Football Match Outcome Prediction Using Machine Learning

By: Eitan Bluer

Course: DS18

Date: 30.3.2025

1. Introduction

Football is one of the most unpredictable sports, yet it also generates vast amounts of structured data that can reveal hidden patterns. In recent years, advanced metrics such as xGoals (expected goals), PPDA (pressing intensity), and deep completions have become essential tools for professional analysis.

This project explores the use of machine learning (ML) to **predict football match outcomes** (home win, draw, away win) by integrating these modern statistical features. Using real historical match data, we aim to transform raw team and player performance indicators into a predictive system that can be applied before matches to forecast results.

The outcome is focused on **game-level predictions**, using both home and away team features, and targets not just prediction accuracy, but also **insight into what drives success on the field**.

What is the probability that a home team will win, lose, or draw a football match based on match statistics and pre-match statistics (Historical)?

This project aims to:

- Identify the most important performance metrics that correlate with match results.
- Build and evaluate machine learning models that classify matches into one of three outcomes.
- Create a deployable system to make predictions in real-time, potentially supporting coaches, analysts, sports media, and betting platforms.

The key focus is not only achieving high predictive power, but also **understanding the "why" behind each result** — revealing the tactical patterns hidden within data.

See in Append 1 the Files structure of the model.

---

3. Project Design & Methodology

3.1 Data Journey

The data used in this project comes from Kaggel Football Database[1] 7 CSV files (appearance, games, leagues, players, shots, teams, teamstats) that can be divided into three categories::

- Players performance
- Teams statistics
- Games statistics

These tables were merged and cleaned to construct a unified match-level dataset. Each row represents a single match, with features engineered for both the home and away teams.

3.2 Data Preparation

Notebook: Football_data_preperation_13_3.ipynb

We merged and cleaned the appearances, shots, games, and team performance datasets to build a match-level dataset for modeling. Players were accurately linked to teams using position order, substitution timing, and goal data.

Key team-level features were engineered—such as xGoals, ppda, shots on target, assists, and key passes—with separate values for home and away teams. The final structured dataset contains one row per match, combining raw and engineered features for both teams.

The final file (team_results.csv) contain 12,860 games

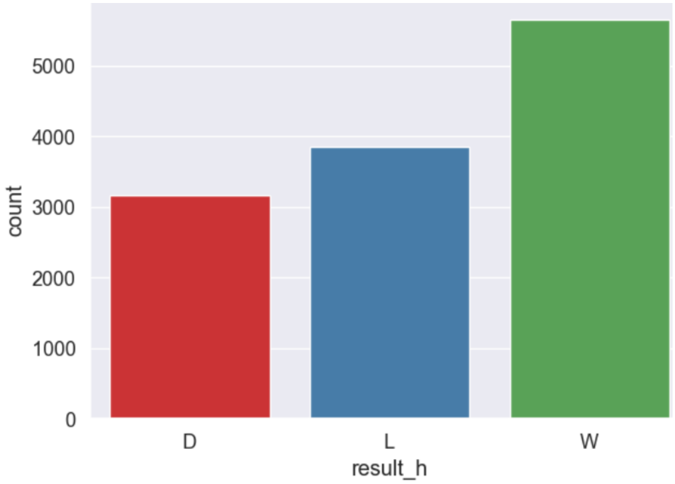
3.3 Exploratory Data Analysis (EDA)

Notebook: EDA_ML_Football.ipynb

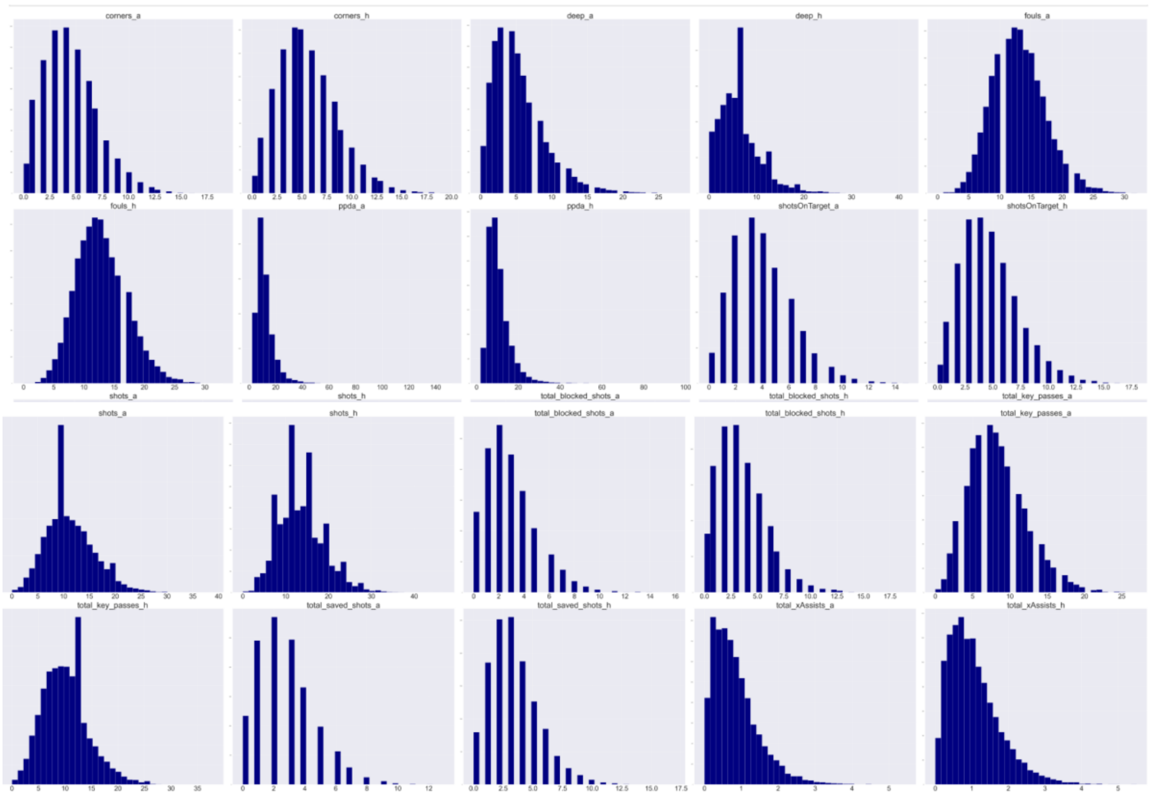
We explored the relationships between match outcomes and key performance metrics using visual tools and statistical tests to identify influential features.

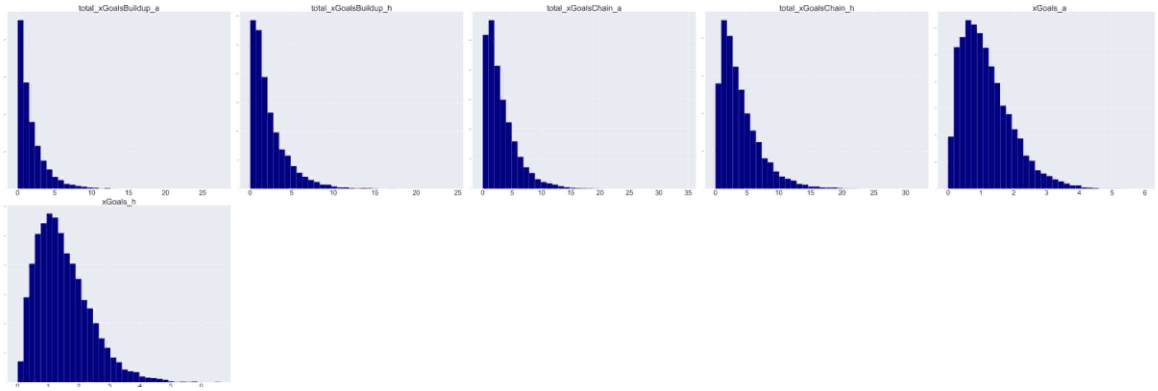
- Included markdown commentary on the class balance of match outcomes (home win, draw, away win).

[1] Football Database < https://www.kaggle.com/datasets/technika148/football-database?select=shots.csv https://www.kaggle.com/datasets/technika148/football-database?select=shots.csv>

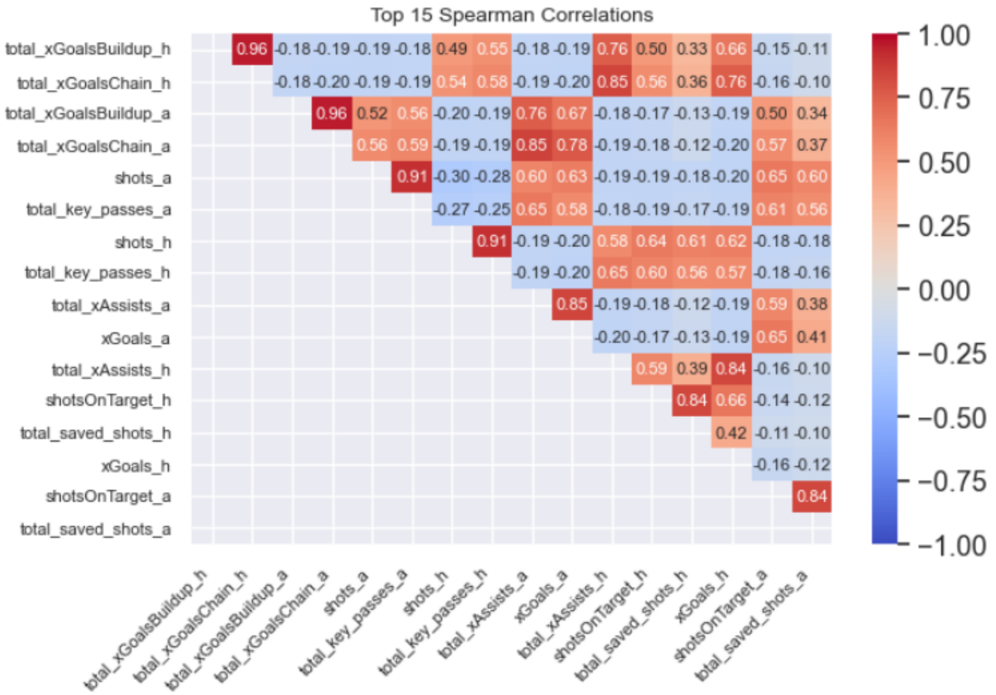


- Checked for normality using skewness and kurtosis tests, confirming that many features are skewed.





- Presented correlation heatmaps and pairplots, revealing strong positive correlations between **xGoals_h**, **teamgoals_h**, **total_assists_h**, and match outcomes.

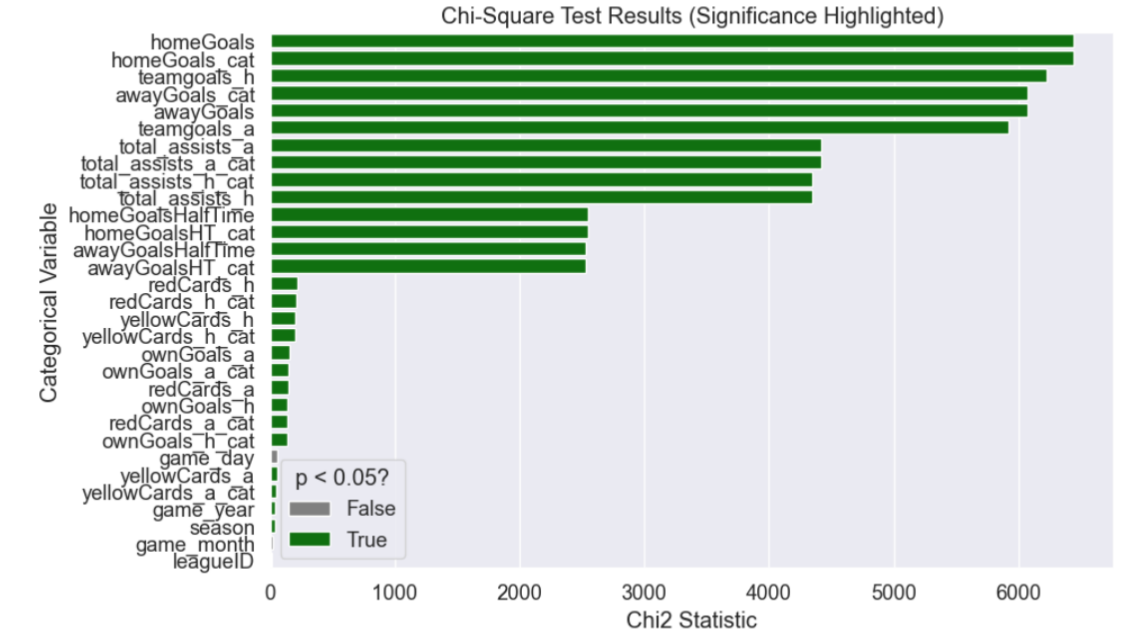


The resulting correlation heatmap revealed several pairs of variables with **high correlation coefficients (ρ > 0.8)**, including:

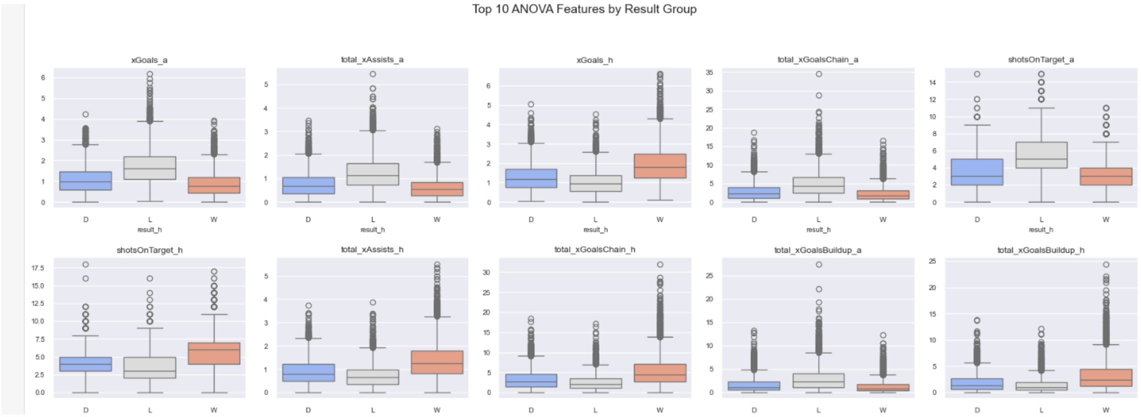
- xGoals_h and shots_on_target_h
- xAssists_total and key_passes_h

- xGoals_chain and xThreat_diff

- Chi-square test with the match results showed **red cards** (redCards_h, redCards_a) significantly impact match results ($p < 0.001$).



- Used visual exploration and ANOVA to highlight influential features like **xGoals**, **deep completions**, **PPDA**, and **shots on target**.
- Found that **xGoals_h** is significantly higher in wins compared to losses ($p < 0.001$); **shots_h** also differ by result ($p < 0.001$).
-



Key findings:

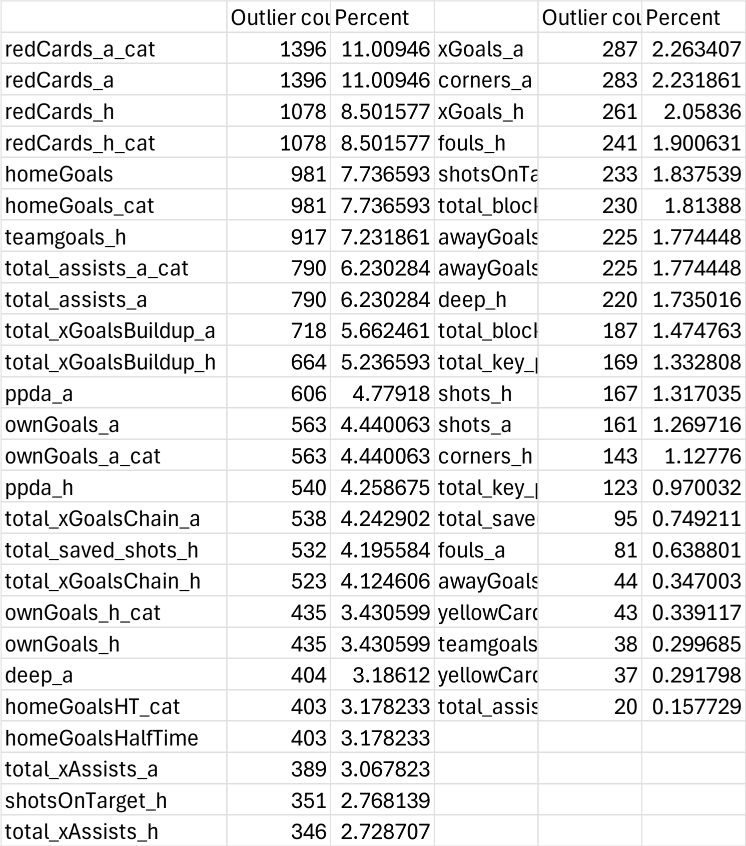
- **xGoals** and **shots on target** are significantly higher in wins.
- Performance Metrics Drive Results: **Goals, assists, xGoals** and **shots on target** are strong predictors of match outcomes.
- Disciplinary Factors Matter: **Red cards** and **pressing metrics (PPDA)** strongly influence outcomes.

3.4 Data Cleansing – Outliers & Missing Values

Notebook: EDA_Football_ML_outliers_and_missing_values.ipynb

We addressed data quality by identifying and handling outliers and missing values to ensure robust model training.

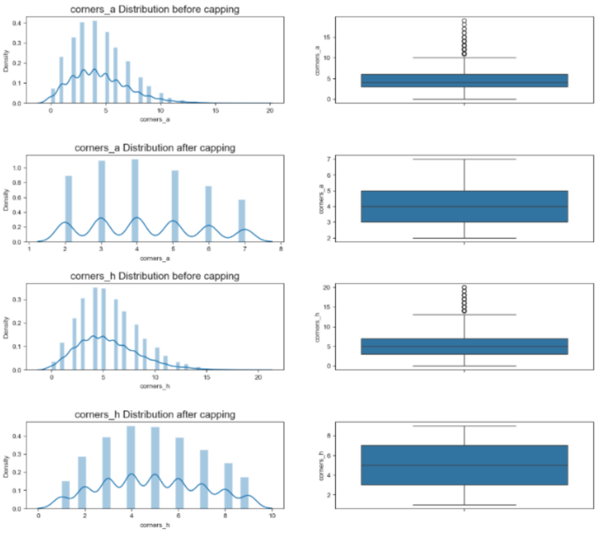
- Detected and removed outliers from 12 features (e.g., **xGoals**, **deep**, **fouls**, **corners**, **blocked/saved shots**) using the **IQR method**.

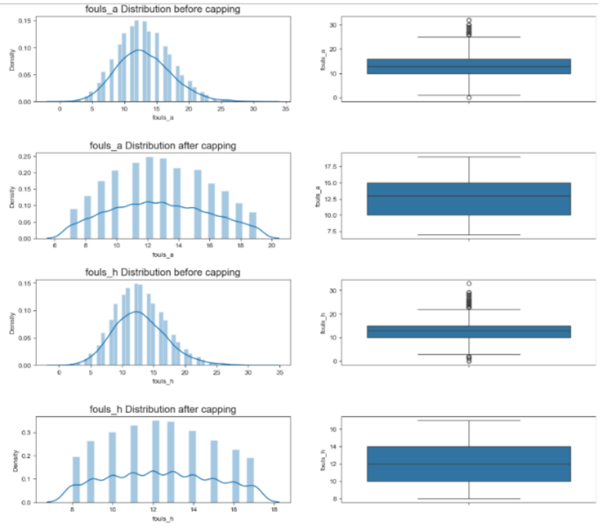


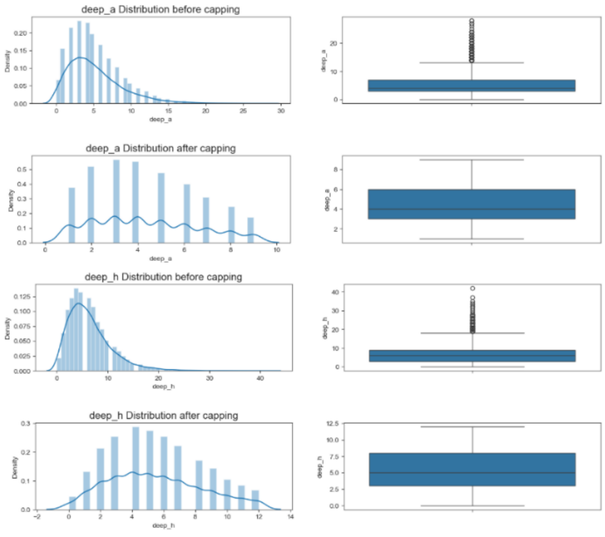
	Outlier cou	Percent		Outlier cou	Percent
redCards_a_cat	1396	11.00946	xGoals_a	287	2.263407
redCards_a	1396	11.00946	corners_a	283	2.231861
redCards_h	1078	8.501577	xGoals_h	261	2.05836
redCards_h_cat	1078	8.501577	fouls_h	241	1.900631
homeGoals	981	7.736593	shotsOnTa	233	1.837539
homeGoals_cat	981	7.736593	total_block	230	1.81388
teamgoals_h	917	7.231861	awayGoals	225	1.774448
total_assists_a_cat	790	6.230284	awayGoals	225	1.774448
total_assists_a	790	6.230284	deep_h	220	1.735016
total_xGoalsBuildup_a	718	5.662461	total_block	187	1.474763
total_xGoalsBuildup_h	664	5.236593	total_key_	169	1.332808
ppda_a	606	4.77918	shots_h	167	1.317035
ownGoals_a	563	4.440063	shots_a	161	1.269716
ownGoals_a_cat	563	4.440063	corners_h	143	1.12776
ppda_h	540	4.258675	total_key_	123	0.970032
total_xGoalsChain_a	538	4.242902	total_save	95	0.749211
total_saved_shots_h	532	4.195584	fouls_a	81	0.638801
total_xGoalsChain_h	523	4.124606	awayGoals	44	0.347003
ownGoals_h_cat	435	3.430599	yellowCar	43	0.339117
ownGoals_h	435	3.430599	teamgoals	38	0.299685
deep_a	404	3.18612	yellowCar	37	0.291798
homeGoalsHT_cat	403	3.178233	total_assis	20	0.157729
homeGoalsHalfTime	403	3.178233			
total_xAssists_a	389	3.067823			
shotsOnTarget_h	351	2.768139			
total_xAssists_h	346	2.728707			

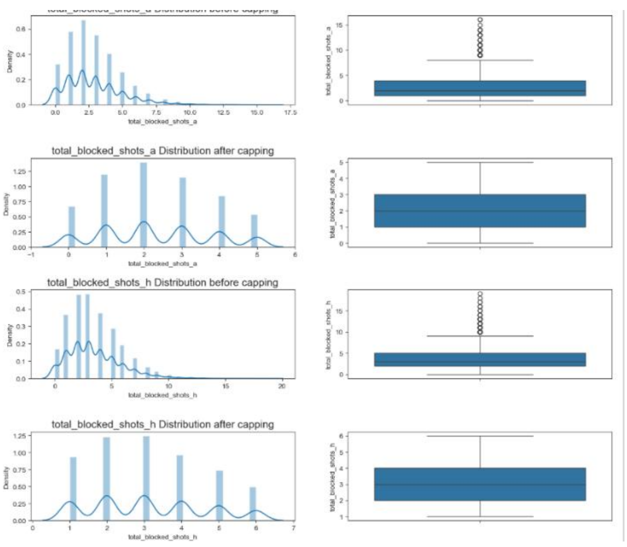
- Visualized before/after effects with **boxplots**, **histograms**, and **correlation analysis**, confirming improved distribution consistency.

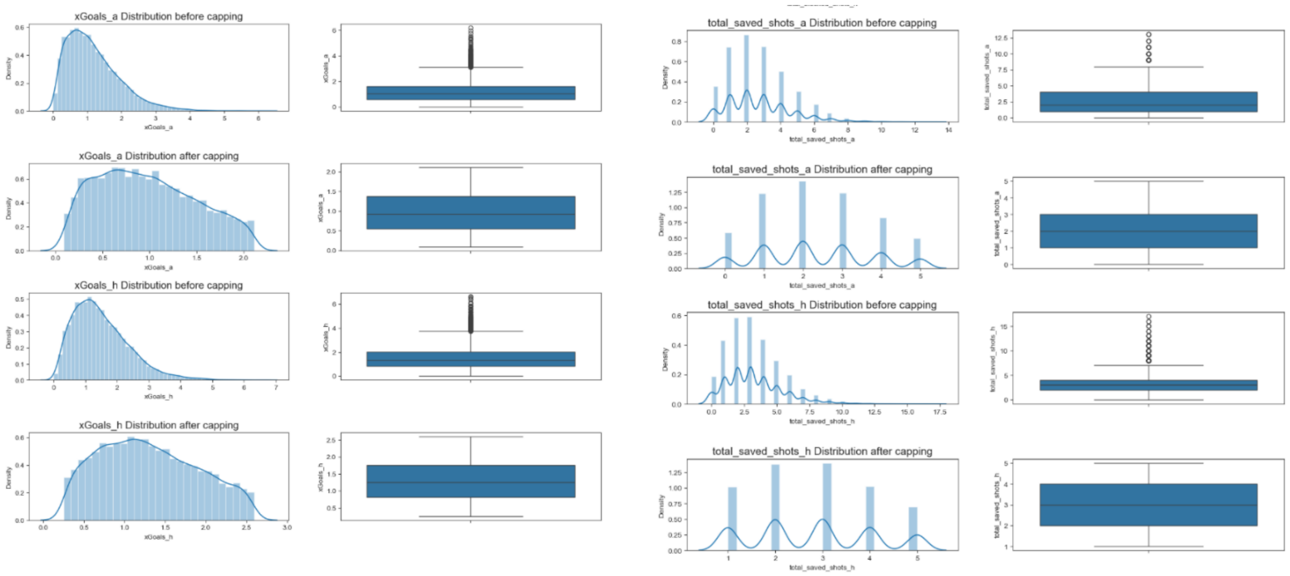
Present 12 feature that we drop their outliers



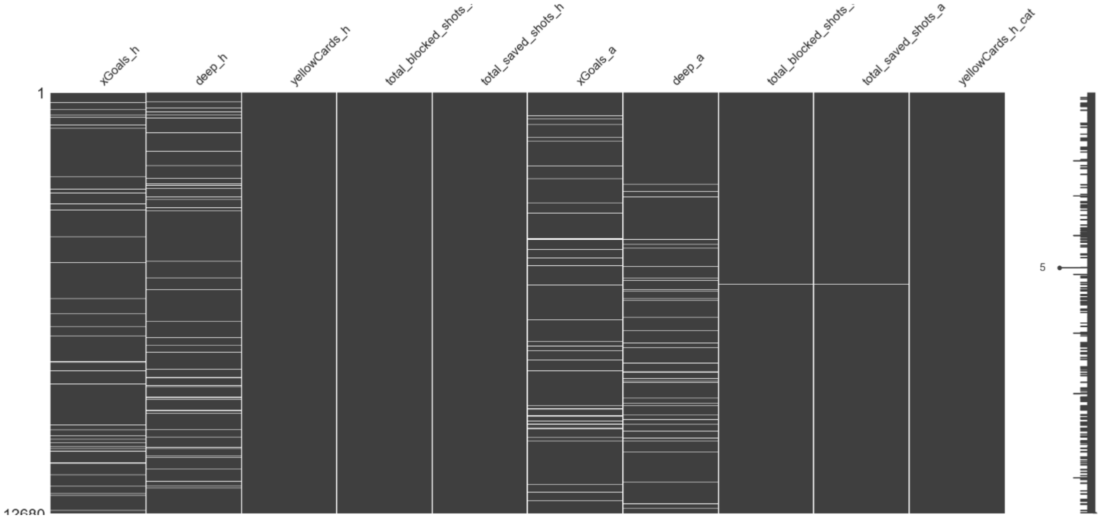








- Missing values were found across several **numeric** features and one **categorical** variable. Though they had minimal distributional impact, we imputed them using **KNN imputation** to preserve the dataset's structure and avoid losing rows.



3.5 Feature Engineering & Feature Selection

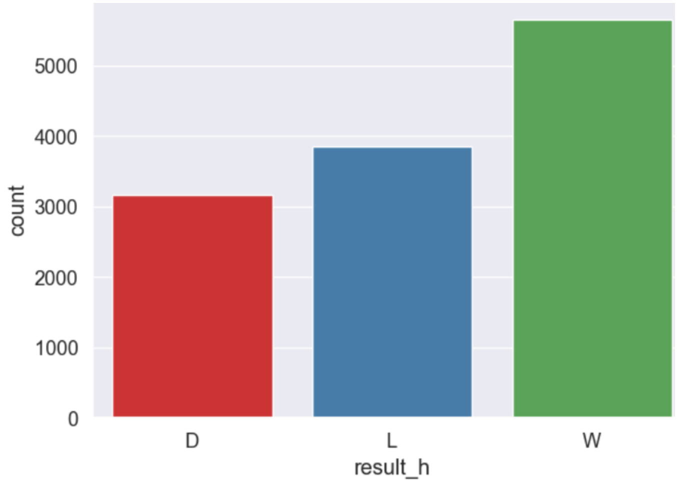
Notebook for **pre-game prediction**: ML_Football_Feature_Engeniring_and_feature_selection.ipynb

Notebook for **in-game prediction**: Prediction_during_the_ganeML_Football_Feature_Engeniring_and_feature_selection.ipynb

This step was divided into two modeling scenarios: **in-game prediction** (using match statistics if the current game without the goals) and **pre-game prediction** (using features derived from the last 5 matches).

- Applied **label encoding** and **one-hot encoding** for categorical variables such as team names and venues.
- Engineered key features including:
  - **goal_difference**
  - **xThreat_diff**
  - **xGoals / xThreat ratio**
  - **xGoals_chain**, **xAssists_total**
- Introduced **rolling averages and ratios** over the last 5 games (e.g., home_xGoals_h_rolling5, home_win_rate_5) to capture team form and momentum.
- Conducted **PCA** for exploratory purposes, but retained original features to maintain interpretability.
- No external data sources were used; all features were derived from internal match and player statistics.

3.6 Imbalanced Data



The target (result matches) seems balance for our needs

3.7 Model Selection & Fine-tuning

Notebook for **pre-game prediction**: Classification Models and Hyperparameter Finetuning.ipynb

Notebook for **in-game prediction**: Prediction_during_game_Classification Models and Hyperparameter Finetuning.ipynb

The Data split in to train (80%) set and test set (20%). We trained several model Models(Logistic Regression Decision Tree Random Forest AdaBoost Gradient Boosting XGBoost SVM Extra Trees) and test their estimation on the test set. In the part blow presents the result for **in-game prediction** and **pre-game prediction**.

A. **in-game prediction result**

This model takes the features that are part of the current game (e.g total assists category, shot on target, XGoals etx.) with out the goals (that have strong relation with the score) and try to predict the game result.

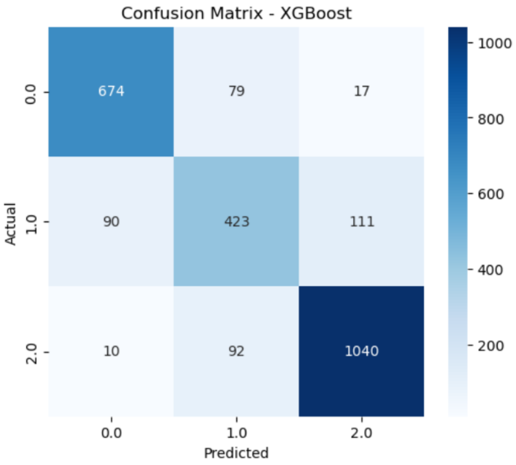
We tested multiple models:

Model	Accuracy	Precision	Recall	f1-score	Log-loss	AUC
Logistic Regression	0.73541	0.70354	0.69915	0.69956	0.599469	0.88263
Decision Tree	0.696372	0.669571	0.66796	0.66863	10.94385	0.75758
Random Forest	0.768533	0.741974	0.73473	0.7352	0.571391	0.90509
AdaBoost	0.73265	0.708296	0.7013	0.70392	1.028839	0.84506
Gradient Boosting	0.7847	0.763293	0.75982	0.76133	0.494114	0.9251
XGBoost	0.820978	0.801544	0.79426	0.79671	0.430875	0.9423
SVM	0.436909	0.145636	0.33333	0.20271	0.884034	0.75514
Extra Trees	0.776814	0.752512	0.74282	0.74378	0.584587	0.91064

The best model was **XGBoost**, with the following results:

- **Accuracy**: 82 %
- **AUC**: 0.942
- **F1-score**: 0.793
- **Log-loss**: 0.391

- **Confusion matrices** for each model to identify per-class prediction quality. The best model confusion matrix below



Hyperparameters were tuned via **RandomizedSearchCV**, and model performance was visualized through confusion matrices and feature importance charts.

- Best Parameters: {'subsample': 0.9, 'n_estimators': 300, 'max_depth': 7, 'learning_rate': 0.05, 'gamma': 1, 'colsample_bytree': 0.7}

Model	**Accuracy**	**Precision**	**Recall**	**f1-score**
XGBoost	0.821	0.802	0.794	0.797
XGBoost After Fine tuning	0.817	0.797	0.791	0.793

B. pre-game prediction result

This model takes the features that are Historical statistics (e.g average of 5 games for shots, shots on target, Key pass etc.) and try to predict the game result before the game started.

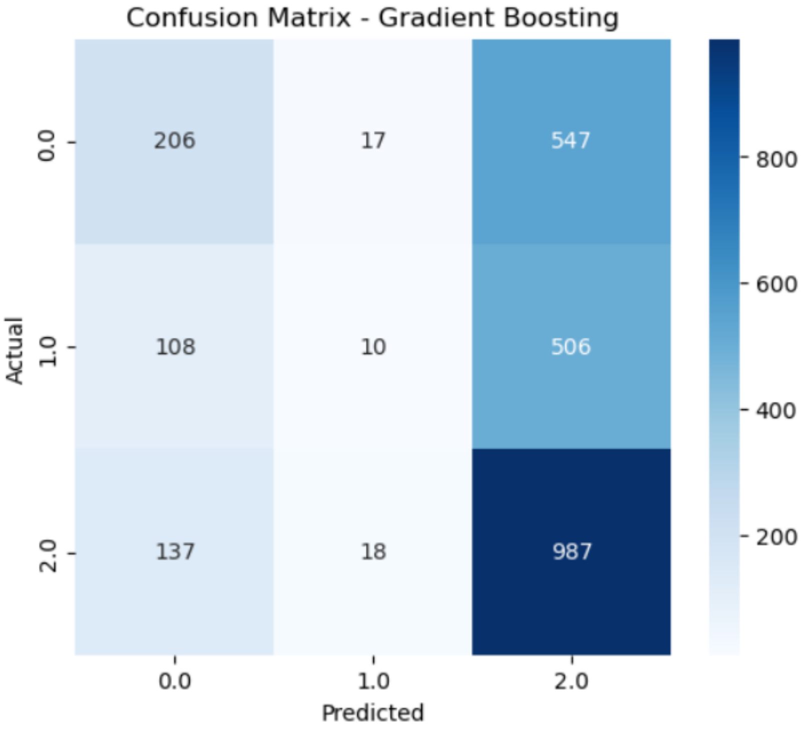
We tested multiple models:

Model	**Accuracy**	**Precision**	**Recall**	**f1-score**	**Log-loss**	**AUC**
Logistic Regression	0.460	0.408	0.353	0.268	1.061	0.530
Decision Tree	0.380	0.356	0.355	0.355	22.343	0.518
Random Forest	0.444	0.362	0.361	0.318	1.065	0.551
AdaBoost	0.465	0.302	0.367	0.300	1.076	0.549
Gradient Boosting	0.475	0.399	0.384	0.331	1.046	0.576
XGBoost	0.455	0.394	0.395	0.380	1.101	0.575
SVM	0.450	0.150	0.333	0.207	1.060	0.542
Extra Trees	0.439	0.363	0.357	0.316	1.084	0.541

The best model was **Gradient Boosting**, with the following results:

- **Accuracy**: 47 %
- **AUC**: 0.575
- **F1-score**: 0.331
- **Log-loss**: 1.046

- **Confusion matrices** for each model to identify per-class prediction quality. Matrix results below (**Gradient Boosting model)** indicate that there is likely a bias in variables entered into the model towards win and loss data than draw results.



Hyperparameters were tuned via **RandomizedSearchCV**, and model performance was visualized through confusion matrices and feature importance charts.

- Best Parameters: {'subsample': 0.9, 'n_estimators': 300, 'min_samples_split': 2, 'min_samples_leaf': 5, 'max_depth': 7, 'learning_rate': 0.1}

-

Model	Accuracy	Precision	Recall	f1-score
Gradient Boosting	0.475	0.399	0.384	0.331
Gradient Boosting fine tuning	0.450	0.381	0.386	0.369

4. Deployment of Model

4.1 Deployment Strategy

This project resulted in two distinct predictive models aimed at forecasting football match outcomes:

1. **In-Game Prediction Model (XGBoost)**
2. **Pre-Game Prediction Model (Gradient Boosting)**

Each model serves a different purpose and is deployed based on the timing of prediction needs.

A. In-Game Prediction Model

- **Use Case**: Real-time match analysis platforms, coaching support tools, and live media commentary.
- **Inputs**: In-match statistics excluding the actual goals (e.g., xGoals, PPDA, key passes, shots on target).
- **Model**: XGBoost Classifier (optimized with RandomizedSearchCV)
- **Deployment Options**:
    - **Web API Endpoint**: Deployed via Flask or FastAPI, allowing real-time match statistics to be sent and predictions returned instantly.
    - **Integration**: Can be embedded into existing analytics dashboards for clubs or broadcasters.
    - **Inference Speed**: Fast prediction time suitable for real-time inference every few minutes during a match.

B. Pre-Game Prediction Model

- **Use Case**: Pre-match forecasting for coaches, analysts, betting markets, and sports journalism.
- **Inputs**: Historical team performance metrics (rolling averages over the past 5 matches).
- **Model**: Gradient Boosting Classifier
- **Deployment Options**:
    - **Batch Prediction System**: Generates predictions for upcoming fixtures daily or weekly.
    - **Web Dashboard**: Enables users to select matches and view predicted outcomes along with top influencing features.
    - **Integration**: Ideal for preview shows, betting recommendation engines, and coaching staff game prep.

Both models were wrapped with joblib and exported as .pkl files to be served in a lightweight environment using Docker for consistency across platforms.

4.2 Model Monitoring & Maintenance

- **Performance Tracking**: Implemented monitoring hooks to log key metrics (accuracy, AUC, log-loss) per match/week.
- **Retraining Triggers**: Periodic retraining is recommended every 10 gameweeks to account for form, injuries, and lineup changes.
- **Feature Drift Monitoring**: Tracked distribution of top features (e.g., xGoals, red cards) to identify data drift over time.

---

5. Conclusions

This project demonstrated that **machine learning can be a powerful tool for understanding and predicting football match outcomes**, leveraging advanced performance metrics and structured historical data.

Key Takeaways:

In-Game Prediction Model

- Achieved high predictive accuracy (**82%**) and strong class separation (AUC **0.94**) using real-time match features.

- Top predictors included **xGoals**, **shots on target**, **PPDA**, and **red cards**, aligning well with tactical intuition.
- Highly suitable for real-time analytics, especially in second-screen or coaching applications.

Pre-Game Prediction Model

- Achieved moderate performance (**47% accuracy**, AUC **0.575**), reflecting the inherent uncertainty in forecasting based on historical form alone.
- Showed promise in identifying team momentum through features like **rolling xGoals**, **key pass averages**, and **win rates**.
- Could benefit from richer data sources in the future (e.g., lineup data, weather, player injuries).

Limitations:

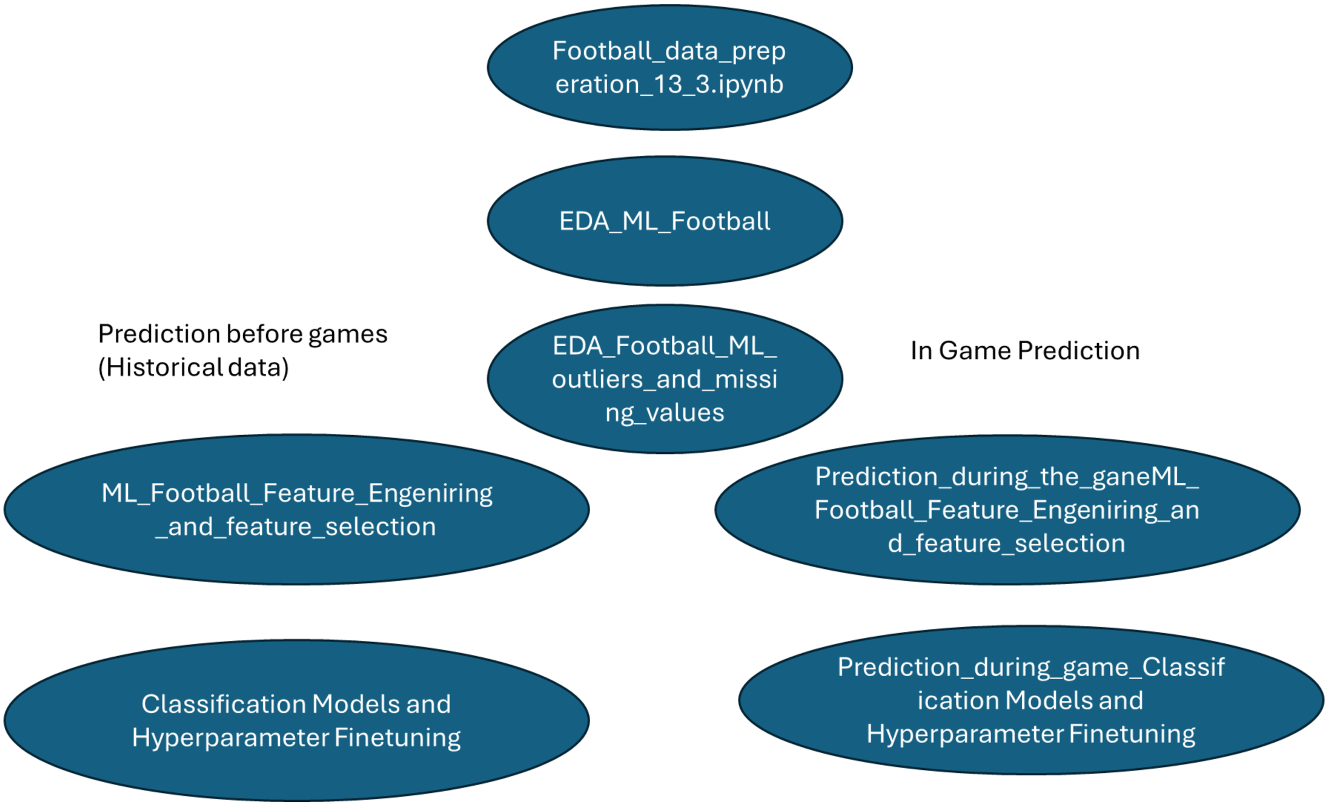
- The pre-game model struggled with predicting draws, likely due to class imbalance or insufficient tactical data.
- No external context (e.g., lineup strength, manager strategies, injuries) was used — incorporating these could boost performance.
- The dataset is historical and may not generalize to future seasons without regular retraining.

Future Work:

- **Ensemble models** combining both pre-game and in-game insights for dynamic prediction updates.
- **Explainability tools** (e.g., SHAP values) to better interpret model decisions and aid coaching staff.
- **Transfer learning** across leagues or competitions to improve generalizability.

Append 1

The model structure in the GoitHub folder



Football_data_preparation_13_3.ipynb

EDA_ML_Football

EDA_Football_ML_outliers_and_missing_values

Prediction before games (Historical data)

ML_Football_Feature_Engeniring_and_feature_selection

Classification Models and Hyperparameter Finetuning

In Game Prediction

Prediction_during_the_ganeML_Football_Feature_Engeniring_and_feature_selection

Prediction_during_game_Classification Models and Hyperparameter Finetuning