**9. Documentation and Reporting**

**Documentation**

Thorough documentation was maintained across each stage of the machine learning workflow using Jupyter notebooks:

**Data Preparation**

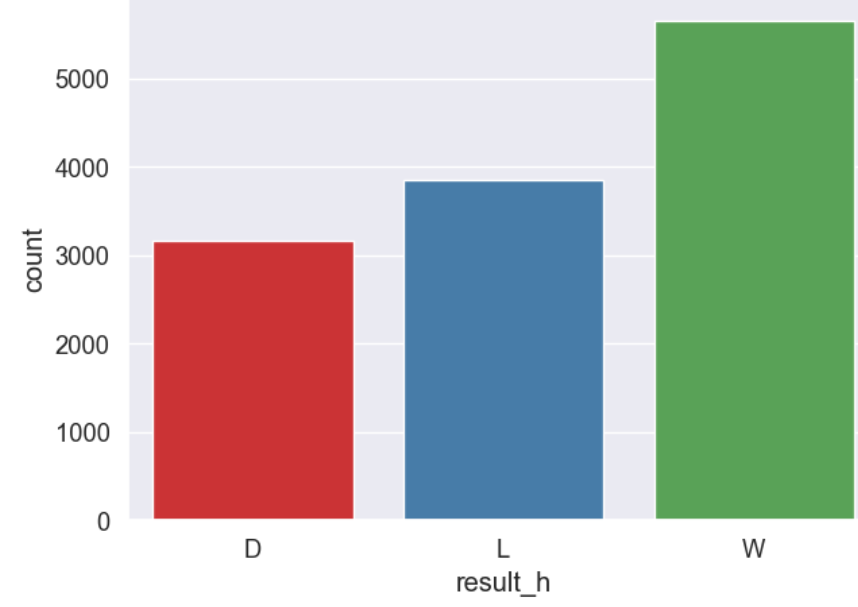
Notebook: Football\_data\_preperation\_13\_3.ipynb

* Described the integration process for multiple raw datasets (appearances, shots, teams, games).
* Detailed validation logic for player-to-team assignment in each match based on positionOrder, substitution, and goal consistency.
* Constructed the final teamstats dataset with merged and engineered variables (e.g., xGoals, deep, ppda, assists, key passes).
* Created structured datasets for modeling, ensuring each row represents a match with home and away team features.

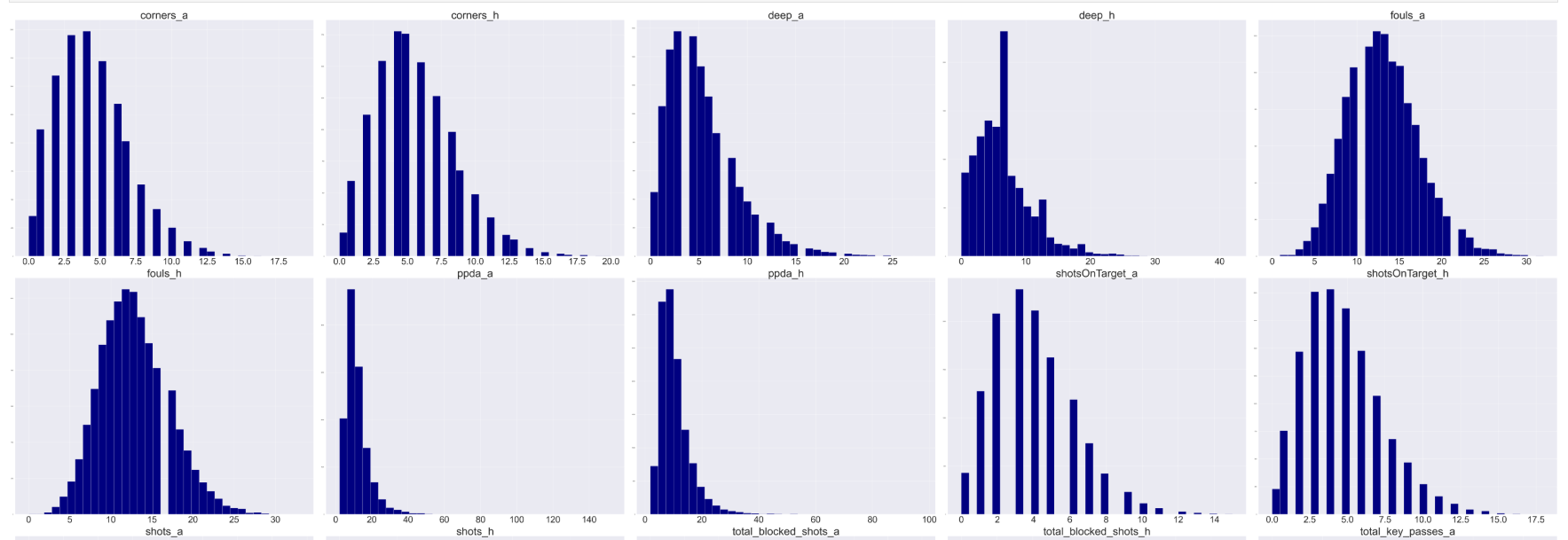
**🔹 Exploratory Data Analysis (EDA)**

Notebook: EDA\_ML\_Football.ipynb

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



* Check Normally Distributed Variables with skewness and kurtosis tests that showed that the data is skewed.



A graph of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

Skewness results:

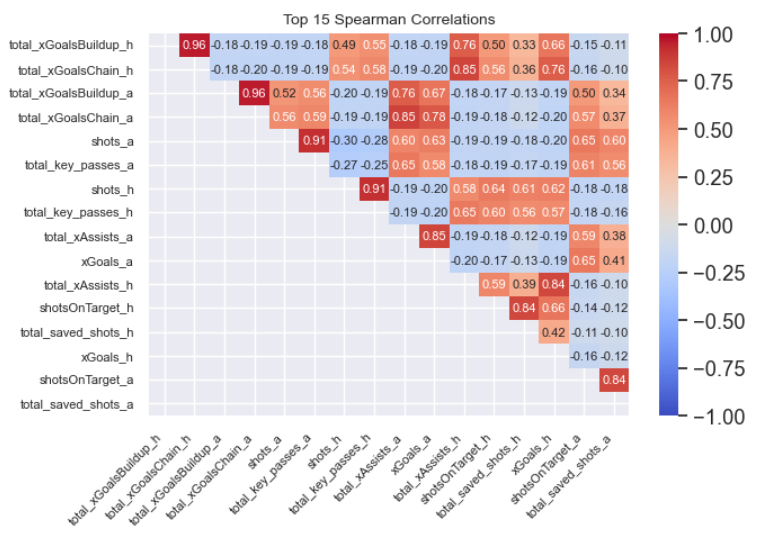
✅ Normally Distributed Variables:

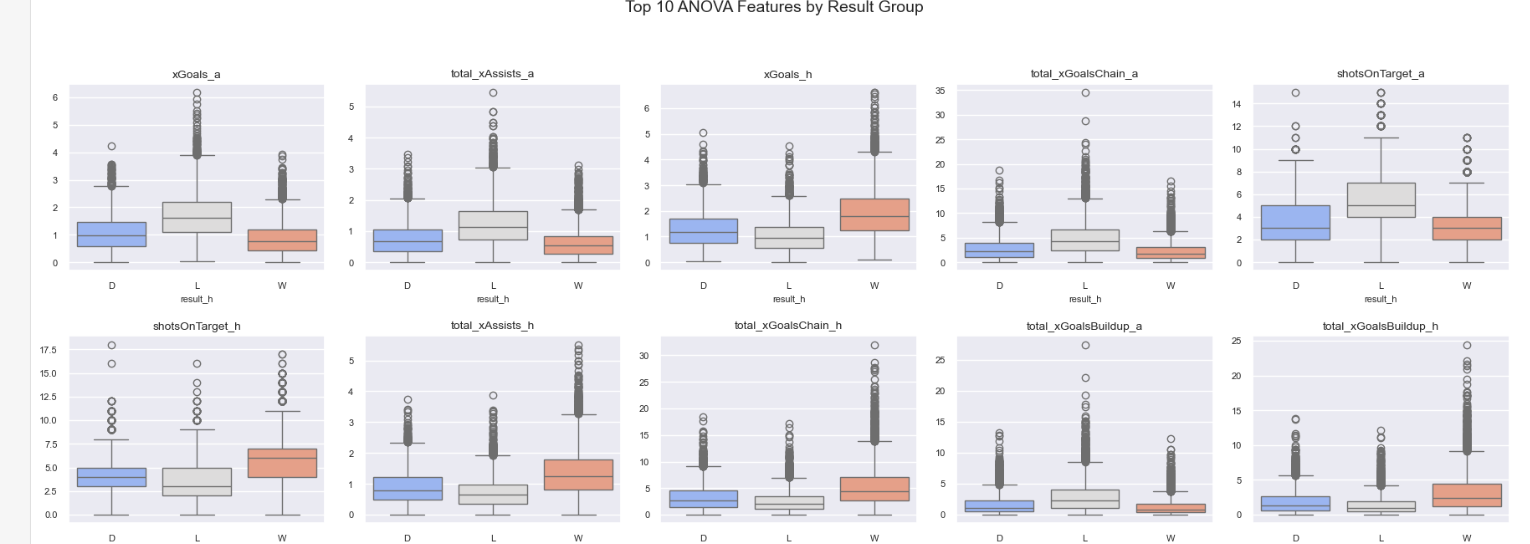
[]

❌ Not Normally Distributed Variables:

['gameID', 'homeTeamID', 'awayTeamID', 'xGoals\_h', 'shots\_h', 'shotsOnTarget\_h', 'deep\_h', 'ppda\_h', 'fouls\_h', 'corners\_h', 'total\_xAssists\_h', 'total\_key\_passes\_h', 'total\_xGoalsChain\_h', 'total\_xGoalsBuildup\_h', 'total\_blocked\_shots\_h', 'total\_saved\_shots\_h', 'xGoals\_a', 'shots\_a', 'shotsOnTarget\_a', 'deep\_a', 'ppda\_a', 'fouls\_a', 'corners\_a', 'total\_xAssists\_a', 'total\_key\_passes\_a', 'total\_xGoalsChain\_a', 'total\_xGoalsBuildup\_a', 'total\_blocked\_shots\_a', 'total\_saved\_shots\_a']

* Presented correlation heatmaps and pairplots to reveal dependencies and relationships between key variables. xGoals\_h, teamgoals\_h, and total\_assists\_h have strong positive correlations with match results.



* Chi-Square Test: Red cards (redCards\_h, redCards\_a) significantly impact match outcomes (p < 0.001).
* Used visual exploration to identify influential features like xGoals, deep, ppda, and shotsOnTarget on the result target (ANNOVA).
* 

Expected goals (xGoals\_h) are significantly higher in wins compared to losses (p < 0.001). Shots (shots\_h) show different distributions between wins and losses (p < 0.001, Kolmogorov-Smirnov test).

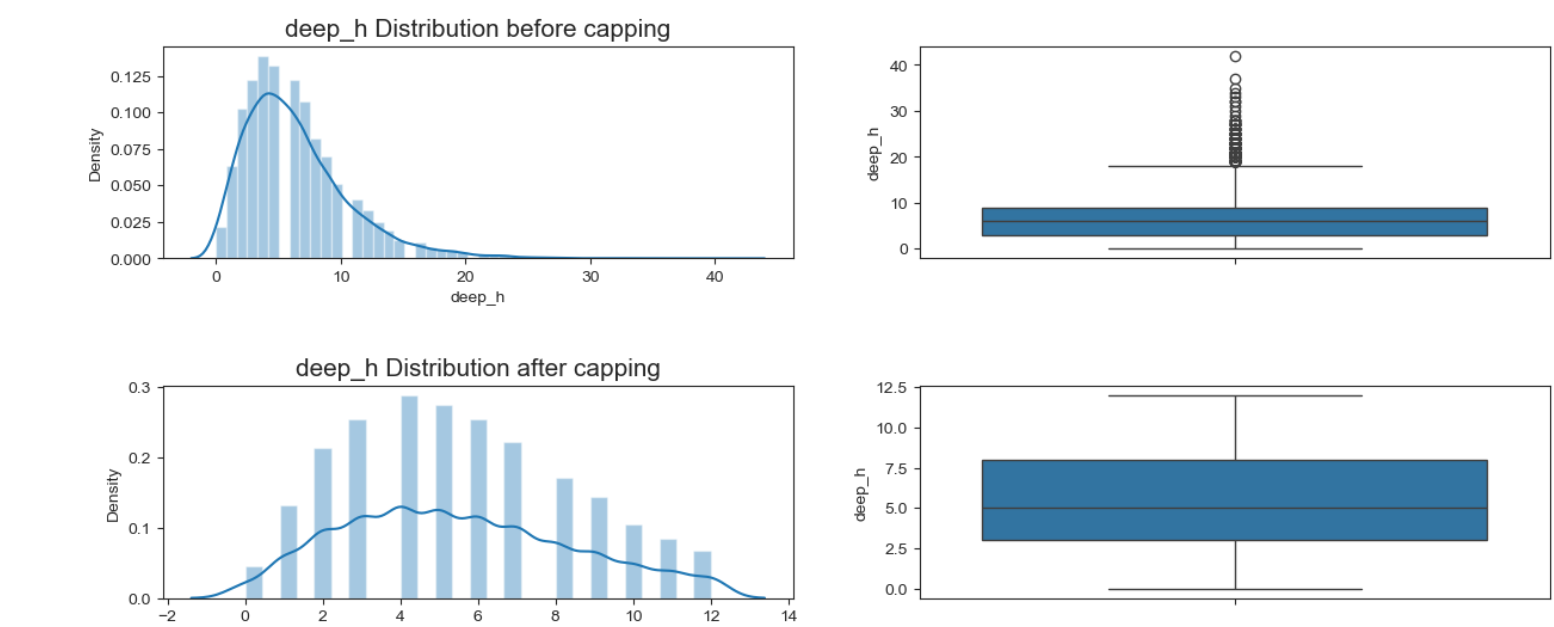
**🔹 Data Cleansing**

Notebook: EDA\_Football\_ML\_outliers\_and\_missing\_values.ipynb

* Identified and removed outliers in metrics such as shots, ppda, and cards using the IQR method.



* Visualized outliers via boxplots, histograms and calculated the change in distribution and in correlation. We decided to drop the outliers of 12 features



('corners\_a','corners\_h','deep\_a','deep\_h','fouls\_a','fouls\_h','total\_blocked\_shots\_a','total\_blocked\_shots\_h','total\_saved\_shots\_a','total\_saved\_shots\_h','xGoals\_a','xGoals\_h')

* Missing Values - We found several features with missing values most of them were numeric and one categorical. none of them effect the distribution but we decided to try and implement them by using KNN model for missing value so we wont loss anything.

A screenshot of a graph

AI-generated content may be incorrect.

**🔹 Feature Engineering**

* **Created new features** derived from player and team performance metrics, including:
* goal\_difference, xGoals\_chain, xAssists\_total, and others.
* Rolling of average 5 years or ratio calculation e.g. home\_xGoals\_h\_rolling5 or home\_win\_rate\_5 (wining ration in last 5 games)
* We didn’t use external data.
* **Categorical encoding**:
* Encoded team names and match venues using one-hot encoding and label encoding where appropriate.
* **Dimensionality reduction**:
* Applied PCA to normalized features for exploratory analysis.
* Decided to retain original features for model interpretability, as PCA components lacked clear meaning.
* We used 25 features for modeling.

**🔹 Model Selection & Evaluation**

Notebook: Classification Models and Hyperparameter Finetuning.ipynb

* Trained and evaluated a diverse range of models (listed below).
* We compute accuracy, F1, AUC, and log-loss to choose the best model.
* Documented model tuning processes using RandomizedSearchCV.
  + The best parameter were {'subsample': 0.9, 'n\_estimators': 200, 'max\_depth': 5, 'learning\_rate': 0.1, 'gamma': 1, 'colsample\_bytree': 0.7}
* Visualized model performance using confusion matrices, and feature importance(see below).

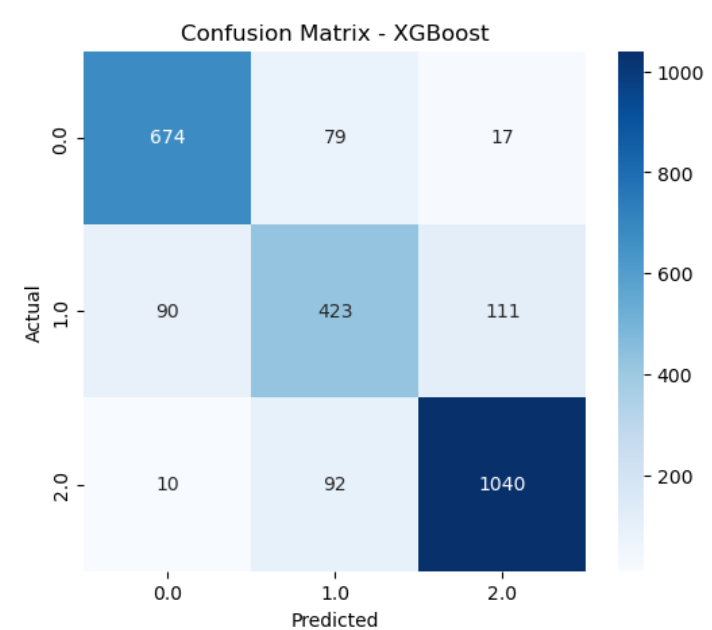
**📊 Reporting**

**Model Performance Summary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **f1-score** | **Log-loss** | **AUC** |
| Logistic Regression | 0.668 | 0.620 | 0.614 | 0.605 | 0.740 | 0.822 |
| Decision Tree | 0.689 | 0.659 | 0.656 | 0.657 | 11.214 | 0.749 |
| Random Forest | 0.772 | 0.742 | 0.737 | 0.737 | 0.569 | 0.910 |
| AdaBoost | 0.740 | 0.709 | 0.700 | 0.703 | 1.029 | 0.850 |
| Gradient Boosting | 0.793 | 0.767 | 0.766 | 0.767 | 0.477 | 0.930 |
| **XGBoost** | **0.843** | **0.824** | **0.821** | **0.823** | **0.391** | **0.949** |
| SVM | 0.450 | 0.150 | 0.333 | 0.207 | 0.996 | 0.667 |
| Extra Trees | 0.780 | 0.750 | 0.741 | 0.741 | 0.575 | 0.917 |

**Visual Assets Created**

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



**Stakeholder Communication**

* **Audience**: Coaches, analysts, sports scientists, media.
* The results have been summarized in a clear and interpretable format for decision-making.
* Visuals and tables can be used in dashboards, presentations, or briefings.
* Insights such as which features contribute most to winning (e.g., xGoals, PPDA, deep completions) are highlighted for actionability.