Predicting Premier League Match Outcomes Using NLP

Authors: Theos Kounias & Marios Christodoulou
University of Cyprus
Date: 02/12/2025

Table of Contents

Abstract

This study investigates whether text based signals extracted from BBC football journalism can help predict Premier League match outcomes. Using a combination of web-scraped match results, fixture lists, and half a season of BBC football articles, we construct models that predict whether a team will win, draw, or lose an upcoming match. Natural language processing including tokenization, lemmatization, sentiment analysis, emotion scoring, TF-IDF extraction and more is combined with historical performance variables such as rolling xG and cumulative points in order to demonstrate both the feasibility and challenges of incorporating real journalistic text into predictive sports modelling, identifying which textual features contribute meaningfully to forecasting match outcomes.

1. Introduction

Predicting the outcome of football matches has long been of interest to statisticians, betting markets, analysts, and fans. Traditional predictive models rely heavily on structured match statistics: goals scored, expected goals (xG), team form, and player availability.

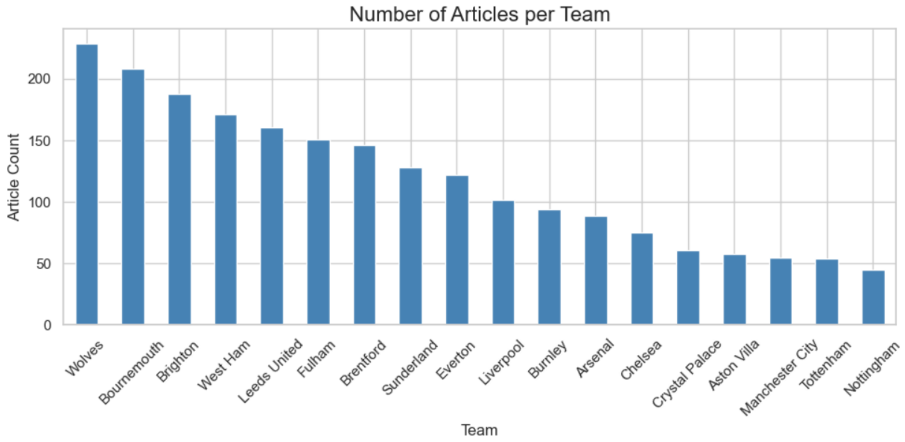
This project investigates whether **news articles published by BBC Sport** contain predictive signals that can enhance performance based forecasting models. We combine scraped BBC football articles with scraped Premier League results and fixtures, preprocess the text, derive multiple linguistic and sentiment features, and integrate them with match statistics to build a predictive classification model for match outcomes.

2. Data Collection and Integration

2.1 Web Scraping of BBC Football Articles

BBC Sport was scraped to collect football related articles containing team references, headlines, publication timestamps, and full article text. The scraping produced a dataset of article texts accompanied by date of article, the team the article talks about, and the article header). Because BBC pages do not always provide consistent timestamps, missing publication dates were interpolated and subsequently forward and backward filled.

To ensure that each article was relevant to the intended team, team name aliases were constructed, and only articles containing **at least two mentions** of the corresponding team were kept.



2.2 Web Scraping of Premier League Fixtures and Results

Match results, fixtures, and expected goals data were obtained by web-scraping using selenium. Each match in the dataset was assigned a unique **MatchID**, and each fixture was duplicated into home-team and away-team rows to facilitate team specific statistics. Some more data cleaning included: standardizing team names, parsing scorelines including atypical Unicode separators, and generating team level match histories sorted chronologically.

2.3 Merging Textual and Match Data

For every match, articles associated with the participating team were searched in a **30-day window prior to the match date**. All articles within this window were concatenated to form a single pre match text document per team per fixture. Rows with zero available articles in the time window were removed to preserve predictive relevance.

3. Text Preprocessing Pipeline

A major contribution of this project is the design of a full linguistic preprocessing chain. Each BBC article text is subjected to a multi stage pipeline, with vocabulary size tracking at each stage.

3.1 Uncontraction of English Text

To normalize informal constructions commonly used in sports writing, contractions such as “don’t”, “he’ll”, “I’m”, “won’t” and others were expanded using a custom regex based uncontraction function.

3.2 Tokenization and Lowercasing

The NLTK word_tokenize method was applied to all texts, producing raw token lists. Tokens were then lowercased to reduce the vocabulary size.

3.3 Noise Removal

Several categories of noise were removed:

- pure punctuation,
- standalone numbers,
- non-alphabetic tokens unless meaningful (e.g., scorelines like “2–1”).

Moreover we used some custom filtering functions in order to prevent accidental deletion of football specific numeric expressions.

3.4 Stopword Removal

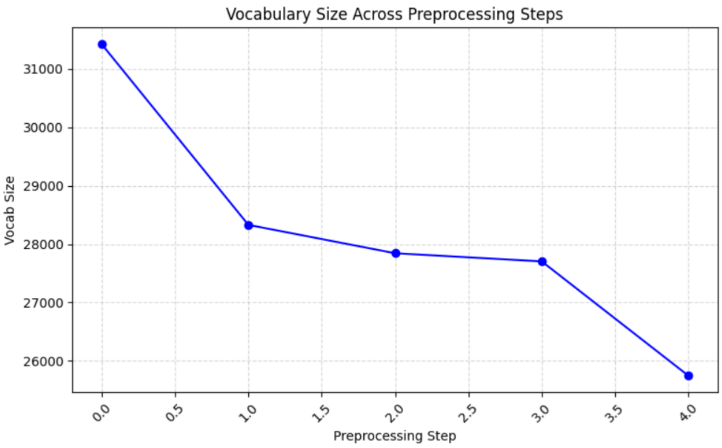
English stopwords from NLTK were removed, reducing grammatical filler while retaining content words relevant to sentiment and event descriptions.

3.5 Lemmatization

Tokens were lemmatized using WordNet’s WordNetLemmatizer, enabling reduction of morphological variants (e.g., “injured”, “injuries” → “injury”), which was vital for our emotion and event keyword matching. Lemmatizarion was decided to be used instead of stemming because we did not want to just chop off endings to preserve a correct language context.

3.6 Vocabulary Tracking

A log table was constructed across all preprocessing steps, tracking total tokens and vocabulary size at each transformation stage. As we can see from the table below, the first and last step were the most effective in reducing the vocabulary size (Step 1: lowercasing and step 2: Lemmatization)



4. Feature Engineering

4.1 Sentiment Features (VADER & TextBlob)

In the generation of sentiment features two complementary sentiment models were applied:

- **VADER**: Which outputs negative/neutral/possitive and a compound sentiment.
- **TextBlob**: To provide polarity and subjectivity.

These captured criticism, optimism, tone, and confidence embedded in BBC's football coverage.

4.2 Emotion Lexicon Features

Using the DepecheMood English lemma database, each token was mapped to multiple emotion categories (For example fear, excitement, positive, negative). Article level emotion vectors were generated by averaging the emotion scores of all matched tokens.

4.3 Event Keyword Features

To detect football-specific contextual signals, the following event categories were engineered these 5 variables by detecting keywords in the article texts:

- **Injury** (e.g., "hamstring", "ruled out")
- **Transfer** (e.g., "deal", "linked", "target")
- **Suspension**
- **Coach/Manager mentions**
- **Uncertainty** (e.g., "might", "risk", "unlikely")

Binary or count based features were produced per category.

4.4 TF-IDF Features

From the cleaned token lists, a TF-IDF model extracted the **top 100 informative terms** (including unigrams and bigrams). These sparse textual features were merged into the final dataset.

4.5 Article Complexity Indicators

To capture readability and stylistic differences between positive vs. negative coverage:

- Flesch reading ease
- Lexical density
- Words per sentence

These metrics may implicitly reflect analytical tone or narrative emphasis.

5. Match-Performance Features

5.1 Rolling xG Features

After sorting matches by team and date, the expected-goals metric was lagged using shift(1) to prevent leakage. Rolling means were computed:

- **xG_last3**: mean xG over previous 3 matches
- **xG_last5**: mean over previous 5 matches

5.2 Cumulative Points Before Match

For each team, cumulative league points were computed using:

- Win = 3 points, Draw = 1 point, Lose = 0 points
- We shifted the summation by 1 in order to remove the current match information and avoid data leakage

These represent form and season momentum.

5.3 Contextual Indicator: Opponent Mention

A binary feature was also created in order to record whether the opponent team was explicitly mentioned in the article text, potentially reflecting heightened media attention or rivalry dynamics.

6. Predictive Modelling

6.1 Problem Definition

The task is a **three-class classification problem**:

- **Win**, **Draw**, **Lose**

Labels were derived programmatically from scorelines using a home/away aware result extraction function.

6.2 Model Selection

We implemented several predictive models to determine which algorithm could best learn from the dataset and make accurate predictions for match outcomes. The models tested include:

- Random Forest
- XGBoost
- Support Vector Machine
- One-vs-Rest Logistic Regression
- Deep Learning Neural Network

6.3 Model Evaluation

The performance metrics for each model on the test set are summarized below:

Model	Accuracy	Notes
Random Forest	0.42	Moderate performance on Wins and Loses, poor on Draws
XGBoost	0.44	Best at detecting Loses, poor for Draws, low for Wins
SVM	0.47	Balanced performance, better detection of Draws and Wins than RF/XGBoost, moderate on Loses.
OvR Logistic Regression	0.50	Balanced performance across classes, modest improvement on minority class recall
Deep Learning Neural Network	0.56	Highest accuracy, strong detection of Wins and Loses, struggles with Draws

7. Discussion

The integration of news articles into predictive football modelling is methodologically complex but promising. The preprocessing pipeline successfully structured highly unstructured journalistic text. Sentiment, emotion, and event keyword features meaningfully augmented traditional football statistics, although the relatively limited number of matches with sufficient text available constrained performance.

Model results suggest:

- textual sentiment correlates more strongly with **wins** than draws or losses,
- “injury” and “uncertainty” keyword categories show predictive relevance,
- rolling xG consistently remains among the most important features.

The poor performance on draws reflects both the rarity and unpredictability of draws in football modelling.

8. Conclusion

This project demonstrates a complete methodological pipeline for merging sports journalism with structured match statistics to predict football outcomes. While accuracy remains modest, the system establishes an extensible foundation for more sophisticated multimodal forecasting models.