UCLouvain

Institute of Statistics, Biostatistics and Actuarial Science

Exploring Measures of Predictive Ability: A Transparent Labeling and Evaluation Framework for Football Analytics

Jury Members:

Dr. Van Oirbeek Robin *Supervisor*
Prof. Dr. Ritter Christian *Reader*

Thesis submitted in partial fulfillment
of the requirements for the Master's degree
in Data Science
(Statistics specialization)
by:

Mwana-Nteba Loïc

Louvain-La-Neuve
June 2025

Preface.

This dissertation has been prepared in partial fulfillment of the requirements for the master degree in data science and statistics delivered by the UCLouvain university.

Contents

Abstract

Machine learning is increasingly shaping the field of sports analytics, providing data-driven methods for evaluating player performance and decision-making in football. This thesis proposes a transparent and reproducible framework for labeling and predicting actions that contribute to high-quality shots, using Wyscout event data from the 2017/2018 Premier League season.

To begin, we propose a rule-based labeling function, `flag_contributing_actions_advanced`, which identifies actions as contributing to a quality shot by examining their temporal proximity and possession continuity relative to the shot. Using this labeling, we derive player-level metrics that capture both offensive and defensive contributions. These are then aggregated into a single metric, the `composite_score`, designed to reflect a player's overall impact on both ends of the pitch. To assess its relevance, the score is compared with external indicators such as player salaries and Ballon d'Or rankings, revealing a moderate correlation and a clear structure aligned with positional roles.

In a second phase, we evaluate the capacity of supervised models—including LightGBM, XGBoost, and Random Forest—to predict these contribution labels using engineered features from event data. We assess the impact of preprocessing choices (imputation, split strategy), tackle class imbalance with weighted loss functions, and interpret predictions using SHAP values and clustering. LightGBM emerges as the best trade-off between performance and stability (PR-AUC = 0.5013, ROC-AUC = 0.8395).

This work highlights the value of interpretable, label-driven modeling pipelines in sports analytics. As a future extension, richer contextual data such as freeze-frames and defensive pressure from StatsBomb Open Data could help refine the model's ability to capture off-ball contributions, particularly in defensive contexts.

Chapter 1

Introduction

In recent years, machine learning (ML) has gained increasing prominence in football analytics, enabling researchers and practitioners to model complex relationships between player actions and match outcomes. While traditional approaches such as Expected Goals (xG) and VAEP assign continuous-valued scores to individual actions, this thesis explores a complementary perspective: the definition and evaluation of interpretable *binary labels* that indicate whether a player action contributes to the creation of a dangerous shot.

The central goal of this study is to evaluate the effectiveness of this binary labeling framework and to assess how it can be used to train and validate machine learning models for the purpose of player performance analysis.

To structure this investigation, we articulate six interrelated research questions, combining classical machine learning concerns with football-specific challenges in labeling, evaluation, and generalization:

RQ1. **[Labeling] Can the design of a labeling pipeline reflect real-world player value?**
This research investigates whether the structure and parameters of a labeling pipeline — particularly those governing quality shot definition, temporal windows, and possession continuity — can yield contribution scores that align with external indicators of player value. Instead of relying on complex tagging schemes for contributions, we adopt a tag-free, rule-based approach for flagging action sequences. Each configuration is assessed using offensive, defensive, and composite metrics, then compared to external benchmarks such as player salaries and team standings.

RQ2. **[Estimator] Which type of model achieves the best predictive results?**
This research question aims to determine which machine learning models are most effective at predicting binary contribution labels. The comparison spans several algorithms, including tree-based methods such as Random Forest, LightGBM, XGBoost, and CatBoost, alongside linear models. Their performance is assessed through standard classification metrics, including accuracy, ROC-AUC, precision, and recall. Particular emphasis is placed on identifying models that offer a suitable trade-off between predictive accuracy and interpretability, in order to support practical deployment in football analytics.

RQ3. **[Preprocessing] How do different preprocessing techniques affect model performance?**

Preprocessing is a crucial step in any ML pipeline. Here, we compare various strategies for handling missing values (e.g., mean, median, or KNN imputation) and examine the influence of sampling methods (e.g., chronological versus random splits) on downstream performance. The goal is to understand how these technical choices shape the robustness and reliability of results.

RQ4. **[Evaluation Alignment] Do the model's predictions align with real-world football metrics?**
To assess practical relevance, we compare model outputs with external football indicators such as player wages, Ballon d'Or rankings, and estimated transfer values. A positive correlation would support the idea that the model captures meaningful

Chapter 2

Literature Review

Modern football clubs are no longer relying solely on goals and assists to assess players. With millions of data points available from every match. Indeed, Machine learning has become a fundamental component of what is known as football analytics. Given the inherent complexity of football and the significant role of chance in the sport, the application of ML techniques enables analysts to more effectively extract insights from voluminous data sets. The applications of machine learning techniques can be grouped into three main areas: injury prediction, performance analysis and talent recruitment [1].

The aforementioned areas facilitate enhancements in football teams' tactical comprehension, training regimens, and player recruitment processes. By limiting their analysis to conventional football metrics, such as goals and assists, analysts can only gain a limited understanding of the impact of individual players. ML enables the utilisation of more intricate data sets, including player positioning and movement patterns throughout a match. Consequently, players, coaches and staff analysts are able to gain a more profound comprehension of the game. This optimises the outcome of each match.

Recent advancements in data collection—particularly the increasing availability of spatiotemporal event logs—have significantly expanded the scope for applying machine learning (ML) in football analytics. These developments allow analysts to move beyond traditional statistical approaches by employing algorithms capable of identifying subtle and nonlinear patterns in player behavior. Models based on decision trees, such as Random Forest, XGBoost, and LightGBM, are especially effective in modeling the complexity of football actions due to their ability to capture interactions and hierarchical decision rules. Meanwhile, deep learning architectures bring additional value by detecting temporal dependencies and sequential dynamics throughout different phases of play. Beyond prediction, such models can also provide tactical insights through interpretability tools like feature importance scores and saliency maps.

In the ensuing sections, a comprehensive exposition of the pivotal applications of ML in the domain of football analytics will be presented. In Section 2.1, a discourse will be held on Injury Prediction Models, with a focus on the utilisation of ML techniques, such as ADTree and XGBoost, for the estimation of players' injury risks through the analysis of training loads, physical assessments, and biomechanical data. Section 2.2 will address Performance Prediction Models, where we illustrate how statistical and ML approaches, such as Poisson-based

models, decision trees, and deep learning frameworks, are used to predict match outcomes, player performance, and tactical dynamics. Following that, Section 2.3 will focus on Talent Identification Models demonstrating how the integration of conventional statistics with advanced action-based metrics and contextual data enables a more comprehensive evaluation of player potential. Finally, Section 2.4 focuses on evaluating actions in football analytics. This section explores the role of contextual variables and discusses the shift from traditional metrics to more advanced, sequence-based models—such as Large Event Models (LEMs)—for assessing the impact of individual player actions. This structured discussion will provide the reader with a clear roadmap of the various machine learning applications in modern football analytics.

2.1 Injury Prediction Models

Predicting the risk of injury makes perfect sense in football, as injuries not only disrupt team performance but can also severely hamper players' careers. ML models enable teams to estimate players' injury risk based on training loads, physical assessments and biomechanical measurements. Models such as ADTree and XGBoost are often used for this task. Reported accuracy rates typically range from 66% to 85%, with most studies framing the task as a binary classification problem—predicting whether a player will sustain a specific type of injury (e.g., muscle or non-contact) within a given time frame based on workload, fitness, and physiological indicators [1]. Teams can therefore tailor their training routine to the unique needs of their players, as well as adapting the training schedule. The aim is to reduce the risk of injury.

Recent studies have validated the potential of ML for injury forecasting in professional teams. For instance [2] utilised decision trees and ensemble methods to analyse GPS training data, encompassing features such as total distance, accelerations, and high-speed running, to predict non-contact injuries in elite players, attaining accuracy rates of approximately 70%. [2]

This is a constantly evolving field, an example of which is the inclusion of long-term data, such as a player's previous injuries and high-speed running statistics, to make prediction models even more effective. What's more, the data is often manually encoded, so there's a risk of human error and inconsistency in event definitions. Optical tracking data—camera-based systems that capture player and ball positions throughout the match can also be used. At first sight, these are more objective, but they can be affected by quality problems such as camera resolution limits or player occlusion, which have an impact on the derived metrics [3]. Furthermore, recent injury models incorporate a range of external and internal load metrics—respectively referring to physical output (e.g., distance run, accelerations) and physiological response (e.g., heart rate, perceived exertion)—including the acute:chronic workload ratio (ACWR), player fatigue scores, and even heart rate variability (when available), to provide a more holistic injury risk profile [4]. The ACWR is calculated as the ratio of acute workload—typically defined as the total training load over one week—to chronic workload, which reflects the average training load over the preceding three to four weeks. The purpose of this calculation is to determine whether an athlete's recent training has exceeded their longer-term capacity. A body of research has repeatedly demonstrated a strong correlation between elevated ACWR values and an increased risk of non-contact injuries. This association

is hypothesised to be attributable to inadequate physiological adaptation to recent increases in training load. Ensemble-based classifiers such as Random Forest and XGBoost are particularly well-suited to this approach, given their capacity to capture nonlinear interactions and model complex patterns in player workload and recovery.

However, injury prediction remains a complex challenge due to a number of factors. The first is the nature of the data itself. A common practice is the use of subjective information, such as health questionnaires, which can be biased by players' desire to participate in matches, making it difficult to obtain an accurate picture of their physical condition [3].

Furthermore, the difficulty of distinguishing correlation from causation complicates the identification of the factors actually responsible for injuries [5]. Finally, restricted access to team data limits researchers' ability to develop more accurate predictive models [3].

This constraint has prompted several researchers to explore alternative approaches that focus on physical and physiological variables routinely collected during training sessions and fitness assessments. For instance, [6] introduced injury risk prediction models that incorporate indicators such as body composition and performance in standardized fitness tests among professional footballers. Their findings also highlight a recurring challenge in this domain: the pronounced class imbalance between injured and non-injured cases. Such imbalance can bias the models toward the dominant class, thereby undermining predictive reliability. Addressing this issue goes beyond simple algorithm adjustments; it necessitates careful application of resampling strategies and a shift in evaluation focus toward metrics like precision-recall curves or the F1 score, which offer a more faithful assessment of model effectiveness in imbalanced scenarios than overall accuracy.

2.2 Performance Prediction Models

Can we anticipate which teams will perform better this season, or which players will shine? Coaches and analysts have long tried to answer these questions with stats like goals or assists. But football is too complex for simple counts. With the rise of ML, we can now model performance based on a combination of player attributes, game context, and team strategy. For example, decision trees and ensemble methods can detect patterns that aren't obvious to the human eye. The use of statistical models to predict match results is one of the earliest applications of statistical analysis in sport. The Dixon and Coles (1997) model [7] uses a Poisson distribution to model the number of goals scored by each team. The Elo model, used for many years in various sports, is a scoring system used to quantify the relative strength of a team or athlete. It works by updating each team's rating after a match based on the actual outcome compared to the expected result—teams gain or lose points depending on whether they outperform or underperform expectations. These systems can be used to predict the outcome of a match and identify favorites in a competition [7]. With the introduction of ML, players can now be classified by position on the pitch (striker, midfielder, defender and goalkeeper), as illustrated in the article [8].

In a first experiment, the aim was to predict a player's position on the basis of his attributes. Using ML algorithms such as Random Forest and Sequential Minimal Optimization (SMO), a classification accuracy of 81.5% was achieved. Players were classified into four different positions. The second experiment involved predicting the number of goals scored in

a season by Lionel Messi and Luis Suarez (two FC Barcelona strikers at the time), using performance data from previous seasons as training data. They compared the results of four different algorithms: Random Forest [9], Logistic Regression [10], MLP Classifier [11] and Linear Support Vector Classifier (SVC) [12] . In the end, Random Forest gave the best results, with an estimate of the number of goals close to the actual number of goals scored during the 2017-2018 season [8]. They repeated the experiment, this time focusing on the number of shots taken by these players during a match. Once again, Random Forest produced the best results.

In [13], two distinct perspectives are explored: the prediction of team performance and the prediction of player performance. The initial experiment is based on two distinct strategies. The first entails forecasting whether a team would achieve a higher ranking in the 2017-2018 season compared to its positions in the preceding two seasons—a binary classification task. The application of the Random Forest technique correctly predicted this outcome in approximately 70%of cases. The second strategy involves simulating matches from the 2018-2019 season to estimate the probabilities of three possible outcomes: home win, draw, or away win. Among the teams analyzed, the model achieved its highest predictive accuracy with the Premier League team at 57%, while the LaLiga (Spain's top football division) team obtained the lowest success rate.

In the second experiment, researchers sought to identify which characteristics and movements during a match can influence the score achieved by a defender. The study consisted of 59 central defenders in the Premier League during the 2016-2017 season. The model used was a linear regression with backward elimination, which yielded an R-squared coefficient of 867%. Consequently, the characteristics that most influence a defender's performance are interceptions and clearances[13].

Today, the most frequently used models are decision trees, random forests and k-nearest neighbors. They can be used to examine game factors such as passing, shooting and possession. The application of these techniques enables to attain accuracy rates ranging from 68% to 93% [1]. This is contingent on the variables under study and the techniques employed. Teams can adapt their tactics according to observed strengths and weaknesses. These models can also be used to estimate factors such as the Rate of Perceived Exertion (RPE), which measures the difficulty felt by a player during a match or training session. Not only does this enable physical preparation to be adapted to improve match performance, it also reduces the risk of injury [1].

With the emergence of Deep Learning (DL) and Large Language Models (LLMs), a better evaluation of actions is possible.An example is presented in [14], where the HATTRICS-OBT model uses a feedforward neural network to assign a value to individual player actions on the pitch. This deep learning model is applied in post-match analysis and estimates each action's contribution to the likelihood of scoring during a possession sequence. This type of model revolutionizes conventional statistical analysis and enables us to better quantify the real impact of players' actions during a match.

Large Event Models (LEMs) are conceptually inspired by the architecture of Large Language Models (LLMs), but differ fundamentally in the nature of the data they process. Rather than working with unstructured text or video, LEMs operate on structured football event data. Their inputs consist of finely detailed and tokenized match events, capturing information such

as the type of action performed, its outcome (e.g., `isGoal`, `isAccurate`), the team involved, time stamps (period, minute, second), spatial coordinates, and match score at the time of the event. These elements are typically sourced from comprehensive datasets like the publicly available WyScout data. In this context, a football match is effectively represented as a sequential stream of discrete tokens, analogous to the way words form a sentence in natural language processing.

Utilising a sequential deep learning approach, LEMs predict each token of an event individually, continuously updating the game state. This facilitates not only the prediction of the subsequent event but also the simulation of entire matches by generating probabilistic sequences of events. While the model can perform iterative inferences with sufficient speed to support near-real-time applications, its primary strength lies in simulation and post-match analysis, enabling the testing of strategies, performance evaluation, and identification of tactical insights. [15]

In essence, while LLMs generally require textual input, LEMs are customised to operate with structured soccer event logs, thus connecting the dots between raw data collection and sophisticated analytical applications in sports analytics.

However, there are several challenges to building accurate and reliable performance models. Firstly, data quality and availability are always an issue. As in the application of injury prediction, noisy, biased or incomplete data can influence model performance. To build more robust models, it is necessary to have access to large quantities of high-quality data, which is rarely available. Secondly, the models developed are frequently constrained by their specific applications. When deployed in disparate contexts, their efficacy is frequently diminished. Consequently, the capacity to adapt to novel circumstances is constrained.

2.3 Talent Identification Models

The field of talent identification is also undergoing a rapid evolution. It is becoming increasingly common for football clubs to seek out young players who exhibit a particular style of play, which necessitates the utilisation of more sophisticated assessment instruments than those based solely on conventional statistics [14].

To support this process, performance data should be gathered from well-established platforms such as Whoscored.com and Sofifa.com, which offer a wide range of statistics, including appearances, minutes played, goals, assists, and completed passes. Nevertheless, relying solely on these general metrics is insufficient for robust talent identification. A more refined approach involves incorporating position-specific features—for instance, attributes like acceleration are particularly relevant for forwards, whereas goalkeepers are better evaluated through measures such as reaction time and positional awareness [8].

Modern talent identification models combine match statistics with richer data sources such as video tracking, wearable sensor data, and even advanced action-based metrics. For example [14] introduce a method for assessing individual player actions by quantifying their expected influence on game outcomes. This approach assigns value to passes, dribbles, tackles, and other actions based on their contribution to increasing a team's goal probability, offering a granular insight into a player's impact that transcends traditional box-score statistics.

Furthermore, these models incorporate contextual information, such as the opposition's strengths and weaknesses, the style of play, and situational variables (e.g., match status and spatial dynamics on the pitch) to more accurately forecast future performance. Machine learning algorithms, including Random Forests, XGBoost, and Neural Networks, are used to merge technical, physical, and even psychological factors, enabling predictions of not only current performance but also a player's future development trajectory [1].

In addition, advanced forecasting models like those presented by [15] have shown that simulating entire sequences of events can further refine our understanding of a player's contributions within the flow of the game. Such models capture the complex interplay of actions in real time, providing a holistic perspective that supports better talent evaluation and recruitment decisions.

Bringing together diverse data sources with advanced machine learning algorithms enables the development of talent identification models that provide a richer and more accurate assessment of a player's potential. This comprehensive strategy is particularly valuable for clubs aiming to construct competitive squads within a football environment that is increasingly shaped by data-driven decision-making.

2.4 Evaluating Actions in Football Analytics

Motivation and Shift in Perspective: In modern football, a player's true value often lies not in goals or assists, but in the subtle actions that shape the flow of the game—recoveries, line-breaking passes, or well-timed duels. Traditional statistics capture only a fraction of this complexity, leaving much of a player's contribution invisible to standard metrics.

To address this gap, analysts are turning toward models that evaluate the *process*, not just the outcome. Instead of simply counting what happened (e.g., goals, assists), the question becomes: *What moved the game forward? What disrupted a promising attack?* This shift in perspective forms the foundation of action-based valuation.

From Events to Sequences: The Role of Context The intuition is straightforward: football is sequential. Valuable plays don't emerge in isolation—they evolve over a series of coordinated actions. A precise through-ball, a successful duel, or an intelligent interception may all contribute to a dangerous shot. Ignoring such context would lead to an incomplete picture of player influence.

To better quantify this buildup, recent models integrate **spatial**, **temporal**, and **tactical context** into player evaluation. Metrics like Expected Goals (xG) [16] began this process by estimating the likelihood of scoring from a shot. More advanced models—Expected Threat (xT) [17], Expected Possession Value (EPV) [18], and VAEP [14]—extend this idea by assigning value to all on-ball actions.

As an example, the VAEP model estimates the variation in a team's probability of scoring or conceding as a result of a given action. Although this provides highly detailed insights into individual contributions, it relies on complex probabilistic modeling and the availability of rich contextual data. Moreover, its outputs are not always easily interpretable for practitioners such as coaches or scouts, which can limit its practical adoption.

A Simpler and Interpretable Alternative: This thesis adopts a more transparent approach: a **binary labeling system** that classifies each event as either *contributing* or *non-contributing* to a high-quality shot. The labels are constructed using modular heuristics based on:

- **Spatial information:** Was the action close to the opponent's goal?
- **Possession continuity:** Did it occur within a short, uninterrupted attacking sequence?
- **Tactical tags:** Was the action tagged as accurate, a key pass, a successful duel, etc.?

The idea is to maintain football interpretability without sacrificing analytical rigor. Instead of modeling probabilities, the labeling relies on well-defined, football-informed rules. These labels then serve as ground truth for machine learning models (see Chapter 3).

Comparison to Existing Approaches: Advanced models such as xT [19] or xOVA [20] require dense context and complex prediction pipelines. By contrast, this binary approach:

- **Improves interpretability**: Labels are discrete and easy to explain to practitioners.
- **Simplifies modeling**: It reduces dependence on large datasets and probabilistic assumptions.
- **Aligns with domain logic**: It reflects the intuitive way in which coaches and analysts break down attacking sequences during match analysis.

While the cost is granularity, the benefit is transparency—a key requirement when models are used for talent identification or tactical assessment.

Design Principles Behind the Labeling System This thesis operationalizes the above intuition through four design choices:

- **Target Outcome Shift:** Actions are valued not for scoring a goal, but for contributing to a high-quality shot.
- **Contextual Adaptation:** The labeling process takes into account spatial positioning, the nature of the event, and the presence of pressure to better mirror the actual match context.
- **Sequence-Based Logic:** A sliding window traces up to W prior actions unless broken by P opponent actions.
- **Discrete Labeling:** Binary classes ("contributing" vs "non-contributing") enable straightforward evaluation and training.

This labeling method forms the backbone of Experiments 1. In Experiment 1, it supports descriptive statistics; in Experiments 2 it defines the prediction targets for supervised models.

Chapter 3

Methodology and Model Foundations

This chapter outlines the complete methodological framework that supports the experimental work carried out in this thesis. It opens with a description of the datasets employed, encompassing both football event data and external metrics related to player valuation. The chapter then introduces the labeling strategy used to identify actions contributing to dangerous shots, before presenting the formulation of a composite performance score. The remaining sections provide a detailed account of the feature engineering process, the predictive models implemented, and the evaluation criteria used to measure model effectiveness.

3.1 Data Source and Preprocessing

3.1.1 Data Source and Scope

The data used in this study originates from the publicly available Wyscout dataset [21], a rich spatiotemporal football event dataset. It provides structured logs of football matches from multiple seasons and competitions. Each record in the dataset corresponds to a discrete match event (e.g., pass, shot, duel), with attributes such as:

Attribute	**Description**
`eventType`, `subEventType`	Categorical codes defining the nature of the action (e.g., pass, shot, foul).
`matchPeriod`, `eventSec`	Time of the event in the match (first or second half, plus seconds elapsed).
`x`, `y`	Normalized spatial coordinates representing the position of the action on the pitch.
`tags`	List of tags associated with the event (e.g., accurate, big chance, assist).
`playerId`, `teamId`	Unique identifiers for the player and team involved in the event.
`matchId`, `competitionId`, `seasonId`	Match metadata for grouping and filtering actions.
`venue`, `date`	Contextual metadata (stadium, date of the match).

Table 3.1.1: Overview of event dataset attributes used in the labeling process

For the current stage of the thesis, the scope has been narrowed to the English Premier League 2017/2018 season. This decision ensures tractability and consistency during

initial experimentation while offering a representative sample of elite-level football. In future phases—especially for Experiment 3 and generalizability—the analysis will be extended to all five major European leagues: Premier League, La Liga, Serie A, Bundesliga, and Ligue 1. The pipeline has been designed to scale seamlessly with this broader scope.

3.1.2 Raw Data Overview and Transformation

The original dataset consists of nested JSON files representing match events. These were transformed into tabular format using the Pandas library. The key transformation and structuring steps included:

- **Flattening nested dictionaries:** Fields such as `tags` and `positions` were unpacked and converted into binary columns (see Table A.1.1 for tag definitions).
- **Normalization of spatial features:** Pitch coordinates were scaled to a standard 100x100 pitch.
- **Feature engineering:**
  - `distance_to_goal`: Euclidean distance from the shot location to the center of the opponent's goal (at coordinates $(100, 50)$ on a normalized pitch).

$$\text{Distance to Goal} = \sqrt{(x_{\text{shot}} - 100)^2 + (y_{\text{shot}} - 50)^2} \tag{3.1.1}$$

  - `angle_to_goal`: Shooting angle calculated between the shot position and the two goalposts located at $(100, 36)$ and $(100, 64)$. Let $\vec{v}_1$ and $\vec{v}_2$ be the vectors from the shot location to each post:

$$\theta = \arccos\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\|\|\vec{v}_2\|}\right) \tag{3.1.2}$$

    The cosine is clipped to $[-1, 1]$ for numerical stability.
- **Player metadata integration:** Player short names, positions, and nationalities were merged into the event-level dataset using Wyscout's player lookup table.
- **Event alignment:** Events were sorted in chronological order and linked to the appropriate team (Team A or Team B) to allow accurate tracking of possession sequences.

Rows with unknown players (`playerId` = 0) were excluded from the dataset to maintain quality and interpretability.

3.1.3 League Filtering and Experimental Setup

Matches were filtered using metadata fields such as `competitionId` and `seasonId` to isolate only Premier League matches from the 2017/2018 season. This subset allows for focused evaluation while keeping computational requirements manageable.

In the experimental pipeline, matches are split into training and test sets. While early experiments use the full season for simplicity, later stages employ stratified splits to avoid data leakage. Future iterations will adopt temporal validation schemes (e.g., training on early-season matches, testing on later rounds) to better simulate real-world forecasting scenarios.

Element	Value
Total matches	380
Unique players	∼550
Unique events	>700,000
Events used	Passes, Shots, Duels, Fouls, etc.
Metadata used	Players, Teams, Matches

Table 3.1.2: Summary of the cleaned dataset for the 2017/2018 Premier League season

3.1.4 Dataset Summary (Premier League 2017/2018)

This cleaned and structured dataset serves as the foundation for the labeling strategies, feature extraction, and modeling approaches described in the upcoming sections. Each player's match actions are enriched with spatial and contextual metadata, making the dataset suitable for both descriptive analysis and predictive modeling tasks.

3.2 Quality Shot Definition

Motivation and Intuition

In football, not all shots carry the same tactical weight. A shot from outside the box under pressure is far less threatening than one taken centrally in open space. For this reason, accurately identifying "quality" shots—those with a high likelihood of scoring is crucial when evaluating offensive contributions.

Traditional models like Expected Goals (xG) estimate scoring probabilities using spatial factors such as shooting distance and angle [16]. While effective, these approaches are often considered opaque "black-box" systems and typically demand substantial data calibration. To enhance interpretability and maintain analytical control, this thesis adopts a transparent, rule-based approach grounded in domain-specific football heuristics to define what constitutes a quality shot.

Definition and Heuristics

A shot is labeled as a *quality shot* if it meets two main conditions:

1. It satisfies **at least one spatial criterion**, and
2. It either includes a **positive tactical tag** or occurs under **low defensive pressure**.

This dual condition ensures that both location and execution context are considered—two dimensions known to impact shot effectiveness.

- **Spatial Criteria (at least one must apply):**
  - The shot is taken from within 20 meters of the goal.
  - The shot angle (relative to the posts) exceeds 0.6 radians.
  - The shot originates from a central penalty zone: $x \geq 80$ and $30 \leq y \leq 70$.

  These criteria are designed to approximate typical xG zones, while preserving a high degree of interpretability. They correspond to pitch areas that have been empirically shown to offer a greater likelihood of scoring.

- **Positive Semantic Tags (at least one required):**
  - `tag_1801` – Accurate shot
  - `tag_201` – Big Chance
  - `tag_302` – Key Pass
  - `tag_703` – Successful Dribble

  These tags indicate either high-quality execution or valuable buildup play, as recognized by Wyscout analysts.

- **Low Defensive Pressure (Fallback Rule):**
  A shot may still be labeled as "quality" if it was taken without significant defensive interference—that is, in the absence of:
  - `tag_701` – High Pressure
  - `tag_1401` – Sliding Tackle
  - `tag_1501` – Duel

  This condition captures open-play chances arising from lapses in defense or transition situations.

- **Set-Piece Adjustments:**
  - Direct free kicks (`subEventId = 33`) are labeled as quality shots if taken within 30 meters of goal.
  - Penalty kicks (`subEventId = 35`) are always labeled as quality shots.

  These adjustments reflect conventional wisdom about the high expected value of well-positioned set pieces.

Implementation and Modularity

The full set of rules is implemented within a modular Python function, `make_quality_shot_fn()`, which supports parameter customization (e.g., radius, angle thresholds, tag selection). This flexible design enables controlled ablation studies and facilitates consistent comparisons across experimental setups. Unlike black-box ML-derived scoring methods, this rule-based approach emphasizes transparency and interpretability, making it particularly suitable for use in scouting, coaching, and downstream analyses of player contributions.

3.3 Core Innovation: Interpretable Tactical Labeling for Football Actions

Once quality shots are identified (see Section 3.2), the next step is to label the actions that contribute to their creation. The goal is to distinguish meaningful actions—such as incisive passes, duels, or recoveries—from those that are not tactically linked to dangerous outcomes.

In this thesis, a binary classification framework is introduced to label each action as either **contributing** (label = 1) or **non-contributing** (label = 0) to the creation of a quality shot. These annotations act as the reference labels for training and evaluating machine learning models. The proposed system is designed to be modular, interpretable, and firmly rooted in football-specific tactical logic.

3.3.1 Sliding Window and Possession Threshold

The labeling function uses a sliding window of $W = 7$ actions and a possession threshold of $P = 2$ to determine whether an action contributes to a high-quality shot. We apply a simple rule: we look at the 7 most recent actions by the same team before the shot occurs. This reflects a common tactical idea in football that most dangerous opportunities are created within a short sequence. However, if the opposing team intervenes and performs more than two consecutive actions, we consider that the attacking move has been disrupted and the chain is broken. We refer to these two parameters as the window ($W = 7$) and the possession threshold ($P = 2$). While several combinations were explored during Experiment 1, we chose this configuration because it offered a good balance between football intuition and empirical robustness.

– **Window (W):** The window defines the *maximum number of prior actions* that can be considered part of the same offensive sequence leading to a quality shot. In our setup, $W = 7$ means that, when a quality shot is detected, we scan back up to 7 actions from the same team to check which ones contributed to its buildup. This reflects a tactical assumption: most meaningful offensive actions occur within a short sequence of events.

– **Possession Threshold (P):** This parameter defines the maximum number of consecutive actions by the opposing team that can occur before an attacking sequence is considered interrupted. When more than $P = 2$ such actions take place prior to the shot, the contribution chain is deemed to be broken.

Although a full grid of (W, P) combinations is evaluated in Experiment 1 (see Section 4.2), the configuration $W = 7$, $P = 2$ is used throughout the thesis as the default setting, striking a balance between football interpretability and empirical robustness.

3.3.2 Key Contribution Conditions

An action is flagged as contributing if it belongs to a sequence leading to a quality shot and satisfies one of the following role-specific tactical filters:

– **Passes (eventId = 8):**
- Tagged as `Assist` (`tag_301`) or `Key Pass` (`tag_302`); or
- Belongs to subevents like Smart Pass, Cross, Launch, or Through Ball (`subEventId` ∈ [80–86]) and is `Accurate` (`tag_1801`).

– **Duels (eventId = 1, subEventId ∈ [10–12]):**
- The duel must be `Won` (`tag_703`) or result in retained possession with accuracy (`tag_702` and `tag_1801`).

– **Interceptions (eventId = 7, subEventId = 72):**
- Must be tagged as `Interception` (`tag_1401`).

– **Clearances:**
- Tagged as `Clearance` (`tag_1501`) and `Accurate` (`tag_1801`).

– **Counter Attacks:**
- Tagged as `Counter Attack` (`tag_1901`) and `Accurate` (`tag_1801`).

These filters ensure that only actions that are tactically significant and technically successful are retained. A bidirectional pass-through ensures fairness: both Team A and Team B are evaluated using the same rules.

3.3.3 Implementation of the Contribution Labeling Function

This subsection provides the full specification and pseudocode of the labeling algorithm used throughout the thesis. It operationalizes the tactical rules described earlier into a modular function used to annotate actions with offensive and defensive contribution flags. The labeling procedure is implemented in a custom function, `flag_contributing_actions_advanced`, which assigns the following outputs for each action:

– `contributes_to_quality` $\in \{0, 1\}$ — whether the action contributes to a quality shot.
– `steps_to_shot` $\in \mathbb{N} \cup \{\text{NaN}\}$ — number of actions before the shot occurs.
– `disrupts_opponent_quality` $\in \{0, 1\}$ — whether the action breaks an opponent's quality shot chain.
The algorithm proceeds as follows:

Algorithm 1 `flag_contributing_actions_advanced`: Flag offensive and defensive contributions

Initialize: `contributes_to_quality = 0`, `steps_to_shot = NaN`, `disrupts_opponent_quality = 0`
foreach *match in dataset* **do**
  Sort events by `eventSec` Identify teams A and B **foreach** *action* a_i *in match* **do**
    Let `team_id` ← team of a_i Initialize: `same_team_count = 0`, `opponent_count = 0`
    `// Forward Contribution Scan`
1   **foreach** *future action* a_j *after* a_i **do**
2   **if** *team of* $a_j \neq$ *team_id* **then**
3   Increment `opponent_count` **if** *opponent_count* $\geq$ *possession_threshold* **then**
4   break
5   **else**
6   Increment `same_team_count` **if** a_j *is a quality shot* **then**
7   Label a_i: `contributes_to_quality = 1` Set `steps_to_shot` break
8   **else if** *same_team_count* $\geq$ *window* **then**
9   break
10   **if** a_i *is a quality shot* **then**
11   Set `contributes_to_quality = 1`, `steps_to_shot = 0`
12   `// Backward Disruption Scan`
13   **if** a_i *is a key action* **then**
14   Let `opponent_team` ← other team Initialize `opponent_sequence = 0` **foreach** *past action* a_k *before* a_i **do**
15   **if** *team of* $a_k \neq$ *opponent_team* **then**
16   break
17   Increment `opponent_sequence` **if** a_k *is a quality shot* **then**
18   break
19   **if** *opponent_sequence* $\geq$ *possession_threshold* **then**
20   Label a_i: `disrupts_opponent_quality = 1`
return dataframe with labeled columns

The labeling framework serves as the cornerstone for the experimental design adopted in this thesis. In Experiment 1, it supports the computation of descriptive metrics, including average offensive and defensive contribution ratios at the player level. In Experiment 2, the offensive label (`contributes_to_quality`) is employed as the binary target variable for supervised learning models.
Both types of labels are generated using the `flag_contributing_actions_advanced` function, which also enables the derivation of a composite score that combines offensive and defensive contributions. Although only the offensive label is used as the prediction target, the defensive annotation (`disrupts_opponent_quality`) remains essential for building performance indicators at the player level. This dual-labeling setup offers analytical flexibility while remaining firmly grounded in football-specific tactical principles.

3.4 Composite Performance Score

While binary contribution labels allow us to detect whether a player action leads to a quality shot, they do not fully capture a player's overall match impact. To address this, we construct a *composite score* that quantifies player performance by integrating both offensive and defensive contributions over time. This aggregated metric is used for player ranking and correlation analyses in Experiment 1, and for training supervision in later modeling tasks.

Motivation and Design

In football, a player's contribution cannot be fully understood through a single dimension. While some players distinguish themselves through offensive actions—such as key passes or assists—others play a crucial role in disrupting the opponent's build-up and preventing dangerous situations. Focusing solely on attacking contributions would lead to an underestimation of defenders and deeper-lying midfielders whose influence is less visible but equally important.

To account for this diversity of impact, we propose a *composite score* that integrates both offensive and defensive contributions into one interpretable measure. This score embodies the idea that players can add value in multiple ways, and that both attacking and defensive efforts are essential in evaluating overall performance.

The score is defined as a weighted average of two core statistics:

$$\text{composite\_score} = \alpha \cdot \text{avg\_contrib\_ratio} + (1 - \alpha) \cdot \text{avg\_disrupt\_ratio} \tag{3.4.1}$$

where:

- `avg_contrib_ratio` captures the average number of actions per match that lead to a quality shot(Full details in 3.7),
- `avg_disrupt_ratio` captures the average number of actions that successfully disrupt the opponent's chance creation (Full details in 3.7.9),
- $\alpha \in [0, 1]$ controls the trade-off between offensive and defensive contributions.

In this thesis, we set $\alpha = 0.7$, thereby assigning more weight to offensive contributions. This choice reflects the tactical reality that opportunities to score are generally less frequent but often more decisive than defensive interventions. Although α was not selected through a formal optimization process, it is consistent with football intuition, as it highlights forward-oriented impact while still accounting for the importance of defensive performance.

Score Calculation and Interpretation

To compute the composite score for each player, we follow four steps:

1. First, all player actions are labeled using the function `flag_contributing_actions_advanced` (see Section 3.3).
2. Next, we aggregate offensive and defensive contributions per player, for each match.
3. We then compute the average number of contributions and disruptions per match.
4. Finally, we apply the weighted formula above to compute each player's composite score.

This approach yields an interpretable metric that captures the typical influence of a player within a match. As expected, forwards tend to exhibit higher values of `avg_contrib_ratio`, reflecting their offensive involvement, whereas central defenders often stand out in terms of `avg_disrupt_ratio` due to their defensive duties. Midfielders usually lie somewhere in between, contributing to both aspects of the game. By varying the parameter α, it becomes possible to examine how different weightings of offensive and defensive contributions influence the overall player rankings.

Scope of Use in Experiments

Throughout this thesis, the composite score plays a central role in linking player behavior to external indicators of value. Specifically:

- **Experiment 1:** It is used to rank players and analyze correlations with real-world metrics such as salary and Ballon d'Or nominations.
- **Experiment 2:** It is used as a proxy to assess whether the model's predicted probabilities of contribution are consistent with players' overall impact on the pitch.

By capturing both offensive creation and defensive disruption in a single number, the composite score offers a robust and football-relevant foundation for player evaluation across all experiments.

3.5 Feature Engineering Pipeline

In order to train predictive models that can classify whether a player action contributes to a quality shot, we construct a structured set of features from the event-level data. This section outlines the selected features, their motivations, and preprocessing steps. Features are derived from spatial, temporal, contextual, and player-specific information. They are selected to balance football interpretability with predictive utility.

Feature Categories

The extracted features are grouped into five categories:

Spatial Features – These capture the location and geometry of the action:

- `x`, `y`: Normalized field coordinates.
- `distance_to_goal`, `angle_to_goal`: Engineered from shot coordinates (see Equations 3.1.1 and 3.1.2).

- `is_central`: Binary indicator for actions within the central shooting zone.

Temporal Features – These describe the timing and sequence of the action:

- `eventSec`: Time in seconds since the start of the half.
- `matchPeriod`: First or second half.
- `sequence_position`: Position of the action within its possession chain.
- `time_since_last_event`: Time gap from the previous event.

Contextual Features – These reflect tactical and semantic metadata:

- `eventId`, `subEventId`: Categorical identifiers of action type.
- `tags`: Binary-encoded tags (e.g., accurate, key pass, duel won).
- `pressure_tag`: Indicates whether the player was under pressure.

Player Metadata – These encode player role and preferred foot:

- `role_name`: Categorical feature (e.g., midfielder, defender).
- `foot`: Preferred foot (left/right).
- `playerId`: Used only for aggregation or grouping (not model input).

Team Context – These features provide match context and team state:

- `teamId`: Team identifier.
- `is_home_team`: Binary feature indicating if the team is playing at home.
- `score_difference`: Current goal differential at time of action.

Encoding and Preprocessing

Categorical variables such as `eventId`, `subEventId`, `role_name`, and `foot` are transformed using one-hot encoding to make them suitable for model training. Numerical features are standardized via z-score normalization to ensure comparability across different scales. Missing data are addressed through imputation techniques:

- Continuous variables are imputed using either the mean or median.
- Categorical variables are imputed using the most frequent category or a dedicated `unknown` label.

In experiments that compare different imputation strategies (see 4.3.3), both mean and KNN-based imputations are evaluated to test the robustness of the models. All preprocessing steps are implemented within `scikit-learn` pipelines to guarantee reproducibility and prevent data leakage during model training and evaluation.

Feature Importance and Selection

The initial selection of features is informed by domain expertise, with a focus on variables that carry strong spatial or temporal significance—such as distance to goal, elapsed time since the last shot, or indicators of possession changes. These features are selected based on their theoretical relevance to offensive contribution and tactical behavior.

Throughout the experimentation phase, SHAP values are primarily employed to enhance interpretability, allowing us to identify which features most strongly influence model predictions. Beyond interpretability, SHAP value magnitudes also serve as a practical tool for feature selection. Specifically, we compute the mean absolute SHAP value for each feature on the test set and rank them accordingly. Features showing consistently low contributions are flagged as potential candidates for removal in subsequent iterations.

Although recursive feature elimination (RFE) was not used in a systematic manner, this SHAP-based filtering approach strikes a balance between model simplicity and explanatory power. It enables us to retain only the most informative features, thus improving parsimony without sacrificing performance.

The final set of selected features forms the basis for training the classification models discussed in the next section.

Input–Output Structure for Modeling

Each labeled action corresponds to one training instance. The input is a feature vector $\mathbf{x} \in \mathbb{R}^K$, where K denotes the total number of engineered features. These features include numerical variables (e.g., distance to goal), binary indicators derived from tags (e.g., accurate, big chance), and one-hot encoded categorical features (e.g., role, event type). Together, these components form a complete vector of dimension K for each sample.

This setup naturally supports standard binary classification pipelines. Tree-based models such as Random Forest, LightGBM, and XGBoost are particularly well-suited for this structured, mixed-type data. The same input–output structure can also be adapted for neural architectures, though tree ensembles are preferred here due to their interpretability and robustness with sparse categorical data.

3.6 Machine Learning Models

Following the implementation of feature engineering and label construction, a range of supervised learning models are used to ascertain whether an action contributes to the creation of a "quality shot". All models are trained using the structured dataset previously described, with categorical coding and standardized numerical inputs. Hyperparameter tuning is performed using `HalvingRandomSearchCV` [22], an efficient resource-aware search strategy (See Section 4.1), and the final evaluation is performed on held-out test data. The selection of models is intended to facilitate an analysis of the influence of different supervised learning approaches on interpretability, robustness to class imbalance, and compatibility with structured event data. In this section, an exposition of the theoretical background pertaining to the various selected models will be provided.

Logistic Regression [23] Logistic Regression is a linear classification model that estimates the probability of a binary outcome, mostly using the logistic (sigmoid) function. In this

context, it is also referred to as logit regression. Given a feature vector $\mathbf{x} \in \mathbb{R}^K$, the model computes the log-odds of the target as a linear combination of the input features:

$$\hat{y} = \frac{1}{1 + \exp(-(\mathbf{w}^\top \mathbf{x} + b))} \tag{3.6.1}$$

where $\mathbf{w}$ is the weight vector and b is the intercept term. The model parameters are learned by minimizing the log-loss, a negative log-likelihood function commonly used for binary classification [23].

Logistic Regression serves as a strong baseline due to its simplicity and interpretability. It is particularly effective when the relationship between input features and the log-odds of the outcome is linear. In our context, the model offers transparency in understanding how specific features (e.g., angle to goal, event type) influence the likelihood of contributing to a quality shot. Although it may not capture nonlinear relationships or complex interactions, it provides a computationally efficient benchmark. Features are standardized and one-hot encoded for interpretability and numerical stability, even though the model itself is not inherently sensitive to scale.

Random Forest [23] Random Forest is an ensemble learning algorithm that constructs a multitude of decision trees using bootstrapped samples of the training data [23]. Each tree $T^{(b)}$ is trained on a bootstrap sample $\mathcal{D}^{(b)}$:

$$\mathcal{D}^{(b)} = \{(x_i, y_i)\}_{i=1}^{n} \quad \text{where } x_i \sim \mathcal{D} \text{ with replacement} \tag{3.6.2}$$

At inference time, predictions from all trees are aggregated via majority voting:

$$\hat{y} = \text{mode}\left\{T^{(1)}(x), T^{(2)}(x), \ldots, T^{(B)}(x)\right\} \tag{3.6.3}$$

Feature importance is computed by averaging the total impurity reduction across all trees:

$$\text{Importance}(f_j) = \frac{1}{B} \sum_{b=1}^{B} \sum_{\text{splits on } f_j} \Delta\text{Impurity}_{f_j}^{(b)} \tag{3.6.4}$$

Random Forest handles mixed-type features, nonlinearities, and missing values naturally, as tree-based models are capable of splitting without explicit imputation in low-missingness contexts. It is also robust to overfitting due to aggregation and feature subsampling.

XGBoost [24] XGBoost (Extreme Gradient Boosting) builds an additive model by sequentially fitting trees to residuals [24]. At iteration t, the prediction is updated as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i), \quad f_t \in \mathcal{F} \tag{3.6.5}$$

where $\mathcal{F}$ is the space of regression trees. The regularized objective function is:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i^{(t)}) + \sum_{t=1}^{T} \Omega(f_t), \quad \text{with} \quad \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \tag{3.6.6}$$

XGBoost supports class imbalance handling via `scale_pos_weight`, and efficiently handles sparse data, missing values, and parallel computation. These properties make it particularly well-suited for structured football event data.

LightGBM [25] LightGBM is a gradient boosting framework that grows trees leaf-wise rather than level-wise [25]. This allows faster convergence and potentially higher accuracy. The objective at step t is:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell\left(y_i,\ \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{3.6.7}$$

The gain from a split is calculated as:

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i\in L} g_i)^2}{\sum_{i\in L} h_i + \lambda} + \frac{(\sum_{i\in R} g_i)^2}{\sum_{i\in R} h_i + \lambda} - \frac{(\sum_{i\in L\cup R} g_i)^2}{\sum_{i\in L\cup R} h_i + \lambda}\right] - \gamma \tag{3.6.8}$$

LightGBM handles categorical variables internally, supports class imbalance via `scale_pos_weight`, and integrates natively with SHAP for post-hoc interpretability.

CatBoost [26] CatBoost is a gradient boosting framework optimized for categorical data [26]. It uses target statistics with permutations to encode categorical variables while avoiding target leakage. A unique contribution is *ordered boosting*, which computes residuals using models trained on preceding observations only. The loss at iteration t is:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} \ell\left(y_i,\ \hat{y}_i^{(t-1)} + f_t(x_i)\right) \tag{3.6.9}$$

CatBoost handles class imbalance via custom loss functions or class weights and is known for its strong performance on structured tabular data with minimal preprocessing.

Each model's mathematical formulation and assumptions are drawn directly from the official papers or canonical texts [23, 24, 25, 26].

3.7 Evaluation Metrics

This section outlines the evaluation metrics used in this study. Definitions of standard classification metrics—such as accuracy, precision, recall, F1-score, and ROC-AUC—are based on the formalizations provided by Rainio et al. [27], a comprehensive survey of evaluation techniques in imbalanced settings. For domain-specific metrics (e.g., contribution and disruption ratios), we provide tailored definitions suited to football analytics.

Accuracy Measures the overall proportion of correctly classified instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.7.1}$$

Precision (Positive Predictive Value) Proportion of predicted positives that are actually contributing:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.7.2}$$

Recall (Sensitivity or True Positive Rate) Proportion of actual contributing actions correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.7.3}$$

F1-score The harmonic mean of precision and recall:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.7.4}$$

ROC-AUC (Receiver Operating Characteristic – Area Under Curve) [28] The ROC-AUC measures the area under the Receiver Operating Characteristic curve, which plots the True Positive Rate (Recall) against the False Positive Rate across various classification thresholds. It provides a metric for how well a model distinguishes between classes: a value of 0.5 indicates random guessing, while 1.0 indicates perfect discrimination.

$$\text{ROC-AUC} \in [0, 1] \tag{3.7.5}$$

Figure 3.7.1 illustrates ROC curves for different classifiers, highlighting the distinction between a perfect model, a random classifier, and real-world models. The image is adapted from [29].

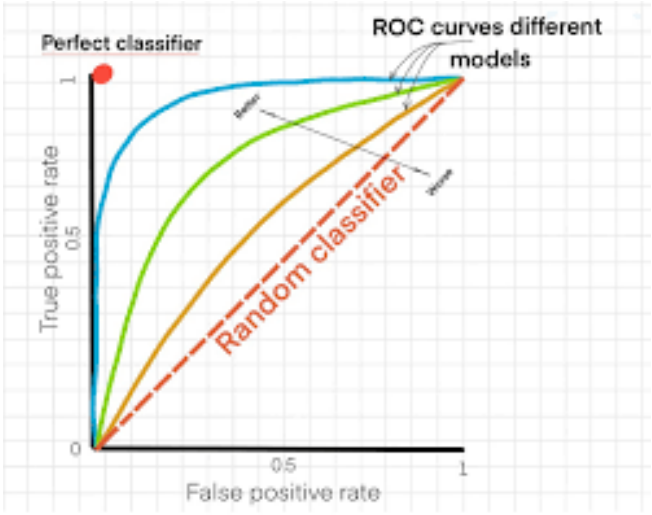


Figure 3.7.1: Illustration of ROC curves comparing the performance of multiple classification models. Source: [29]

PR-AUC (Precision-Recall Area Under Curve) [30] The PR-AUC quantifies the area under the curve that plots precision against recall across all possible classification thresholds. Unlike ROC-AUC, which considers both true positives and false positives, PR-AUC focuses specifically on the trade-off between precision and recall. This makes it a valuable metric for evaluating how well a model balances false positives and false negatives.

Figure 3.7.2 provides a visual illustration of a typical precision-recall curve. The shaded region represents the PR-AUC value.

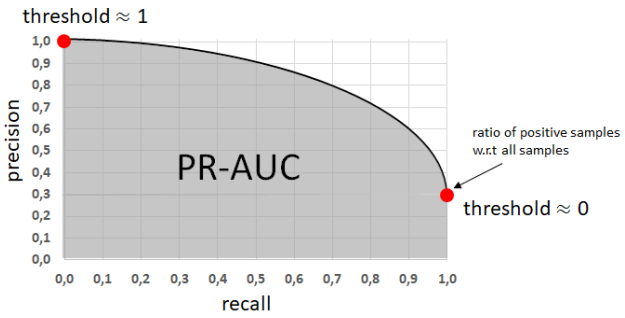


Figure 3.7.2: Illustration of a Precision-Recall curve and corresponding PR-AUC. Adapted from [31].

LogLoss (Logarithmic Loss) [32] Measures the negative log-likelihood of predicted probabilities. Penalizes overconfident and wrong predictions:

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{3.7.6}$$

Contribution Ratio (Domain-Specific) The *contribution ratio* quantifies a player's offensive involvement by measuring the proportion of meaningful actions that directly result in high-quality shots. This is computed per game and averaged across a season:

$$\begin{aligned} \text{Contribution Ratio}_{\text{game}} &= \frac{\#\text{Contributing Actions}}{\#\text{Total Actions}} \\ \text{Avg. Contribution Ratio}_{\text{player}} &= \frac{1}{T}\sum_{t=1}^{T}\text{Contribution Ratio}_t \end{aligned} \tag{3.7.7}$$

where T is the number of games played. An action is labeled as contributing if it occurs within a defined window before a high-quality shot and is not interrupted by a possession loss.

Disruption Ratio (Domain-Specific) The *Disruption Ratio* is a defensive metric specifically designed to quantify how often a player manages to interrupt the opponent's buildup play. It measures the share of a player's actions that effectively disrupt the opposing team's progression toward a quality shot or scoring opportunity:

$$\text{Disruption Ratio}_{\text{game}} = \frac{\#\text{Disrupting Actions}}{\#\text{Total Actions}} \tag{3.7.8}$$

$$\text{Avg. Disruption Ratio}_{\text{player}} = \frac{1}{T}\sum_{t=1}^{T}\text{Disruption Ratio}_t \tag{3.7.9}$$

In this formulation, T represents the number of matches analyzed, and $\text{Disruption Ratio}_t$ corresponds to the disruption ratio computed for match t. An action is considered disruptive if it breaks an ongoing possession phase from the opponent—particularly when that sequence shows signs of leading to a quality chance. Aggregating these events across matches makes it possible to identify players who consistently excel at neutralizing dangerous plays.

Chapter 4

Experiments

In this chapter, we discuss the implementation and operation of our main experiments. The introduction to this chapter describes the pipeline common to all the experiments. It should be noted that variations in the pipeline specific to each experiment are specified. The specific objectives, configurations and analyses of each experiment are then presented.

4.1 Training Procedure

The training pipeline adopted throughout this thesis follows a standardized and reproducible multistep process, ensuring consistency across experiments and enabling fair comparisons between models and preprocessing strategies.

- **Dataset Creation:** The datasets were generated by applying the labeling functions to the raw event data (see Experiment 1), followed by structured feature extraction (as described in Section 3.5). Each action was converted into a feature vector containing spatial, temporal, and contextual descriptors.
- **Train/Test Splits:** Two splitting strategies were used to prepare the data:
  - *Random Stratified Split:* Applied during the initial phases of Experiment 2 to maintain a balanced distribution of the binary labels across training and test sets, particularly important given the strong class imbalance (approximately 95% non-contributing labels).
  - *Chronological Championship Week Split:* Designed to simulate real-world forecasting scenarios. Matches were ordered by matchday, with the first 30 rounds used for training and the final 8 for testing. This approach prevents temporal leakage and mirrors the challenge of predicting future match events from past data.

  In both cases, the training and test sets were constructed without overlap. For the random split, 80% of the data was used for training and 20% for testing, ensuring stratification by class to preserve the original label proportions.
- **Cross-Validation and Hyperparameter Tuning:** Hyperparameters were optimized using `HalvingRandomSearchCV` [22], a resource-efficient successive halving method. The number of candidate configurations (`n_candidates`) was dynamically adjusted based on

the size of the search space for each model. All configurations were evaluated via 3-fold stratified cross-validation to ensure robustness and generalization.

- **Preprocessing Pipeline:**
  All preprocessing steps were implemented within `scikit_learn.pipeline.Pipeline` objects to prevent data leakage and streamline model deployment. A `Column_Transformer` was used to handle numerical and categorical features separately:

  – *Numerical features:* Imputed using either the mean or a KNN-based method (`KNN_Imputer`), then standardized with `Standard_Scaler`.

  – *Categorical features:* Imputed with the most frequent value and encoded using `One_Hot_Encoder`, with `handle_unknown='ignore'` to handle previously unseen categories during inference.

  This preprocessing routine was consistently applied across all experimental configurations and integrated directly into the full model pipeline.

- **Hyperparameter Grids for HalvingRandomSearchCV (Experiment 2):**
  A dedicated hyperparameter grid was defined for each model to guide the randomized search process (see Appendix A.6 for full details):

  – **Logistic Regression:** `penalty` (`l1`, `l2`), `C` (12 values from 0.0001 to 100), `solver` (`liblinear`, `saga`).

  – **Random Forest:** `n_estimators` (100, 300, 500), `max_depth` (None, 10, 20, 40), `min_samples_split` (2, 5, 10), `min_samples_leaf` (1, 2, 4), `max_features` (`sqrt`, `log2`, None).

  – **XGBoost:** `n_estimators`, `max_depth`, `learning_rate`, `subsample`, `colsample_bytree`, `reg_alpha`, `reg_lambda`, `min_child_weight`, `gamma`, `max_delta_step`.

  – **LightGBM:** `n_estimators`, `learning_rate`, `max_depth`, `num_leaves`, `min_data_in_leaf`, `subsample`, `colsample_bytree`, `reg_alpha`, `reg_lambda`.

  – **CatBoost:** `iterations`, `learning_rate`, `depth`, `l2_leaf_reg`, `border_count`, `bagging_temperature`.

  Each model was tested under various preprocessing settings (mean/median/KNN imputation) and both train/test split strategies, leading to the training of several hundred pipelines.

- **Model Evaluation:** Final model performance was assessed on the held-out test set using standard classification metrics as described in Section 3.7: Accuracy, Precision, Recall, F1-score, ROC-AUC, PR-AUC, and LogLoss. Additionally, SHAP-based interpretability analysis is planned as a post-hoc evaluation for the best-performing models.

In recent years, player valuation models in football have increasingly adopted continuous scoring frameworks to assess the impact of on-ball actions. One well-known approach is the VAEP (Valuing Actions by Estimating Probabilities) model, developed by Decroos et al. [14], which assigns value to each action based on the change it induces in the likelihood of scoring or conceding shortly thereafter. While this probabilistic methodology provides fine-grained

insights into player contributions, it also comes with challenges related to modeling complexity, data intensity, and limited interpretability for end users.

This study proposes a simpler, rule-based labeling system. Each event is labeled as either contributing or non-contributing to a high-quality shot. This binary approach emphasizes interpretability and transparency, making it well-suited for practical scouting and coaching use. Moreover, these interpretable labels serve as the foundation for predictive modeling in Experiments 2.

4.2 Experiment 1: Evaluation of Labeling Quality and Composite Score

The goal of this experiment is to test whether simple, interpretable labels can reflect real-life player performance. The hypothesis is that players who consistently contribute to quality shots (according to our labels) should also be recognized in the football world (e.g., salary, Ballon d'Or ranking). To evaluate this, we assess both the tactical validity of the binary labeling function and the usefulness of a composite score that integrates offensive and defensive impact metrics. The analysis addresses research questions RQ1 (validity of labels) and RQ4 (alignment of model outputs with real-world indicators).

4.2.1 Methodology and Setup

A total of 117 players from the 2017/2018 Premier League season were selected, with attention to positional diversity and impact profiles (see Table A.3.1). The analysis relies on the quality shot definition (Section 3.2), the contribution and disruption ratios (Section 3.7), and the composite score formula (Section 3.4).

4.2.2 Labeling Function and Parameters

As introduced in Section 3.3.3, the labeling function `flag_contributing_actions_advanced` assigns binary labels to actions occurring within a defined temporal and possession window preceding a quality shot. For the purposes of Experiment 1, we varied the following hyperparameters:

- `window` $\in \{3, 5, 7, 10\}$: maximum number of consecutive actions by the same team traced backward.
- `possession_threshold` $\in \{1, 2, 3\}$: maximum number of consecutive opponent actions tolerated before the contribution sequence is considered broken.

Semantic tags (e.g., `tag_101` for goals, `tag_201` for big chances) were used only to validate final shots, but deliberately excluded from the logic that identifies contributing actions. This decision ensures robustness and tactical interpretability, avoiding inconsistencies in Wyscout annotations.

4.2.3 Composite Score Distribution by Role

The distribution of composite scores across player roles reveals a structured and tactically plausible hierarchy. As shown in Figure 4.2.1, forwards exhibit the highest median scores (around 0.09), reflecting their dominant involvement in finishing actions and final-third sequences. Midfielders present a much wider interquartile range (approximately 0.04 to 0.075), due to their varied roles ranging from deep-lying disruptors to advanced creators. Defenders cluster at the lower end of the spectrum (mostly below 0.05), though outliers like Marcos Alonso indicate offensive tendencies among some full-backs. This stratification validates the

labeling scheme, as it captures expected role-specific contributions without introducing artificial skew.

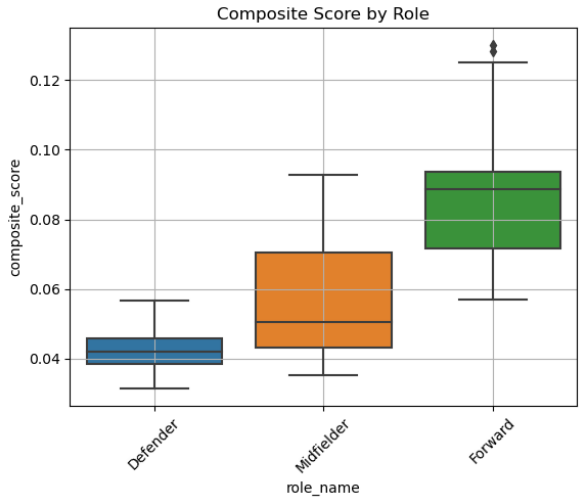


Figure 4.2.1: Distribution of the average composite score by position on the field. Forwards rank highest; defenders cluster at the low end.

4.2.4 Sub-role Patterns

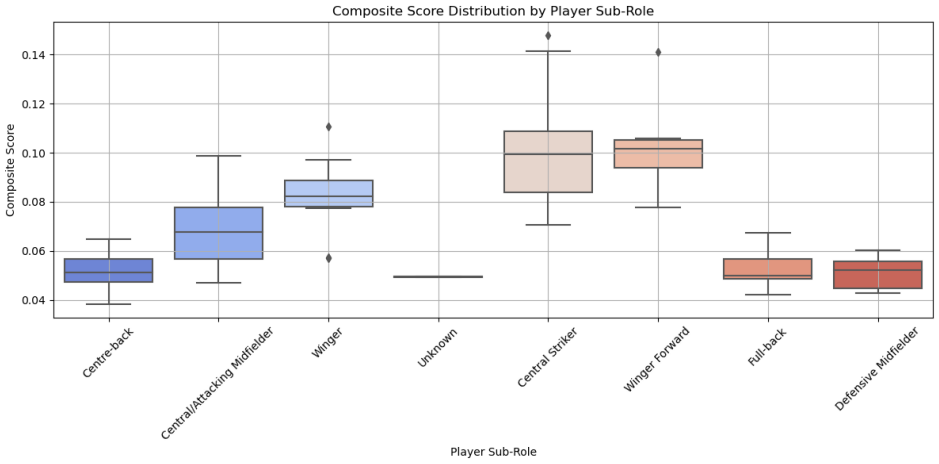


Figure 4.2.2: Composite score distribution by sub-role. Central strikers and winger-forwards dominate, with a sharp drop-off among defenders.

To further explore our initial findings, we conducted a more detailed analysis by examining player sub-roles. This level of granularity reveals additional nuances: winger-forwards and central strikers tend to achieve the highest composite scores, whereas players in deeper roles—such as full-backs and defensive midfielders—generally score lower. Notably, certain outliers, such as attacking full-backs, challenge conventional positional expectations and illustrate the importance of role-specific contributions.

Elite Midfielders Case Study This pattern of sub-role differentiation prompted a closer look at specific player groups. Midfielders, given their diverse tactical roles, offer an ideal case

study. Among central creators, De Bruyne and Özil top the rankings, consistent with their attacking contributions. Kanté ranks lower due to his predominantly defensive responsibilities. These observations illustrate the extent to which the score captures intra-role variation.

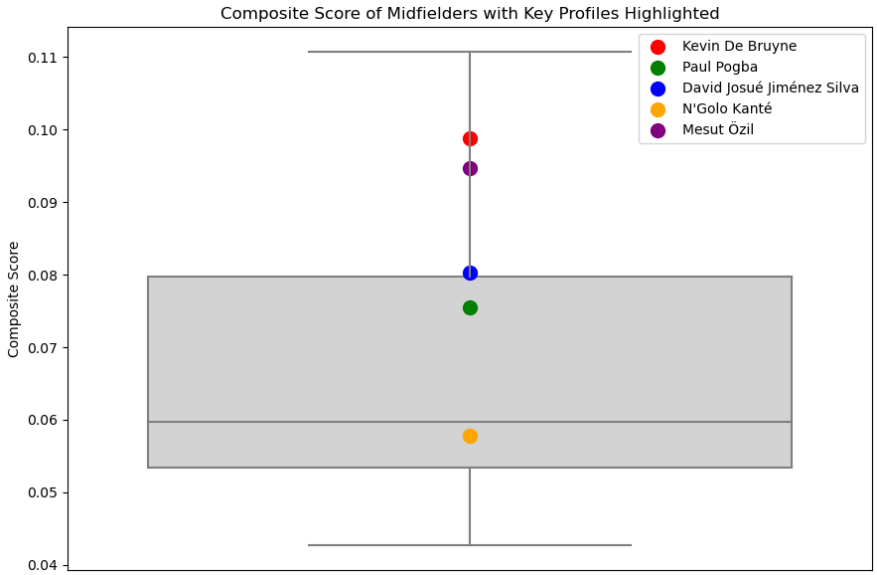


Figure 4.2.3: Composite score distribution of midfielders, with elite profiles highlighted. De Bruyne and Özil dominate among central creators.

4.2.5 Top Players and External Validation

The ranking presented in Table 4.2.1 provides a tangible illustration of how the composite score captures player influence. Leading figures such as Harry Kane, Sergio Agüero, and Mohamed Salah achieve the highest values, which aligns with their consistent offensive performances during the 2017/2018 season. The presence of both Kane and Salah among the Ballon d'Or top 10 further reinforces the relevance of the metric, suggesting that it is sensitive to performances recognized at the highest competitive level.

Further down the ranking, Eden Hazard emerges as a compelling case, combining a high composite score with an 8^{th} place finish in the Ballon d'Or and a substantial salary—illustrating convergence between internal and external performance signals. Conversely, the case of Alexis Sánchez reveals a mismatch: despite commanding one of the league's highest wages, his composite score remains relatively modest and is not accompanied by major accolades, highlighting potential inefficiencies in the relationship between remuneration and on-field impact.

Beyond these individual examples, the distribution of player roles within the top 15 underscores a structural trend. The list is heavily dominated by forwards, with only a single midfielder included. This concentration suggests that the composite score is particularly sensitive to actions occurring in advanced areas of the pitch, especially those directly contributing to chance creation and goal scoring.

This tendency is also reflected in Figure 4.2.4, where defenders appear clustered in the lower-left quadrant, indicative of more limited offensive involvement as captured by the metric. While the Spearman correlation between composite score and salary is moderate ($\rho = 0.39$), outliers such as Richarlison (high score, low salary) and Sánchez (low score, high salary) illustrate that monetary valuation does not always align with actual in-game contribution.

Player	Club	Role	Avg Comp Ratio	Salary (£)	Ballon d'Or
Harry Kane	Tottenham	Forward	0.1301	5.2M	10
Sergio Leonel Agüero del Castillo	Manchester City	Forward	0.1283	8.3M	–
Mohamed Salah Ghaly	Liverpool	Forward	0.1251	6.2M	6
Alexandre Lacazette	Arsenal	Forward	0.1023	9.5M	–
Romelu Lukaku Menama	Manchester United	Forward	0.0947	9.4M	–
Gabriel Fernando de Jesus	Manchester City	Forward	0.0941	3.9M	–
Álvaro Borja Morata Martín	Chelsea	Forward	0.0932	8.8M	–
Jamie Vardy	Leicester	Forward	0.0929	5.2M	–
Wilfried Zaha	Crystal Palace	Midfielder	0.0929	5.2M	–
Eden Hazard	Chelsea	Forward	0.0898	11.6M	8
Raheem Shaquille Sterling	Manchester City	Forward	0.0897	7.8M	–
Alexis Alejandro Sánchez Sánchez	Manchester United	Forward	0.0895	18.2M	–
Roberto Firmino Barbosa de Oliveira	Liverpool	Forward	0.0887	5.2M	–
Heung-Min Son	Tottenham	Forward	0.0851	4.4M	–
Sadio Mané	Liverpool	Forward	0.0843	5.2M	–

Table 4.2.1: Top 15 Players by Average Composite Ratio with External Validation

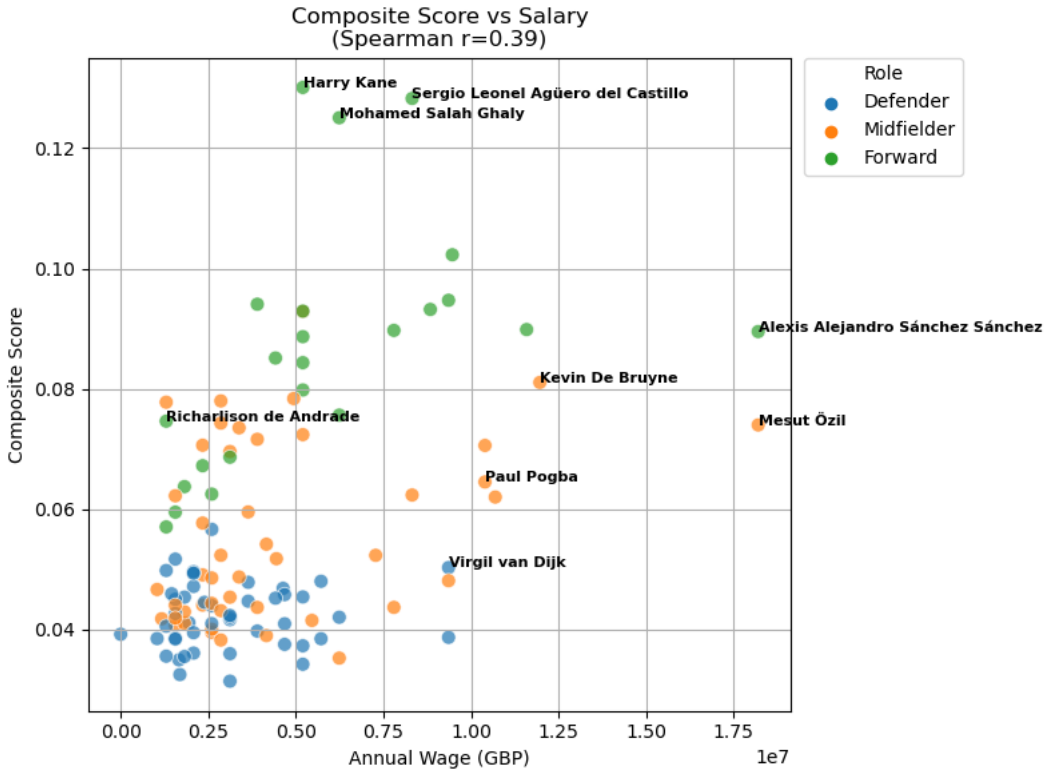


Figure 4.2.4: Composite score vs. salary with highlighted players. While Kane, Salah, and Agüero are high earners with strong performance, Richarlison appears underpaid relative to his impact. Conversely, Alexis Sánchez shows a low composite score despite a very high salary.

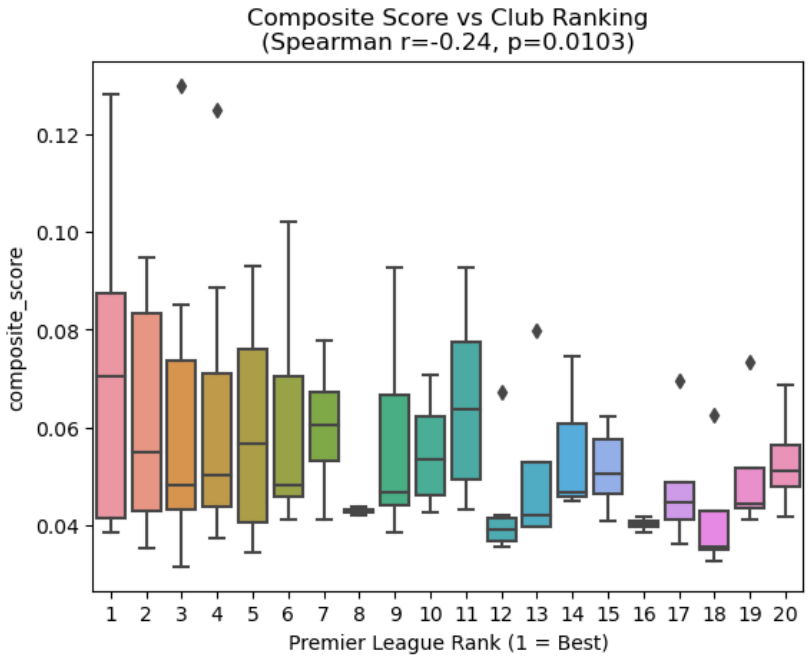


Figure 4.2.5: Composite score vs. club ranking (1 = best team). A moderate negative Spearman correlation ($\rho = -0.24$) indicates that higher-scoring players tend to be concentrated in stronger teams.

This interpretation is further reinforced by Figure 4.2.5, which displays a moderate yet statistically significant negative Spearman correlation ($\rho = -0.24$) between composite score and club ranking. In practical terms, this suggests that players with higher composite scores are more frequently associated with top-tier clubs—a result that aligns with the intuition that impactful players contribute to both offensive efficiency and defensive solidity.

Moreover, the accompanying boxplot reveals a clear downward trend in composite score distribution from rank 1 to rank 20, indicating that players from lower-ranked teams are generally less involved in high-value actions. This trend supports the idea that club performance context plays a role in shaping both the frequency and visibility of meaningful contributions.

Taken together, the evidence suggests that the composite score captures a quantifiable layer of player value, especially among attacking profiles. Its partial agreement with Ballon d'Or outcomes supports its validity, while the exceptions highlight areas where data-driven insights may challenge established narratives.

4.2.6 Sensitivity to Labeling Parameters

We assessed how the labeling output responds to variations in the parameters `window` (length of forward sequence) and `possession_threshold` (max opponent interruptions allowed).

Figure 4.2.6 shows that increasing `window` from 3 to 10 consistently raises the average contribution ratio (from ≈ 0.045 to 0.068), and the composite score accordingly. This confirms that longer windows capture more indirect contributions, especially for creators and build-up players.

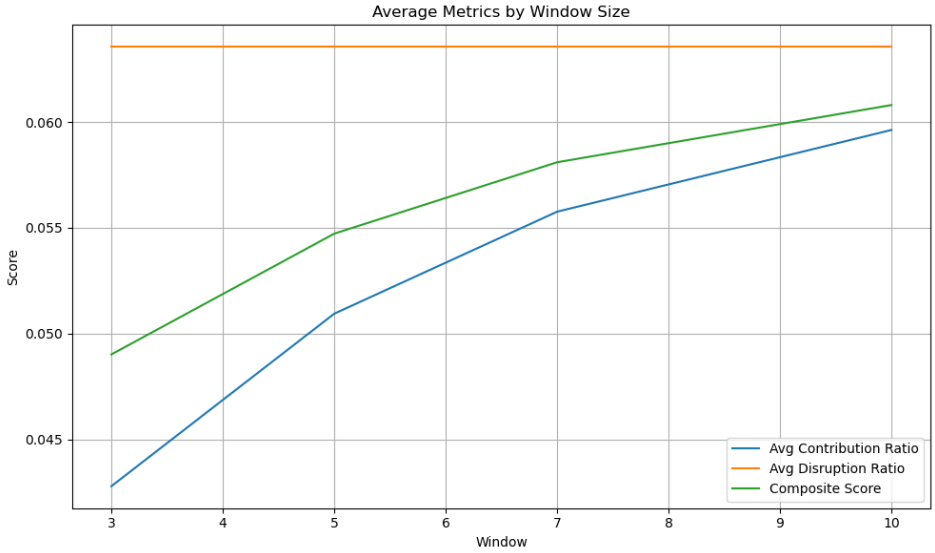


Figure 4.2.6: Evolution of average contribution ratio and composite score across window sizes. Longer sequences capture more offensive involvement.

The specific behavior of the disruption component is discussed in Section 4.2.6, where we analyze its sensitivity to both tactical context and labeling parameters.

Beyond average values, the distribution of composite scores shown in Figure 4.2.7 reveals a steady upward shift in median scores as the window size increases. From $W = 5$ onward, the emergence of high-scoring outliers suggests that longer action sequences begin to capture more intricate attacking patterns—often involving elite playmakers or forwards with sustained influence. The relative stability of the interquartile range across window sizes indicates that this effect is concentrated among a specific group of players, rather than resulting in a uniform increase across the board.

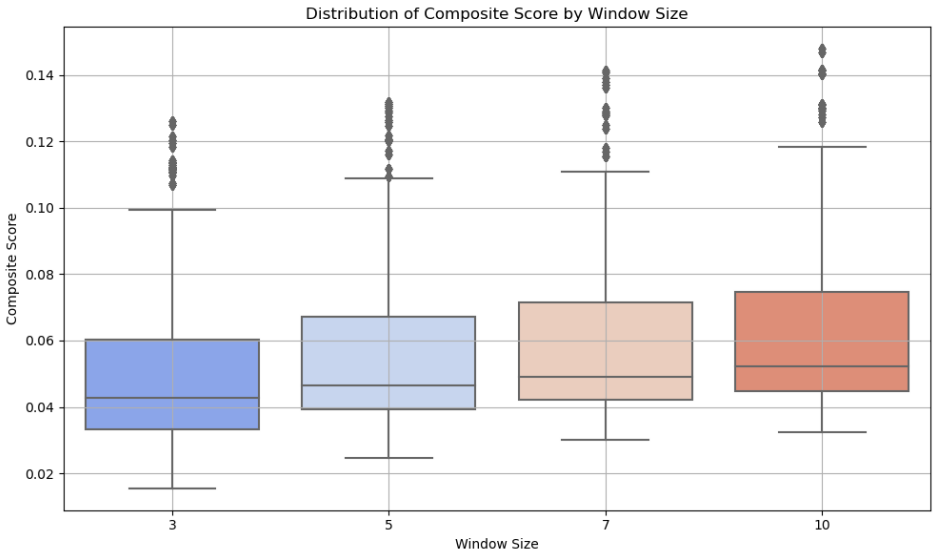


Figure 4.2.7: Distribution of composite scores across window sizes. Higher windows increase the median score and reveal high-value outliers.

To visualize the interaction between both parameters, Figure 4.2.8 presents a heatmap of average composite scores across all (window, possession) pairs. The best empirical configuration is $(10, 3)$, yielding a mean composite score of 0.0637. However, we retain $(7, 2)$ as our default setting due to its interpretability, corresponding to typical offensive sequences of 2–3 passes before a shot.

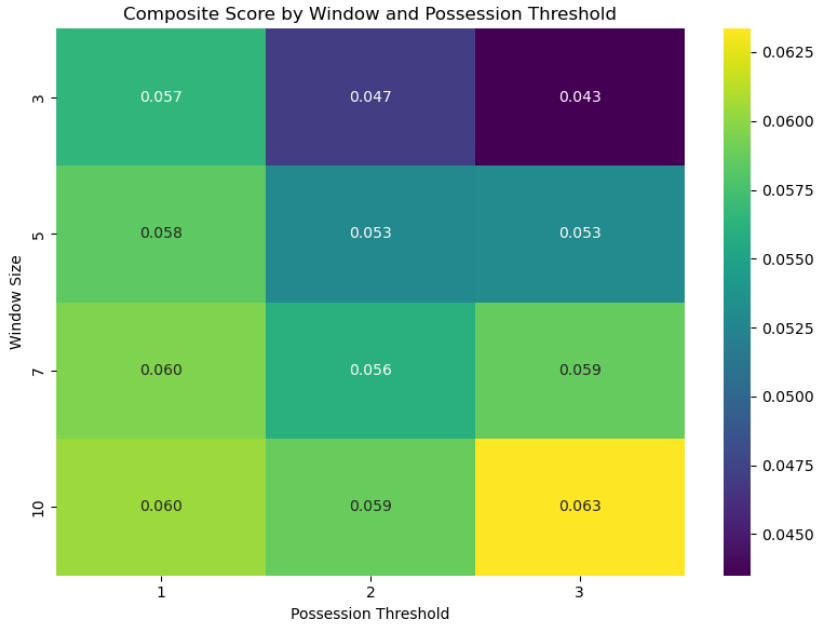


Figure 4.2.8: Heatmap of composite score across combinations of window size and possession threshold. Best empirical performance at $(10, 3)$.

Role-Specific Sensitivity Building on the static sub-role analysis, we now explore how labeling parameters—especially `window`—interact with different player profiles. To examine how the labeling setup affects different player types, we plot the evolution of composite scores across window sizes for five contrasting profiles: Harry Kane, Mohamed Salah, Kevin De Bruyne, Eden Hazard, and Marcos Alonso (Figure 4.2.9).

Kane and Salah plateau early—Kane peaks at $W = 5$, Salah around $W = 7$—suggesting that their contributions are immediate and close to the goal. De Bruyne and Hazard continue improving up to $W = 10$, reflecting involvement in longer attacking sequences. Marcos Alonso's curve remains flat and low across all windows, as expected for a defender.

These differences confirm that the window parameter functions as a tactical lens, bringing to light distinct behavioral patterns associated with different positional roles.

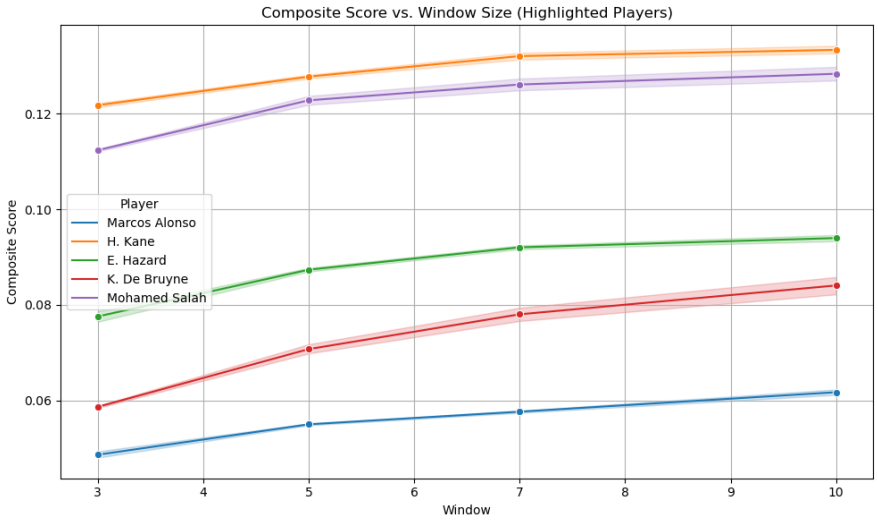


Figure 4.2.9: Composite score across window sizes for key players. Forwards plateau early; creators benefit from longer sequences.

A complementary focus on centre-backs (Figure 4.2.10) shows how defensive roles vary under the same labeling rules. Marcos Alonso's consistently higher scores reflect his attacking involvement as a wing-back, with frequent forward contributions inflating his activity profile. In contrast, Virgil van Dijk maintains lower values across window sizes—an outcome not of lower performance, but of a style centered on positioning, anticipation, and minimal intervention. This highlights a key limitation of event-based data: it may fail to capture the defensive impact of players whose quality lies in preventing actions rather than directly contesting them.

Harry Maguire displays an intermediate trajectory, characteristic of a high-intervention defensive profile marked by frequent involvement in both aerial and ground duels. His responsiveness to changes in window size suggests that his defensive actions are more evenly distributed across different phases of play. The variation observed among defenders highlights how individual playing styles and tactical responsibilities influence the outputs of the labeling framework—where elevated activity levels may reflect either assertive defending or offensive license, while lower scores may conceal forms of elite defensive control that are less disruptive by nature.

Although these patterns offer valuable insights, they also point to a limitation of disruption-based metrics in fully capturing the spectrum of defensive excellence. This issue is examined more closely in the following section, which addresses structural biases inherent in event-based annotation systems.

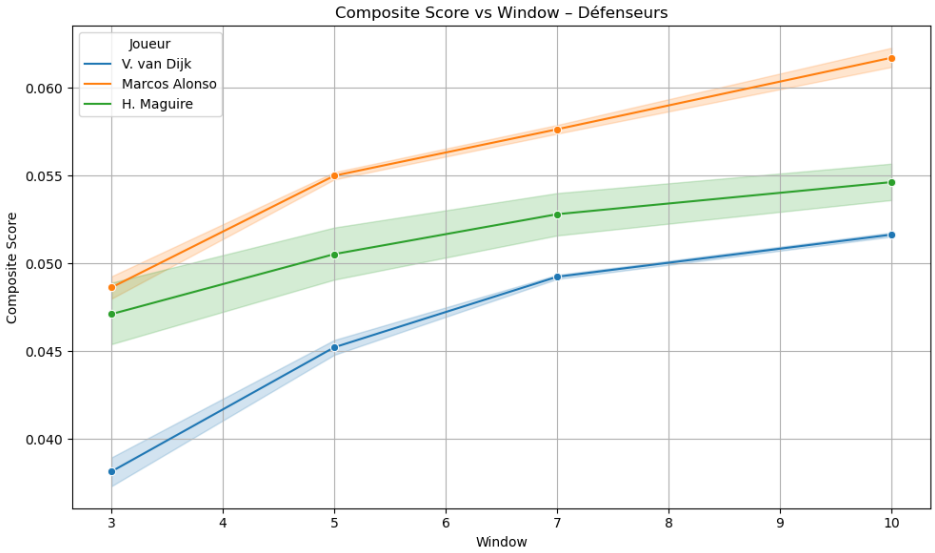


Figure 4.2.10: Composite score by window size for defenders. Alonso benefits from offensive sequences; Van Dijk and Maguire remain lower.

Limits of the Disrupt Ratio and Defensive Style Bias Disruption scores are context-dependent: defenders in dominant teams (e.g., Van Dijk, Otamendi) show lower disrupt ratios—not due to lack of quality, but due to limited defensive opportunity.

Figure 4.2.11 presents boxplots of disruption ratios grouped by club ranking quartiles. Players from top-five clubs systematically exhibit the lowest disruption values, whereas those from lower-ranked teams show average ratios exceeding 0.08.

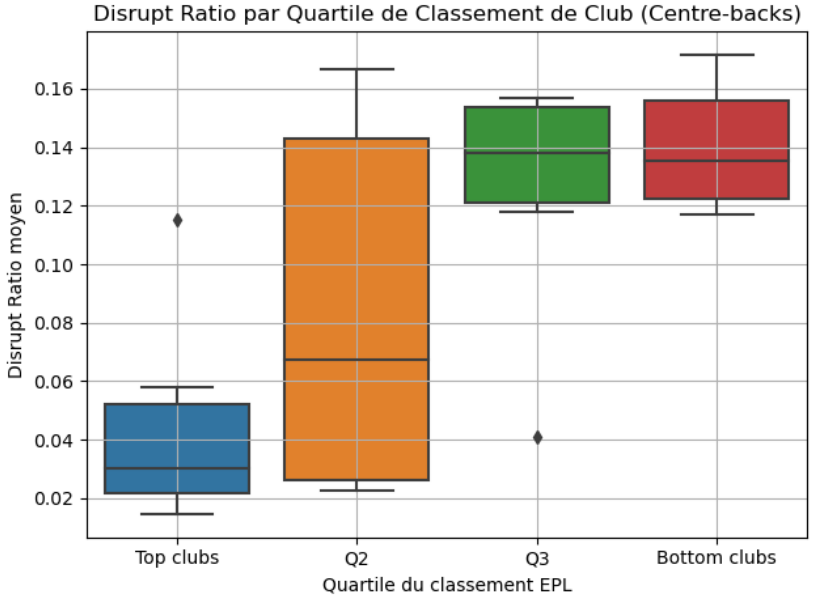


Figure 4.2.11: Disrupt ratio across club ranking quartiles. Defenders from lower-ranked teams face more defensive sequences, increasing disruption potential.

This confirms that disruption-based metrics should be interpreted relative to team context and tactical setup.

Moreover, the structural bias observed in disruption metrics is not only contextual but also influenced by parameter choices. Specifically, the disrupt ratio is highly sensitive to the labeling threshold, which amplifies variability across player profiles. As illustrated in Figure 4.2.12, increasing the possession threshold leads to a noticeable decline in average disrupt ratios—from 0.058 at $p = 1$ to below 0.03 at $p = 3$. This pattern suggests that loosening the definition of possession continuity diminishes the model's capacity to identify defensive interventions, particularly in high-press contexts. In contrast, a stricter setting (e.g., $p = 1$) captures a greater number of meaningful disruptions.

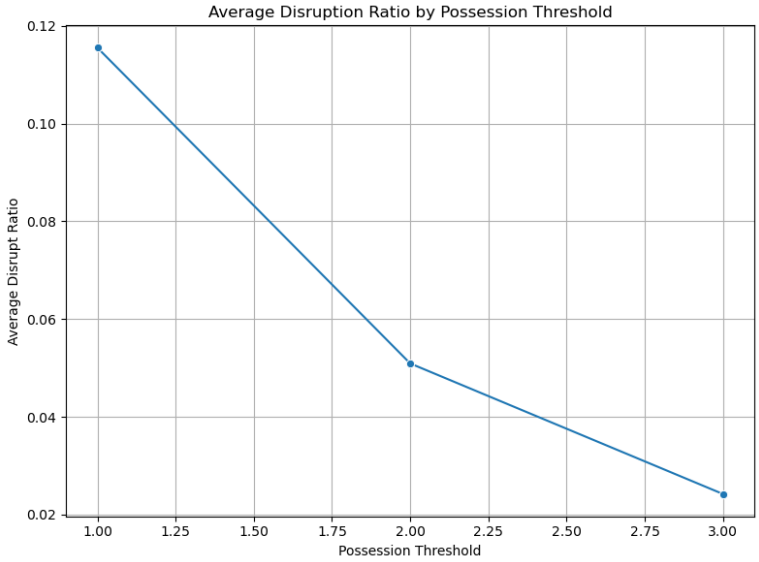


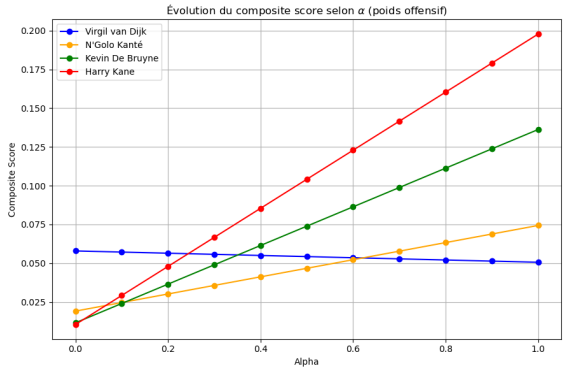
Figure 4.2.12: Average disrupt ratio across possession thresholds. A strict definition ($p = 1$) detects more defensive disruptions.

While the disrupt ratio provides useful information about high-intervention defensive profiles, it presents notable blind spots. Figure B.4.1 (Appendix B.4) ranks the top 10 centrebacks by disrupt ratio. All of them—such as Ahmed Hegazy, Michael Keane, and Shane Duffy—belong to lower-ranked clubs frequently exposed to defensive pressure.

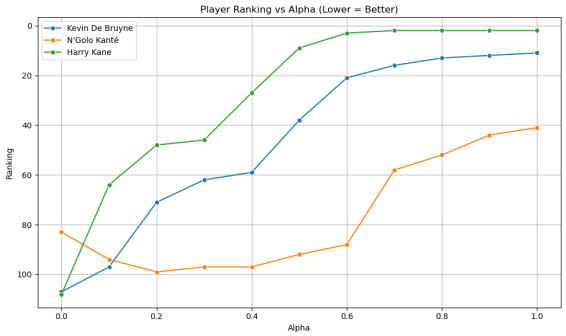
Notably, Virgil van Dijk is absent from this list despite his elite status. This omission illustrates a structural limitation of the Wyscout dataset: it captures physical defensive actions (tackles, interceptions, duels) but fails to account for non-ball behaviors like positional anticipation, shadowing, or prevention of dangerous sequences.

As a result, defenders excelling through strategic positioning rather than frequent interventions like Van Dijk tend to be undervalued. This reinforces the need to interpret `disrupt_ratio` within a contextual framework that accounts for team dominance and defensive playing style.

Strategic Weighting via α To assess how the composite score reflects different player profiles, we test its sensitivity to the weighting parameter $\alpha \in [0, 1]$. Figure 4.2.13 shows the absolute score progression and ranking evolution of four representative players.



(a) Composite score evolution. Kane and De Bruyne benefit from higher α; Van Dijk and Kanté favor low α.



(b) Ranking evolution. Kane rises with offensive weighting; Van Dijk drops. Kanté remains stable mid-ranking.

Figure 4.2.13: Impact of α on absolute composite scores and player rankings for four selected individuals. Varying α shifts their valuation, illustrating how different roles and tactical profiles influence scoring dynamics.

Together, these figures illustrate that α functions as more than a mere technical parameter; it acts as a strategic lever. Adjusting its value shifts the emphasis of the evaluation from disruption (at lower α) to offensive contribution (at higher α), thereby allowing the composite score to be tailored to specific tactical objectives or scouting requirements.

Figure 4.2.14 isolates Van Dijk's sensitivity to the α parameter. As α increases, his score decreases linearly, confirming his defensive-heavy profile. This validates the idea that weighting schemes allow evaluators to tailor the composite metric to the tactical focus—offensive or defensive—of a scouting context.

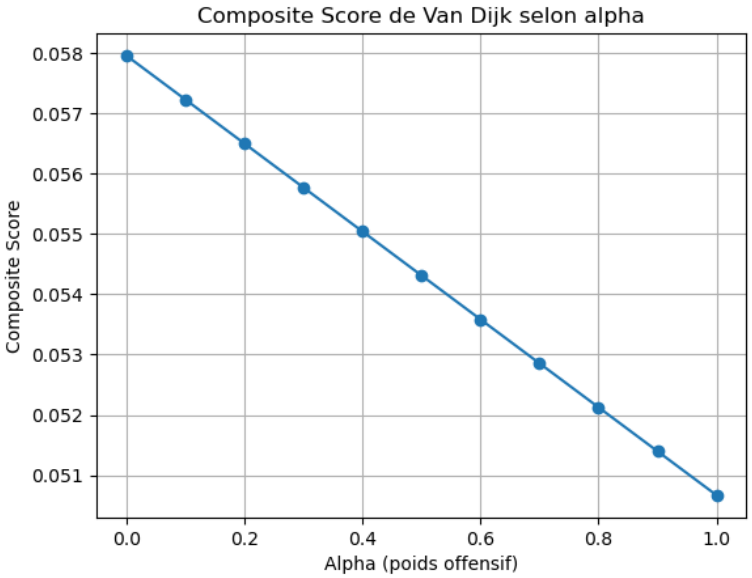


Figure 4.2.14: Composite score of Van Dijk across α values. His score decreases as offensive weighting increases.

Although Van Dijk's composite score remains modest across all window sizes, it shows a

slight upward trend as window increases (see Figure B.3.1 in Appendix B.3). This suggests that even highly defensive players can be marginally impacted by temporal context, but the increase is much flatter than for creators or attacking roles.

Conclusion Experiment 1 confirms that:

- **RQ1**: The labeling system yields tactically meaningful, explainable, and robust metrics.
- **RQ4**: The composite score demonstrates strong alignment with external performance indicators, providing objective differentiation particularly for players with hybrid roles or those who are traditionally undervalued.

In summary, the combination of internal consistency (by role and player type), external alignment (with salary and awards), and parameter sensitivity analysis supports the validity of our labeling framework. While offensive profiles are more easily captured, the score remains sufficiently nuanced to distinguish creators, disruptors, and hybrids. The moderate correlations observed—especially with salary and club ranking—suggest that the composite score captures key dimensions of player value, while also exposing role-based asymmetries and tactical biases present in event data. These insights motivate its use as a learning target in subsequent predictive experiments.

The following chapter builds on these insights by evaluating how well predictive models can learn to infer these contribution labels from contextual, temporal, and spatial features.

4.3 Experiment 2: Predicting Contribution Labels from Event Data

This experiment builds upon the standardized training pipeline described in Section 4.1, evaluating five supervised classifiers across multiple preprocessing strategies and data split configurations. The objective is to assess the feasibility of automatically predicting whether an action contributes to a quality shot, using spatial, temporal, and contextual features extracted from raw football event data.

We frame this as a binary classification task, where the target label corresponds to the contribution flag introduced in Experiment 1. The analysis focuses on three core dimensions: (i) the comparative performance of machine learning estimators (RQ2), (ii) the impact of preprocessing choices such as imputation and train/test splitting strategy (RQ3), and (iii) a deeper interpretability analysis of prediction errors, probability distributions, and model behavior.

Performance is assessed using standard evaluation metrics including PR-AUC, ROC-AUC, and F1-score, with special emphasis placed on stability across different random seeds, computational efficiency, and consistency with football domain semantics. The primary objective of this experiment is to determine the most appropriate model architecture for downstream tasks such as player ranking, scouting, and tactical recommendation.

4.3.1 Global Model Comparison

Model performance and PR-AUC comparison across seeds. In highly imbalanced classification settings such as ours, it is crucial to evaluate both the average model performance and its variability across different seeds. Figure C.3.1 displays the distribution of PR-AUC

scores for all model configurations. These distributions appear visually consistent. This indicates that model performance remains stable under random reinitialization, a desirable property in real-world pipelines.

Turning to average performance, a Kruskal-Wallis test revealed significant differences in PR-AUC across classifiers ($H = 39.27$, $p < 0.001$). Dunn's post-hoc test with Holm correction (Table 4.3.1) confirms that LightGBM and XGBoost significantly outperform Logistic Regression ($p < 0.001$). LightGBM also scores higher than CatBoost ($p = 0.024$) and Random Forest ($p = 0.0108$). These results, visualized in Figure 4.3.1, support the hypothesis that boosting-based models are better suited for learning rare, high-impact football actions.

Model	**CatBoost**	**LightGBM**	**LogReg**	**RF**	**XGBoost**
CatBoost	1.0000	0.9785	0.0001	0.1239	0.9785
LightGBM	0.9785	1.0000	0.0000	0.0108	0.9785
Logistic Regression	0.0001	0.0000	1.0000	0.1239	0.0000
Random Forest	0.1239	0.0108	0.1239	1.0000	0.1239
XGBoost	0.9785	0.9785	0.0000	0.1239	1.0000

Table 4.3.1: Adjusted p-values from Dunn's test (Holm correction) comparing PR-AUC across classifiers. Values below 0.05 indicate significant differences.

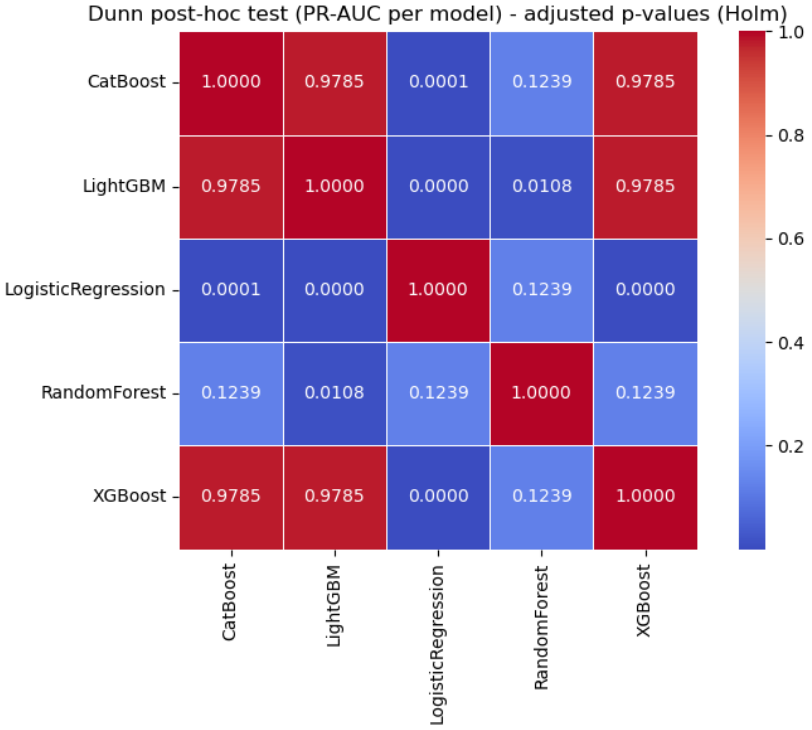


Figure 4.3.1: Heatmap of pairwise Dunn test results on PR-AUC scores. Dark blue cells ($p < 0.05$) highlight statistically significant differences.

Mean performance across configurations. To complement the previous analysis, Table 4.3.2 reports average `PR-AUC` and `ROC-AUC` scores (with standard deviations) for all model and preprocessing combinations. The best configuration `LightGBM_knn_random` achieves a

PR-AUC of \$0.5013 and ROC-AUC of \$0.8395, outperforming all alternatives. Other boosting models (XGBoost, CatBoost) also perform well, while Logistic Regression lags behind.

Model Configuration	**PR-AUC (mean ± std)**	**ROC-AUC (mean ± std)**
CatBoost_knn_chrono	0.4769 ± 0.0003	0.8229 ± 0.0001
CatBoost_knn_random	0.4839 ± 0.0017	0.8256 ± 0.0022
CatBoost_mean_chrono	0.4768 ± 0.0011	0.8227 ± 0.0002
CatBoost_mean_random	0.4848 ± 0.0013	0.8270 ± 0.0016
LightGBM_knn_chrono	0.4767 ± 0.0022	0.8223 ± 0.0020
LightGBM_knn_random	**0.5013 ± 0.0007**	**0.8395 ± 0.0010**
LightGBM_mean_chrono	0.4777 ± 0.0064	0.8214 ± 0.0070
LightGBM_mean_random	0.5006 ± 0.0014	0.8397 ± 0.0014
LogReg_knn_chrono	0.3064 ± 0.0000	0.7835 ± 0.0000
LogReg_knn_random	0.3189 ± 0.0000	0.7867 ± 0.0000
LogReg_mean_chrono	0.3064 ± 0.0000	0.7835 ± 0.0000
LogReg_mean_random	0.3189 ± 0.0000	0.7867 ± 0.0000
RandomForest_knn_chrono	0.4668 ± 0.0020	0.8039 ± 0.0019
RandomForest_knn_random	0.4744 ± 0.0080	0.8155 ± 0.0059
RandomForest_mean_chrono	0.4668 ± 0.0020	0.8039 ± 0.0019
RandomForest_mean_random	0.4744 ± 0.0080	0.8156 ± 0.0060
XGBoost_knn_chrono	0.4755 ± 0.0056	0.8218 ± 0.0075
XGBoost_knn_random	0.4884 ± 0.0049	0.8330 ± 0.0025
XGBoost_mean_chrono	0.4732 ± 0.0062	0.8189 ± 0.0079
XGBoost_mean_random	0.4922 ± 0.0078	0.8351 ± 0.0040

Table 4.3.2: Average PR-AUC and ROC-AUC scores (± standard deviation) across three seeds. Boosting-based models like LightGBM and XGBoost dominate both metrics.

Robustness to random seeds. Table 4.3.3 reports the standard deviation of performance metrics for each model across multiple random seeds. Random Forest and CatBoost demonstrate the highest stability in terms of F1-score and ROC-AUC. Although LightGBM and XGBoost exhibit slightly greater variability in PR-AUC, they consistently deliver strong average results. Logistic Regression, while showing stable behavior, underperforms relative to the other models.

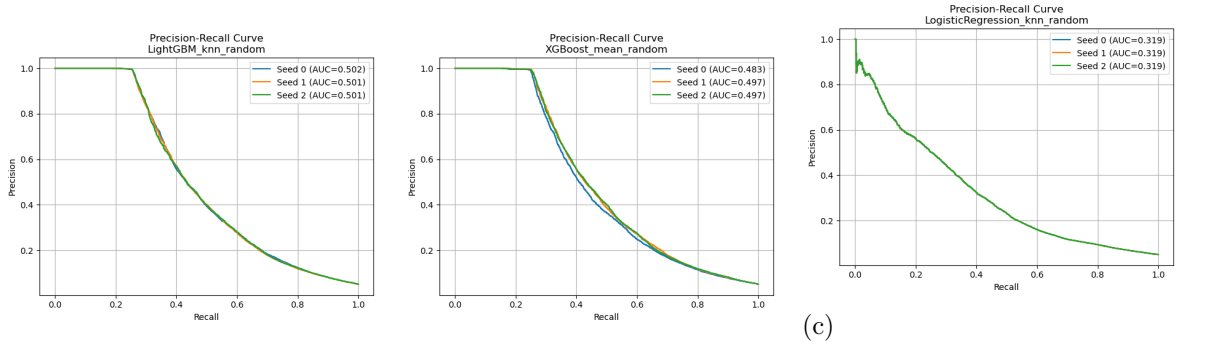
Model	**F1 (std)**	**PR-AUC (std)**	**ROC-AUC (std)**
CatBoost	0.0021	0.0040	0.0022
LightGBM	0.0035	0.0127	0.0098
Logistic Regression	0.0041	0.0065	0.0016
Random Forest	0.0014	0.0063	0.0072
XGBoost	0.0053	0.0100	0.0088

Table 4.3.3: Standard deviation of key performance metrics across seeds. CatBoost and Random Forest show the most stable behavior.

Taken together, these results reinforce the superiority of boosting-based models in terms of both predictive accuracy and stability when identifying rare, high-impact football actions. LightGBM, in particular, offers an effective trade-off between strong performance and compu-

tational efficiency, positioning it as an excellent candidate for practical deployment in football analytics applications.

Precision-Recall curves grouped by seed. To complement the statistical findings, we plotted the precision-recall (PR) curves for representative models trained on random splits and averaged across three seeds. As shown in Figure 4.3.2, LightGBM and XGBoost achieve superior performance across nearly all probability thresholds, with low variance between seeds. Logistic Regression, while stable, consistently underperforms across the entire recall spectrum—highlighting its limited discriminative capacity in this highly imbalanced setting.



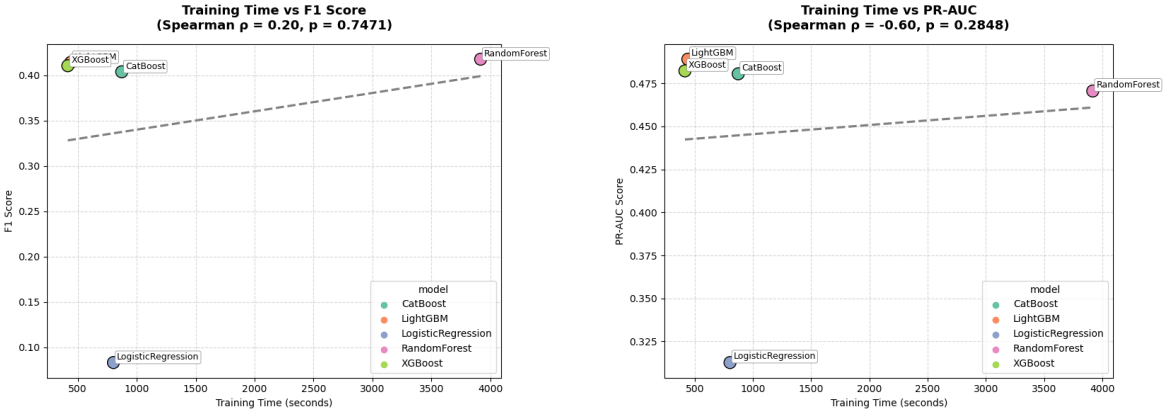
(a) `LightGBM_knn_random` (b) `XGBoost_mean_random` (c) `LogisticRegression_knn_random`

Figure 4.3.2: Grouped Precision-Recall curves for three models under random split. LightGBM and XGBoost consistently outperform Logistic Regression across seeds. See C.4 for the other models

Training time versus predictive value. Beyond accuracy and stability, computational cost is a critical consideration in applied settings such as scouting platforms or match preparation workflows. Figure 4.3.3 presents the relationship between training time and two key metrics: F1 score (often used for monitoring during model selection) and PR-AUC (preferred in imbalanced classification tasks).

Spearman rank correlations reveal no statistically significant association between training duration and performance metrics: $\rho = 0.20$ ($p = 0.7471$) for the F1 score, and $\rho = -0.60$ ($p = 0.2848$) for PR-AUC. This indicates that longer training times do not necessarily translate into improved model quality.

Notably, Random Forest exhibits relatively high computational demands despite offering only moderate gains in predictive accuracy, whereas Logistic Regression is extremely efficient but falls short in performance. On the other hand, LightGBM and CatBoost strike a favorable balance by combining strong predictive capabilities with reasonable training times. These characteristics make them well-suited for deployment in real-time or frequently updated machine learning workflows.



(a) Training Time vs F1 Score

(b) Training Time vs PR-AUC

Figure 4.3.3: Trade-off between training time and performance metrics. Boosting models maintain high scores with moderate computation time, unlike Random Forest (slow) or Logistic Regression (weak).

Conclusion. The findings of this section converge toward a clear conclusion: gradient boosting models, particularly LightGBM and XGBoost, offer the best combination of predictive accuracy, stability across seeds, and computational efficiency. Their robustness under repeated training, superior PR-AUC and ROC-AUC scores, and consistent behavior across probability thresholds make them ideal for deployment in football analytics contexts.

In contrast, Logistic Regression—although fast and consistent—struggles to model the complexity inherent in high-impact event-level contributions. While Random Forest and CatBoost represent viable alternatives, they do not match the combination of discriminative strength and computational efficiency offered by other methods.

Therefore, **LightGBM** stands out as the most reliable and cost-effective model for predicting contribution labels in this dataset. Its consistent advantage justifies its selection in downstream applications, including scouting, tactical analysis, and player ranking pipelines. All the following results in this thesis will therefore be obtained from this model.

4.3.2 Comparison of Train/Test Split Strategy

Rationale and design. Model evaluation protocols must reflect the deployment context. In sports analytics, the objective is often to forecast future match events based on past data. To simulate this setting, we compare two split strategies:

- **Random stratified split** ensures class balance and is commonly used for generic ML benchmarking.
- **Chronological split** respects temporal ordering by training on matchdays 1–30 and testing on the final 8 matchdays.

The latter mimics real-world usage, where a model trained on earlier games is deployed to predict outcomes in later stages of the season. Our aim is to determine whether the simpler random split leads to overestimated performance due to temporal leakage.

Statistical comparison of split types. Before applying statistical tests, we verified whether the PR-AUC and ROC-AUC distributions per split followed normality assumptions. Histograms and Q-Q plots (see Appendix C.5.1) revealed clear deviations from normality—confirmed by Shapiro-Wilk tests (Table 4.3.4). For all four cases (PR-AUC and ROC-AUC under both splits), $p < 0.001$, justifying the use of non-parametric testing.

Metric (Split)	W-statistic	p-value
PR-AUC (Chrono)	0.555	< 0.0001
PR-AUC (Random)	0.613	< 0.0001
ROC-AUC (Chrono)	0.821	0.0002
ROC-AUC (Random)	0.827	0.0002

Table 4.3.4: Shapiro-Wilk test for normality. All distributions deviate significantly from normality ($p < 0.05$).

We then applied the Mann–Whitney U test to assess performance differences. Results in Table 4.3.5 show statistically significant performance drops under chronological splits for both PR-AUC ($p = 0.0004$) and ROC-AUC ($p = 0.0032$). These differences are also visible in the boxplots of Figure 4.3.4.

Metric	Chrono Median [IQR]	Random Median [IQR]	Mann–Whitney U (p-value)
PR-AUC	0.4715 [0.4653–0.4773]	0.4832 [0.4683–0.4957]	211.0 ($p = 0.0004$)
ROC-AUC	0.8164 [0.8026–0.8230]	0.8265 [0.8111–0.8367]	250.0 ($p = 0.0032$)

Table 4.3.5: Comparison of PR-AUC and ROC-AUC distributions (sample size: $n = 60$ models per split). Median scores are consistently lower under chronological splits.

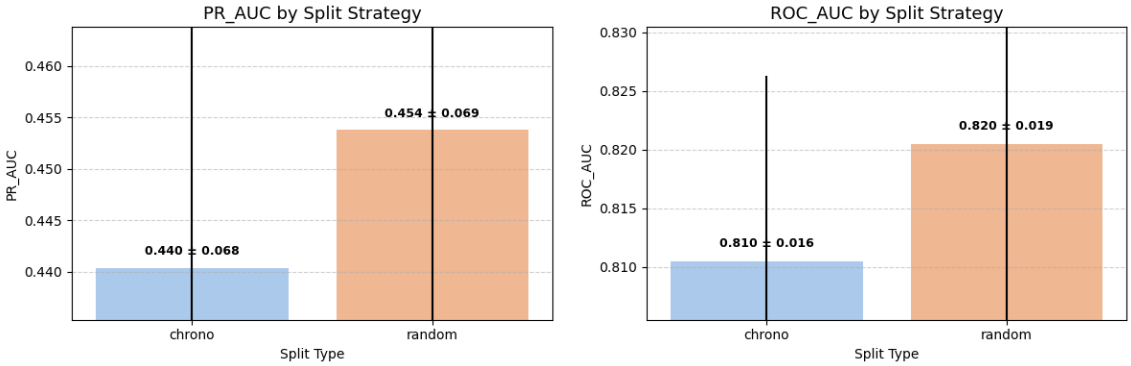


Figure 4.3.4: Distribution of PR-AUC and ROC-AUC scores by split strategy. Despite substantial overlap, central tendencies indicate a small but statistically significant difference.

Interpretation and implications. These results offer valuable guidance for practitioners. Although random stratified splits are straightforward to implement and often yield higher performance metrics, they tend to introduce an optimistic bias by allowing models to leverage patterns unlikely to recur in future matches.

Conversely, chronological splits impose a more stringent and realistic evaluation framework. The observed decrease in performance—particularly noticeable in PR-AUC—highlights the inherent difficulty of generalizing within a dynamic context characterized by evolving line-

ups, tactical variations, and differing opponents. Crucially, this does not imply that models are inherently "worse," but rather that their effectiveness may be overestimated when evaluated using random splits.

Conclusion. The analysis reinforces the hypothesis that temporally consistent evaluation is crucial for forecasting tasks in sports analytics. Although random stratified splits are simpler to implement and tend to produce slightly higher median scores, they risk introducing optimistic bias by mixing information across different time periods. In contrast, chronological splits—despite overlapping score distributions—offer a more faithful representation of real-world deployment scenarios, where models must generalize to unseen future data.

While the observed differences between splitting strategies may appear modest visually, they are statistically significant according to Mann–Whitney U tests. This underscores the necessity of pairing visual examination with rigorous statistical testing when assessing model robustness. Consequently, we advocate for the use of temporally aware evaluation protocols in any modeling framework intended to inform real-world decisions in scouting, training, or tactical planning.

4.3.3 Comparison of Imputation Strategies

Motivation. Missing values are common in football event data due to incomplete tracking or inconsistent annotation. Our pipeline addresses this issue through two widely used strategies:

- **Mean imputation**, which replaces missing numerical values with the average across the training set;
- **K-Nearest Neighbors (KNN)** imputation, which fills gaps using the average of similar samples.

The objective of this section is to assess whether the choice of imputation method significantly influences model performance, particularly with respect to PR-AUC and ROC-AUC metrics.

Statistical analysis. We began by examining the distribution of scores produced by each imputation strategy to determine adherence to normality assumptions. Visual assessments using histograms and Q-Q plots indicated noticeable deviations, which were confirmed by Shapiro-Wilk tests yielding p-values below 0.05. These results justified the application of non-parametric statistical methods.

Prior to comparing imputation methods, we evaluated the extent of missing data within the dataset. Out of the 90 features in the final preprocessed dataset (`data_exp2`), only `prev_x` and `prev_y` contained missing entries—each with precisely 300 missing values among 642,820 rows (approximately 0.0591%). The total number of missing cells across all features was minimal, with $1,140$ missing values out of $74,605,400$ cells, resulting in a global missingness rate of 0.0015% (see Appendix A.10.1). This negligible level of missingness suggests it is unlikely to bias model training or evaluation.

Subsequently, the Mann–Whitney U test was employed to compare PR-AUC and ROC-AUC scores between the two imputation approaches. As presented in Table 4.3.6, no statistically significant differences were found ($p = 1.000$ for PR-AUC and $p = 0.9352$ for ROC-AUC). These findings are further supported by the boxplots depicted in Figure 4.3.5, which show nearly identical distributions for both methods.

Metric	Mean Imputer (Mean ± Std)	KNN Imputer (Mean ± Std)	Mann–Whitney U (p-value)
PR-AUC	0.4472 ± 0.0692	0.4469 ± 0.0690	450.0 ($p = 1.000$)
ROC-AUC	0.8154 ± 0.0185	0.8155 ± 0.0182	444.0 ($p = 0.9352$)

Table 4.3.6: Comparison of predictive scores between mean and KNN imputation. No significant difference was observed.

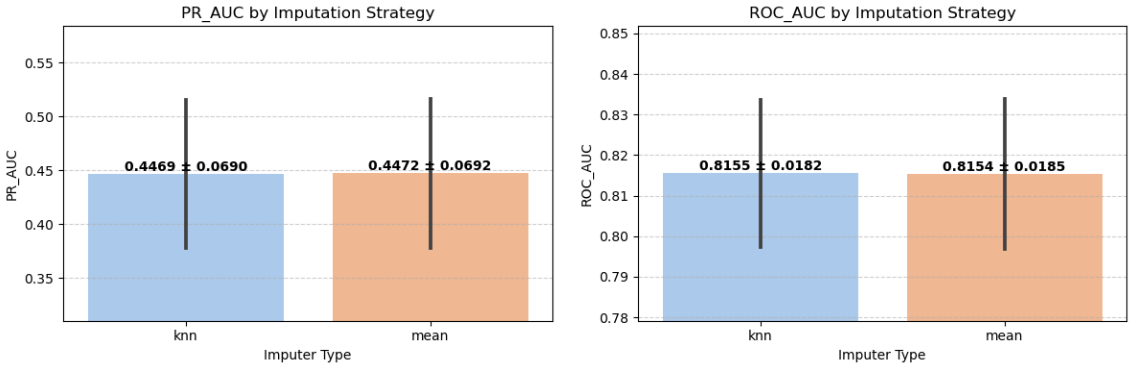


Figure 4.3.5: Boxplots of PR-AUC and ROC-AUC by imputation strategy. The similarity of distributions suggests negligible impact of the imputer on performance.

Distributions of model performance. To further substantiate this conclusion, we examined the distributions of PR-AUC and ROC-AUC scores under each imputation strategy. Figures 4.3.6 and 4.3.7 reveal similar distributional shapes, with comparable measures of central tendency and variability. The considerable overlap observed in these distributions reinforces the conclusion that neither imputation method confers a systematic advantage.

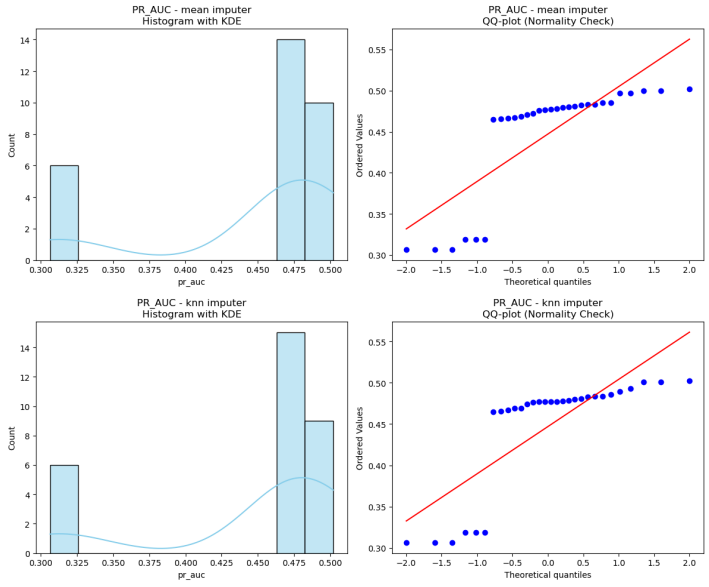


Figure 4.3.6: Distributions of PR-AUC scores under mean (left) and KNN (right) imputation. No significant skew or divergence is observed.

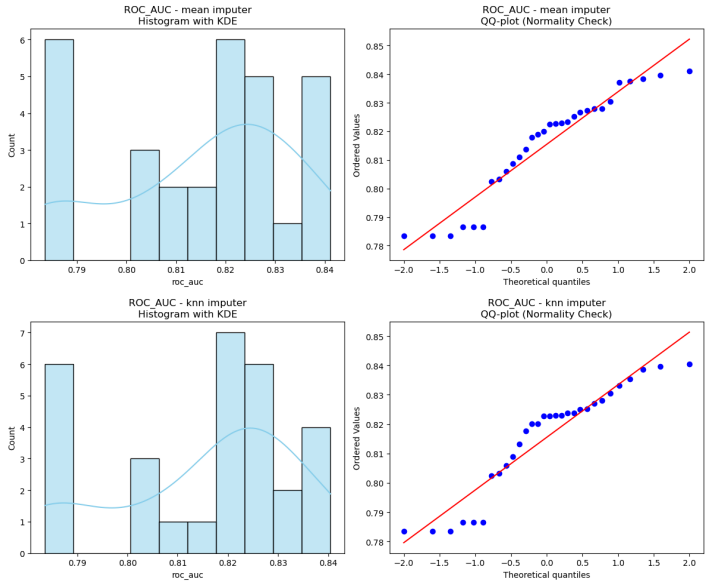


Figure 4.3.7: Distributions of ROC-AUC scores under mean (left) and KNN (right) imputation. Visual inspection supports the statistical equivalence.

Conclusion. The results corroborate the hypothesis that the model pipeline exhibits robustness to the choice of imputation method. Considering the exceptionally low proportion of missing data (less than 0.06% within the affected features), it is unsurprising that the selection of imputer does not materially impact downstream model performance. Although KNN imputation is often regarded as a more advanced technique, it offers no discernible benefit over mean imputation in this context. Consequently, the simplicity and computational efficiency of mean imputation position it as a preferred default, particularly in scenarios where retraining speed or real-time inference is critical.

Overall, this analysis reinforces our broader conclusion regarding **RQ3**: when missingness is minimal, model predictive accuracy remains largely unaffected by imputation strategy, allowing the use of simpler methods without compromising performance.

4.3.4 Feature Importance Analysis

Identifying which features most strongly influence the model's predictions is essential for ensuring interpretability, validating domain alignment, and guiding future model simplifications. In this section, we analyze the best-performing configuration—LightGBM with KNN imputation and chronological split—using two complementary approaches: internal gain-based importance (Figure 4.3.8) and SHAP value analysis (Figure 4.3.9).

Gain-based feature importance. According to LightGBM's internal gain metric, spatial and contextual variables emerge as the most informative features. In particular, `num__angle__to__goal` and `num__distance__to__goal` dominate the ranking, both exceeding a cumulative gain of 100 000. These are closely followed by positional features such as `num__end__coordinates_x/y`, and temporal indicators like `num__time__since__last__shot` and `num__eventSec`.

This ranking is not only statistically robust but also tactically insightful. Factors such as angle and distance to goal naturally influence shooting threat, while endpoint coordinates and

timing encapsulate the dynamic context surrounding each action. These findings support the hypothesis that the model effectively emphasizes fundamental indicators of offensive value, aligning well with established football intuition.

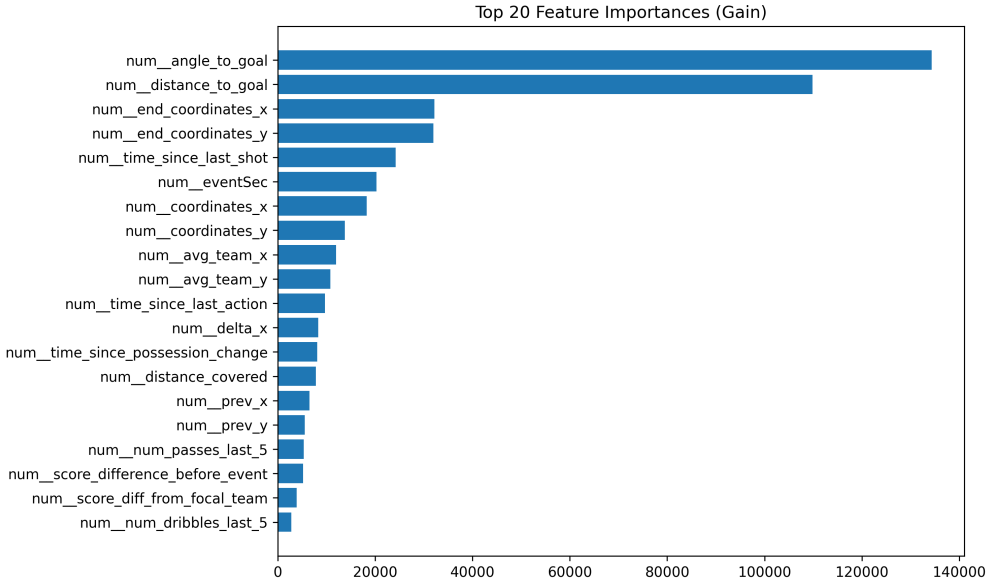


Figure 4.3.8: Top 20 feature importances based on LightGBM's gain metric. Key spatial and temporal features dominate the model's internal decision process.

SHAP-based explanation. To complement the overall gain-based feature importance, we analyze local feature contributions using SHAP values. The summary plot in Figure 4. 3.9 highlights the prominent influence of spatial and temporal features. The top-ranked features—`num__distance_to_goal`, `num__end_coordinates_x/y`, `num__eventSec`, and `num__angle_to_goal`—largely coincide with those identified through gain, reinforcing confidence in the model's internal consistency.

Crucially, SHAP also clarifies the directionality of feature effects: lower values of `distance_to_goal` and narrower `angle_to_goal` (representing close-range, tight-angle situations) tend to increase the predicted probability of contribution. Likewise, actions occurring earlier in the sequence (lower `eventSec`) exert greater influence, suggesting that impactful action sequences often begin prior to entering the final third of the pitch.

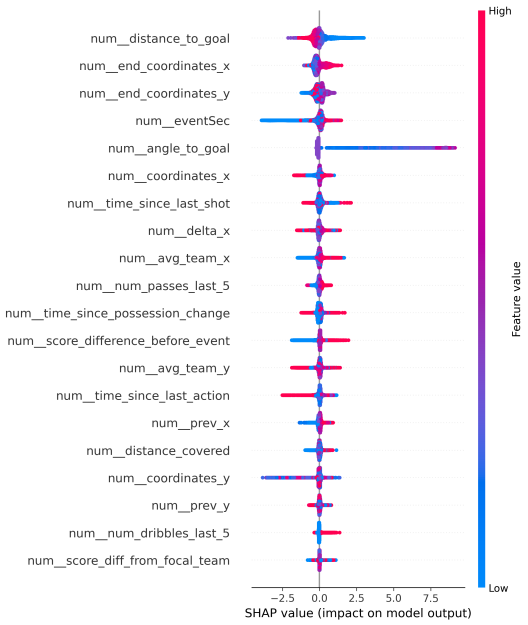


Figure 4.3.9: SHAP summary plot for the LightGBM model. Each dot represents a SHAP value for one sample; red indicates high feature values, blue low values. Horizontal dispersion reflects the magnitude of influence.

Implications. Taken together, these analyses confirm that the model has successfully internalized patterns that align with both football logic and domain expectations. The consistency between gain-based rankings and SHAP interpretations strengthens the credibility of the feature engineering pipeline. Moreover, the interpretability of the top predictors supports potential operational uses, such as tactical reporting or decision support for analysts.

From a practical perspective, this insight paves the way for model simplification. Focusing on a reduced set of top-ranked features has the potential to preserve predictive performance while enhancing model interpretability and computational efficiency—factors that are particularly crucial in real-time or resource-limited environments.

4.3.5 Error Diagnosis

Normalized Confusion Matrix. To better understand the nature of model errors, we computed the normalized confusion matrix on the test set (Figure 4.3.10). The model achieves perfect identification of non-contributing actions (True Negative Rate = 1.00), while recall for contributing actions is limited to 0.27 (True Positive Rate). This asymmetry stems directly from the strong class imbalance: only 5.6% of all test actions are labeled as contributing. In this context, detecting over a quarter of these rare positives represents a nearly **5-fold lift** over random guessing, which is a meaningful gain in practice—especially under a strict definition of contribution.

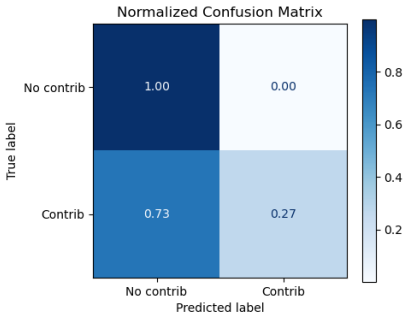


Figure 4.3.10: Normalized confusion matrix of the best model. Absolute counts: TN = 122,006, FP = 85, FN = 4,793, TP = 1,746. The model detects non-contributing actions with perfect precision and identifies 27% of contributors.

Probability Distribution by True Class. To assess the model's discriminative power, we visualized the distribution of predicted probabilities by true class (Figure 4.3.11). Negative instances cluster tightly around low probabilities, while positives show a bimodal pattern—receiving either confidently low or high scores. The Kolmogorov–Smirnov (KS) distance between the two distributions is 0.537, confirming substantial separability. This diagnostic is particularly useful for fine-tuning thresholds or designing ranking filters in practical settings.

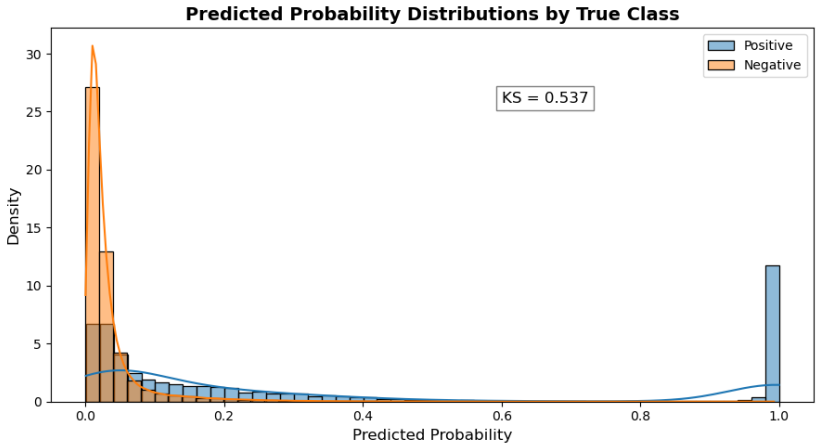
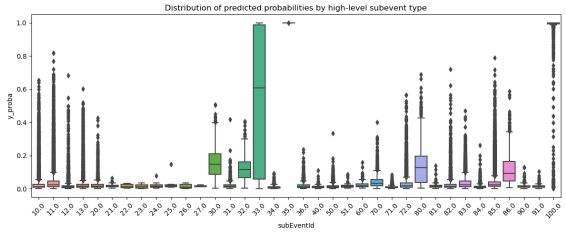
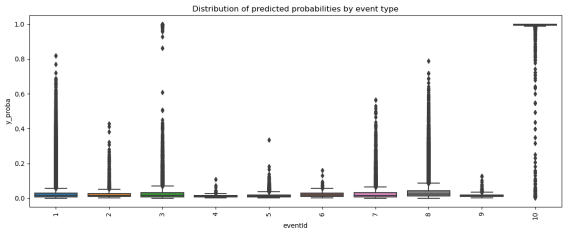


Figure 4.3.11: Distribution of predicted probabilities by class. KS = 0.537 indicates meaningful separation between positive and negative labels.

Biases by Action Type. To further investigate the semantic knowledge learned by the model, we analyzed predicted probabilities aggregated by event and subevent categories. As illustrated in Figures 4.3.12a and 4.3.12b, the model consistently assigns high probabilities to subEventIds associated with finishing actions (e.g., `Shot`, `Free kick shot`). In contrast, actions such as `Clearance` and `Throw-in` are assigned probabilities close to zero, reflecting their limited role in sequences leading to quality shots. These observations indicate that the model has effectively captured meaningful domain-specific distinctions.



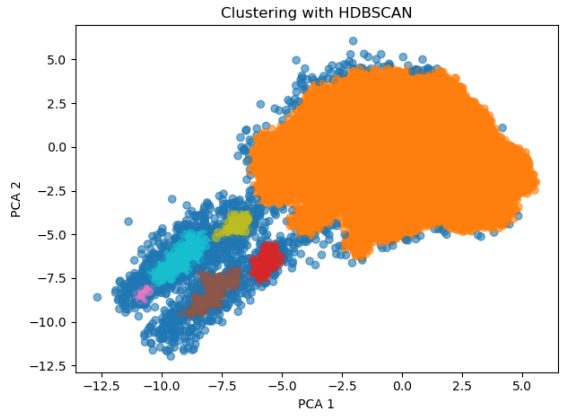
(a) Predicted probability by `subEventId`.



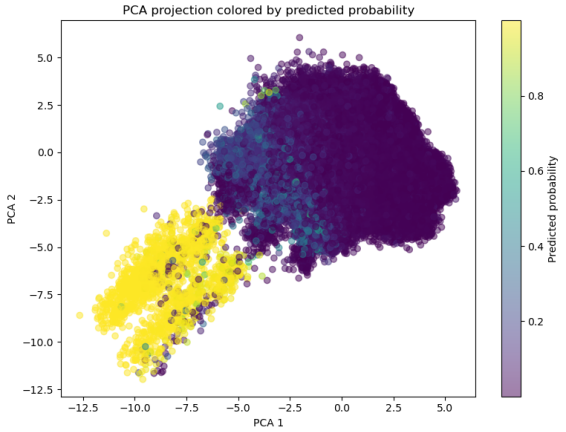
(b) Predicted probability by `eventId`.

Figure 4.3.12: Predicted probabilities grouped by event and subevent types. Offensive actions like shots and through passes yield high scores, while defensive or neutral actions receive near-zero values.

Clustering on Predicted Behavior. To explore latent structure in predicted behavior, we applied PCA to the standardized action features and clustered the resulting 2D representation using HDBSCAN, a hierarchical density-based algorithm. As shown in Figure 4.3.13a, HDBSCAN discovers a large dominant cluster along with several compact and semantically meaningful smaller clusters. These smaller groups correspond to high-probability actions such as well-positioned shots, while the large cluster mostly aggregates low-impact events like routine passes. When coloring the PCA projection by predicted probability (Figure 4.3.13b), a clear gradient emerges—reinforcing that the model encodes tactically intuitive behavioral signals.



(a) HDBSCAN clustering on PCA-reduced action features.



(b) PCA projection colored by predicted probability.

Figure 4.3.13: PCA-based projections of player actions, where HDBSCAN clustering differentiates low- and high-probability actions into distinct groups. Coloring by predicted likelihood exhibits a continuous gradient, consistent with tactical expectations.

Note: DBSCAN-based clustering results are included in Appendix C.6 for comparison.

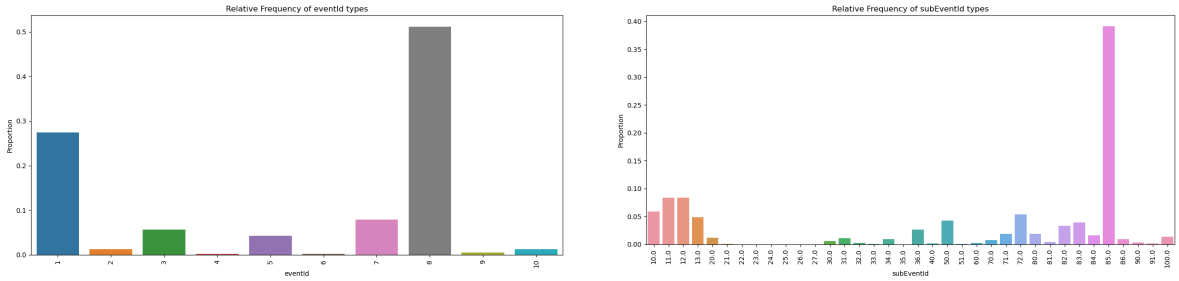
Semantic Composition of Clusters. To better understand the behavioral meaning of each HDBSCAN cluster, we analyzed the composition of their dominant event types and positional characteristics. Cluster 0, the largest group (133,692 actions), mostly contains passes with low

predicted probability, while clusters 1 to 5 predominantly contain shots with varying likelihood and spatial patterns. Notably, cluster 3 groups shots taken very close to goal with extremely high predicted probability (mean $p = 0.99$, $x \approx 91.6$), indicating ideal scoring situations (e.g., one-on-ones or penalties). Full statistics are provided in Table 4.3.7.

Cluster	Size (n)	Mean Proba	Main Event	x Mean	y Mean
-1	979	0.7678	Shot	82.4	48.8
0	133692	0.0390	Pass	48.7	49.9
1	125	0.9125	Shot	80.7	48.4
2	218	0.9416	Shot	85.8	48.4
3	34	0.9867	Shot	91.6	47.1
4	173	0.9127	Shot	80.5	48.2
5	434	0.9540	Shot	86.5	49.2

Table 4.3.7: Summary of cluster composition (event type, position, and predicted probability).

Action Type Distribution. Lastly, we analyzed the relative frequency of different action types. As depicted in Figure 4.3.14, the dataset is predominantly composed of low-impact actions, with passes (eventId 8) being the most frequent, followed by duels and various supporting events. This inherent class imbalance helps explain the model's conservative scoring behavior: high contribution probabilities remain rare due to the infrequency of such impactful events. Acknowledging this context is essential for accurate interpretation of the model's outputs.



(a) Relative frequency of `eventId`.

(b) Relative frequency of `subEventId`.

Figure 4.3.14: Distribution of event and subevent categories in the dataset. The dominance of low-impact actions justifies the overall rarity of high predicted contribution scores.

Chapter 5

Conclusion

5.1 Conclusion

5.1.1 Overall Conclusion

In summary, this thesis presents a scalable and explainable framework for valuing football player actions using interpretable binary labels. The core contribution lies in the design and validation of the `flag_contributing_actions_advanced` function, a rule-based labeling mechanism that captures whether an action contributes to a quality shot based on temporal and possession-based criteria. This method offers a tactically intuitive, reproducible, and transparent alternative to black-box valuation models such as VAEP [14].

Building upon the proposed labeling framework, this study introduces the `composite_score`, a unified metric that integrates offensive contributions and defensive disruptions through a weighted average. This score offers a comprehensive assessment of player impact, uncovering systematic patterns across roles, sub-roles, and tactical profiles. Case studies and correlation analyses with salary and team rankings further validate the metric's external relevance, while sensitivity analyses highlight its flexibility to accommodate varying scouting and tactical objectives.

On the predictive front, the thesis demonstrates that contemporary gradient boosting methods—most notably LightGBM—can accurately and reliably infer contribution labels, even in the presence of significant class imbalance. The modeling pipeline incorporates rigorous preprocessing, temporally aware validation protocols, and feature importance evaluations, thereby ensuring both robustness and interpretability.

By effectively linking tactical domain knowledge with supervised learning techniques, this work establishes a reproducible, interpretable, and extensible framework for football player evaluation. It provides analysts and data scientists with a solid foundation for practical downstream applications, including scouting, tactical analysis, and recruitment decision support.

5.1.2 Answers to Research Questions

RQ1: Can a labeling pipeline reflect real-world player value?
Experiment 1 demonstrates that a simple, rule-based labeling system can yield tactically coherent and externally valid metrics. The binary classification of actions—based on temporal and possession-based windows—produced contribution and disruption ratios that align with real-world indicators such as player salaries and Ballon d'Or rankings (see Section 4.2). The resulting composite score, which combines offensive and defensive impact, revealed structured

patterns across roles and sub-roles (Figures 4.2.1 and 4.2.2). For instance, elite forwards (e.g., Kane, Salah) and central creators (e.g., De Bruyne) were accurately ranked, while the method also surfaced undervalued profiles such as Marcos Alonso or Richarlison (Table 4.2.1). Parameter sensitivity analyses further confirmed the system's robustness and tactical interpretability (Figures 4.2.6 and 4.2.13), validating its potential as a reliable labeling framework.

RQ2: Which model performs best for predicting contribution labels?
Among the five supervised classifiers evaluated in Experiment 2, LightGBM consistently outperformed its competitors across all performance metrics (see Table 4.3.2), achieving the highest mean PR-AUC (0.5013) and ROC-AUC (0.8395). Statistical analyses, including the Kruskal–Wallis test followed by Dunn's post-hoc comparisons (Table 4.3.1), confirmed that both LightGBM and XGBoost significantly surpass Random Forest, Logistic Regression, and CatBoost. Additionally, LightGBM strikes a favorable balance between predictive accuracy and training efficiency (Figure 4.3.3), establishing it as the most appropriate choice for real-time or large-scale football analytics applications.

RQ3: How do preprocessing choices impact model performance?
Two preprocessing dimensions were evaluated: imputation and train/test split strategy (Sections 4.3.2 and 4.3.3). First, the comparison between mean and KNN imputation yielded no statistically significant performance difference ($p = 1.000$ for PR-AUC, Table 4.3.6), despite the latter being computationally more intensive. Given the extremely low missingness rate ($< 0.0015\%$), mean imputation is recommended for its simplicity and speed. Second, the split strategy analysis showed that chronological splits provide a more realistic estimate of real-world performance than random stratified splits. Although the latter yield higher scores, they introduce optimistic bias due to temporal leakage (Table 4.3.5), reinforcing the importance of temporally consistent evaluation in deployment scenarios.

RQ4: Can the composite score serve as a holistic measure of player performance?
The composite score—computed as a weighted sum of contribution and disruption ratios (Section 3.4)—proved effective in capturing multidimensional player value. Role-specific analysis showed it distinguishes creators, disruptors, and hybrids (Figures 4.2.3 and 4.2.10). Its moderate correlations with salary and club ranking support its external validity (Figures 4.2.4 and 4.2.5), while case studies confirmed its alignment with known football narratives. Furthermore, the score's sensitivity to the α parameter (Figures 4.2.13 and 4.2.14) enables tactical customization, making it an adaptable tool for scouting, player comparison, and recruitment.

5.2 Limitations

Despite the encouraging results obtained throughout this study, several limitations must be acknowledged. First, the scope of the dataset was deliberately restricted to the 2017/2018 Premier League season. While this choice ensured homogeneity and allowed for computational feasibility, it may reduce the generalizability of the findings across other leagues, seasons, or footballing cultures. Extending the analysis to multiple competitions would be necessary to validate the broader applicability of the labeling and modeling framework.

Secondly, the contribution labels were constructed using predefined heuristics based on a fixed window size and possession threshold. Although these parameters were justified both tactically and empirically, they represent a simplification of real game dynamics. In particular, long build-up plays or secondary attacking phases might not be fully captured by the current

rule-based approach, potentially underestimating the value of certain actions.

Thirdly, the feature set was primarily composed of spatial, temporal, and categorical indicators directly available from event data. More granular context—such as off-ball positioning, opponent pressure levels, or tactical formations—was not incorporated due to data limitations. As a result, the model may fail to account for important hidden variables that influence action outcomes.

Another limitation stems from the binary nature of the target variable. By simplifying contribution to a categorical "yes" or "no" label, the approach overlooks varying degrees of influence that could be better represented by continuous valuation models. This dichotomous classification may reduce the expressiveness of predictions, particularly in borderline or ambiguous cases.

Finally, although various techniques were employed to mitigate the impact of class imbalance (e.g., `scale_pos_weight`, use of `PR-AUC`), the extreme skew in the target distribution remains a challenge. It may affect both model calibration and the stability of performance estimates, particularly for rare but tactically significant positive cases.

These limitations do not undermine the insights developed throughout this thesis; rather, they outline avenues for future improvements aimed at enhancing tactical granularity, predictive robustness, and generalizability across different contexts.

Furthermore, the exclusive reliance on Wyscout event data—despite its detailed capture of on-ball actions—constrains the integration of richer contextual features such as off-ball positioning, pressing intensity, and freeze-frame analyses. These additional dimensions are crucial for comprehensively representing defensive influence and tactical nuances, particularly for players occupying roles with lower intervention frequency.

A promising direction for extending this research involves validating the current findings with StatsBomb's open dataset [33], which offers enriched metrics including defensive pressure and positional heatmaps. Incorporating such data could substantially improve the descriptive accuracy and tactical relevance of the model, especially for defensive profiles that remain underrepresented in traditional event-based annotation frameworks.

5.3 Future Work

Enhancing and validating feature representations for robust player ranking. Future research may investigate how engineered and context-aware features—such as spatial zones, event abstractions, and tactical context—can improve model performance. Approaches inspired by KU Leuven's action abstraction [14] offer promising directions. In parallel, evaluating how these enriched representations interact with data augmentation techniques could help assess the stability and consistency of the composite player ranking across different modeling setups.

Extending the analysis to multiple leagues and seasons. Future research should focus on assessing the robustness and generalizability of the models across the five major European leagues—Premier League, La Liga, Serie A, Bundesliga, and Ligue 1. Expanding the dataset to include multiple seasons would also facilitate temporal validation and the evaluation of model stability over time. The current pipeline has been developed to accommodate such scaling with minimal adaptation.

Developing explainable interfaces for practitioners.

Building upon the SHAP-based interpretability analysis presented in Section 4.3.4, subsequent work could investigate methods for integrating these explanations into intuitive tools tailored for scouts and coaches. Potential applications include interactive dashboards, recommendation engines, or simulation platforms that combine model predictions with domain-specific football insights.

Bibliography

[1] Markel Rico-González et al. "Machine learning application in soccer: a systematic review". In: *Biology of Sport* 40.1 (2023), pp. 249–263. ISSN: 0860-021X. DOI: 10.5114/biolsport.2023.112970. URL: https://www.termedia.pl/doi/10.5114/biolsport.2023.112970 (visited on 10/20/2024).

[2] Alessio Rossi et al. "Effective injury forecasting in soccer with GPS training data and machine learning". en. In: *PLOS ONE* 13.7 (July 2018). Publisher: Public Library of Science, e0201264. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0201264. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0201264 (visited on 03/25/2025).

[3] Jesse Davis et al. "Methodology and evaluation in sports analytics: challenges, approaches, and lessons learned". In: *Machine Learning* 113.6977-7010 (2024). Published online: 17 July 2024. DOI: 10.1007/s10994-024-06585-0. URL: https://link.springer.com/article/10.1007/s10994-024-06585-0 (visited on 10/20/2024).

[4] Alan Griffin et al. "The Association Between the Acute:Chronic Workload Ratio and Injury and its Application in Team Sports: A Systematic Review". en. In: *Sports Medicine* 50.3 (Mar. 2020). Company: Springer Distributor: Springer Institution: Springer Label: Springer Number: 3 Publisher: Springer International Publishing, pp. 561–580. ISSN: 1179-2035. DOI: 10.1007/s40279-019-01218-2. URL: https://link-springer-com.proxy.bib.uclouvain.be:2443/article/10.1007/s40279-019-01218-2 (visited on 03/25/2025).

[5] Elia Morgulev, Ofer H. Azar, and Ronnie Lidor. "Sports analytics and the big-data era". In: *International Journal of Data Science and Analytics* 5.4 (June 1, 2018), p. 221. ISSN: 2364-4168. DOI: 10.1007/s41060-017-0093-7. URL: https://doi.org/10.1007/s41060-017-0093-7 (visited on 10/18/2024).

[6] Francisco Martins et al. "Predictive Modeling of Injury Risk Based on Body Composition and Selected Physical Fitness Tests for Elite Football Players". eng. In: *Journal of Clinical Medicine* 11.16 (Aug. 2022), p. 4923. ISSN: 2077-0383. DOI: 10.3390/jcm11164923.

[7] Guillermo Durán. "Sports scheduling and other topics in sports analytics: a survey with special reference to Latin America". In: *TOP* 29.1 (Apr. 1, 2021), pp. 146–149. ISSN: 1863-8279. DOI: 10.1007/s11750-020-00576-9. URL: https://doi.org/10.1007/s11750-020-00576-9 (visited on 10/18/2024).

[8] Konstantinos Apostolou and Christos Tjortjis. "Sports Analytics algorithms for performance prediction". In: *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*. 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA). July 2019, pp. 1–4. DOI: 10.1109/IISA.

2019.8900754. URL: https://ieeexplore.ieee.org/abstract/document/8900754 (visited on 10/14/2024).

[9] Scikit-learn Developers. *Ensemble Methods - Forests of Randomized Trees*. Consulted on: 2024-11-09. 2024. URL: https://scikit-learn.org/stable/modules/ensemble.html#forest.

[10] Scikit-learn Developers. *Logistic Regression - Scikit-learn*. Consulted on: 2024-11-09. 2024. URL: https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html.

[11] Scikit-learn Developers. *MLP Classifier - Scikit-learn*. Consulted on: 2024-11-09. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html.

[12] Scikit-learn Developers. *Linear Support Vector Classification - Scikit-learn*. Consulted on: 2024-11-09. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html.

[13] Victor Chazan Pantzalis and Christos Tjortjis. "Sports Analytics for Football League Table and Player Performance Prediction". In: *2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA*. 2020 11th International Conference on Information, Intelligence, Systems and Applications (IISA. July 2020, pp. 1–8. DOI: 10.1109/IISA50023.2020.9284352. URL: https://ieeexplore.ieee.org/abstract/document/9284352 (visited on 11/09/2024).

[14] Tom Decroos et al. "Actions Speak Louder Than Goals: Valuing Player Actions in Soccer". In: (2018). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1802.07127. URL: https://arxiv.org/abs/1802.07127 (visited on 11/08/2024).

[15] Tiago Mendes-Neves, Luís Meireles, and João Mendes-Moreira. *Forecasting Events in Soccer Matches Through Language*. Apr. 26, 2024. DOI: 10.48550/arXiv.2402.06820. arXiv: 2402.06820. URL: http://arxiv.org/abs/2402.06820 (visited on 10/20/2024).

[16] Alex Rathke. "An examination of expected goals and shot efficiency in soccer". en. In: *Journal of Human Sport and Exercise* 12.Proc2 (2017). ISSN: 1988-5202. DOI: 10.14198/jhse.2017.12.Proc2.05. URL: http://hdl.handle.net/10045/68771 (visited on 03/25/2025).

[17] Karun Singh. *Introducing Expected Threat (xT)*. Consulté le: 2024-11-09. 2019. URL: https://karun.in/blog/expected-threat.html.

[18] Lotte Bransen and Jan Van Haaren. "Measuring Football Players' On-the-Ball Contributions from Passes During Games". en. In: *Machine Learning and Data Mining for Sports Analytics*. ISSN: 1611-3349. Springer, Cham, 2019, pp. 3–15. ISBN: 978-3-030-17274-9. DOI: 10.1007/978-3-030-17274-9_1. URL: https://link-springer-com.proxy.bib.uclouvain.be:2443/chapter/10.1007/978-3-030-17274-9_1 (visited on 03/25/2025).

[19] Daniel Cervone et al. "A Multiresolution Stochastic Process Model for Predicting Basketball Possession Outcomes". In: *Journal of the American Statistical Association* 111.514 (2016), pp. 585–599.

[20] Aditya Singh et al. "xOVA: Explainable Offensive Value Added for Soccer Player Evaluation". In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 3189–3197.

[21] Koen Vossen. *Wyscout Soccer Match Event Dataset (based on Wyscout data from Figshare)*. Original data from: Paolo Cintia, Luca Pappalardo, Alessio Rossi et al., *Soccer Match Event Dataset*, Figshare, 2019. Available at: https://figshare.com/collections/Soccer_match_event_dataset/4415000/2. Accessed: 2024-11-09. 2024. URL: https://github.com/koenvo/wyscout-soccer-match-event-dataset.

[22] Scikit-learn Developers. *sklearn.model$_s$election.HalvingRandomSearchCV*. Consulted on: 2024-11-09. 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.HalvingRandomSearchCV.html.

[23] Gareth James et al. *An Introduction to Statistical Learning: with Applications in Python*. 2nd. Springer, 2021. ISBN: 9781071614174.

[24] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

[25] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://papers.nips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

[26] Liudmila Prokhorenkova et al. "CatBoost: Unbiased Boosting with Categorical Features". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html.

[27] Oona Rainio, Jarmo Teuho, and Riku Klén. "Evaluation metrics and statistical tests for machine learning". en. In: *Scientific Reports* 14.1 (Mar. 2024). Publisher: Nature Publishing Group, p. 6086. ISSN: 2045-2322. DOI: 10.1038/s41598-024-56706-x. URL: https://www.nature.com/articles/s41598-024-56706-x (visited on 03/28/2025).

[28] Tom Fawcett. "An introduction to ROC analysis". In: *Pattern Recognition Letters* 27.8 (2006), pp. 861–874.

[29] Data Science Enthusiast. *AUC-ROC Curve*. https://medium.com/@data.science.enthusiast/auc-roc-curve-ae9180eaf4f7. Accessed: 2024-11-09. 2019.

[30] Takaya Saito and Marc Rehmsmeier. "The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets". In: *PLOS ONE* 10.3 (2015), e0118432.

[31] Siddhi Prabhu. *Gaining an Intuitive Understanding of Precision and Recall*. https://medium.com/data-science/gaining-an-intuitive-understanding-of-precision-and-recall-3b9df37804a7. Accessed: 2024-11-09. 2020.

[32] Scikit-learn developers. *Scikit-learn: Machine Learning in Python – Log loss*. https://scikit-learn.org/stable/modules/model_evaluation.html#log-loss. Accessed: 2025-05-23.

[33] StatsBomb. *StatsBomb Open Data*. Consulté le: 2024-11-09. 2024. URL: https://github.com/statsbomb/open-data.

[34] Premier League. *2017/18 Season Review: City break records and Salah lights up the league*. Consulté le: 2025-04-26. 2018. URL: https://www.premierleague.com/history/season-reviews/79/report.

[35] GOAL. *Luka Modric and the 2018 Ballon d'Or final rankings*. Accessed: 2025-04-16. 2018. URL: https://www.goal.com/en/lists/luka-modric-and-the-2018-ballon-dor-final-rankings/g4abmah6vk6u1qvmfansvvf5y.

Appendix A

Supplementary Tables

A.1 Mapping of Tags to Descriptions

Table A.1.1: Mapping of tags to their names and descriptions.

Tag	Label	Description
101	Goal	A successful attempt where the ball crosses the goal line within the frame.
102	Own goal	A goal inadvertently scored by a player into their own team's net.
301	Assist	The final pass or action leading directly to a goal by a teammate.
302	Key pass	A pass that directly leads to a shot on goal, not necessarily resulting in a goal.
1901	Counter attack	A rapid offensive move initiated immediately after regaining possession.
401	Left	Action executed using the player's left foot.
402	Right	Action executed using the player's right foot.
403	Head/body	Action executed using the head or other body parts, excluding the feet.
1101	Direct	A set-piece (e.g., free kick) taken with a direct attempt on goal.
1102	Indirect	A set-piece requiring a secondary touch before a goal can be scored.
2001	Dangerous ball lost	Loss of possession in a manner that presents an immediate threat.
2101	Blocked	An attempt (e.g., shot or pass) obstructed by an opponent.
801	High	A ball played at a height above the average player's head.
802	Low	A ball played at or below knee height.
1401	Interception	Gaining possession by anticipating and intercepting an opponent's pass.
1501	Clearance	A defensive action to remove the ball from the defensive area.
201	Opportunity	A significant chance to score, often referred to as a "big chance."

Tag	Label	Description
1301	Feint	A deceptive movement to mislead an opponent.
1302	Missed ball	An unsuccessful attempt to make contact with the ball.
501	Free space right	Movement into unmarked space on the right side of the field.
502	Free space left	Movement into unmarked space on the left side of the field.
503	Take on left	Attempt to dribble past an opponent on the left side.
504	Take on right	Attempt to dribble past an opponent on the right side.
1601	Sliding tackle	A tackle executed by sliding on the ground to dispossess an opponent.
601	Anticipated	Successfully predicting and acting upon an opponent's move.
602	Anticipation	The act of predicting an opponent's move, regardless of outcome.
1701	Red card	A dismissal from the game for a serious offense.
1702	Yellow card	A caution issued for misconduct.
1703	Second yellow card	A second caution leading to a red card dismissal.
1201	gb	Goal bottom center: shot placement at the bottom center of the goal.
1202	gbr	Goal bottom right: shot placement at the bottom right corner.
1203	gc	Goal center: shot placement at the center of the goal.
1204	gl	Goal left: shot placement at the left side of the goal.
1205	glb	Goal bottom left: shot placement at the bottom left corner.
1206	gr	Goal right: shot placement at the right side of the goal.
1207	gt	Goal top center: shot placement at the top center of the goal.
1208	gtl	Goal top left: shot placement at the top left corner.
1209	gtr	Goal top right: shot placement at the top right corner.
1210	obr	Out bottom right: shot missed to the bottom right of the goal.
1211	ol	Out left: shot missed to the left side of the goal.
1212	olb	Out bottom left: shot missed to the bottom left of the goal.
1213	or	Out right: shot missed to the right side of the goal.
1214	ot	Out top: shot missed over the top of the goal.
1215	otl	Out top left: shot missed over the top left corner.
1216	otr	Out top right: shot missed over the top right corner.
1217	pbr	Post bottom right: shot hit the bottom right post.
1218	pl	Post left: shot hit the left post.
1219	plb	Post bottom left: shot hit the bottom left post.
1220	pr	Post right: shot hit the right post.
1221	pt	Post top: shot hit the top post.
1222	ptl	Post top left: shot hit the top left post.
1223	ptr	Post top right: shot hit the top right post.
901	Through	A pass played between defenders into space for a teammate to run onto.

Tag	Label	Description
1001	Fairplay	An action demonstrating sportsmanship.
701	Lost	Loss of possession or unsuccessful action.
702	Neutral	An action with no significant positive or negative outcome.
703	Won	Successful action resulting in positive outcome.
1801	Accurate	An action executed with precision, meeting its intended target.
1802	Not accurate	An action failing to meet its intended target.

Table A.1.1: Mapping of tags to their names and descriptions.

A.2 Mapping of Events and Subevents

Table A.2.1: Mapping of event and subevent IDs to their labels and descriptions.

Event ID	Subevent ID	Event Label	Subevent Label
1	10	Duel	Air duel: Contest for the ball in the air between two players.
1	11	Duel	Ground attacking duel: Offensive player attempts to get past a defender on the ground.
1	12	Duel	Ground defending duel: Defensive player challenges an attacker on the ground.
1	13	Duel	Ground loose ball duel: Contest for a loose ball on the ground.
2	20	Foul	Foul: Infringement of the rules resulting in a free kick or penalty.
2	21	Foul	Hand foul: Handling the ball deliberately.
2	22	Foul	Late card foul: Foul committed after the ball has been played, often resulting in a card.
2	23	Foul	Out of game foul: Foul committed when the ball is out of play.
2	24	Foul	Protest: Player shows dissent or protests a decision.
2	25	Foul	Simulation: Attempt to deceive the referee by feigning injury or foul.
2	26	Foul	Time waste: Deliberate delay of the game to gain advantage.
3	30	Free Kick	Free kick: Restart of play from a foul, taken from the spot of the infringement.
3	31	Free Kick	Free kick cross: Free kick delivered into the penalty area.
3	32	Free Kick	Free kick shot: Direct attempt on goal from a free kick.
3	33	Free Kick	Free kick short: Short pass from a free kick to maintain possession.
4	40	Goalkeeper leaving line	Goalkeeper leaving line: Goalkeeper moves off the goal line, often to intercept or challenge.
5	50	Goalkeeper	Goalkeeper: General goalkeeper actions not specified elsewhere.
5	51	Goalkeeper	Goalkeeper leaving line: Goalkeeper advances from the goal line to engage play.

Event ID	Subevent ID	Event Label	Subevent Label
5	52	Goalkeeper	Goalkeeper out of game: Goalkeeper is out of position or not involved in play.
6	60	Offside	Offside: Player is in an offside position when the ball is played to them.
7	70	Others on the ball	Others on the ball: Actions involving the ball not categorized elsewhere.
8	80	Pass	Pass: General pass from one player to another.
8	81	Pass	Cross: Ball played from wide areas into the penalty area.
8	82	Pass	Hand pass: Ball passed using the hand, typically by the goalkeeper.
8	83	Pass	Head pass: Ball passed using the head.
8	84	Pass	Launch: Long ball played forward, often from defense to attack.
8	85	Pass	Simple pass: Short, straightforward pass to a teammate.
8	86	Pass	Smart pass: Creative or incisive pass breaking defensive lines.
8	87	Pass	High pass: Ball played at a height above the ground.
8	88	Pass	Low pass: Ball played along the ground.
8	89	Pass	Lay-off: Pass made by a player receiving the ball and immediately passing it to a teammate.
8	90	Pass	Flick-on: Ball redirected with a light touch, often with the head.
8	91	Pass	Through ball: Pass played between defenders into space for a teammate to run onto.
9	100	Shot	Shot: Attempt to score by directing the ball towards the goal.
9	101	Shot	Head shot: Shot taken using the head.
9	102	Shot	Left foot shot: Shot taken with the left foot.
9	103	Shot	Right foot shot: Shot taken with the right foot.
9	104	Shot	Other body part shot: Shot taken with a body part other than the foot or head.
9	105	Shot	Penalty: Shot taken from the penalty spot.
9	106	Shot	Free kick shot: Direct shot on goal from a free kick.
9	107	Shot	Volley: Shot taken before the ball touches the ground.
9	108	Shot	Half volley: Shot taken just after the ball bounces.
9	109	Shot	Overhead kick: Acrobatic shot where the player kicks the ball over their own head.
10	110	Save attempt	Save attempt: Goalkeeper's attempt to prevent the ball from entering the goal.
10	111	Save attempt	Diving save: Goalkeeper dives to stop the ball.
10	112	Save attempt	Standing save: Goalkeeper saves the ball without diving.
10	113	Save attempt	Reflex save: Quick reaction save, often from close range.
10	114	Save attempt	Parry: Goalkeeper deflects the ball away from goal.
10	115	Save attempt	Punch: Goalkeeper punches the ball away, typically from crosses.
10	116	Save attempt	Catch: Goalkeeper catches the ball to gain possession.

Event ID	Subevent ID	Event Label	Subevent Label
11	120	Take on	Take on: Attempt to dribble past an opponent.
11	121	Take on	Successful take on: Dribble past an opponent successfully.
11	122	Take on	Unsuccessful take on: Attempt to dribble past an opponent fails.
12	130	Clearance	Clearance: Defensive action to remove the ball from the defensive area.
13	140	Interception	Interception: Player anticipates and intercepts an opponent's pass.
14	150	Recovery	Recovery: Regaining possession of the ball after it was lost.
15	160	Ball out of the field	Ball goes out of play over the sidelines or end lines.

Table A.2.1: Mapping of event and subevent IDs to their labels and descriptions.

A.3 Subset player Experiment 1

Full Name	Role Name	Sub Role
Jan Vertonghen	Defender	Centre-back
Christian Dannemann Eriksen	Midfielder	Central/Attacking Midfielder
Johann Berg Guðmundsson	Midfielder	Winger
Erik Pieters	Defender	Unknown
Virgil van Dijk	Defender	Centre-back
Dušan Tadić	Midfielder	Winger
Davy Pröpper	Midfielder	Central/Attacking Midfielder
Mesut Özil	Midfielder	Central/Attacking Midfielder
Álvaro Borja Morata Martín	Forward	Central Striker
Francesc Fàbregas i Soler	Midfielder	Central/Attacking Midfielder
Alexis Alejandro Sánchez Sánchez	Forward	Winger Forward
Ignacio Monreal Eraso	Defender	Full-back
José Salomón Rondón Giménez	Forward	Central Striker
Kieran Gibbs	Defender	Full-back
Laurent Koscielny	Defender	Centre-back
Ryan Bertrand	Defender	Full-back
Oriol Romeu Vidal	Midfielder	Defensive Midfielder
Romelu Lukaku Menama	Forward	Central Striker
Michael Keane	Defender	Centre-back
Paul Pogba	Midfielder	Central/Attacking Midfielder
Luis Antonio Valencia Mosquera	Defender	Full-back
Ashley Young	Defender	Full-back
Jordan Brian Henderson	Midfielder	Defensive Midfielder
Jonjo Shelvey	Midfielder	Central/Attacking Midfielder
Yohan Cabaye	Midfielder	Central/Attacking Midfielder
Marc Albrighton	Midfielder	Winger
Marcos Alonso Mendoza	Defender	Full-back
Dale Stephens	Midfielder	Defensive Midfielder
Jack Cork	Midfielder	Central/Attacking Midfielder
Martin Olsson	Defender	Full-back

Continued on next page

Full Name	Role Name	Sub Role
Shane Duffy	Defender	Centre-back
Kyle Walker	Defender	Full-back
Pablo Javier Zabaleta Girod	Defender	Full-back
David Josué Jiménez Silva	Midfielder	Central/Attacking Midfielder
James Milner	Midfielder	Central/Attacking Midfielder
Sergio Leonel Agüero del Castillo	Forward	Central Striker
Kyle Naughton	Defender	Full-back
Chris Brunt	Midfielder	Defensive Midfielder
Glenn Murray	Forward	Central Striker
Wilfried Zaha	Midfielder	Winger
Fabian Delph	Midfielder	Central/Attacking Midfielder
Andros Townsend	Midfielder	Winger
Mark Noble	Midfielder	Defensive Midfielder
Aaron Cresswell	Defender	Full-back
James McArthur	Midfielder	Winger
Harry Maguire	Defender	Centre-back
Harry Kane	Forward	Central Striker
Jamal Lascelles	Defender	Centre-back
Jonathan Hogg	Midfielder	Defensive Midfielder
Kieran Trippier	Defender	Full-back
Tom Carroll	Midfielder	Defensive Midfielder
Lewis Dunk	Defender	Centre-back
Ashley Barnes	Forward	Central Striker
Chris Wood	Forward	Central Striker
Matt Ritchie	Midfielder	Winger
Steve Cook	Defender	Centre-back
Simon Francis	Defender	Centre-back
Charlie Daniels	Defender	Full-back
Joshua King	Forward	Central Striker
Raheem Shaquille Sterling	Forward	Winger Forward
Moussa Sidi Yaya Dembélé	Midfielder	Unknown
Jamie Vardy	Forward	Central Striker
Bamidele Alli	Midfielder	Central/Attacking Midfielder
Marko Arnautović	Forward	Central Striker
Antonio Rüdiger	Defender	Centre-back
Sead Kolašinac	Defender	Full-back
Joël Andre Job Matip	Defender	Centre-back
Heung-Min Son	Forward	Winger Forward
Christopher Schindler	Defender	Centre-back
Pascal Groß	Midfielder	Central/Attacking Midfielder
Roberto Firmino Barbosa de Oliveira	Forward	Central Striker
Emre Can	Midfielder	Central/Attacking Midfielder
Shkodran Mustafi	Defender	Centre-back
Dejan Lovren	Defender	Centre-back
Alexandre Lacazette	Forward	Central Striker
César Azpilicueta Tanco	Defender	Centre-back
Jordan Ayew	Forward	Winger Forward
Idrissa Gana Gueye	Midfielder	Defensive Midfielder
Eden Hazard	Forward	Winger Forward
Sadio Mané	Forward	Winger Forward
Kurt Happy Zouma	Defender	Centre-back

Continued on next page

Full Name	**Role Name**	**Sub Role**
Riyad Mahrez	Midfielder	Winger
Abdoulaye Doucouré	Midfielder	Defensive Midfielder
N'Golo Kanté	Midfielder	Defensive Midfielder
Kevin De Bruyne	Midfielder	Central/Attacking Midfielder
Christian Benteke Liolo	Forward	Central Striker
Xherdan Shaqiri	Midfielder	Winger
Granit Xhaka	Midfielder	Central/Attacking Midfielder
Mathias Jattah-Njie Jørgensen	Defender	Centre-back
Aaron Mooy	Midfielder	Central/Attacking Midfielder
Nicolás Hernán Otamendi	Defender	Centre-back
Nemanja Matić	Midfielder	Defensive Midfielder
Cédric Ricardo Alves Soares	Defender	Full-back
Joe Allen	Midfielder	Central/Attacking Midfielder
José Holebas	Defender	Full-back
Henrikh Mkhitaryan	Midfielder	Winger
Fernando Luiz Rosa	Midfielder	Central/Attacking Midfielder
Ahmed Hegazy	Defender	Centre-back
Mohamed Salah Ghaly	Forward	Winger Forward
Luka Milivojević	Midfielder	Central/Attacking Midfielder
Nathan Aké	Defender	Centre-back
Ben Davies	Defender	Full-back
Héctor Bellerín Moruno	Defender	Full-back
Eric Dier	Midfielder	Defensive Midfielder
Andrew Robertson	Defender	Full-back
Leroy Sané	Midfielder	Winger
Alfie Mawson	Defender	Centre-back
DeAndre Yedlin	Defender	Full-back
Davinson Sánchez Mina	Defender	Centre-back
Joe Gomez	Defender	Full-back
Onyinye Wilfred Ndidi	Midfielder	Defensive Midfielder
Wesley Hoedt	Defender	Centre-back
Lewis Cook	Midfielder	Defensive Midfielder
Gabriel Fernando de Jesus	Forward	Central Striker
Ben Chilwell	Defender	Full-back
Richarlison de Andrade	Forward	Winger Forward

Table A.3.1: List of Players and their Roles

A.4 Premier League Standings [34]

Table A.4.1: Premier League 2017/2018 Final Standings

Rank	Team
1	Manchester City
2	Manchester United
3	Tottenham Hotspur
4	Liverpool
5	Chelsea
6	Arsenal
7	Burnley
8	Everton
9	Leicester City
10	Newcastle United
11	Crystal Palace
12	Bournemouth
13	West Ham United
14	Watford
15	Brighton
16	Huddersfield
17	Southampton
18	Swansea City
19	Stoke City
20	West Bromwich Albion

A.5 Ballon d'or Ranking [35]

Rank	Player
1	Luka Modrić
2	Cristiano Ronaldo
3	Antoine Griezmann
4	Kylian Mbappé
5	Lionel Messi
6	Mohamed Salah
7	Raphaël Varane
8	Eden Hazard
9	Kevin De Bruyne
10	Harry Kane
11	N'Golo Kanté
12	Neymar
13	Luis Suárez
14	Thibaut Courtois
15	Paul Pogba
16	Sergio Agüero
17	Gareth Bale
18	Karim Benzema
19	Ivan Rakitić
19	Sergio Ramos
21	Edinson Cavani
22	Marcelo
23	Sadio Mané
24	Keylor Navas
25	Alisson Becker
25	Mario Mandžukić
25	Jan Oblak
25	Ivan Perišić

Table A.5.1: Ballon d'Or 2018 Final Rankings

A.6 Hyperparameter Search Spaces

The following table summarizes the hyperparameter grids used with `HalvingRandomSearchCV` for each model:

Model	Hyperparameter Grid
Logistic Regression	`penalty`: ['l1', 'l2'] `C`: [0.0001, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100] `solver`: ['liblinear', 'saga']
Random Forest	`n_estimators`: [100, 300, 500] `max_depth`: [None, 10, 20, 40] `min_samples_split`: [2, 5, 10] `min_samples_leaf`: [1, 2, 4] `max_features`: ['sqrt', 'log2', None]
XGBoost	`n_estimators`: [100, 300, 500] `max_depth`: [3, 6, 10] `learning_rate`: [0.01, 0.05, 0.1] `subsample`: [0.6, 0.8, 1] `colsample_bytree`: [0.6, 0.8, 1] `reg_alpha`: [0, 0.5, 1] `reg_lambda`: [0, 0.5, 1] `min_child_weight`: [1, 3, 5] `gamma`: [0, 1] `max_delta_step`: [0]
LightGBM	`n_estimators`: [100, 300, 500] `learning_rate`: [0.01, 0.05, 0.1] `max_depth`: [-1, 6, 10] `num_leaves`: [31, 63, 127] `min_data_in_leaf`: [20, 50, 100] `subsample`: [0.6, 0.8, 1.0] `colsample_bytree`: [0.6, 0.8, 1.0] `reg_alpha`: [0, 0.1, 1.0] `reg_lambda`: [0, 0.1, 1.0]
CatBoost	`iterations`: [100] `learning_rate`: [0.01, 0.05, 0.1] `depth`: [4, 6, 8] `l2_leaf_reg`: [1, 3, 5] `border_count`: [32, 64, 128] `bagging_temperature`: [0, 1, 5]

Table A.6.1: Hyperparameter search spaces for each model used in Experiment 2.

A.7 Best Defenders

The table below lists the top 10 defenders ranked by their average disruption ratio during the 2017/2018 Premier League season. This metric captures their defensive influence by quantifying how often they successfully interrupted opponent sequences that led to quality shot opportunities.

Player	Role	Avg. Disruption Ratio
M. Keane	Defender	0.1668
E. Pieters	Defender	0.1381
K. Gibbs	Defender	0.1223
A. Valencia	Defender	0.1044
J. Vertonghen	Defender	0.1024
V. van Dijk	Defender	0.0999
A. Young	Defender	0.0990
L. Koscielny	Defender	0.0969
R. Bertrand	Defender	0.0957
Nacho Monreal	Defender	0.0700

Table A.7.1: Top 10 defenders by average disruption ratio (2017/2018 EPL season).

A.8 Tags Combination Frequencies

Tag Combination	Count
[1801]	278049
[1802]	47568
[701, 1802]	44968
[703, 1801]	44579
[702, 1801]	32342
[]	61481
[1401]	12961
[1401, 1801]	11281
[502, 701, 1802]	5418
[501, 703, 1801]	5334
[901, 1802]	3959
[504, 701, 1802]	3706
[503, 703, 1801]	3688
[503, 701, 1802]	3516
[504, 703, 1801]	3425
[402, 801, 1802]	2344
[301, 402, 801, 1801]	356
[201, 401]	291
[301, 1801]	190
[502, 701, 1601, 1802]	159

Table A.8.1: Top 20 Tag Combinations by Frequency in the Dataset (Sorted by Count)

A.9 Full Player Composite Score Statistics

playerId	mean	std	range	shortName	fullName	role_name
3319	0.07	0.01	0.04	M. Özil	Mesut Özil	Midfielder
38021	0.07	0.01	0.04	K. De Bruyne	Kevin De Bruyne	Midfielder

Continued on next page

Table A.9.1 – continued from previous page

playerId	mean	std	range	shortName	fullName	role_name
8325	0.12	0.01	0.04	S. Agüero	Sergio Leonel Agüero del Castillo	Forward
38031	0.08	0.01	0.03	C. Benteke	Christian Benteke Liolo	Forward
8317	0.06	0.01	0.04	David Silva	David Josué Jiménez Silva	Midfielder
25854	0.05	0.01	0.03	K. Zouma	Kurt Happy Zouma	Defender
7919	0.04	0.01	0.03	M. Keane	Michael Keane	Defender
8422	0.10	0.01	0.03	W. Zaha	Wilfried Zaha	Midfielder
119951	0.05	0.01	0.03	Ahmed Hegazy	Ahmed Hegazy	Defender
9637	0.07	0.01	0.03	J. King	Joshua King	Forward
245364	0.08	0.01	0.04	L. Sané	Leroy Sané	Midfielder
14911	0.09	0.01	0.04	Son Heung-Min	Heung-Min Son	Forward
25413	0.10	0.01	0.04	A. Lacazette	Alexandre Lacazette	Forward
120353	0.12	0.01	0.03	Mohamed Salah	Mohamed Salah Ghaly	Forward
8653	0.05	0.01	0.03	H. Maguire	Harry Maguire	Defender
105339	0.04	0.01	0.03	Fernandinho	Fernando Luiz Rosa	Midfielder
11066	0.09	0.01	0.03	R. Sterling	Raheem Shaquille Sterling	Forward
25571	0.07	0.01	0.03	J. Ayew	Jordan Ayew	Forward
3350	0.06	0.01	0.03	Fàbregas	Francesc Fàbregas i Soler	Midfielder
49876	0.05	0.01	0.03	G. Xhaka	Granit Xhaka	Midfielder
8416	0.06	0.01	0.02	G. Murray	Glenn Murray	Forward
54	0.07	0.01	0.03	C. Eriksen	Christian Dannemann Eriksen	Midfielder
222220	0.05	0.01	0.03	A. Robertson	Andrew Robertson	Defender
77548	0.05	0.01	0.03	J. Allen	Joe Allen	Midfielder
15808	0.08	0.01	0.03	Roberto Firmino	Roberto Firmino Barbosa de Oliveira	Forward
340386	0.09	0.01	0.03	Gabriel Jesus	Gabriel Fernando de Jesus	Forward
265366	0.04	0.01	0.03	W. Ndidi	Onyinye Wilfred Ndidi	Midfielder
25393	0.04	0.01	0.03	D. Lovren	Dejan Lovren	Defender
15215	0.04	0.01	0.02	C. Schindler	Christopher Schindler	Defender
8464	0.03	0.01	0.03	F. Delph	Fabian Delph	Midfielder
9097	0.05	0.01	0.03	L. Dunk	Lewis Dunk	Defender
8313	0.04	0.01	0.03	P. Zabaleta	Pablo Javier Zabaleta Girod	Defender
9123	0.06	0.01	0.02	A. Barnes	Ashley Barnes	Forward
70086	0.04	0.01	0.03	N. Otamendi	Nicolás Hernán Otamendi	Defender
18550	0.05	0.01	0.03	E. Can	Emre Can	Midfielder
9227	0.07	0.01	0.03	M. Ritchie	Matt Ritchie	Midfielder
8833	0.05	0.01	0.02	J. Lascelles	Jamal Lascelles	Defender
25707	0.09	0.01	0.03	E. Hazard	Eden Hazard	Forward
25747	0.08	0.01	0.03	S. Mané	Sadio Mané	Forward
7855	0.04	0.01	0.03	L. Koscielny	Laurent Koscielny	Defender
268776	0.04	0.01	0.02	W. Hoedt	Wesley Hoedt	Defender
9277	0.04	0.01	0.02	S. Cook	Steve Cook	Defender
3577	0.07	0.01	0.02	S. Rondón	José Salomón Rondón Giménez	Forward
8277	0.03	0.01	0.03	K. Walker	Kyle Walker	Defender
14870	0.04	0.01	0.03	J. Matip	Joël Andre Job Matip	Defender
7899	0.05	0.01	0.02	Oriol Romeu	Oriol Romeu Vidal	Midfielder
127537	0.04	0.01	0.02	L. Milivojević	Luka Milivojević	Midfielder
167145	0.04	0.01	0.03	Bellerín	Héctor Bellerín Moruno	Defender
13484	0.07	0.01	0.02	D. Alli	Bamidele Alli	Midfielder
8242	0.05	0.01	0.02	S. Duffy	Shane Duffy	Defender
11152	0.04	0.01	0.03	M. Dembélé	Moussa Sidi Yaya Dembélé	Midfielder
7936	0.06	0.01	0.03	P. Pogba	Paul Pogba	Midfielder

Continued on next page

Table A.9.1 – continued from previous page

playerId	mean	std	range	shortName	fullName	role_name
7905	0.09	0.01	0.02	R. Lukaku	Romelu Lukaku Menama	Forward
8319	0.05	0.01	0.03	J. Milner	James Milner	Midfielder
14869	0.04	0.01	0.03	S. Kolašinac	Sead Kolašinac	Defender
302518	0.04	0.01	0.02	L. Cook	Lewis Cook	Midfielder
8013	0.05	0.01	0.02	M. Albrighton	Marc Albrighton	Midfielder
31528	0.04	0.01	0.03	N. Kanté	N'Golo Kanté	Midfielder
377071	0.08	0.01	0.02	Richarlison	Richarlison de Andrade	Forward
7964	0.04	0.01	0.03	J. Henderson	Jordan Brian Henderson	Midfielder
9206	0.07	0.01	0.02	C. Wood	Chris Wood	Forward
136441	0.04	0.01	0.03	B. Davies	Ben Davies	Defender
3560	0.04	0.01	0.02	Nacho Monreal	Ignacio Monreal Eraso	Defender
8336	0.03	0.01	0.02	K. Naughton	Kyle Naughton	Defender
15526	0.06	0.01	0.02	P. Groß	Pascal Groß	Midfielder
14748	0.03	0.01	0.02	A. Rüdiger	Antonio Rüdiger	Defender
246866	0.04	0.01	0.02	A. Mawson	Alfie Mawson	Defender
3361	0.09	0.01	0.02	A. Sánchez	Alexis Alejandro Sánchez Sánchez	Forward
20612	0.04	0.01	0.02	S. Mustafi	Shkodran Mustafi	Defender
383	0.07	0.01	0.03	D. Tadić	Dušan Tadić	Midfielder
55979	0.04	0.01	0.02	M. Jørgensen	Mathias Jattah-Njie Jørgensen	Defender
8945	0.05	0.01	0.02	K. Trippier	Kieran Trippier	Defender
48	0.04	0.01	0.02	J. Vertonghen	Jan Vertonghen	Defender
107	0.04	0.01	0.02	E. Pieters	Erik Pieters	Defender
210044	0.04	0.01	0.02	E. Dier	Eric Dier	Midfielder
8192	0.03	0.01	0.02	M. Olsson	Martin Olsson	Defender
8717	0.13	0.01	0.02	H. Kane	Harry Kane	Forward
8471	0.08	0.01	0.02	A. Townsend	Andros Townsend	Midfielder
370	0.05	0.01	0.02	V. van Dijk	Virgil van Dijk	Defender
257762	0.03	0.01	0.02	D. Sánchez	Davinson Sánchez Mina	Defender
7967	0.06	0.01	0.02	J. Shelvey	Jonjo Shelvey	Midfielder
105338	0.07	0.01	0.02	H. Mkhitaryan	Henrikh Mkhitaryan	Midfielder
70389	0.04	0.01	0.02	Cédric Soares	Cédric Ricardo Alves Soares	Defender
25553	0.04	0.01	0.02	Azpilicueta	César Azpilicueta Tanco	Defender
9279	0.04	0.01	0.02	S. Francis	Simon Francis	Defender
70122	0.04	0.01	0.02	N. Matić	Nemanja Matić	Midfielder
3324	0.10	0.01	0.02	Álvaro Morata	Álvaro Borja Morata Martín	Forward
8032	0.06	0.01	0.02	Marcos Alonso	Marcos Alonso Mendoza	Defender
257899	0.04	0.01	0.02	J. Gomez	Joe Gomez	Defender
8623	0.05	0.01	0.02	J. McArthur	James McArthur	Midfielder
28292	0.05	0.01	0.02	A. Doucouré	Abdoulaye Doucouré	Midfielder
8897	0.04	0.01	0.02	J. Hogg	Jonathan Hogg	Midfielder
8561	0.04	0.01	0.02	M. Noble	Mark Noble	Midfielder
466	0.05	0.01	0.02	D. Pröpper	Davy Pröpper	Midfielder
252365	0.04	0.01	0.02	D. Yedlin	DeAndre Yedlin	Defender
12829	0.09	0.01	0.02	J. Vardy	Jamie Vardy	Forward
25706	0.04	0.01	0.02	I. Gueye	Idrissa Gana Gueye	Midfielder
7938	0.04	0.00	0.02	A. Valencia	Luis Antonio Valencia Mosquera	Defender
8086	0.04	0.00	0.02	D. Stephens	Dale Stephens	Midfielder
8125	0.04	0.00	0.02	J. Cork	Jack Cork	Midfielder
26150	0.07	0.00	0.02	R. Mahrez	Riyad Mahrez	Midfielder
93	0.08	0.00	0.02	J. Guðmundsson	Johann Berg Guðmundsson	Midfielder

Continued on next page

Table A.9.1 – continued from previous page

playerId	mean	std	range	shortName	fullName	role_name
7885	0.04	0.00	0.02	R. Bertrand	Ryan Bertrand	Defender
7939	0.04	0.00	0.02	A. Young	Ashley Young	Defender
8370	0.05	0.00	0.02	C. Brunt	Chris Brunt	Midfielder
9285	0.04	0.00	0.02	C. Daniels	Charlie Daniels	Defender
62389	0.04	0.00	0.02	A. Mooy	Aaron Mooy	Midfielder
350976	0.04	0.00	0.02	B. Chilwell	Ben Chilwell	Defender
49872	0.07	0.00	0.02	X. Shaqiri	Xherdan Shaqiri	Midfielder
7853	0.04	0.00	0.01	K. Gibbs	Kieran Gibbs	Defender
7988	0.05	0.00	0.02	Y. Cabaye	Yohan Cabaye	Midfielder
8582	0.04	0.00	0.01	A. Cresswell	Aaron Cresswell	Defender
14703	0.08	0.00	0.01	M. Arnautović	Marko Arnautović	Forward
92899	0.04	0.00	0.01	J. Holebas	José Holebas	Defender
8976	0.04	0.00	0.01	T. Carroll	Tom Carroll	Midfielder
134102	0.04	0.00	0.01	N. Aké	Nathan Aké	Defender

Table A.9.1: Detailed composite score statistics for all players. The table reports the mean, standard deviation, and range of the composite metric, along with player identity and role.

A.10 Missingness

Feature	**Missing Values**	**Missing (%)**
prev_x	300	0.0591
prev_y	300	0.0591
Total	**1140**	**0.0015% (global)**

Table A.10.1: Overview of missingness in `data_exp2`. Missing values are rare and confined to three temporally sensitive features.

A.11 Features description

Feature	Description
role_code	Encoded player role (e.g., defender, midfielder, forward)
team_game_state	Team's current match state (winning, drawing, losing)
is_team_losing_before_event	Was the team losing just before the event?
is_clutch_moment	High-pressure moment (e.g., late game, crucial situation)
distance_to_goal	Distance from the ball to the goal at the moment of the event
angle_to_goal	Available shooting angle toward the goal
eventSec	Game time (in seconds) at which the event occurred
coordinates_x	X position of the action on the field
coordinates_y	Y position of the action on the field
end_coordinates_x	X position where the action ends (e.g., pass target)
end_coordinates_y	Y position where the action ends
time_since_last_action	Time since the previous recorded action
possession_change	Did possession change just before this action?
time_since_possession_change	Time since the last change in possession
time_since_last_shot	Time since the last shot attempt
prev_x	X position of the previous action
prev_y	Y position of the previous action
distance_covered	Distance moved since the previous action
avg_team_x	Average X position of the player's teammates
avg_team_y	Average Y position of the player's teammates
num_passes_last_5	Number of passes in the last 5 team actions
num_dribbles_last_5	Number of dribbles in the last 5 team actions
score_difference_before_event	Raw score difference before the action
score_diff_from_focal_team	Score difference from the player's team perspective
delta_x	Change in X between current and previous action
is_shot	Whether the event is a shot
is_pass	Whether the event is a pass
is_dribble	Whether the event is a dribble
same_team_as_prev	Is the actor on the same team as in the previous event?
is_in_final_third	Is the event located in the final third of the pitch?
is_central_channel	Is the event located in the central zone?
is_defensive_player	Is the player typically in a defensive role?

Table A.11.1: Description of contextual and spatio-temporal features used for clustering and prediction.

A.12 Gain importances

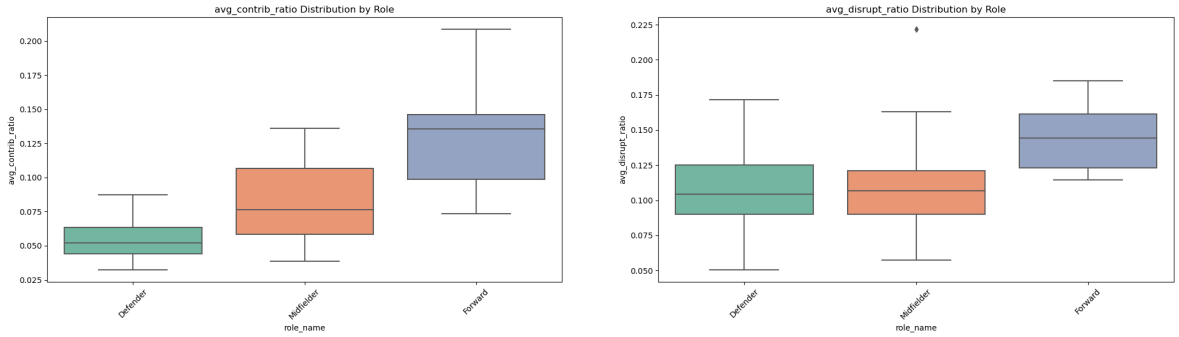
Rank	Feature	Gain Importance
1	num__angle_to_goal	134,342.73
2	num__distance_to_goal	109,867.75
3	num__end_coordinates_x	32,164.74
4	num__end_coordinates_y	31,975.42
5	num__time_since_last_shot	24,221.97
6	num__eventSec	20,287.65
7	num__coordinates_x	18,255.59
8	num__coordinates_y	13,769.66
9	num__avg_team_x	11,999.68
10	num__avg_team_y	10,837.30

Table A.12.1: Top 10 features ranked by gain-based importance in the best LightGBM model (KNN imputation, chronological split).

Appendix B

Additional Figures

B.1 Role-Based Score Distributions

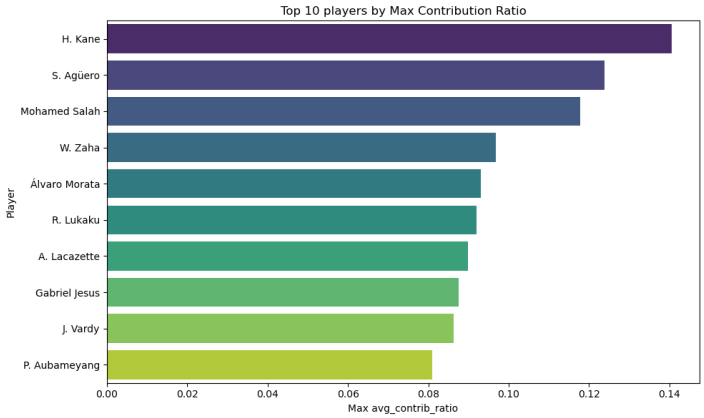


(a) Average contribution ratio by role

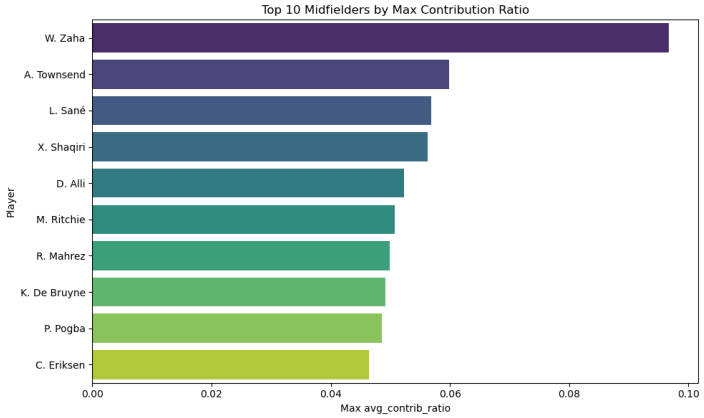
(b) Average disruption ratio by role

Figure B.1.1: Distribution of average contribution and disruption ratios (`avg_contrib_ratio` and `avg_disrupt_ratio`) across player roles. These metrics reflect different tactical responsibilities.

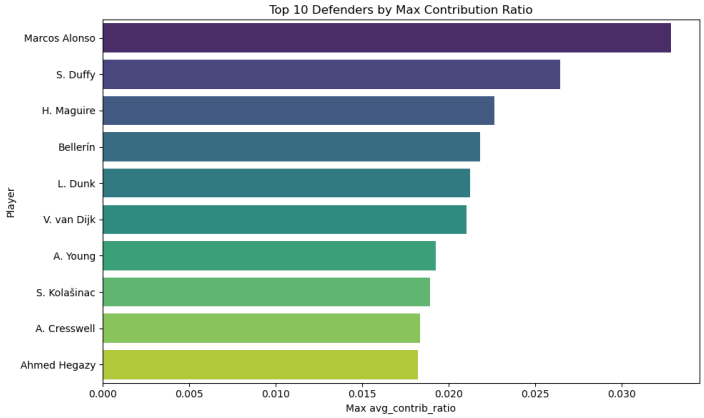
B.2 Top Players Composite Score by Role



(a) Top 10 Forwards by Max Contribution Ratio



(b) Top 10 Midfielders by Max Contribution Ratio



(c) Top 10 Defenders by Max Contribution Ratio

Figure B.2.1: Top 10 players by Max Contribution Ratio, grouped by role.

B.3 Van Dijk composite score by window

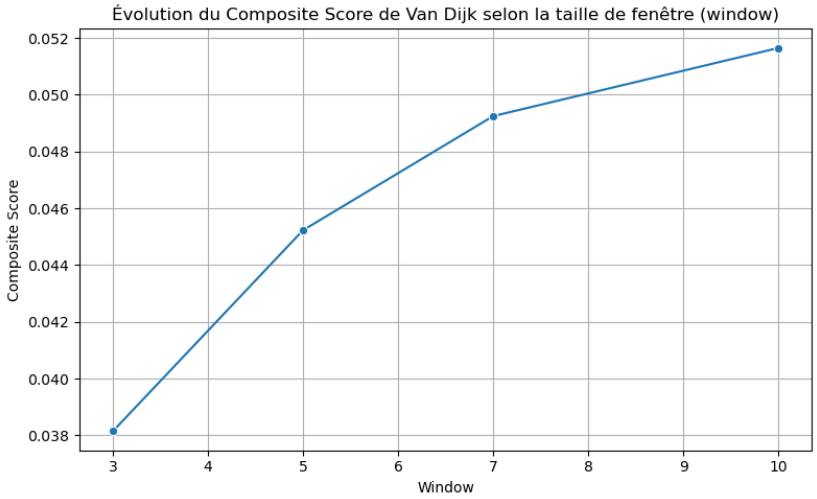


Figure B.3.1: Composite score of Van Dijk across window sizes. Slight increase but remains consistently lower than offensive profiles.

B.4 Top Centre-Backs by Disruptive Defensive Actions

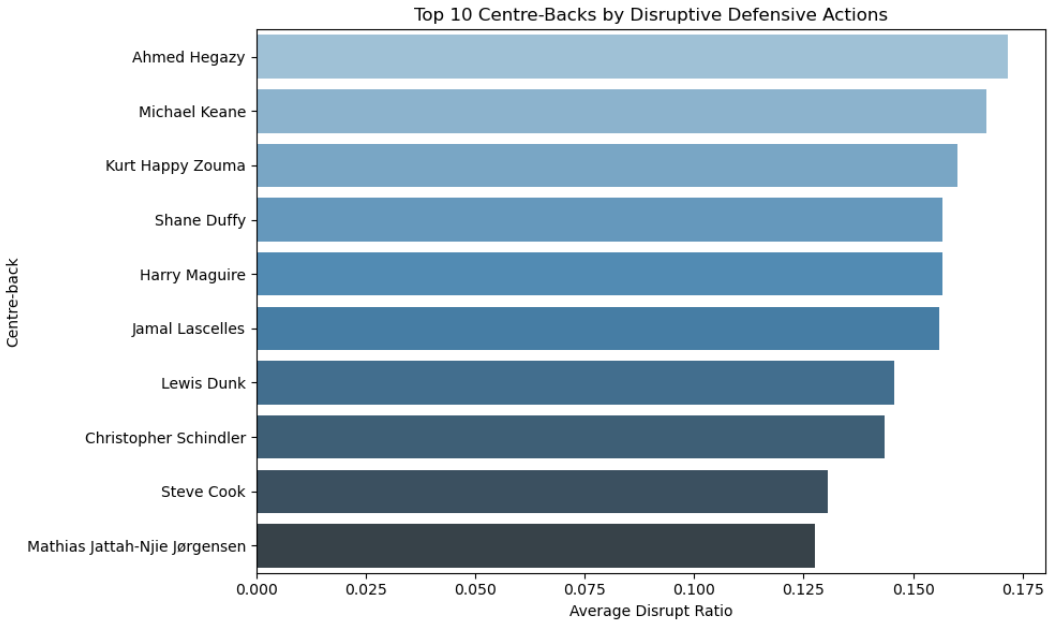


Figure B.4.1: Top 10 centre-backs based on average disruptive defensive action ratio (`avg_disrupt_ratio`).

Appendix C

Additional Statistical Analyses

C.1 Global Performance Summary

Model	F1	F1 Std	Precision	Prec. Std	Recall	Recall Std	ROC AUC	ROC Std	PR AUC	PR Std	Accuracy	Acc. Std	Fit Time (s)	CV-F1 Mean
CatBoost	0.4037	0.0021	0.9897	0.0026	0.2536	0.0018	0.8245	0.0022	0.4806	0.0040	0.9629	0.0010	873.17	0.3812
LightGBM	0.4141	0.0035	0.9595	0.0134	0.2640	0.0036	0.8307	0.0098	0.4891	0.0127	0.9630	0.0009	440.64	0.3935
LogisticRegression	0.0832	0.0041	0.8143	0.0291	0.0439	0.0024	0.7851	0.0016	0.3127	0.0065	0.9521	0.0013	802.62	0.1213
RandomForest	0.4176	0.0014	0.9463	0.0089	0.2680	0.0017	0.8097	0.0072	0.4706	0.0063	0.9630	0.0009	3917.13	0.3872
XGBoost	0.4104	0.0053	0.9568	0.0200	0.2613	0.0056	0.8272	0.0088	0.4823	0.0100	0.9628	0.0010	415.58	0.3943

Table C.1.1: Summary of average performance metrics across classifiers (aggregated over all configurations). Includes F1-score, precision, recall, ROC-AUC, PR-AUC, and training time.

C.2 F1-score

Table C.2.1: Shapiro-Wilk normality test applied to F1-scores per model. Significant p-values ($p < 0.05$) indicate a deviation from normality. Logistic Regression, CatBoost, and Random Forest fail the normality test, justifying the use of non-parametric methods.

Model	W-statistic	p-value	Normal distribution
Logistic Regression	0.650	0.0003	No
Random Forest	0.856	0.0433	No
XGBoost	0.878	0.0831	Yes (approx.)
LightGBM	0.930	0.3842	Yes
CatBoost	0.772	0.0046	No

Table C.2.2: Adjusted p-values from Dunn's post-hoc test (Holm correction) comparing F1-scores between models. Values below 0.05 (see diagonal) indicate statistically significant differences in F1-score distribution. This supports the superiority of ensemble methods like Random Forest and LightGBM.

Model	CatBoost	LightGBM	LogReg	RF	XGBoost
CatBoost	1.0000	0.0240	0.1920	0.0001	0.1920
LightGBM	0.0240	1.0000	0.0000	0.2078	0.3676
Logistic Regression	0.1920	0.0000	1.0000	0.0000	0.0008
Random Forest	0.0001	0.2078	0.0000	1.0000	0.0575
XGBoost	0.1920	0.3676	0.0008	0.0575	1.0000

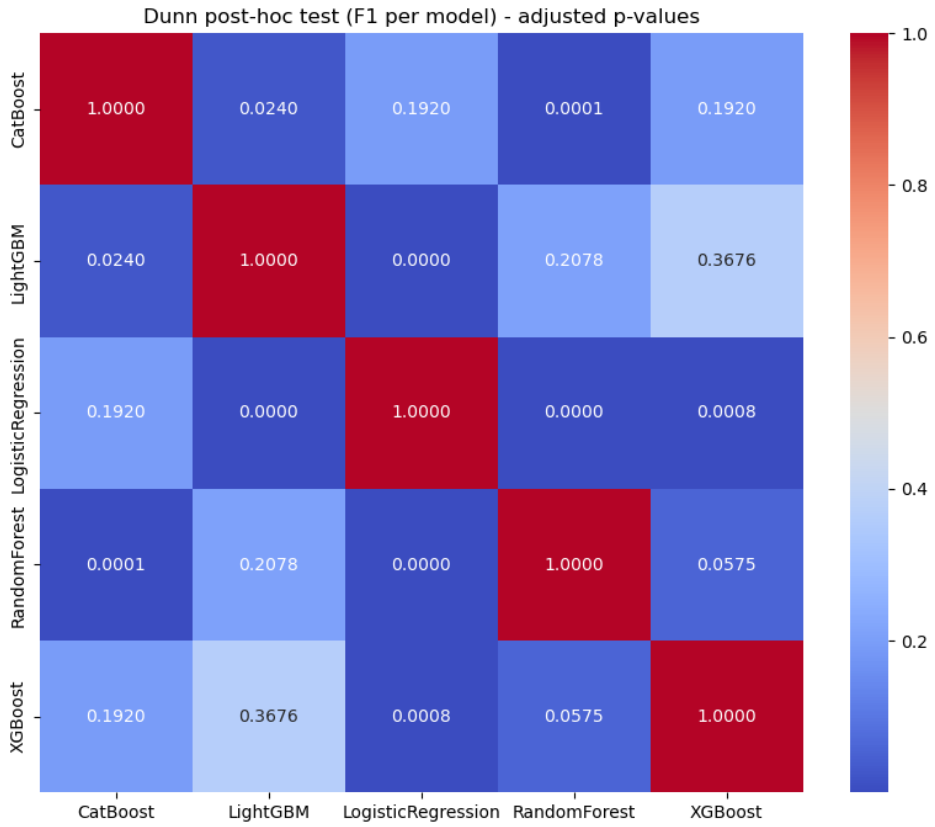


Figure C.2.1: Heatmap of adjusted p-values from Dunn's post-hoc test comparing F1-scores between models. Dark cells (values < 0.05) indicate statistically significant differences. Random Forest significantly outperforms Logistic Regression and CatBoost.

C.3 Boxplot AUC

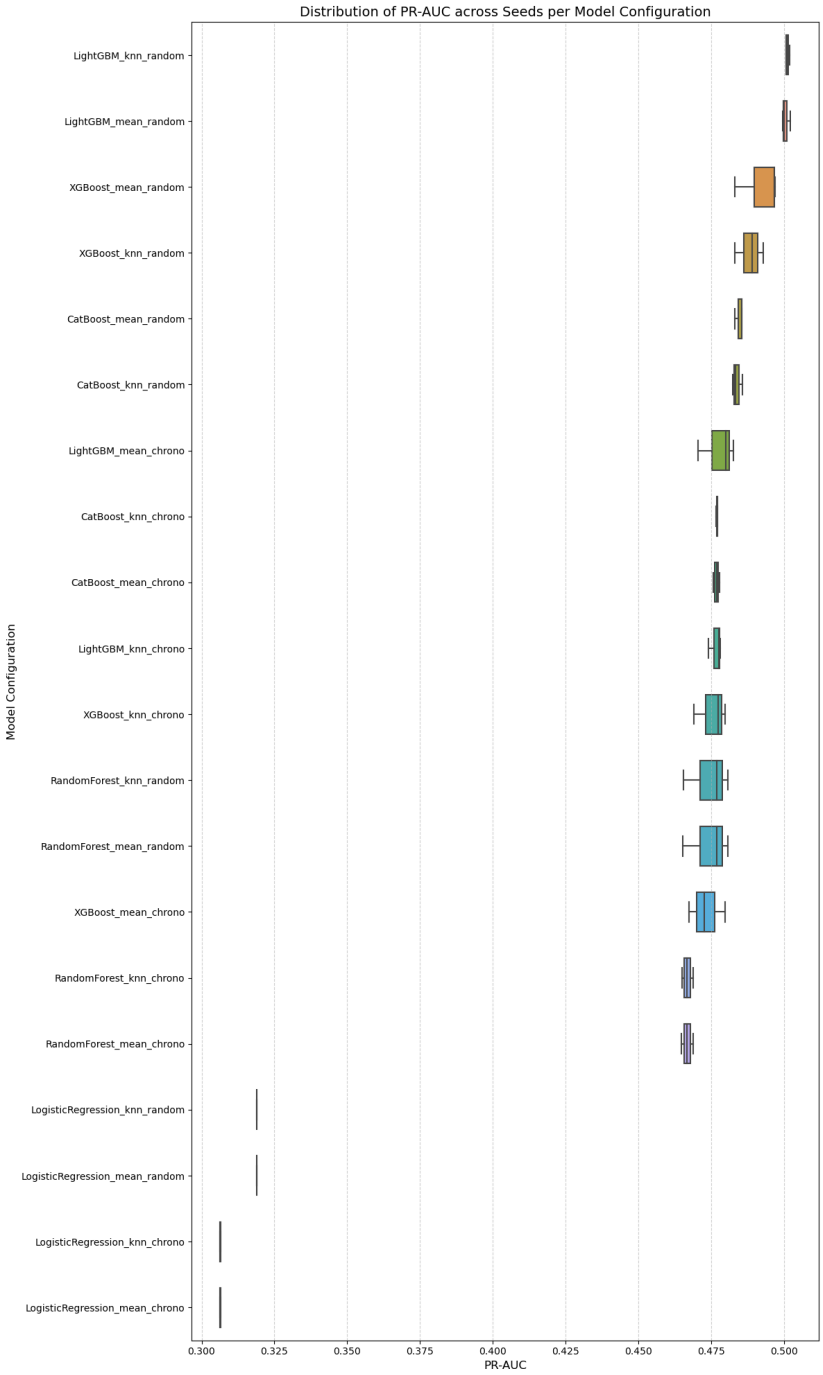
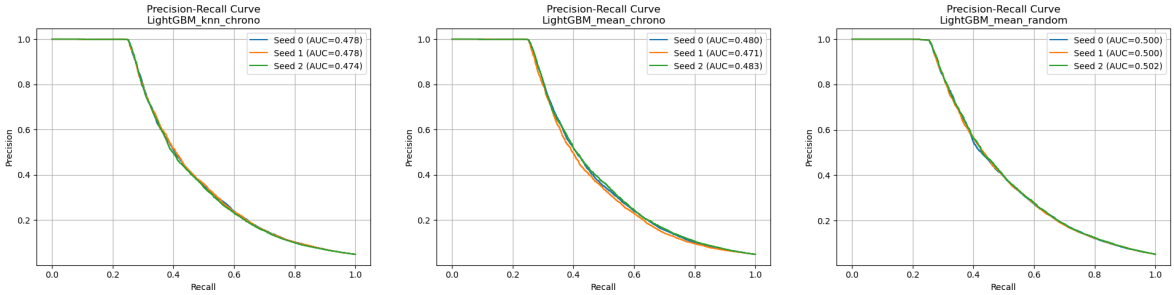


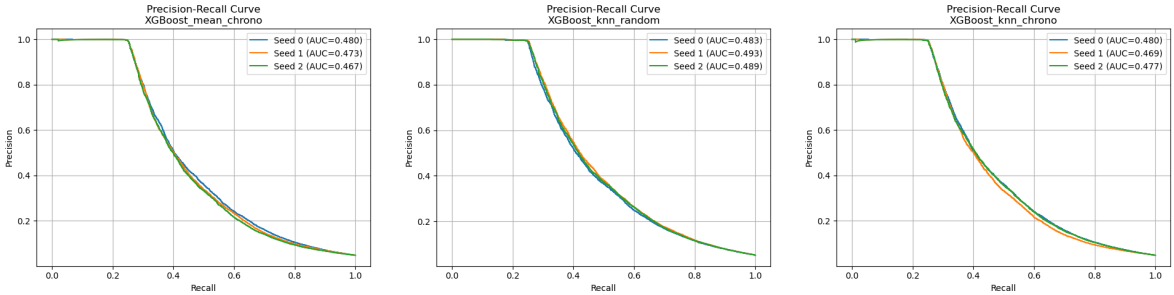
Figure C.3.1: Distribution of PR-AUC scores across seeds for different models.

C.4 Precision-Recall curves other models



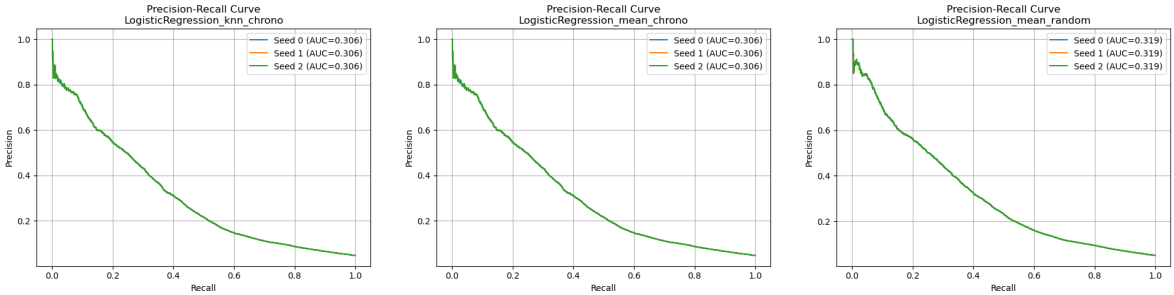
(a) `LightGBM_knn_chrono` (b) `LightGBM_mean_chrono` (c) `LightGBM_mean_random`

Figure C.4.1: Grouped Precision-Recall curves for other LGBM configuration



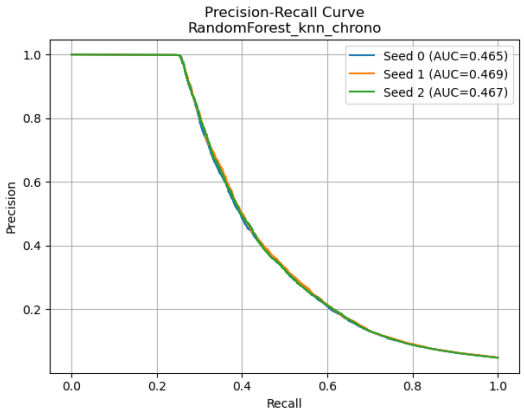
(a) `XGBoost_mean_chrono` (b) `XGBoost_knn_random` (c) `XGBoost_knn_chrono`

Figure C.4.2: Grouped Precision-Recall curves for other XGBoost configuration

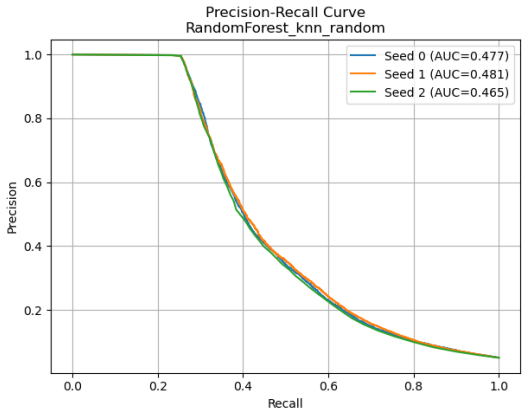


(a) `LogisticRegression_knn_chrono` (b) `LogisticRegression_mean_chrono` (c) `LogisticRegression_mean_random`

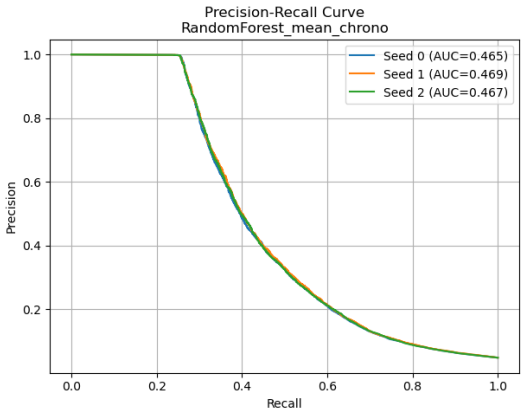
Figure C.4.3: Grouped Precision-Recall curves for Logistic Regression configurations



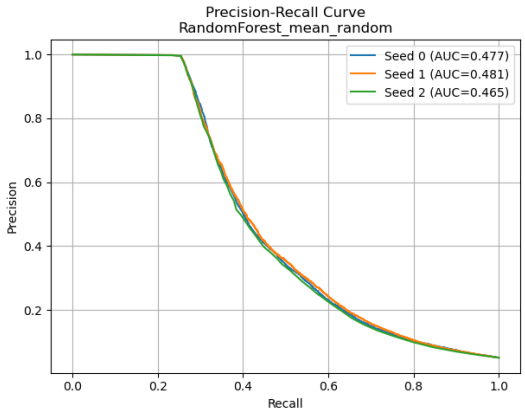
(a) `RandomForest_knn_chrono`



(b) `RandomForest_knn_random`

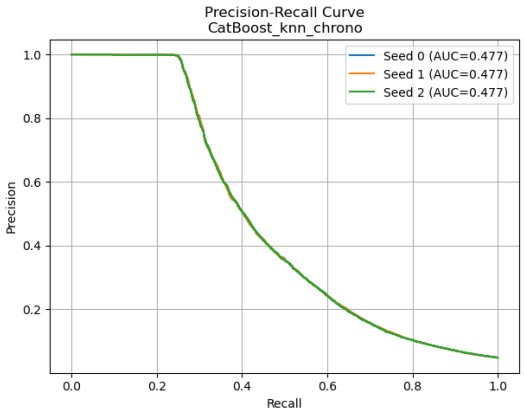


(c) `RandomForest_mean_chrono`

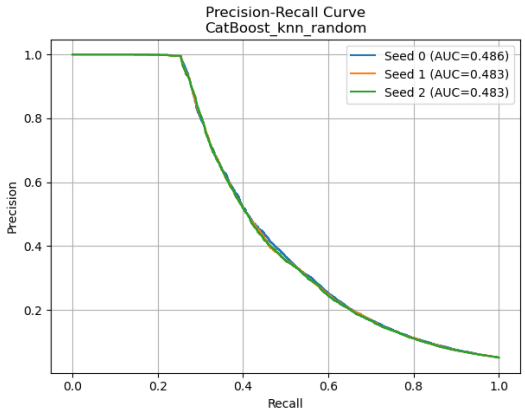


(d) `RandomForest_mean_random`

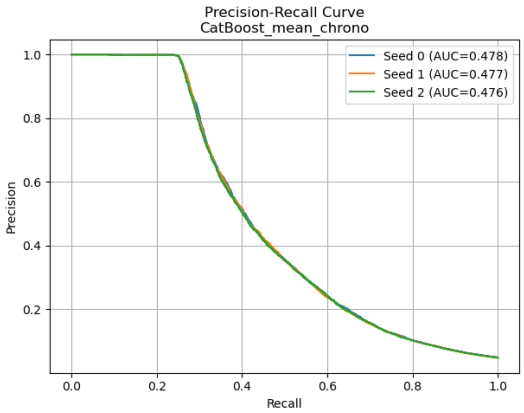
Figure C.4.4: Grouped Precision-Recall curves for Random Forest configurations



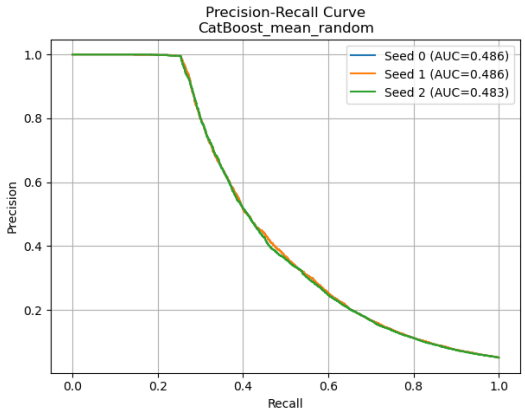
(a) `CatBoost_knn_chrono`



(b) `CatBoost_knn_random`



(c) `CatBoost_mean_chrono`



(d) `CatBoost_mean_random`

Figure C.4.5: Grouped Precision-Recall curves for CatBoost configurations

C.5 Split distrisbutions

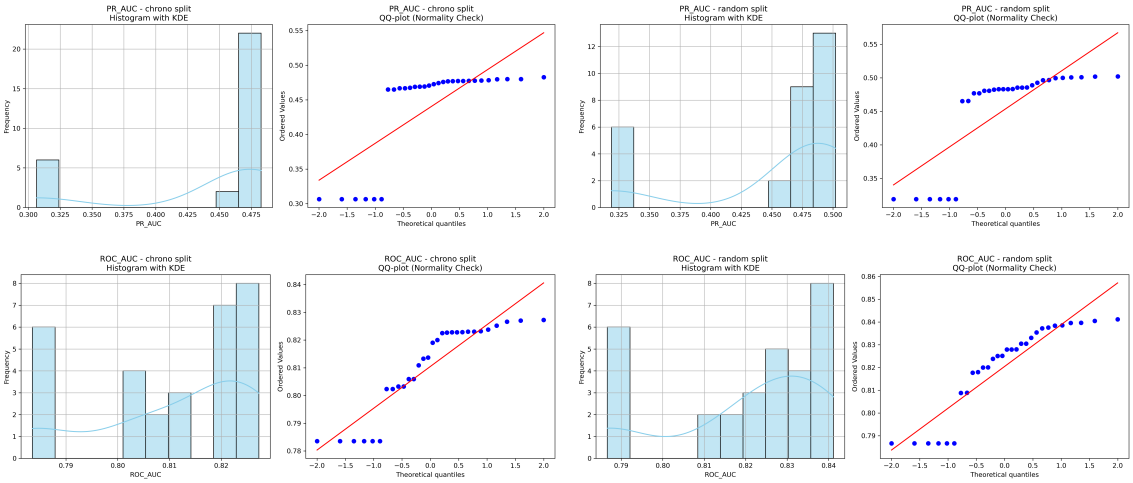


Figure C.5.1: Histograms and Q-Q plots for PR-AUC and ROC-AUC across split strategies: chrono (left) and random (right). All four distributions exhibit significant deviations from normality.

C.6 Appendix: DBSCAN Clustering

Figure C.6.1 shows the results of DBSCAN clustering on PCA-reduced action features. While DBSCAN identifies a dominant central cluster, it fails to clearly separate meaningful behavioral patterns compared to HDBSCAN.

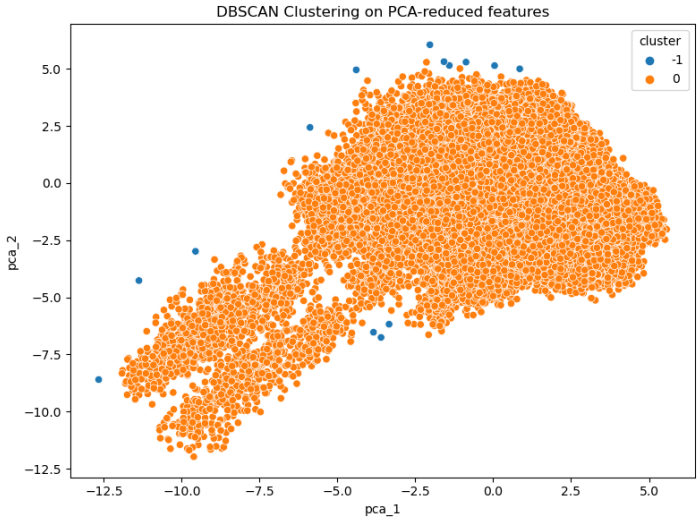


Figure C.6.1: DBSCAN clustering on PCA-reduced features. Most points belong to a large central cluster.

C.7 Appendix: Detailed Composition of HDBSCAN Clusters

Figure C.7.1 shows the proportion of each event type (e.g., pass, shot, duel) across clusters. As expected, cluster 0 includes over 50% of all passes, while clusters 1–5 are almost exclusively shots. Cluster 3 stands out for grouping only highly probable shots close to goal.

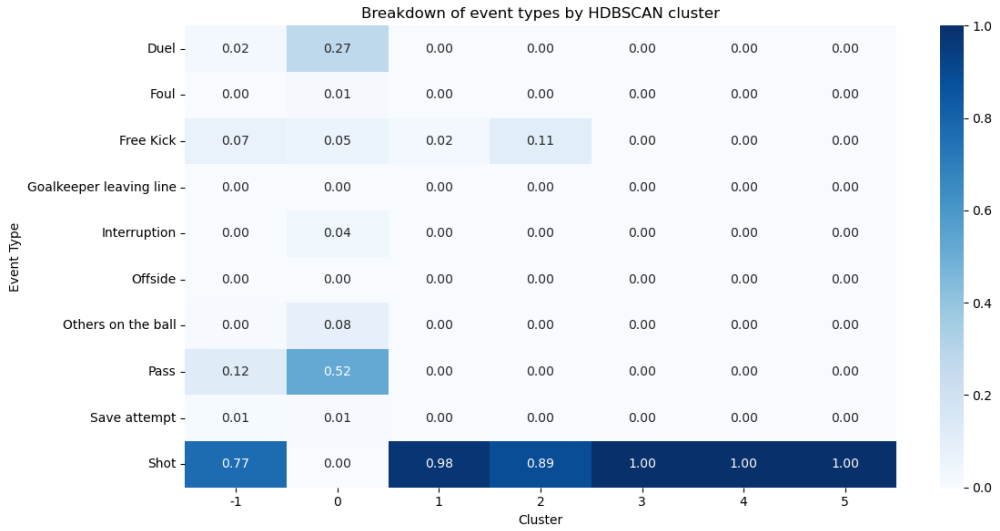


Figure C.7.1: Breakdown of event types by HDBSCAN cluster (normalized per cluster).

Appendix D

Full Code

The full project code is publicly available on the following GitHub repository: `https://github.com/loic-mnt/LDATS2840MasterThesis.git`

Key extracts are provided in this appendix to illustrate the main functions used.

D.1 flag_contributing_actions_advanced

```python
def flag_contributing_actions_advanced(
    df,
    window=3,
    possession_threshold=2
):
    """
    Flags contributions to quality shots using a fixed number of same-team actions (window),
    and flags defensive disruptions using backward possession-aware logic.

    Parameters:
        df (pd.DataFrame): Wyscout event data with 'is_quality_shot' and 'is_key_action'.
        window (int): Max number of same-team actions to evaluate (forward).
        possession_threshold (int): Max number of opponent actions before breaking sequence.

    Returns:
        pd.DataFrame with:
            - contributes_to_quality (1/0)
            - steps_to_shot (int)
            - disrupts_opponent_quality (1/0)
    """
    df = df.copy()
    df['contributes_to_quality'] = 0
    df['steps_to_shot'] = np.nan
    df['disrupts_opponent_quality'] = 0

    df = df.sort_values(['matchId', 'eventSec']).reset_index(drop=True)

    for match_id in df['matchId'].unique():
        match_df = df[df['matchId'] == match_id]
        match_indices = match_df.index.tolist()
        team_ids = match_df['teamId'].unique()
```

```
        if len(team_ids) < 2:
            continue

        for idx in match_indices:
            current_team = df.at[idx, 'teamId']
            same_team_count = 0
            opponent_count = 0

            # ---------- Forward: Offensive Contribution ----------
            for step in range(1, len(df) - idx):
                next_idx = idx + step
                if next_idx >= len(df):
                    break
                if df.at[next_idx, 'matchId'] != match_id:
                    break

                next_team = df.at[next_idx, 'teamId']
                if next_team == current_team:
                    same_team_count += 1
                    if df.at[next_idx, 'is_quality_shot'] == 1:
                        df.at[idx, 'contributes_to_quality'] = 1
                        df.at[idx, 'steps_to_shot'] = step
                        break
                    if same_team_count >= window:
                        break
                else:
                    opponent_count += 1
                    if opponent_count >= possession_threshold:
                        break

            # Always mark the shot itself
            if df.at[idx, 'is_quality_shot'] == 1:
                df.at[idx, 'contributes_to_quality'] = 1
                df.at[idx, 'steps_to_shot'] = 0

            # ---------- Backward: Defensive Disruption ----------
            if df.at[idx, 'is_key_action'] == 1:
                opponent_team = [t for t in team_ids if t != current_team][0]
                opponent_sequence = 0
                for step in range(1, idx + 1):
                    prev_idx = idx - step
                    if prev_idx < 0 or df.at[prev_idx, 'matchId'] != match_id:
                        break
                    if df.at[prev_idx, 'teamId'] != opponent_team:
                        break
                    opponent_sequence += 1
                    if df.at[prev_idx, 'is_quality_shot'] == 1:
                        break
                if opponent_sequence >= possession_threshold:
                    df.at[idx, 'disrupts_opponent_quality'] = 1

    return df
```

Listing D.1: Fonction `flag_contributing_actions_advanced`: identifying offensive contributions and defensive disruptions

D.2 add_composite_score

```python
def add_composite_score(df, alpha=0.7):
    df = df.copy()
    if 'avg_contrib_ratio' in df.columns and 'avg_disrupt_ratio' in df.columns:
        df['composite_score'] = (
            alpha * df['avg_contrib_ratio'] +
            (1 - alpha) * df['avg_disrupt_ratio']
        )
    else:
        raise ValueError("Both avg_contrib_ratio and avg_disrupt_ratio must be in the DataFrame.")
    return df
```

Listing D.2: Function `add_composite_score`: computation of a weighted performance score combining offensive and defensive contributions

D.3 Experiment 1 code

```python
def run_single_experiment(df, tracked_players, params):
    # Label generation
    compute_shot_fn = make_quality_shot_fn(
        distance_threshold=params['distance'],
        angle_threshold=params['angle'],
        free_kick_distance_threshold=params['fk_dist'],
        tags_required=params['quality_tags']
    )
    df = compute_shot_fn(df)
    df = compute_is_key_action(df)
    df = flag_contributing_actions_advanced(
        df,
        window=params['window'],
        possession_threshold=params['possession']
    )

    # Contribution stats
    stats, df_game_ratios = compute_player_contrib_stats(df)
    stats = stats[stats['playerId'].isin(tracked_players)]
    df_game_ratios = df_game_ratios[df_game_ratios['playerId'].isin(
        tracked_players)]

    # Add hyperparameter metadata
    for k, v in params.items():
        stats[k] = str(v) if isinstance(v, list) else v
        df_game_ratios[k] = str(v) if isinstance(v, list) else v

    return stats, df_game_ratios
```

Listing D.3: Function `run_single_experiment`: applies a parameterized labeling and computes contribution statistics

```python
def run_experiment_grid(df, tracked_players, param_grid, results_dir):
    os.makedirs(results_dir, exist_ok=True)
```

```
3
4     partial_path = os.path.join(results_dir, "experiment1_results_partial_opt.
          csv")
5     final_path = os.path.join(results_dir, "experiment1_results_opt.csv")
6
7     results = []
8     is_first_write = True
9
10    for params in tqdm(param_grid, desc="Running experiment grid"):
11        try:
12            result = run_single_experiment(df.copy(), tracked_players, params)
13            results.append(result)
14
15            # Append to partial CSV (ensures headers only once)
16            result.to_csv(
17                partial_path,
18                mode='a',
19                header=is_first_write,
20                index=False
21            )
22            is_first_write = False
23
24        except Exception as e:
25            print(f"Skipping param set {params} due to error: {e}")
26            continue
27
28    if results:
29        final_df = pd.concat(results, ignore_index=True)
30        final_df.to_csv(final_path, index=False)
31        return final_df
32    else:
33        print(" No results generated. Check your experiment logic.")
34        return pd.DataFrame()
```

Listing D.4: Function `run_experiment_grid`: executes a full grid search across labeling configurations and logs partial results

```
# Param Grid Setup
distances = [16, 18, 20]
angles = [0.6]
fk_distances = [25, 30]
windows = [3, 5, 7, 10]
possessions = [1, 2, 3]

quality_tag_sets = [
    ['tag_201'],                          # Big Chance
    ['tag_101'],                          # Goal
    ['tag_201', 'tag_101'],              # Big Chance + Goal
]

key_tag_sets = [
    ['tag_301'],                          # Assist
    ['tag_302'],                          # Key Pass
    ['tag_703'],                          # Duel Won
]
```

Listing D.5: Parameter grid for contribution labeling configurations explored in Experiment 1

D.4 Experiment 2 code

```
def log_step(message):
    print(f"[{datetime.now().strftime('%Y-%m-%d %H:%M:%S')}] {message}", flush
        =True)

def save_model_to_gcs(model, model_name, bucket_path="gs://thesis-loic-data/
    Models/"):
    tmp = os.path.join(local_path, model_name)
    joblib.dump(model, tmp)
    gcs_dest = os.path.join(bucket_path, f"{model_name}.pkl")
    os.system(f"gsutil cp {tmp} {gcs_dest}")
    log_step(f" Model {model_name} saved to {gcs_dest}")

def save_csv_to_gcs(df, filename, bucket_path="gs://thesis-loic-data/Results/"
    ):
    tmp = os.path.join(local_path, filename)
    df.to_csv(tmp, index=False)
    print(f"DEBUG:{tmp} exist? {os.path.exists(tmp)}")
    gcs_dest = os.path.join(bucket_path, filename)
    print(f"DEBUG: moving {tmp} to {gcs_dest}")
    os.system(f"gsutil cp {tmp} {gcs_dest}")
    log_step(f" CSV results saved to {gcs_dest}")
```

Listing D.6: Utility functions for logging and saving models or CSVs to Google Cloud Storage (GCS)

```
def train_single_model(X_train, y_train, X_test, y_test,
                       model_name, model_proto, param_grid,
                       categorical_features, numerical_features, all_features,
                       split_type, imputer_name, imputer,
                       bucket_path, scoring,
                       seed_base=42,
                       repeat_idx=0,
                       verbose=False):
    try:
        start_time = time.time()
        random_seed = seed_base + repeat_idx
        np.random.seed(random_seed)

        log_step(f"       Start | Split={split_type} | Model={model_name} |
            Imputer={imputer_name} | Seed={random_seed}")

        model = clone(model_proto)
        if hasattr(model, "random_state"):
            model.set_params(random_state=random_seed)

        preprocessor = ColumnTransformer(transformers=[
            ('num', Pipeline([
                ('imputer', imputer),
```

```
                ('scaler', StandardScaler())
            ]), numerical_features),
            ('cat', Pipeline([
                ('imputer', SimpleImputer(strategy='most_frequent')),
                ('encoder', OneHotEncoder(handle_unknown='ignore'))
            ]), categorical_features)
        ])

        pipeline = Pipeline([
            ('preprocessor', preprocessor),
            ('clf', model)
        ])

        ...
        % (code truncated for readability in print version; full function
            available at GitHub)
        ...

        return {
            'model_id': f"{model_name}_{imputer_name}_{split_type}_{repeat_idx
                }",
            ...
            'best_params': search.best_params_,
        }

    except Exception as e:
        ...
```

Listing D.7: Function `train_single_model`: training a supervised model with preprocessing, hyperparameter tuning, and GCS saving

```
def train_all_parallel(
    splits,
    categorical_features, numerical_features, all_features,
    bucket_path="gs://thesis-loic-data",
    results_path="gs://thesis-loic-data/Results/results_exp2.csv",
    save_path="/tmp/DataExp2/results",
    n_jobs=12,
    n_repeats=1,
    seed_base=1000,
    verbose=True,
    return_summary=False
):
    os.makedirs(save_path, exist_ok=True)
    log_step("START EVAL ALL PARAM")

    imputers = {
        'mean': SimpleImputer(strategy='mean'),
        'knn': KNNImputer(n_neighbors=5)
    }

    models = {
        'LogisticRegression': LogisticRegression(max_iter=3000, solver='
            liblinear'),
        'RandomForest': RandomForestClassifier(n_jobs=1),
        'XGBoost': XGBClassifier(eval_metric='logloss', objective='binary:
            logistic', n_jobs=1),
```

```
25          'LightGBM': LGBMClassifier(objective='binary', n_jobs=1, verbosity=-1)
                ,
26          'CatBoost': CatBoostClassifier(verbose=0, allow_writing_files=False)
27      }
28
29      log_step("Build param grid")
30      param_grids = {
31          'LogisticRegression': {...},
32          'RandomForest': {...},
33          'XGBoost': {...},
34          'LightGBM': {...},
35          'CatBoost': {...}
36      }
37
38      ...
39      % Function truncated     full version available on GitHub
```

Listing D.8: Function `train_all_parallel`: parallel execution of model training with preprocessing and GCS persistence