated method for extracting data from websites by programmatically loading HTML pages, parsing specific elements (such as tables, statistics, or links), and converting them into structured data for analysis or model training.
Random Forest – an ensemble machine learning algorithm composed of multiple decision trees, used to model complex relationships in structured data and improve classification accuracy through averaging.

Data Mining Goals

Our data-mining objective is to build a supervised classification model that predicts whether a team will win, draw, or lose an upcoming match. Random Forest serves as our initial algorithm because it performs strongly on structured numerical data without requiring feature scaling, tolerates noise, and captures nonlinear relationships between football metrics. For the project to succeed, the model must produce predictions before matches and consistently outperform simple baselines, such as random guessing or predicting the most common outcome. If these conditions are met, we consider both our business and our data-mining goals achieved.

Data understanding

Gathering Data

To achieve our goal, we must gather data that reflects how teams have performed in the past. Match results alone are insufficient because the scoreline does not capture the full complexity of team performance. A team may dominate possession, generate high-quality chances, or defend exceptionally well without these aspects being fully reflected in the final result. Therefore, our project requires a richer dataset that covers multiple dimensions of performance, including attacking, passing, and defensive indicators.

Outline Data Requirements

To meet our modelling goals, we require two main categories of data: match-level data and team-level performance data.

Match-level data provides the foundation for the prediction task and consists of one record per Premier League fixture. The essential fields include the teams involved, goals scored, expected goals, and the match date.

Team-level performance data captures how each team performs across various aspects of play. This includes metrics such as goals and expected goals, shot creation, possession, progressive passing. For completeness, we also require the corresponding opponent statistics against each team, which quantify what each team concedes to its opponents. These metrics help account for the relative strength of opposition and provide a more balanced representation of team performance.

Verify Data Availability

We examined several publicly available football statistics platforms to verify that the necessary data exists. FBref.com proved to be the most suitable and reliable source due to its consistent structure, rich statistical coverage, and clear separation of match-level and team-level tables. The data is provided in well-defined HTML tables that are easy to scrape and process.

The structure of FBref's Premier League data is stable across seasons, which simplifies automated scraping and ensures compatibility when merging datasets. Our web scraper successfully retrieved all required match-level and team-level variables, confirming that the platform provides complete and accessible coverage for our purposes.

Define Selection Criteria

For this project, we restrict the dataset to Premier League fixtures only, excluding domestic cup matches and European competitions. These competitions involve uneven opponent quality and different strategic contexts, which would introduce noise. We retain only variables directly relevant to on-pitch performance, such as goals, expected goals, shooting metrics, possession, passing quality, and defensive actions, while excluding non-predictive fields like referee names, stadium attendance, kit colours, or weather descriptions. From these performance indicators, we keep only the metrics that meaningfully contribute to modelling team strength and predicting match outcomes.

Describing Data

Our dataset consists of match-level and team-level Premier League data collected from FBref.com. The match-level data covers all fixtures from the 2020/21 season onward and includes teams, goals, expected goals, and match dates, giving roughly 1920 matches across 6 seasons. The team-level data provides season-level statistics for all 20 clubs, offering a detailed view of their attacking and defensive performance throughout each season. For each team, we have around 25 different attributes, covering results, expected goals, possession, progressive passing, shooting, chance creation, and defensive concession metrics.

Exploring Data

During data exploration, we examined variable ranges, distributions, and summary statistics. Goals and expected goals showed consistent ranges across seasons, with no extreme outliers. Team-level indicators such as shots, expected goals, possession and shot creating actions displayed realistic variation across clubs. Correlation analysis suggested that these stats are strongly linked to the outcome of a match. Team names, dates, and match IDs matched correctly across datasets, and no missing values were found in key fields. Overall, the dataset appears coherent and ready for modelling.

Verifying Data Quality

Data quality checks confirmed that the scraped data is complete and reliable for modelling. All match-level fields from the 2020/21 season onward were available, correctly formatted, and contained no missing or duplicated fixtures. The team-level table was consistently structured, with all attributes present for the current season. However, not all teams appear in every season because some clubs are relegated and others promoted, meaning historical team-level data varies in length across clubs. Minor issues, such as occasional team name formatting differences, were easily corrected. No invalid numerical values or broken rows were found. Overall, the data meets all requirements and is suitable for use in model development.

Project plan

Our project aims to develop a machine-learning system capable of predicting Premier League match outcomes using historical fixtures and team performance statistics. Below is our structured project plan with tasks, workload distribution, methods, and important notes.

Task 1 - Data Scraping (18h total / 6h per member)

- Scrape fixtures, standings and squad statistics from FBref.com
- Resolve scrape-blocking issues (403 errors)
- Validate downloaded CSVs and maintain iterations

Task 2 - Data Cleaning & Integration (15h total / 5h per member)

- Clean and standardise team names
- Fix inconsistent columns and missing values
- Merge multi-season datasets into a unified structure

Task 3 - Feature Engineering (15h total / 5h per member)

- Build statistical difference features (xG diff, GD diff, Pts/MP diff)
- Add form metrics and per-match indicators
- Evaluate feature relevance via correlations

Task 4 - Model Development & Evaluation (18h total / 6h per member)

- Train logistic regression and tune hyperparameters
- Test alternative models (RandomForest, XGBoost)
- Analyse calibration curves and confusion matrices

Task 5 - Prediction Interface (6 h total / 2 h per member)

- Implement function to predict match probabilities
- Validate results with known fixtures

Task 6 - Report, Poster & Final Presentation (18 h total / 6 h per member)

- Write project report
- Prepare visual results, graphs and predictions
- Design and create the final poster and slides

Tools and methods:

We plan to use Python, primarily pandas and NumPy for data cleaning and integration, and scikit-learn (Logistic Regression) for modelling. Data scraping is done using cloudscraper and pandas HTML parsing. Development is carried out in Jupyter Notebook with GitHub for version control. Visualisations and debugging rely on matplotlib.