Predicting Football Match Outcomes using Event Data and Machine Learning Algorithms

Peter Hassard
Ulster University
Belfast

***Abstract*—This paper demonstrates how machine learning algorithms can be leveraged to predict match outcomes from 2015/16 football season for matches in the top professional leagues of 5 countries (Spain – La Liga, England – Premier League, Italy – Serie A, Germany – 1. Bundesliga and France – Ligue 1). The aim is to derive informative features based on game play activity and events to demonstrate how to predict the result of the match.**

Keywords— StatsBomb, Machine Learning, XGBoost.

I. INTRODUCTION

Football, the world's most popular sport, presents a complex challenge in predicting match outcomes, a task traditionally fraught with limited success. Recent advances in data availability have spurred a data-driven analytical revolution, aiming to unearth underlying correlations between on-pitch events and match results. This paper taps into StatsBomb's expansive repository of event-level data from European leagues [1], employing exploratory data analysis and feature engineering to discern patterns that could predict outcomes effectively.

Exploratory data analysis of the available data examines the impact of dribbling, passing, and shooting actions, enriched by spatial data, to uncover their predictive power on match results. This granular approach, informed by prior research and StatsBomb's data insights into possession values and event impacts. The aid of machine learning algorithms Random Forest [2] and eXtreme Gradient Boosting (XGBoost) [3] use exceptional performance in handling the complexity of football match data. By focusing on these algorithms, we aim to refine the predictive model, enhancing its accuracy and reliability.

Recent studies explore the strategic use of space in football, showing that control over key zones, especially the middle offensive area, significantly influences match outcomes. Casal's work (2017 [4], 2019 [5]) delves into how various factors—like build up type, ball possession, and attack strategies—affect scoring chances. Similarly, Klem [6] and Zou [7] highlight the predictive value of in-play metrics, stressing the critical role of positional data in forecasting despite football's unpredictability. These insights underscore the tactical importance of ball control across the pitch for winning strategies.

StatsBomb analyses football match outcomes by assessing possession events—Carry, Dribble, Pass, and Shots—focusing on how these actions impact goalscoring probabilities. This analysis considers the start and end locations and the play context, offering a nuanced understanding of in-game dynamics. For comprehensive details on these events and others, see the supplementary material, "StatsBomb Open Events Structure and Data Specification v4.0.0." [8].

Shots, crucial for scoring goals, predominantly occur within 17% of the pitch area; Fig. 1 illustrates 16/96 grid squares contain over 95% of the shot events. Despite comprising less than 1% of match events. Analysing the locations of shots, along with dribbles, carries, and passes, enhances our understanding of game outcomes.

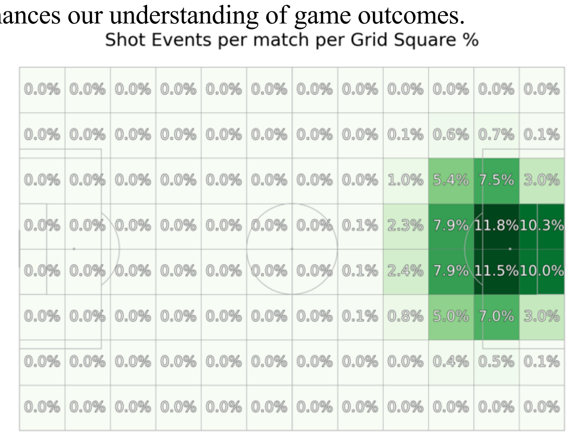

Fig. 1. The percentage per zone of the number of shot events in our dataset.

The use of expected goals (xG) and expected threat (xT) in football analysis is on the rise. xG assigns a probability to shots based on factors like distance, angle, and shot type, to predict goal likelihood. Advanced xG models, like StatsBomb's, also consider player positions and shot height, offering deeper insights into team and player performance beyond traditional metrics [9]. This approach enables a more nuanced assessment of scoring chances.

Expected Threat (xT) is a football metric that assesses a player's offensive contribution by dividing up the pitch using a grid, where each part of the grid is assigned, a value based on the scoring probability [10]. This approach, developed by Karun Singh in 2018, it calculates shooting and moving probabilities for each grid, while factoring in how players move or pass the ball. xT rates offensive contributions for teams in possession. xT is split into two components: xT from passes and xT from carries, recognizing that key actions leading to scoring opportunities and valuing strategic ball movement [11].

II. EVENT DATA DESCRIPTION AND FEATURES

StatsBomb's Python library, statsbombpy [12], was used to access detailed match and event data, including unique match IDs for efficient data import and analysis. Description of the data provided on each match in a competition is in Table 1.

The dataset which provided the match details and event metrics across 115 columns, was selectively narrowed down to 14 key columns (Table 2) this study focused on event locations, durations, and actions like passes, carries, and shots. This approach enables an attentive examination of in-possession events, analyzing the data by mapping each event with the home or away team involved in the match. This summary retains the essence of accessing, selecting, and utilising StatsBomb data for football match analysis.

TABLE I. STATSBOMB COMPETITION DATASET

Competition Dataset Columns		
Column Name	***Data Type***	***Description***
Match_id	Int	The unique identifier for the match
Competition	String	The competition or league the match was apart of
Match_week	Int	The week of the season the match took place in
Home_team	String	The team that played at their home venue
Away_team	String	The team that travel to the home team's venue
Home_score	Int	The number of goals the home team scored in the match
Away_score	Int	The number of goals the away team scored in the match

Table. 1. The columns of the Competition Dataset, data type and description.

Each row in the competition dataset refers to a match or fixture that is played between a home team and an away team. The relevant event (events with the same match id) can be mapped to a team in possession as either the home team or the away team from the corresponding row in the competition table.

TABLE II. STATSBOMB EVENT DATASET

Event Dataset Columns		
Column Name	***Data Type***	***Description***
Match_id	Int	The unique identifier for the match, corresponds to the match_id provided in the Match Dataset
Index	Int	The number indicating the chronology order of the labelled event in the match
Possession_team	String	The team that is in possession of the ball when the event occurs
Type	String	Type of event (e.g., Shot, Carry, Pass)
Duration	String	Duration of the event (in seconds)
Location	Pair of Floats	The horizontal (1st value) and vertical (2nd value) of where an event happens on the pitch. E.g.(60,40 is the center spot in the middle of pitch)
Dribble_outcome	String	Details the outcome of the dribble. E.g. complete or incomplete. Note: a completed dribble is when the player is not tackled after taking on an opponent
Pass_outcome	String	Details the outcome of the pass. E.g. incomplete, out, etc. note: nan is a complete pass.
Shot_outcome	String	Outcome of a shot event (e.g., Goal, Saved)
Shot_statsbomb_xg	Float	Expected goals (xG) for shot events, valued between 0 and 1. This is StatsBomb's calculated and recorded value
Carry_end_location	Pair of Floats	The horizontal (1st value) and vertical (2nd value) of where a carry event ends on the pitch.
Pass_end_location	Pair of Floats	The horizontal (1st value) and vertical (2nd value) of where a pass event ends on the pitch.
Shot_end_location	Pair of Floats	The horizontal (1st value) and vertical (2nd value) of where a pass event ends on the pitch.
Pass_length	Float	A calculated length of the pass from location to pass_end_location. This is StatsBomb's calculated and recorded value

Table. 2. The columns of the Event Dataset, their data type and description.

Fourteen key columns in the dataset have been distilled to develop over 20 significant match-related features to enhance the predictive power of a machine learning algorithm for match outcomes. Within the vast event dataset, unnecessary data are filtered out, focusing on types like "Pass", "Carry", and "Shot", which provided essential spatial information. This selection streamlines the dataset to a manageable subset, focusing on over 3 million rows from 1823 games, ensuring relevant details are captured without including superfluous data.

The StatsBomb dataset maps pitch locations with precise coordinates: horizontally (x) from 0 to 120 and vertically (y) from 0 to 80. Fig. 2 provides a visualization of the key locations on the pitch and their corresponding x and y values.

This precision delineates the pitch into thirds—defensive (x<40), midfield ($40 \leqslant x \leqslant 80$), and attacking (>80)—and details specific zones like the attacking penalty box (x>102; $18 \leqslant y \leqslant 62$), facilitating accurate event mapping and analysis.

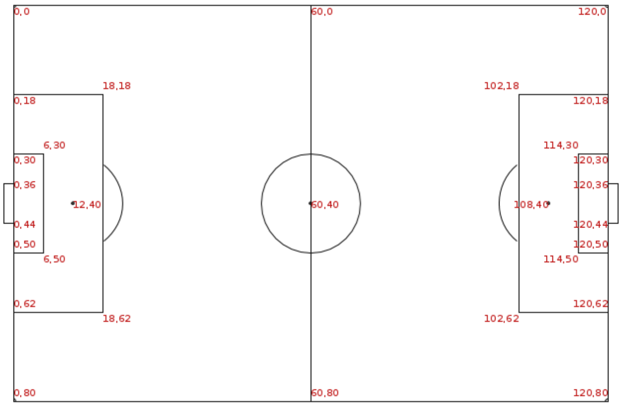

Fig. 2. Statsbomb provided pitch with labels of key locations on a football pitch, the defending goal is always on the left (x value of 0) and the attacking goal is always on the right (x value of 120).

Event durations indicate team possession times, allowing for possession percentage calculation. Passes and carries, recorded with start and end locations, contribute to team possession through successful completions.

Karun Singh's expected Threat (xT) model [13] quantifies the strategic value of these actions, assigning higher xT to moves that bring play closer to the opponent's goal. This method involves dividing the pitch into grids, with xT values adjusted based on ball movement across these zones, providing a nuanced understanding of possession dynamics and goal-scoring potential. The xT value of each grid square is visualized in Fig 3.

There are over 2,600,000 pass and carry events provided in the StatsBomb event dataset. Illustrations of these events and resulting values can be assigned to real events using the soccermatics [14] python library which is built on the matplotlib framework [15].

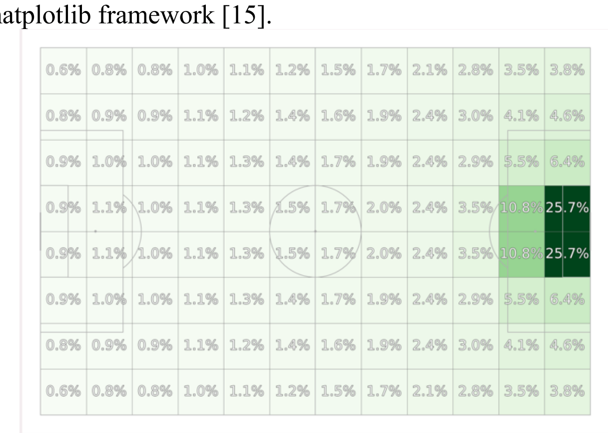

Fig. 3. A visualisation of the values of expected threat provided by Karun Singh [13]. With the goal a team is attacking on the right.

Expected Threat (xT) reflects the dynamic chances of scoring, increasing with actions closer to the opponent's goal. Since not all forward movements increase xT, highlighting the strategic play through passes and carries leading to shots. xT evaluates the potential of sequences of events (also called possession chains) ending in shots, aiding in assessing team performance by their positioning and threat generation.

Shots are assessed by StatsBomb's expected goals (xG) metric[9], factoring in shot location, defensive pressure, ball height, player contact, and goalie positioning. This pre-calculated xG, integral to assessing goal likelihood, requires no further calculations for this study.

Passes, carries and shots can be graphically illustrated using the Python Soccermatics library, Fig. 4 showcases an example of positive and negative xT values resulting in a goal. In this possession chain, possession was first gained from a throw in. Passes illustrated using in purple arrows, carries using orange and the shot is represented using gold.

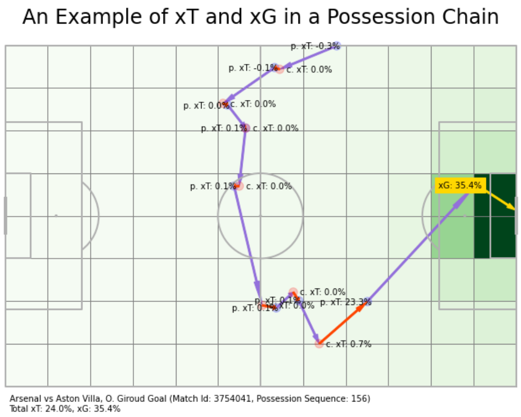

Fig. 4. A series of events in a possession chain that involved passes, carries and a shot which turned out to be a goal.

This study analyzes statistical features from 1823 games, focusing on data rather than player or team reputation. Insights from event data, alongside game knowledge and previous research, have led to identifying features categorized into four distinct groups, detailed in the subsequent chapter.

III. Feature Engineering

Utilizing historical data from the StatsBomb dataset, this paper engineers features to predict match outcomes, focusing on in-possession events. These features are categorized into **General**, **Dribble/Carry**, **Passing, and Shooting**, quantifying team performance. Table 3 details the features used. The utility of these features is assessed by analyzing their correlation with match results and goals scored, considering the actions of both home and away teams.

In Table 3 the match id distinguishes the match the event is in relation to, and the possession team name distinguishes the team the event benefits. These features all produce numeric values.

The features, centered around ball possession, offer insights for deeper data analysis. Using Seaborn's [16] Python library, a correlation matrix highlights key feature relationships. This matrix differentiates between home and away team features, correlating them with match outcomes (win, draw, loss) to uncover impactful trends. Fig. 5 shows a selection of the most interesting and relevant features based on their correlation values. The top of the correlation matrix includes features relevant to the home team and the bottom includes the features relevant to the away team. The results are indicated by H for home team win, D for draw, and A for away team win. The non-diagonal elements represent the correlation between pairs of different variables.

TABLE III. Calculated Features

	Calculated Features Columns			
	Feature Name	***Columns from Event Dataset used***	***Description of their calculation***	***Associated Features***
General Features	Possession	Duration, Location	Takes a sum of the duration of each event for each team. This will give a total value of how long a team was in possession of the ball in a match.	Possession Defending third, Possession Attacking third, Possession Percentage Share
	Events count	Location	Takes a count of the number of events each possession team has.	Events count Defending third, Events count Attacking third, Events count in the box
	Events distance from team goal	Location	This takes the mean distance of the events for that team from their own goal (location value (0,40)). And the opponents goal, (120,40)	Events distance from opponents' goal
Dribble / Carry Features	Dribbles count	Type, Dribble_outcome	Counts the total number events of type "Dribble" for each team.	Dribbles Completed, Dribbles Completed Percentage
	Carries count	Type, Location, Carry_end_location	Counts the total number events of type "Carry" for each team.	Carries Length, Carries Mean Length, Carries count Defending third, Carries count attacking third
	Expected Threat (xT) from Carries	Type, Location, carry_end_location	Accumulates the total value of xT calculated from each carry a team has in the match.	Total xT from Passes and Carries
Passing Metrics	Passes count	Type, Location Pass_outcome, Pass_length, Pass_end_location	Counts the total number events of type "Pass" for each team. Field tilt Compares each team's calculated passes in the final third and works out the percentage relevant to each team. FT of A = Att 3rd passes of A / (Att 3rd passes of A + B) %.	Passes Completed, Passes Completed Percentage, Passes Length, Passes Mean Length, Passes Defending third, Passes Attacking third, Field Tilt
	Expected threat (xT) from Passes	Type, Location, Pass_end_location, Pass_outcome	Accumulates the total value of xT calculated from each completed pass a team has in the match.	Total xT from Passes and Carries
Shooting Metrics	Shots count	Type	Counts the total number events of type "Shot" for each team.	Shots on-target count, Shots on-target percentage
	Expect Goals (xG) from Shots	Type, Shot_statsbomb_xg	Taking a sum of the Statsbomb provided xG values	Mean xG

Table. 3. A summary table of the main calculated features and brief description of how they were calculated. As well as similar features linked to the main calculated feature.

Correlation matrices were created for all the categorized features and where the key features were extracted and displayed in Fig. 5.

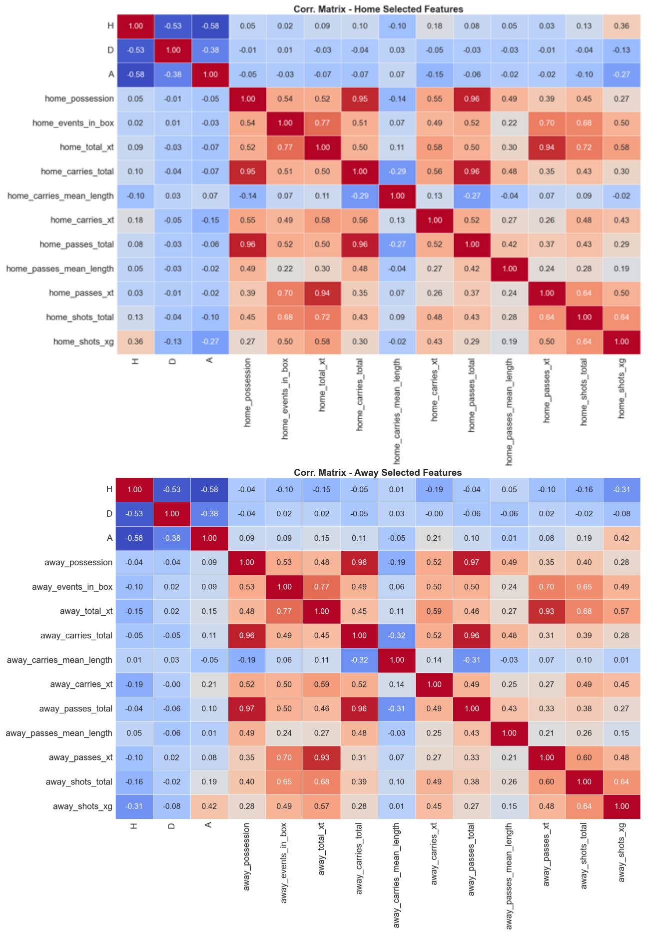

Fig. 5. Correlation matrices showing the connection between results and goal scores using selected features.

Home team data shows a strong link between possession and expected threat (xT) at 0.52, with the highest correlation (0.96) found between possession and carries xT. Home shots xG also correlates significantly with home wins (0.36), while longer carries negatively correlate with frequency (-0.39). Similarly, away team features mirror home trends, with away shots xG (0.42) and carries xT showing strong positive correlations with winning. Despite common beliefs, possession alone doesn't correlate directly with winning, highlighting the need for nuanced metrics like xT for predictive accuracy. These insights form a robust foundation for developing machine learning models.

Other opportunities for feature engineering could have included possession chain analysis as well as looking at other metrics such as blocks, saves, fouls, free kicks, corners, and many more which have been omitted here. This demonstrates many opportunities for further work on developing calculations. However, these features provided a great platform to build machine learning models, and which lead to even more accurate outcomes.

IV. Machine Learning Algorithm Variations

The study uses feature vectors from match data and engineered features, applying various machine learning models to predict match outcomes, aiming for the highest accuracy. Using the sklearn library [17], the data—1823 matches—was split into training (1458) and testing (365) sets. Models tested include Support Vector Machine, Random Forest, and XGBoost; each predicting home wins, draws, and away wins. The process involves one-hot encoding of outcomes and evaluating model accuracy based on prediction probabilities across these categories.

TABLE IV. Calculated Results

Calculated Features Columns				
Model	***Accuracy (H)***	***Accuracy (D)***	***Accuracy (A)***	***Overall Accuracy***
Support Vector Machine (SVM) Linear Kernel	0.773	0.740	0.814	0.658
Support Vector Machine (SVM) RBF Kernel	0.553	0.789	0.699	0.488
Random Forest (RF)	0.696	0.781	0.767	0.625
Extreme Gradient Boosting (XGBoost)	0.748	0.712	0.781	0.611

Table. 4. A summary table of the considered machine learning algorithms for predicting the outcome of a match.

Table 4 highlights the performance of various machine learning models, with Support Vector Machine (SVM) with Linear Kernel, Random Forest (RF), and Extreme Gradient Boosting (XGBoost) emerging as the top performers. These were all obtained using the default parameters set by Sklearn, notably for the SVM model with a linear kernel its default has no limit to the maximum number of iterations it uses on its training data, this significantly increases the runtime of fitting model compared to the others.

To evaluate model performance, each algorithm was assessed using a confusion matrix and classification report. The confusion matrix details true positives, false positives, true negatives, and false negatives, while the classification report provides metrics such as precision (correct positive predictions relative to total positive predictions), recall (correct positive predictions relative to all actual positives), and F1-score (harmonic mean of precision and recall). These tools offer insights into each model's accuracy and its ability to handle class imbalance, with macro and weighted averages highlighting performance across different classes.

The SVM model with a linear kernel was the most accurate at predicting match outcomes, achieving a 65.8% accuracy rate. This suggested definitive linear boundaries generated within the data, which also implies that the key features of football matches can directly affect the outcome matches. However, its performance dropped to 44.9% when limited to 1,000,000 iterations, indicating that while effective, its suitability for larger datasets or real-time predictions may be limited due to potential efficiency issues and the risk of infinite loops. Adjustments to the data normalization process and model parameters, like the hyperplane tolerance and regularization parameter C, could enhance model efficiency and effectiveness.

The Random Forest model ranked second in accuracy at 62.5% and performed well in predicting draws. Despite its strengths, the model faced challenges with draw predictions, indicated by lower precision and recall rates. As an ensemble of decision trees, Random Forest's effectiveness can be fine-tuned by adjusting the number of trees, or n_estimators, using sklearn's GridSearchCV. The balance between too few trees, risking underfitting, and too many, causing overfitting and increased computational demands, is crucial. Optimal tree counts were found to be 250 for home wins, 500 for away wins, and 2500 for draws, with diminishing returns beyond these numbers.

Fig. 6 shows these optimal number of trees. This enforces the fact that the random forest's precision and recall being very low for Draws the number of trees just keeps increasing and may never find an optimal value. Using the new parameters for the model results in higher accuracies for predicting just Home Wins (71.5%), Away Wins (77.8%) and Draws (79.5%) in isolation but this resulted in a small decrease to overall accuracy (61.3%).

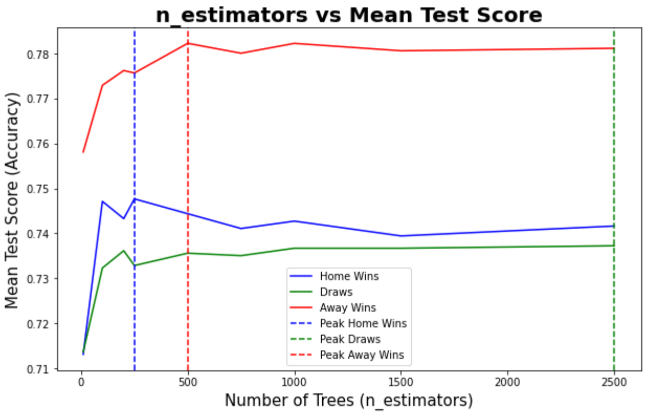

Fig. 6. A line graph plotting the Mean Test score of various n_estimator values to help decide the optimal hyperparameters for the RF model.

XGBoost achieved a 61.1% accuracy, featuring capabilities like handling missing data and regularization to curb overfitting. It allows fine-tuning through parameters such as **n_estimators** (tree number), **learning_rate** (to control learning speed and reduce overfitting), and **max_depth** (limiting tree depth). Default settings are 100 trees, a 0.3 learning rate, and a max depth of 6.

Random Forest and XGBoost both excel in football score prediction through iterative decision tree construction, each addressing the previous errors for enhanced accuracy. Random Forest uses parallel trees with bagging and feature randomness to mitigate overfitting, diversifying its model by sampling data and features. Conversely, XGBoost builds trees sequentially, focusing on correcting past mistakes to minimize bias and loss, thus gradually improving predictions.

Optimizing XGBoost for the dataset involved adjusting parameters: increasing n_estimators to 1000, decreasing the learning_rate to 0.01, and reducing max_depth to 3. This was achieved using an element of trial and error as sklearn's GridSearchCV and RandomizedSearchCV provided little or no significant improvements to overall accuracy. These changes led to an approved 64.4% accuracy. The optimized results are detailed in the confusion matrix and classification report in Table 5.

The XGBoost model achieves 64% accuracy in predicting football match outcomes, with high F1-scores for home (0.75) and away wins (0.69), but a lower score (0.26) for draws, suggesting difficulty in this area due to possible class imbalance or feature issues. Improvements might include class balancing or enhanced feature engineering. This model, particularly effective for definitive outcomes, shows promise for predicting future matches based on historical data.

TABLE V. XGBoost Confusion Matrix and Classification Report

		Actual Values		
Predicted Values		***H***	***D***	A
	H	141	27	9
	D	35	18	24
	A	9	27	141

Class	***Precision***	***Recall***	***F1-Score***	***Support***
H	*0.72*	0.80	0.75	177
D	*0.31*	0.23	0.26	77
A	*0.70*	0.68	0.69	111
Macro Average	*0.57*	0.57	0.57	365
Weighted Average	*0.62*	0.64	0.63	365
Accuracy			0.64	365

Table. 5. The XGBoost confusion matrix and classification report.

The next section will showcase how using a team's average values from their previous matches can help predict matches that are about to happen.

V. Predicting Matches Using Team Average

Simulated match predictions utilize engineered features, aggregating each team's previous match features to calculate mean values. This method involves combining home and away team data from past games to form new average feature values for upcoming matches, enhancing prediction accuracy.

Predictions start from a team's second game, utilizing mean values from previous match feature categories **General**, **Dribble/Carry**, **Pass**, and **Shot**. With each subsequent game, these averages update, incorporating all prior games' data. The XGBoost model, trained on matches up to the current one, uses this evolving dataset for predictions, requiring at least one past game for accurate forecasting.

The dataset covers 1823 matches across 171 dates, with a median of 6 and a mean of 10 games per date, ranging from 1 to 28 games. Collecting the match data in into **Match Weeks**, provided better sequential analysis of the results. Fig 7 shows the accuracy for each of the relevant match weeks.

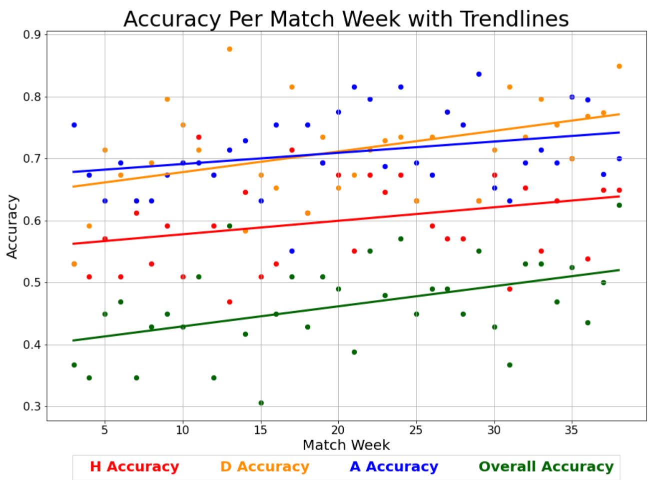

Fig. 7. A line graph plotting the Accuracy per match week using XGBoost.

As the dataset grows by 49 games per match week, it enhances the algorithm's training, notably improving draw prediction accuracy by 0.33% per week. This incremental data input boosts overall prediction accuracy by 0.32% per week, illustrating the algorithm's potential as a long-term predictive tool for football match outcomes.

The algorithm's accuracy peaked at 0.53% in the final match week, notably with a 90% prediction success rate for Spanish La Liga final match week matches. Detailed results and team performances are presented in Table 6. Predictions were made for all 5 leagues.

TABLE VI. THE FINAL MATCH WEEK OF LA LIGA PREDICTION

Competition	***Home Team***	***Away Team***	***Result***	***Prediction***
Spain - La Liga	Valencia	Real Sociedad	A	A
Spain - La Liga	Atlético Madrid	Celta Vigo	H	H
Spain - La Liga	Athletic Club	Sevilla	H	H
Spain - La Liga	RC Deportivo La Coruña	Real Madrid	A	A
Spain - La Liga	Granada	Barcelona	A	A
Spain - La Liga	Espanyol	Eibar	H	H
Spain - La Liga	Málaga	Las Palmas	H	H
Spain - La Liga	Sporting Gijón	Villarreal	H	A
Spain - La Liga	Rayo Vallecano	Levante UD	H	H
Spain - La Liga	Real Betis	Getafe	H	H

Table. 6. The Table showing the Home Team, Away Team and the Result and the Algorithms prediction for Match week 38 in La Liga.

Including real team labels in the algorithm underscores the practicality of this research and demonstrates how engineered features could impact real match outcomes.

VI. CONCLUSION AND FUTURE WORK

This research has made significant strides in predicting football match outcomes using event data and machine learning. By meticulously analysing football match event data and engineering relevant features, the research has demonstrated that certain on-pitch actions and strategic manoeuvres significantly correlate with match outcomes.

This research outlined the insights gained from the recorded data of match events alongside showcasing the benefits and draw backs of potential algorithms. Such as how XGBoost parameters can be altered to provided more optimized solutions.

This does only begin to explore the potential of using Machine Learning to predict the outcome of football matches and despite the achievements the research opens potential avenues to explore further, by expanding on the features, optimizing the algorithms further and building on more data. This may become more successful if it used data from more seasons and from more leagues to increase the number of data points the algorithms can train with. Also, there may be more patterns emerge from team's form, fixture history and specific players' influences.

ACKNOWLEDGMENT

To complete this work StatsBomb provided the sufficient data for free, allowing me to conduct the research accordingly. Thank you to Ulster University and my supervisor Dr Dermot Kerr for supporting me throughout the project.

REFERENCES

[1] StatsBomb, "The 2015/16 Big 5 Leagues Free Data Release." Dec. 2023. [Online]. Available: https://statsbomb.com/news/the-2015-16-big-5-leagues-free-data-release-ligue-1/

[2] L. Breiman, "Random Forests," *Mach Learn*, vol. 45, no. 1, pp. 5–32, 2001.

[3] XGBoost Documentation. Accessed: Dec. 01, 2024. [Online]. Available: https://xgboost.readthedocs.io/

[4] C. A. Casal, et al, "Possession zone as a performance indicator in football. The game of the best teams," *Front Psychol*, vol. 8, no. JUL, Jul. 2017, doi: 10.3389/fpsyg.2017.01176.

[5] C. A. Casal, et al "Possession in football: More than a quantitative aspect - A mixed method study," *Front Psychol*, vol. 10, no. MAR, 2019, doi: 10.3389/fpsyg.2019.00501.

[6] M. Klemp, et al, "In-play forecasting in football using event and positional data," *Sci Rep*, vol. 11, no. 1, Dec. 2021, doi: 10.1038/s41598-021-03157-3.

[7] Q. Zou, K. Song, and J. Shi, "A Bayesian in-play prediction model for association football outcomes," *Applied Sciences (Switzerland)*, vol. 10, no. 8, 2020, doi: 10.3390/APP10082904.

[8] StatsBomb Open Events Structure and Data Specification v4.0.0, *Github*. May 08, 2019. Accessed: Mar. 01, 2024. [Online]. Available: github.com/statsbomb/open-data/blob/master/doc/Open%20Data%20Events%20v4.0.0.pdf

[9] Statsbomb, "What Are Expected Goals (xG)?" Accessed: Dec. 01, 2023. [Online]. Available: What Are Expected Goals (xG)?

[10] Statsbomb, "What is Expected Threat (xT)? Possession Value Models Explained." Accessed: Dec. 01, 2023. [Online]. Available: statsbomb.com/articles/football/what-is-expected-threat-xt-possession-value-models-explained

[11] Towards AI, "Sports Analytics 101 — Expected Threats (xT)", Accessed: Dec. 01, 2023. [Online]. Available: https://towardsai.net/p/sports-analytics-101-expected-threats-xt

[12] StatsBomb, "statsbombpy." Github, 2023. Accessed: Dec. 19, 2023. [Online]. Available: https://github.com/statsbomb/statsbombpy

[13] Karun Singh, "Introducing Expected Threat (xT)." Accessed: Aug. 01, 2023. [Online]. Available: https://karun.in/blog/expected-threat.html

[14] A. A. (code) David Sumpter, "soccermatics." 2022. Accessed: Dec. 1, 2023. [Online]. Available: https://soccermatics.readthedocs.io/en/latest/

[15] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput Sci Eng*, vol. 9, no. 3, pp. 90–95, 2007, doi: 10.1109/MCSE.2007.55.

[16] M. Waskom, "seaborn: statistical data visualization," *J Open Source Softw*, vol. 6, no. 60, p. 3021, Apr. 2021, doi: 10.21105/joss.03021.

[17] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.