



# UNIVERSITÄT LEIPZIG

Institut für Informatik  
Fakultät für Mathematik und Informatik  
Abteilung Datenbanken

## **Sports Analytics im Handball: Erstellen eines Modells zum Berechnen von erwarteten Toren auf Basis von Raum-Zeit Sensordaten**

Masterarbeit

vorgelegt von:  
Andreas Kretschmer

Matrikelnummer:  
3760558

Betreuer:  
Prof. Dr. Erhard Rahm  
Dr. Christian Saal, Florens Rohde

© 2023

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

---

## Zusammenfassung

In den letzten Jahren ist der Bereich der Sports Analytics stark gewachsen. Sportarten wie Baseball, Basketball, Fußball und Eishockey sind Vorreiter und beschäftigen sich ausführlich mit diesem Thema. Im Handball ist der Bereich Sports Analytics noch nicht sonderlich ausgeprägt. Aber auch im Handball gibt es erste Schritte zur umfassenden Datenanalyse und neuen Metriken, die einen Wettbewerbsvorteil bieten können.

Dieses Projekt stellt eine Version einer dieser neuen Metriken vor, die auf die Sportart Handball angewendet wird. Diese Metrik sind die erwarteten Tore. Die Metrik approximiert die Wahrscheinlichkeit, dass ein Wurf zu einem Tor wird. Es werden zwei verschiedene Möglichkeiten vorgestellt, wie dieser Wert berechnet werden kann. Einer beinhaltet nur die Abwurfpositionen. Der andere beinhaltet zusätzliche Kontextinformationen zu jedem Wurf. Anschließend wird auf die Wahrscheinlichkeiten eingegangen, die die zwei Modelle liefern. Zusätzlich dazu werden mögliche Interpretationen der Modelle betrachtet und Anwendungsmöglichkeiten gezeigt, die zur Bewertung der Spieler- oder Mannschaftsleistung verwendet werden können.

Es wird ebenfalls gezeigt wie sich einzelne Merkmale auf die Ergebnisse der Modelle auswirken um die Modelle interpretierbar zu machen und es wird gezeigt, dass zusätzliche Kontextinformationen relevant für die berechneten Wahrscheinlichkeiten sind und das Modell verbessern können.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>V</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Zielsetzung . . . . .	1
1.2. Aufbau . . . . .	2
<b>2. Related Work</b>	<b>3</b>
2.1. Sports Analytics . . . . .	3
2.2. Sports Analytics in Handball . . . . .	4
<b>3. Grundlagen</b>	<b>7</b>
3.1. Data Mining Aufgaben . . . . .	7
3.1.1. Klassifikation . . . . .	7
3.1.2. Metriken . . . . .	8
3.1.3. k-Fold Cross Validation . . . . .	9
3.1.4. Kalibrierung . . . . .	10
3.2. Shapley Additive Explanations-Framework . . . . .	11
<b>4. Methodik</b>	<b>13</b>
4.1. Business Understanding . . . . .	14
4.2. Data Understanding . . . . .	14
4.2.1. Ereignisdaten . . . . .	15
4.2.2. Positionsdaten . . . . .	16
4.3. Data Preparation . . . . .	17
4.3.1. Kombinieren der Datensätze . . . . .	17
4.3.2. Erstellen zusätzlicher Merkmale . . . . .	18
4.3.3. Bereinigen der Daten . . . . .	24
4.3.4. Datensätze . . . . .	25
4.4. Modellierung . . . . .	25
4.5. Implementation . . . . .	28
<b>5. Evaluation</b>	<b>30</b>
5.1. Training der Modelle . . . . .	30
5.2. Testen der Modelle . . . . .	31
5.3. Kalibrierung . . . . .	32
5.4. Interpretation . . . . .	35
<b>6. Anwendungsfälle</b>	<b>41</b>
6.1. Analyse einzelner Würfe . . . . .	42

6.2. Leistungsanalyse eines Spiels . . . . .	44
<b>7. Limitation</b>	<b>47</b>
<b>8. Zusammenfassung und Ausblick</b>	<b>50</b>
8.1. Zusammenfassung . . . . .	50
8.2. Ausblick . . . . .	51
<b>Literatur</b>	<b>52</b>
<b>Erklärung</b>	<b>57</b>
<b>A. Ordnerstruktur</b>	<b>A-I</b>
<b>B. Dateischemata</b>	<b>B-I</b>
B.1. Schema der Spiel-Liste . . . . .	B-I
B.2. Schema der Wurf-Liste . . . . .	B-II
B.3. Schema der Positionsdaten . . . . .	B-III
B.4. Schema der Trägheitsdaten . . . . .	B-IV

## **Abkürzungsverzeichnis**

**AUC** Area under Curve.

**CRISP-DM** Cross Industry Standard Process for Data Mining.

**ECE** Expected Calibration Error.

**EFB** Exclusive Feature Bundling.

**GOSS** Gradient-based One-Side Sampling.

**HBL** LIQUI MOLY Handball Bundesliga.

**LIME** Local Interpretable Model-Agnostic Explanations.

**ROC** Receiver Operating Characteristics.

**SHAP** SHapley Additive exPlanations.

**SPAM** Shot Position Average Model.

**xG** eXpected Goals oder erwartete Tore.

## Abbildungsverzeichnis

4.1. Crisp-DM Modell . . . . .	13
4.2. Erstellprozess neuer Merkmale . . . . .	19
4.3. Darstellung der berechneten Winkel . . . . .	20
4.4. Beispiel für Abwurf- und Spielerpositionen . . . . .	21
4.5. Beispiel für eine Empty-Net- und Powerplay-Spielsituation . . . . .	22
5.1. Kalibrierungskurve des XGBoost-Modells . . . . .	32
5.2. Nachkalibrierungskurven des XGBoost-Modells . . . . .	34
5.3. Kalibrierungskurve des LightGBM-Modells . . . . .	35
5.4. Detaillierte Bedeutung der Merkmale des XGBoost-Modells . . . . .	36
5.5. Vereinfachte Bedeutung der Merkmale des XGBoost-Modells . . . . .	36
5.6. Detaillierte Bedeutung der Merkmale des LightGBM-Modells . . . . .	38
5.7. Vereinfachte Bedeutung der Merkmale des LightGBM-Modells . . . . .	39
5.8. Vier Wurf-Beispiele mit dazugehörigem xG-Wert . . . . .	40
6.1. Heatmap der xG-Werte . . . . .	41
6.2. Analyse der Schätzung einzelner xG-Werte . . . . .	43

# Tabellenverzeichnis

4.1. Beispiel der Ereignisdaten . . . . .	15
4.2. Übersicht zu der Anzahl der Würfe und Tore . . . . .	15
4.3. Übersicht der Spiele mit Positionsdaten . . . . .	16
4.4. Beispiel der Positionsdaten . . . . .	16
4.5. Beispiel der Trägheitsdaten . . . . .	17
4.6. Beispiel der Rundung der Zeitstempel . . . . .	18
4.7. Übersicht der verwendeten Merkmale . . . . .	23
4.8. Merkmalsübersicht nach Datensatz . . . . .	25
4.9. Übersicht der Modell-Hyperparameter . . . . .	27
5.1. Trainingsergebnisse . . . . .	30
5.2. Evaluationsergebnisse . . . . .	31
5.3. Kalibrierungsergebnisse des XGBoost-Modells . . . . .	34
5.4. Kalibrierungsergebnisse des LightGBM-Modells . . . . .	34
6.1. Vergleich der Wurfstatistik mit dem xG-Wert eines Spiels pro Spieler . . . . .	45
6.2. Vergleich der Wurfstatistik mit dem xG-Wert eines Spiels pro Team . . . . .	46

# 1. Einleitung

In den letzten Jahren hat der Bereich der Sport Analytics einen enormen Aufschwung erfahren und wird immer wichtiger für die Entscheidungsfindung im Sport [1]. Dies wird ermöglicht durch neue Technologien mit denen viele Daten einfach und detailreich erfasst werden können. Seit 2019 werden durch Kinexon umfassende Daten der Spieler und aller Spielbälle der LIQUI MOLY Handball Bundesliga (HBL) aufgezeichnet. Die Aufzeichnung erfolgt mithilfe von Sensoren. Diese Daten werden auch als Raum-Zeit Sensordaten bezeichnet [2]. Bis dahin basierte die Entscheidungsfindung auf Erfahrung von Trainern anstatt auf objektiven und informationsbasierten Erkenntnissen [3]. Durch die konstante mögliche Überwachung und Analyse der Daten auf Basis von Data Mining Methoden können Betreuer und Trainer in ihren Entscheidungen unterstützt werden [2].

Im Handball werden die Anzahl der Würfe und die Anzahl der Tore oft als Statistik für die Qualitätsbewertung herangezogen. Die Qualität der Würfe wird dabei nicht mitberücksichtigt. Das Konzept der xG ist im Handball bisher nicht weit verbreitet. Bisherige Modelle berücksichtigen nur die Abwurfposition aber keine Informationen über den Kontext des Wurfes. Allerdings bieten die Daten von Kinexon eine Vielzahl von Möglichkeiten, die genutzt werden können, um ein Modell bereitzustellen, welches auf Raum-Zeit Sensordaten basiert und somit zusätzliche Kontextinformationen beinhaltet. Dazu gehören neben der Wurfposition auch Geschwindigkeiten der Spieler und Informationen über die Spielsituation.

In dieser Masterarbeit soll das Konzept der xG im Handball genauer untersucht werden. Ziel ist es, ein Modell zu entwickeln, das die Wahrscheinlichkeit berechnet, dass ein bestimmter Wurf ein Tor erzielt. Dabei sollen Die Raum-Zeit Sensordaten verwendet werden, um die Kontextinformationen zu berechnen. Diese können beispielsweise die Verteidigerdichte vor dem Werfer, die Position des Torwarts oder Überzahlsituationen sein. Um dieses Ziel zu erreichen, werden zunächst verschiedene Methoden der Sportanalyse und der statistischen Modellierung untersucht. Anschließend werden verfügbare Datenquellen analysiert und die erforderlichen Daten aufbereitet. Danach wird ein Modell entwickelt und anhand von Testdaten validiert. Schließlich werden die Ergebnisse der Modellierung diskutiert und mögliche Anwendungen in der Praxis des Handballsports erörtert.

Diese Masterarbeit soll somit einen Beitrag zur Weiterentwicklung der Sportanalyse im Handball leisten und aufzeigen, wie das Konzept der xG auch in dieser Sportart genutzt werden könnte, um Entscheidungen zu treffen und Leistungen zu bewerten.

## 1.1. Zielsetzung

Ziel dieser wissenschaftlichen Arbeit soll es sein, ein Modell zu erstellen, das die Qualität eines jeden Wurfes quantifiziert. Dieses soll die Wahrscheinlichkeit eines erwarteten Tores bei einem Wurf angeben. Dabei sollen zwei verschiedene Arten von Modellen erstellt werden. Zum einen soll es ein Modell geben, welches nur auf den Wurfpositionen basiert. Dieses wird als Vergleichsmodell dienen. Zum anderen soll ein zweites Modell erstellt werden, welches auf zusätzlichen Kontextinformationen basiert. Diese beiden Modelle werden anschließend miteinander verglichen. Des weiteren soll auf die



Interpretation der beiden Modelle und Anwendungsszenarien eingegangen werden. Daraus lassen sich folgende Forschungsfragen formulieren, die im Zuge dieser Arbeit beantwortet werden:

**RQ1** Kann ein xG-Modell durch zusätzliche Kontextinformationen verbessert werden?

**RQ2** Welche Eigenschaften sind für die Berechnung des xG-Wertes ausschlaggebend?

**RQ3** Wie kann man das xG-Modell interpretieren?

**RQ4** Ist der xG-Wert eine geeignete Metrik für die Leistungsbewertung im Handball?

### 1.2. Aufbau

Zuerst soll ein Überblick über den bisherigen Forschungsstand gegeben werden. Zusätzlich dazu sollen die Grundlagen von Sports Analytics, xG und den Data Science Methoden, die für die zu erzeugenden Modelle genutzt werden, aufgezeigt werden. Anschließend wird die Methodik erläutert. Dies beinhaltet das allgemeine Vorgehen bei der Bearbeitung dieser Arbeit sowie die Beschreibung der verwendeten Daten, die Berechnung der Merkmale, die für die Erstellung des Modells ausgewählt werden, die Bereinigung der Daten und die Modellierung der Modelle. Sobald dies erfolgt ist, sollen die Modelle mit Hilfe eines Trainings- und Testdatensatzes trainiert und evaluiert werden. Anschließend werden beide Modelle interpretiert und Anwendungsszenarien für die Ergebnisse der Modelle aufgezeigt. Abschließend wird eine Einordnung der Ergebnisse und eine Limitation der Arbeit erfolgen.

## 2. Related Work

Dieses Kapitel beschäftigt sich mit bisherigen Arbeiten zum Thema Sports Analytics. Unter Sports Analytics versteht man die Datenanalyse mit verschiedenen Data Mining Methoden von Sportdaten. Zuerst werden Arbeiten zu erwarteten Toren aus verschiedenen Sportarten vorgestellt. Zusätzlich dazu wird ein Überblick über Forschungsarbeiten aus dem Handballumfeld und auf die ersten Anfänge der Datenanalyse im Sport gegeben.

Im Sport lassen sich die Forschungsthemen in die drei Kategorien Sportwissenschaften, Sportmedizin und Sporttaktiken einteilen. In diesem Kapitel sowie in dieser Arbeit wird sich auf die Sporttaktiken konzentriert.

### 2.1. Sports Analytics

Eine der ersten Publikationen zu Sports Analytics ist „Baseball Abstract“ [4] von Bill James aus 1981, welches als Beginn von Sports Analytics gilt. Auch wenn James zum größten Teil keine grundlegenden Instrumente der Statistik, wie beispielsweise Modellanpassung oder grafische Darstellungen verwendete, war seine Arbeit einflussreich. Die Arbeit widerlegte Thesen im Baseball, welche lange Zeit als Überzeugung galten. James belegt seine Hypothesen mithilfe von Daten zu den Spielern und Spielen [4]. Darüber hinaus legte seine Arbeit den Grundstein für die Arbeit von Lewis, der mit seinem Buch „Moneyball: The Art of Winning an Unfair Game“ [5] mehrere Konzepte von James aufgreift und beschreibt, wie Billy Beane diese in der praktischen Anwendung adaptiert. Billy Beane war zu diesem Zeitpunkt der Generaldirektor des Major League Baseball Teams Oakland Athletics aus Amerika. Mithilfe der Konzepte formte er ein erfolgreiches Team, obwohl er nur ein kleines Budget zur Verfügung hatte [5]. Eine der neueren wissenschaftlichen Veröffentlichungen zu Sports Analytics ist das Buch „Handbook of Statistical Methods and Analyses in Sport“ [6] von Albert, Swartz, Koning und Glickman, welches einen Überblick über statistische Methoden in den großen Sportarten gibt und auf Probleme und Herausforderungen in der Forschung eingeht [6].

Eines der ersten Modelle zu erwarteten Toren wurde von Pollard, Ensum et al. 2004 [7] veröffentlicht. Es wurde ein Modell entwickelt, welches die Wahrscheinlichkeit für ein Tor eines Schusses im Fußball angibt. Das Modell basiert auf einer logistischen Regressionsanalyse und den drei Merkmalen Distanz zum Tor, Winkel zum Torpfosten und der Raum des schießenden Spielers zu dem nächsten Verteidiger. Als Datensatz wurde 1096 Schüsse verwendet. Pollard et al. argumentierten bereits 2004, dass es noch weitere Eigenschaften geben könnte, die den Schuss beeinflussen. Dennoch bietet das Modell eine Basis für verschiedene Auswertungen, wie beispielsweise das Quantifizieren der Effektivität von Taktiken oder das Quantifizieren der Effektivität von Spielern oder auch von Torhütern [7]. Die Arbeit von Wright et al. von 2011 [8] befasst sich mit weiteren Faktoren, die den Erfolg eines Schusses im Fußball ausmachen. Hierfür werden mehrere Eigenschaften von 1788 Schüssen untersucht. Die Schüsse werden manuell erfasst. Eigenschaften zu jedem Schuss sind beispielsweise die Position des Torhüters, die Art des Schusses und die vorangegangenen Pässe [8]. Der Begriff „expected Goals“ wurde das erste Mal 2012 von Macdonald [9] in einer Publikation über Eishockey erwähnt. In seiner Arbeit verwendet er Ridge Regression, um den Beitrag eines

Spielers für die erwarteten Tore seines Teams pro 60 Minuten zu schätzen. Obwohl Fußball und Eishockey unterschiedliche Sportarten sind, ähneln sich diese dennoch in Bezug auf das Modell für erwartete Tore. In beiden Sportarten versucht ein Spieler durch einen Schuss ein Tor zu erzielen und ein Torhüter versucht dies zu verhindern. Für das Modell verwendet Macdonald die Schüsse der letzten vier vollen Saisons der National Hockey League. Als zusätzliche Eigenschaften des Schusses benutzt Macdonald Teamstatistiken, die die Leistung des Teams quantifizieren. Diese sind beispielsweise Tore, Schüsse, Fehlschüsse und Umschaltspiele [9].

In den Jahren nach der ersten Erwähnung von erwarteten Toren wurden zahlreiche Modelle von Firmen oder der Onlinecommunity erstellt und stetig verbessert. Beispiele dafür sind Caley et al. [10], Cintia et al. [11], Green et al. [12] oder Trainor et al. [13]. Hier wurden vor allem zusätzliche Eigenschaften des Schusses mit in die Entwicklung des Modells mit einbezogen, wie beispielsweise mit welchem Körperteil der Schuss erfolgte oder ob der Schuss Teil eines speziellen Spielzuges war. Darüber hinaus werden diverse statistische Methoden in Betracht gezogen, um die Modelle zu erstellen.

Neuere Modelle beziehen sogenannte Raum-Zeit-Daten mit ein. Raum-Zeit Sensordaten beinhaltet Informationen über die Position jedes einzelnen Spielers auf dem Spielfeld anstatt nur die Daten des Spielers mit dem Ball. Lucey et al. [14] stellte 2015 ein logistisches Regressionsmodell für die Wahrscheinlichkeit eines Tores vor, welches zusätzlich auf Eigenschaften aus Raum-Zeit Sensordaten basiert. Die Eigenschaften zu jedem Schuss werden in die Kategorien Position, Kontext, Angriff und Verteidigung unterteilt. Beispiele dafür sind die Nähe von Verteidigern, die Spielgeschwindigkeit oder die Interaktion von umliegenden Spielern. Die Eigenschaften wurden aus Ereignisdaten und Trackingdaten von knapp 9700 Schüssen extrahiert [14].

Ähnlich ist die Lösung von Eggels et al. von 2016 [15]. Hier wurden ebenfalls Ereignis- und Trackingdaten verwendet, um die Eigenschaften eines Schusses zu extrahieren. Zusätzlich dazu wurden auch die Fähigkeiten des Schützen und des Torwartes als Eigenschaften mit verwendet. Diese stammen aus der öffentlich zugänglichen Datenbank des Videospiels EA Sports FIFA. Darüber hinaus wurden verschiedene statistische Methoden verwendet. Neben der logistischen Regression wurden drei baumbasierte Klassifikatoren untersucht. Diese waren Random Forest, Entscheidungsbäume und AdaBoost [15].

Die Arbeit von [16] nutzte ebenfalls Raum-Zeit-Daten, um sogenannte Off-Ball Torchancen zu quantifizieren und somit den xG-Wert weiter mit Kontext anzureichern. Somit können spielentscheidende Situationen identifiziert und analysiert werden [16]. Aktuelle Arbeiten zu erwarteten Toren in Fußball und Eishockey beschäftigen sich vor allem mit zusätzlichen Merkmalen, um die Schüsse genauer zu beschreiben. Diese verwenden ebenfalls Raum-Zeit Sensordaten, um die Merkmale zu berechnen. Hier zu nennen sind die Arbeiten von Hewitt et al. [17], Mead et al. [18], Anzer et al. [19], Madrero Pardo [20] und Paerels et al. [21].

## 2.2. Sports Analytics in Handball

Im Handball sind Sport Analytics Anwendungen bisher noch nicht so weit fortgeschritten, wie in anderen Sportarten. Derzeit gibt es zwei verschiedene xG-Modelle. Zum einen gibt es das Shot Po-

sition Average Model (SPAM) [22] von Pallamano. Zum anderen existiert ein vorgestelltes Modell von Julian Rux [23]. Beides sind Blog-Beiträge und sind gleich aufgebaut. In beiden Fällen wird das Spielfeld in verschiedene Bereiche aufgeteilt. Pro Bereich werden alle Würfe betrachtet, die aus dem gewählten Bereich erfolgt sind. Der xG-Wert pro Bereich ist dann der prozentuale Anteil der Tore der Würfe aus diesem Bereich. Die beiden Arbeiten unterscheiden sich in den verwendeten Datensätzen. Während Rux knapp 75000 Würfe der HBL verwendete, verwendete Pallamano hingegen 30000 Würfe aus der Handball Champions League, Weltmeisterschaft, Europameisterschaft und anderen Turnieren.

Andere Arbeiten im Bereich Sport Analytics in Handball befassen sich mit der Einführung und Entwicklung von verschiedenen Leistungs-Metriken für den Handball [24]. Hier ist vor allem die Arbeit zu Spieler-Leistung-Idizes von Schwenkreis [3] zu nennen. Dieser stellt ein Informationsmodell vor, welches die Leistung der Spieler oder eines Teams bewertet, indem diese Punkte für Aktionen während des Spiels verdienen oder verlieren [3].

Des Weiteren gibt es Arbeiten von Hatzimanouil et al. [25] und Meletakos et al. [26] zur Wurf-Effektivität von Spielern. Die Arbeit von Hatzimanouil et al. [25] beschäftigt sich mit der Wurfeffektivität der Spieler je nach Abwurf und nach Spielerposition. Hierbei wurden Würfe aus 25 Spielen der griechischen Liga untersucht. Die Würfe wurden per Video manuell ausgewertet. Für alle Würfe wurden insgesamt fünf Merkmale festgehalten. Diese sind die Wurf-Effektivität zwischen 6 und 9 Metern, das Ergebnis des Wurfs (Tor, neben das Tor, gehalten), der Bereich des Einschlags im Tor, die Distanz zum Tor und die Anzahl der Paraden des Torhüters. Die gesammelten Daten wurden anschließend mit einem Chi-Quadrat-Test analysiert. Die Ergebnisse zeigen, dass die meisten Würfe in einer Distanz von 6 bis 9 Metern zum Tor erfolgen. Würfe mit einer Distanz größer als 9 Meter sind sehr selten. Des Weiteren wird festgehalten, dass die Wurf-Effektivität zwischen den einzelnen Spielerpositionen unausgeglichen ist. Im Gegensatz zu den vorher genannten Arbeiten, wurden in dieser Arbeit Daten verwendet, die ausschließlich manuell erfasst wurden [25].

Zusätzlich zu den Leistungsbewertungen, gibt es mehrere Arbeiten, die sich mit Einflussfaktoren für den Erfolg eines Teams oder Spielers beschäftigen. Diese beziehen sich auf die physische Leistungsfähigkeit der Spieler, wie beispielsweise die gesamte gelaufene Distanz, Beschleunigung, Entscheidungsverhalten oder Reaktionszeit [27][28][29][30]. Hier stellt sich heraus, dass die Aktivitätenprofile der Spieler einen Einfluss auf den Erfolg des Teams haben. Unter Aktivitätenprofile versteht man eine Reihe von physischen Leistungen, die ein Spieler während eines Spiels leistet [28]. Eine Optimierung dieser Aktivitätenprofile durch ein abgestimmtes Training kann die Leistung eines Spielers zusätzlich positiv beeinflussen [31].

Neben der Leistungsanalyse der Spieler oder Teams gibt es mehrere Arbeiten zu der Spielanalyse. Die Arbeit von Schwenkreis von 2018 [32] befasst sich mit der automatischen Erkennung von Taktiken. Es wird ein Modell vorgestellt, welches mithilfe einer Serie von Bildern versucht, taktische Bewegungen eines Teams automatisch zu erkennen [32].

Da die automatische Datenerfassung im Handball noch relativ neu ist und die Mannschaften durch ihr Budget limitiert sind, wird derzeit an Möglichkeiten gearbeitet, die Teams dabei zu unterstützen. Eine mögliche Lösung ist eine Anwendung, die von Schwenkreis et al. 2020 [33] vorgestellt wurde. Die Anwendung stellt, neben der Möglichkeit Daten zu erfassen, Methoden des Data Minings

[34] bereit, um Informationen zu den gesammelten Daten bereitzustellen. Dazu zählen beispielsweise detaillierte Informationen zu den Toren, zur Wurfeffektivität aus verschiedenen Distanzen, zu verschiedenen Spielsituationen oder zur Schätzung des Spielergebnisses [33][34].

## 3. Grundlagen

Dieses Kapitel befasst sich mit den Grundlagen des Data Minings, die in dieser Arbeit verwendet werden. Hierbei werden die verschiedenen Modellverfahren, verwendete Metriken und Trainingsverfahren erläutert. Zusätzlich dazu wird ein Framework vorgestellt, um die erstellten Modelle zu interpretieren. Hier wird vor allem auf das Prinzip eingegangen, auf dem das Framework basiert.

### 3.1. Data Mining Aufgaben

Allgemein befasst sich Data Mining mit dem Generieren von Wissen aus gesammelten Daten. Als Beobachtungen kann man gesammelte Daten zu einem bestimmten Ereignis bezeichnen. Jede Beobachtung besteht aus Merkmalen, die die Beobachtung beschreiben. Die verwendeten Methoden im Data Mining lassen sich in drei verschiedene Kategorien unterteilen. Diese wären überwachtes Lernen, unüberwachtes Lernen und Reinforcement Learning. Unter unüberwachtem Lernen versteht man das Ziehen von Schlussfolgerungen aus Daten ohne einen bestimmten Zielwert pro Beobachtung. Reinforcement Learning sind Methoden, mit denen ein Modell trainiert wird, indem es Belohnungen oder Bestrafungen für jede ausgeführte Aktion erhält und somit sein Verhalten entsprechend anpasst. Unter überwachtem Lernverfahren versteht man Methoden, die versuchen aus Daten mit jeweils einem Zielwert pro Beobachtung, Schlussfolgerungen zu ziehen [35, S. 26–28].

#### 3.1.1. Klassifikation

Klassifikation ist eine Art von Data Mining Aufgaben. Darunter versteht man die Aufgabe, eine Funktion  $f$  zu erlernen, die eine Beobachtung mit einer Menge an Merkmalen  $X$  zu einer vordefinierten Klasse  $y$  zuordnet. Die Klasse ist unser Zielwert oder auch abhängige Variable. Da es einen Zielwert gibt gehört die Klassifikation zu den überwachten Lernverfahren [35, S. 28-29]. Im Fall des xG-Modells gibt es die zwei Klassen „Tor“ und „kein Tor“. Somit ist  $y_i \in \{KeinTor, Tor\}$ . In dieser Arbeit werden verschiedene Klassifikations-Algorithmen verwendet. Diese werden hier kurz erläutert.

- **Logistische Regression:** Die logistische Regression gehört zu den grundlegendsten Methoden für eine binäre Klassifikation. Das resultierende Modell schätzt die Wahrscheinlichkeit des Auftretens eines Ereignisses wie zum Beispiel Tor oder kein Tor. Es wird eine Logit-Funktion auf die Odds angewendet. Unter Odds versteht man die Wahrscheinlichkeit, dass ein Ereignis eintritt, geteilt durch die Wahrscheinlichkeit für das nicht Eintreten des Ereignisses. Die Schätzung der Koeffizienten des Modells erfolgt nach dem Maximum-Likelihood-Verfahren [35, S. 172-176].
- **Random Forest:** Das Random Forest-Verfahren gilt als Tree-Ensemble Methode. Es besteht aus mehreren unkorrelierten Entscheidungsbäumen. Ein Entscheidungsbaum ist ein nichtparametrischer Lernalgorithmus des überwachten Lernens für Klassifizierungs- und Regressionsprobleme. Er besteht aus einer Wurzel, Zweigen, inneren Knoten und Blattknoten. Der

Wurzelknoten ist der Start jedes Baumes. Davon gehen Zweige zu inneren Knoten ab. Beide Knotenarten führen Bewertungen durch und treffen Entscheidungen. Die Entscheidungen basieren auf den Merkmalen, die für das Modell verwendet werden. Jede Entscheidung dient dazu, homogene Teilmengen zu bilden, die durch die Blattknoten abgebildet werden und die möglichen Ergebnisse des Modells darstellen. Um die optimalen Teilungspunkte innerhalb des Baums zu finden, wird eine Greedy-Suche durchgeführt. Der Aufteilungsprozess wird iterativ durchgeführt, bis der größte Teil des Trainingsdatensatzes durch den Entscheidungsbaum klassifiziert werden kann [35, S. 328-338].

Beim Random Forest-Verfahren werden mehrere solcher Bäume erstellt. Jeder Baum wird auf Basis einer zufälligen Teilmenge des Trainingsdatensatzes erstellt. Zudem wird beim Erstellen jedes inneren Knoten im Baum, nur eine zufällige Teilmenge der Merkmale aus dem Trainingsdatensatz verwendet. Durch diese beiden Zufallsfaktoren wird die Varianz des Klassifikators verringert und eine Überanpassung vermieden. Das Ergebnis der Klassifikation ist der Mittelwert der geschätzten Wahrscheinlichkeiten aller Bäume, dass eine Beobachtung zu einer bestimmten Klasse gehört [36].

- **XGBoost:** Ähnlich wie das Random Forest Verfahren basiert XGBoost ebenfalls auf mehreren Entscheidungsbäumen, die in der Gemeinschaft das Ergebnis einer Klassifikation ermitteln. Der Unterschied liegt im Training. Beim Erstellen der Bäume kommt das sogenannte Gradient Boosting zum Einsatz. Deshalb zählt der XGBoost Algorithmus zu den Gradient Boosting Methoden. Zuerst wird ein Klassifikations-Baum erstellt. Auf Basis dieses Baums wird die Differenz zwischen geschätztem Wert und tatsächlichem Wert ermittelt. Anschließend wird ein neuer Baum erstellt, der den Fehler aus dem vorangegangenen Baum mit berücksichtigt. Es werden solange Bäume hinzugefügt, bis keine Verbesserung mehr in der Schätzung erzielt wird [37].
- **LightGBM:** Ähnlich wie XGBoost ist LightGBM ebenfalls ein Gradient Boosting Verfahren, welches auf Entscheidungsbäumen basiert. LightGBM basiert auf dem gleichen Algorithmus wie XGBoost. Allerdings löst LightGBM Probleme mit der Laufzeit und dem benötigten Speicher bei großen Datensätzen. LightGBM verwendet zwei Tricks um diese Probleme zu lösen. Zum einen wird die Methode Gradient-based One-Side Sampling (GOSS) und zum anderen die Methode Exclusive Feature Bundling (EFB) verwendet. Unter GOSS versteht man, dass nur bestimmte Beobachtungen aus dem Trainingsdatensatz für die Ermittlung des bestmöglichen nächsten Blattes herangezogen werden. Diese Beobachtungen werden anhand eines Gradienten ausgewählt. Die Beobachtungen mit einem großen Gradienten, also mit einem großen Fehler zwischen geschätzter und tatsächlicher Klasse, werden mit einer höheren Wahrscheinlichkeit ausgewählt. Unter EFB versteht man eine Methode, mit der die Merkmale, die für den nächsten Zweig in Frage kommen, verkleinert werden. Dabei werden spärlich besetzte Merkmale zusammengefasst und somit die Anzahl der Merkmale reduziert [38].

### 3.1.2. Metriken

Das Ziel eines Modells sollte sein, so gute Schätzungen wie möglich zu generieren. Dies gilt vor allem mit Daten, die nicht während des Trainings verwendet werden und somit für das Modell

unbekannt sind. Es gibt verschiedene Metriken, um die Leistung eines Modells zu messen. Welche man verwendet ist abhängig von dem Zweck des Modells. Im folgenden werden die Metriken aufgezeigt und erläutert, die in dieser Arbeit verwendet werden. Diese sind der logarithmische Verlust und die Area under Curve (AUC).

Der logarithmische Verlust, oder auch log Loss, ist eine Metrik, die die Genauigkeit eines Klassifikators mithilfe der geschätzten Wahrscheinlichkeiten für jede mögliche Klasse berechnet. Andere Metriken betrachten nur die Wahrscheinlichkeit einer Klasse. Man kann dies auch als die Kreuzentropie zwischen der Verteilung der Schätzung und dem tatsächlichen Wert bezeichnen. Die Kreuzentropie misst die Summe der Unvorhersehbarkeit der tatsächlichen Verteilung und einer zusätzlichen Unvorhersehbarkeit, falls eine andere Verteilung, als die tatsächliche, angenommen wird. Der logarithmische Verlust wird formal definiert durch:

$$LL = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij}). \quad (3.1)$$

Bei einem binären Klassifikationsproblem kann die Definition wie folgt vereinfacht werden

$$LL = -\frac{1}{N} \sum_{i=1}^N y_i \log(p_i) + (1 - y_i) \log(1 - p_i). \quad (3.2)$$

$N$  steht dabei für die Anzahl der Beobachtungen in der Testmenge,  $M$  ist die Menge an möglichen Klassen,  $y_{ij}$  ist ein binärer Indikator, ob die Klasse  $j$  die tatsächliche Klasse für die Beobachtung  $i$  ist und  $p_{ij}$  ist die Wahrscheinlichkeit des Modells für die Klasse  $j$  für die Beobachtung  $i$ . Der logarithmische Verlust hat den Vorteil, dass er die Konfidenz von Klassifikationen mit berücksichtigt. Fehlklassifikationen fließen als Strafen in das Ergebnis mit ein, wodurch sich die Metrik auch bei unausgeglichene Datensätzen eignet [39].

Die Area under Curve (AUC) beschreibt die Fläche unter der Receiver Operating Characteristics (ROC) Kurve. Die ROC-Kurve ist ein Plot der True-Positive-Rate gegen die False-Positive-Rate bei verschiedenen Schwellenwerten für die Klassifikation. Der AUC-Wert kann interpretiert werden als die Wahrscheinlichkeit, dass eine zufällige True-Positive Beobachtung durch ein Modell höher bewertet wird als eine True-Negative Beobachtung. Der Wert kann dabei zwischen 0 und 1 liegen. Ein Wert unter 0.5 wäre allerdings schlechter als ein Modell, welches zufällig Beobachtungen klassifiziert. Alle Modelle, die einen AUC-Wert unter 0.5 erzielen, sollten nicht verwendet werden. Der AUC-Wert hat zwei Vorteile. Zum einen ist er invariant gegenüber verschiedenen Skalen. Er misst, wie gut Klassifizierungen eingeordnet werden. Die eigentlichen geschätzten Werte werden nicht betrachtet. Ein weiterer Vorteil ist, dass der AUC-Wert invariant gegenüber verschiedenen Schwellenwerten ist, die für die Klassifikation verwendet werden. Das liegt daran, dass die Qualität der Klassifikation unabhängig von dem gewählten Schwellenwert gemessen wird [40].

### 3.1.3. k-Fold Cross Validation

Cross Validation wird verwendet, um eine Überanpassung von Modellen beim Training zu verhindern. Dafür wird der Trainingsdatensatz in einen Trainings- und Validierungsdatensatz aufgeteilt.



Der Validierungsdatensatz wird verwendet, um das Modell nach der Trainingsphase zu testen. Allgemein wird der Trainingsdatensatz in  $k$  Submengen mit gleicher Größe zufällig aufgeteilt. Diese nennt man auch Faltungen. Aus den  $k$  Submengen wird eine Submenge als Validierungsdatensatz ausgewählt. Die restlichen  $k-1$  Submengen dienen als Trainingsdatensatz. Das Training wird  $k$  Mal wiederholt, wobei jede Submenge nur einmal als Validierungsdatensatz verwendet wird. Das Ergebnis ist in der Regel der Durchschnitt aller Validierungsergebnisse. Dies verringert die Variabilität der Modellgenauigkeit. Gängige Werte für  $k$  sind 5 oder 10 [35, S. 203-205].

### 3.1.4. Kalibrierung

Die Klassifizierungsergebnisse der Modelle können als Wahrscheinlichkeiten wiedergegeben werden. Dies geschieht in der Regel durch die Berechnung der Wahrscheinlichkeit, dass eine Beobachtung zu einer bestimmten Klasse gehört. Die Wahrscheinlichkeit der Klassenzugehörigkeit kann auch interpretiert werden als Konfidenz des Modells, dass eine gegebene Beobachtung zu einer bestimmten Klasse gehört. Damit die geschätzten Wahrscheinlichkeiten auch als Wahrscheinlichkeiten interpretiert werden können, muss geprüft werden, ob diese auch den tatsächlichen Wahrscheinlichkeiten entsprechen. Über eine Kalibrierungskurve lässt sich dies für ein Modell prüfen und visuell darstellen. Ein Beispiel und eine Erläuterung der Kalibrierungskurve findet sich in Kapitel 5.3. Um die Kalibrierung des Modells nach dem Training zu verbessern gibt es sogenannte Re-Kalibrierungsmethoden. Zwei Hauptmethoden existieren, mit denen sich die Wahrscheinlichkeiten des Modells zu Posterior-Wahrscheinlichkeiten zuordnen lassen. Diese sind Platt- oder auch Sigmoid-Kalibrierung und die Isotonic Regression [41].

- **Platt-Kalibrierung:** Platt schlug vor, die Wahrscheinlichkeiten eines Modells zu Posterior-Wahrscheinlichkeiten mithilfe einer Sigmoid-Funktion zu transformieren [42].
- **Isotonic Regression:** Einen allgemeineren Vorschlag machte Zadrozny mithilfe von Isotonic Regression [43]. Sie ist allgemeiner in dem Sinne, dass die Abbildungsfunktion nicht mehr eine Sigmoid-Funktion, sondern nur eine monoton steigende Funktion sein muss [41].

Es kann gezeigt werden, dass die Platt-Kalibrierung bei Datensätzen mit relativ wenigen Beobachtungen bessere Ergebnisse erzielt als die Isotonic Regression. Wenn der Datensatz hingegen größer wird, erzielt die Isotonic Regression bessere Ergebnisse [41].

Die Ergebnisse des unkalibrierten Modells unterscheiden sich zu dem des kalibrierten Klassifikators. Die Hauptaufgabe des kalibrierten Klassifikators ist, möglichst genaue Schätzungen der tatsächlichen Wahrscheinlichkeiten zu erzielen. Anders gesagt vergleicht der kalibrierte Klassifikator die Wahrscheinlichkeiten des unkalibrierten Klassifikators mit den tatsächlichen Wahrscheinlichkeiten. Da a priori keine tatsächlichen Wahrscheinlichkeiten vorhanden sind, werden ähnliche Wahrscheinlichkeiten des unkalibrierten Klassifikators in Körbe zusammengefasst. Pro Korb wird anschließend der Anteil jeder Klasse berechnet. In dieser Arbeit wird pro Korb der Anteil der tatsächlich erzielten Tore berechnet. Der Anteil wird anschließend mit der durchschnittlichen geschätzten Wahrscheinlichkeit des gleichen Korbs verglichen. Eine gute Kalibrierung zeichnet sich dadurch aus, dass die tatsächlichen Wahrscheinlichkeiten und die durchschnittlichen geschätzten Wahrscheinlichkeiten eines Korbs gleich beziehungsweise sehr ähnlich sind [41].

## 3.2. Shapley Additive Explanations-Framework

Zu verstehen, wie ein Modell bestimmte Schätzungen berechnet, kann für die Interpretation des Modells genau so kritisch sein, wie die Genauigkeit des Modells. Es gibt verschiedene Ansätze, die versuchen Modelle zu erklären. Mögliche Implementationen der Ansätze bieten beispielsweise das Local interpretable model-agnostic explanations-Framework (LIME) [44], Anchors [45] oder das Shapley Additive Explanations-Framework (SHAP) [46]. Im folgenden Kapitel wird die Theorie hinter SHAP beispielhaft erläutert. SHAP hat den Vorteil, dass die Implementation effizient ist. Sie ist besonders effizient in Bezug auf Baum basierte Modelle wie LightGBM, Random Forest oder XGBoost [47]. Des weiteren sind mit SHAP globale Modellinterpretationen möglich. Mit LIME oder mit Anchor sind nur lokale Modellinterpretationen möglich. Anders ausgedrückt lassen sich mit LIME oder Anchors nur eine einzelne Schätzungen zu einer Beobachtung auf einmal erklären [48]. Darüber hinaus baut SHAP auf LIME auf und inkludiert deshalb die Vorteile von LIME [48].

SHAP weist jedem Merkmal eines Modells eine Wichtigkeit für eine bestimmte Schätzung zu. Diese Wichtigkeit nennt man auch Shapley-Werte. Das Prinzip von SHAP basiert auf Shapley-Werten und auf der kooperativen Spieltheorie. In der Spieltheorie spielen Spieler ein Spiel und erhalten dafür einen bestimmten Gewinn. Eine einzelne Schätzung eines Klassifikationsproblems ist das Spiel, welches gespielt wird. Die Schätzung für dieses Klassifikationsproblem ist der Gewinn des Spiels. Die Spieler sind die Merkmalswerte, die für die Schätzung verwendet werden. Die Spieler arbeiten zusammen, um den geschätzten Gewinn zu erzielen. In einem Beispiel haben wir die Merkmale *Distanz*, *Winkel* und *Powerplay*. Diese haben die Werte 9 Meter für die *Distanz*, 30° für den *Winkel* und Falsch für *Powerplay*. Diese Merkmalswerte sind die Spieler, die zusammenarbeiten. Zusammen erzielen sie einen Gewinn von 0.5. Der durchschnittliche geschätzte Gewinn über alle Beobachtungen ist 0.75. Das Ziel der Shapley-Werte ist es, den Unterschied zwischen der geschätzten Wahrscheinlichkeit (0.5) und der durchschnittlichen geschätzten Wahrscheinlichkeit über alle Beobachtungen (0.75) zu erklären. Anders gesagt wird versucht die Differenz von -0.25 (0.5-0.75) zu erklären. Es stellt sich die Frage, wie die Shapley-Werte für einzelne Merkmale berechnet werden. Der Shapley-Wert für ein Merkmal ist der durchschnittliche marginale Beitrag eines Merkmalswertes über alle möglichen Kombinationen aus Merkmalen hinweg. Wenn man also den Shapley-Wert für den Merkmalswert von *Powerplay* berechnen möchte, dann kombinieren wir den Wert mit verschiedenen anderen Merkmalswerten für die restlichen Merkmale. In unserem Beispiel wäre das die *Distanz* und der *Winkel*. Für die Merkmalswerte des Merkmals *Distanz* wird zufällig eine andere Beobachtung aus unseren verfügbaren Daten gezogen und der ursprüngliche Wert 9 Meter mit dem Wert aus der zufällig gezogenen Probe ersetzt. Anschließend wird mit dem neuen Wert die Wahrscheinlichkeit vom Modell berechnet. Nehmen wir an, der Wert für *Distanz* aus der zufällig gezogenen Probe ist 5 Meter. Die restlichen Merkmale *Winkel* (30°) und kein *Powerplay* bleiben gleich. Die geschätzte Wahrscheinlichkeit ist 0.80. Anschließend ersetzen wir den Wert des Merkmals *Powerplay* ebenfalls mit dem Wert aus der gezogenen Probe. Dieses mal ist es ein *Powerplay* und wir berechnen die Wahrscheinlichkeit mithilfe des Modells neu. Die berechnete Wahrscheinlichkeit ist 0.9. Der Beitrag, der Das Merkmal *Powerplay* auf die Wahrscheinlichkeit hat, ist also 0.1 (0.9-0.8). Die Schätzung basiert auf den Werten der zufällig gezogenen Proben. Je öfters man diese Berechnung mit verschiedenen Werten durchführt und den durchschnittlichen Beitrag der einzelnen Merkmale davon berechnet, umso präziser wird der Shapley-Wert. Diese Schritte werden

für alle möglichen Kombinationen aus Merkmalswerten und Merkmalen wiederholt. Der Shapley-Wert ist der Durchschnitt aller marginalen Beiträge zu allen möglichen Kombinationen. Wenn man die Shapley-Werte für alle Merkmalswerte berechnet, erhält man die vollständige Verteilung der Schätzungen (abzüglich des Durchschnitts) für die Merkmalswerte [48].

Das SHAP-Framework implementiert die Berechnung der Shapley-Werte für verschiedene Data Mining Modelle und bietet darüber hinaus verschiedene Möglichkeiten die Ergebnisse visuell darzustellen [46].

## 4. Methodik

Um mit der Datenmenge an Ereignis- und auch Trackingdaten arbeiten zu können, benötigt es Methoden des Data Minings. Unter Data Mining versteht man das Extrahieren von nützlichen Informationen aus Daten. Um dies zu erfüllen, verwendet man Methoden aus der Statistik, dem maschinellen Lernen, der Datenbank- und Datenverwaltung, der Mustererkennung oder der künstlichen Intelligenz [49]. Ein Vorgehensmodell für Data Mining, welches branchenübergreifend als Standardprozess gilt, ist CRISP-DM [50]. In Abbildung 4.1 ist beispielhaft der Prozess für CRISP-DM dargestellt.

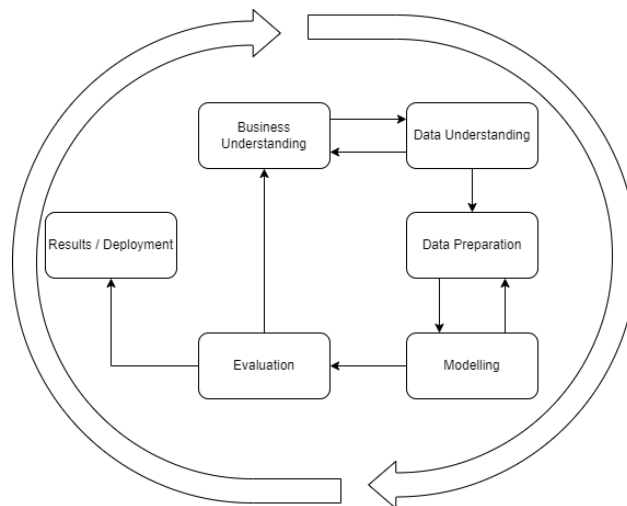


Abbildung 4.1.: Crisp-DM Modell

Im Folgenden werden die einzelnen Schritte des CRISP-DM Vorgehen näher erläutert:

1. Business Understanding: Die initiale Phase fokussiert sich auf das Verständnis der Projektziele aus der Geschäftssicht und das Konvertieren dieser in eine Definition eines Data Mining Problems.
2. Data Understanding: Diese Phase befasst sich mit dem Sammeln der Daten. Anschließend werden die Daten gesichtet, um sich mit diesen vertraut zu machen. So können erste Qualitätsprobleme identifiziert werden und ein erstes Verständnis der Daten entwickelt werden.
3. Data Preparation: Diese Phase umfasst alle Aktivitäten, um den finalen Datensatz zu erstellen.
4. Modelling: Hier werden verschiedene Modellverfahren ausgewählt und angewendet, und ihre Parameter werden optimiert.
5. Evaluation: An diesem Punkt hat man ein oder mehrere Modelle erstellt. Die Modelle und das Erstellen der Modelle werden überprüft, ob sie die Ziele des Projektes wie gewünscht erfüllen können.
6. Results: Die Ergebnisse eines ausgewählten Modells werden präsentiert und reflektiert.

Diese Schritte können in mehreren Iterationen wiederholt werden.

## 4.1. Business Understanding

Aus geschäftlicher Sicht ist es für einen Verein, einen Scout oder Trainer interessant die Wurfähigkeit eines Spielers oder eines Teams zu bewerten. Wie in Kapitel 1.1 beschrieben, ist das Ziel dieser Projektarbeit zwei Modelle zu entwickeln, welche die Wahrscheinlichkeit eines Treffers für einen Wurf berechnet. Dafür wird ein Klassifikator benötigt, der einen Wurf als Tor oder kein Tor klassifiziert und die Konfidenz der Klassenzugehörigkeit ausgibt. Dabei handelt es sich um ein binäres Klassifikationsproblem.

## 4.2. Data Understanding

In diesem Kapitel werden die vorhandenen Datensätze vorgestellt. Dabei wird darauf eingegangen, wie diese erfasst werden. Es stehen zwei verschiedene Arten von Daten zur Verfügung. Zum einen gibt es Ereignisdaten. Diese bestehen aus zwei verschiedenen Listen. Die erste Liste, umfasst eine Liste von Spielen. Die andere Liste enthält alle Würfe aufs Tor. Die Daten beider Listen beschränken sich dabei auf drei verschiedene Saisons der HBL [51]. Die betrachteten Saisons sind die Saison 2019-2020, 2020-2021 und 2021-2022. Aus der Saison 2019-2020 sind nur die Spiele verfügbar, die bis März 2020 ausgetragen wurden, da der Ligabetrieb aufgrund der Corona-Situation vorzeitig eingestellt wurde. Zum anderen stehen, durch eine Kooperation mit den beiden Teams SC DHFK Leipzig und Bergischer HC, die Positionsdaten (Raum-Zeit Sensordaten) aller Spieler pro Spiel aus den genannten Saisons zur Verfügung. Die Positionsdaten der anderen Teams sind nicht frei zugänglich, weshalb diese nicht verwendet werden können. Für dieses Projekt wird die Liste der Würfe aus den Ereignisdaten die Basis für die hier verwendeten Datensätze bilden. Dies hat mehrere Gründe. Zum einen lassen sich die Würfe nicht so leicht aus den Positionsdaten extrahieren. Zum anderen wären die Würfe, die extrahierbar wären, viel weniger, da die Trackingdaten nur von zwei Teams vorliegen.

Da nur von zwei Teams die Daten vorliegen, ist die Idee zwei Datensätze zu erstellen. Der erste Datensatz ( $P_t$ ) wird nur Daten aus der Liste der Würfe beinhalten. Das dazugehörige Modell soll als Basis für einen Vergleich dienen. Wenn man nur Ereignisdaten für ein Modell verwendet, hat es den Nachteil, dass keine Informationen ohne den Spielball erfasst werden. Diese sind aber ebenfalls wichtig im Spielverlauf, da die Spieler die meiste Zeit ohne Ball agieren [52]. Die hier verwendeten Ereignisdaten enthalten nur Würfe und es fehlen Informationen zu Schlüsselmomenten, wie beispielsweise das Erzeugen von Raum für einen Spieler durch einen Pass. Deshalb wird es einen zweiten Datensatz ( $P_t, K_t$ ) geben. Dieser wird die Liste der Würfe um zusätzliche Merkmale erweitern. Somit erhält ein Wurf zusätzlichen Kontext. Diese zusätzlichen Merkmale sollen aus den Positionsdaten erzeugt werden und somit zusätzliche Informationen inklusive Informationen zu Ereignissen ohne Ball beinhalten. Das dazugehörige Modell soll dann mit dem Modell des ( $P_t$ ) Datensatzes verglichen werden, um zu prüfen, ob zusätzliche Merkmale helfen akkuratere Schätzungen durch das Modell zu erstellen.

Die Daten werden erfasst und bereitgestellt von Kinexon. Kinexon ist eine Firma, die Trackingsysteme herstellt und bereitstellt. Kinexon bietet Lösung für verschiedene Sportarten, für die Industrie und für Entertainmentzwecke an, um Daten in Echtzeit zu erfassen [2].

Im folgenden werden die vorhandenen Daten vorgestellt, die von Kinexon bereitgestellt werden. Dabei wird nur auf Spalten aus den Dateien eingegangen, die in der Arbeit verwendet werden. In jeder Datei sind zusätzliche Spalten verfügbar, die eventuell nicht gefüllt werden, oder für die weitere Bearbeitung nicht relevant sind und deshalb hier nicht mit aufgeführt werden. Eine komplette Übersicht der verfügbaren Spalten, findet sich in Anhang B.

#### 4.2.1. Ereignisdaten

Die Ereignisdaten sind in zwei Listen aufgeteilt. Diese sind eine Liste zu den Spielen und eine Liste der Würfe der genannten Seasons. Die Liste der Spiele hat 357 Spiele gelistet. Sie enthält Informationen zu Datum, Spieltag, Heim- und Auswärtsmannschaft, der Saison und eine eindeutige Spiel-Id. Die Liste der Würfe enthält 30196 Einträge. Diese werden in der Regel automatisch durch das Trackingsystem von Kinexon erfasst und der HBL bereitgestellt. In den Fällen, wo das System einen Wurf nicht erkennt, wird dieser manuell durch einen Beobachter nachgetragen. Tabelle 4.2 zeigt die Anzahl der verfügbaren Daten zu Beginn dieses Projektes. Jeder Eintrag der Wurfliste enthält Informationen zu x- und y-Koordinaten des Abwurfs, den y- und z-Koordinaten des Einschlags des Balls im und um das Tor, die Distanz zum Tor, die Geschwindigkeit des Balls, das werfende Team, eine Id des Werfers, eine Id des verteidigenden Torhüters, die Id des Spiels, die Spielminute, einen Zeitstempel in Sekunden, den Erfolg des Wurfs und noch weitere Informationen. Ein Beispiel dazu findet sich in Tabelle 4.1.

Spaltenname	Zeile 1	Zeile 2	Zeile 3
player_id	651	280	164
timestamp	2021-10-30 16:32:42	2021-10-30 16:32:49	2021-10-30 16:33:32
success	0	1	1
shot_position_y	1.714587	5.067685	-0.98129
shot_position_x	10.600447	-17.752932	14.830914
distance	9.554654	5.543532	5.261405
game_clock	1:08	1:15	1:58
goalkeeper_id	272.0	439.0	272.0
id	122606	122664	122811
league_id	1290342.0	874068.0	544378.0
goalkeeper_league_id	317365.0	1315846.0	317365.0
team	SC DHFK Leipzig	Bergischer HC	SC DHFK Leipzig
team_id	7962	6268	7962
match_id	28449996.0	28449996.0	28449996.0

Tabelle 4.1.: Beispiel der Ereignisdaten. Dargestellt sind hier die Spalten der Datei in der ersten Spalte der Tabelle.

	Alle	Tore	Keine Tore	Spiele
Würfe	30196	11741	18455	357

Tabelle 4.2.: Anzahl der Würfe, davon Tore, davon keine Tore und die Anzahl der Spiele

### 4.2.2. Positionsdaten

Für dieses Projekt stehen nur die Positionsdaten von den beiden Bundesliga-Mannschaften SC DHfK Leipzig und vom Bergischer HC zur Verfügung. Somit werden nur die Spiele betrachtet, in denen beide Mannschaften gegeneinander gespielt haben. In Tabelle 4.3 ist eine Übersicht der entsprechenden Spiele aufgezeigt. Die Positionsdaten bestehen ebenfalls aus zwei verschiedenen Lis-

Datum	Spieltag	Heim-Mannschaft	Gast-Mannschaft	Saison
26.09.2019	7	SC DHfK Leipzig	Bergischer HC	2019-2020
14.02.2021	18	Bergischer HC	SC DHfK Leipzig	2020-2021
06.06.2021	27	SC DHfK Leipzig	Bergischer HC	2020-2021
30.10.2021	9	Bergischer HC	SC DHfK Leipzig	2021-2022
22.05.2022	31	SC DHfK Leipzig	Bergischer HC	2021-2022

Tabelle 4.3.: Übersicht der Spiele zwischen SC DHfK Leipzig und Bergischer HC

ten pro Team. Zum einen aus einer Positions-Datei und zum anderen aus einer Trägheits-Datei. Die Positions-Datei enthält Informationen jedes Spielers und des verwendeten Spielballs. Alle 50 Millisekunden wird pro Spieler und Ball ein Eintrag in der Datei angelegt. Jeder Eintrag enthält neben Informationen zur Aufzeichnungszeit, Sensor ID und Spieler ID auch Informationen zur Geschwindigkeit und ob ein Spieler im Ballbesitz ist. Aufgezeichnet werden die Daten durch einen Sensor. Jeder Spieler trägt während dem Spiel einen Sensor. Die Daten des Balls werden ebenfalls durch einen Sensor erfasst, der im Ball verbaut ist. Pro Spiel können bis zu drei verschiedene Bälle verwendet werden. Alle haben einen Sensor verbaut. Nur einer der drei Bälle ist zeitgleich auf dem Spielfeld und erzeugt somit Protokolleinträge. Die Positions-Datei hat pro Spiel und Team zwischen 600000 und 1200000 Einträge. Ein Beispiel hierzu findet sich in Tabelle 4.4. Die Einträge in der Trägheits-Datei enthalten ebenfalls Informationen zu jedem Spieler und des verwendeten Balls in Form der Sensor ID. Jeder Eintrag in der Datei besteht neben der Information zur Aufzeichnungszeit, aus der Position jedes einzelnen Spielers und des aktiven Spielballs. Die Positionen werden durch eine x- und y-Koordinate repräsentiert. Die Trägheits-Daten werden ebenfalls durch einen Sensor protokolliert. Dieser Sensor erfasst die Daten in einem Intervall von 66 Millisekunden. Die Trägheits-Datei hat pro Spiel und Team ebenfalls zwischen 600000 und 1200000 Einträge. Ein Beispiel findet sich in Tabelle 4.5.

Spaltenname	Zeile 1	Zeile 2	Zeile 3
ts in ms	1653220728550	1653223374450	1653226471550
sensor id	45740	151932	43564
mapped id	1147	379	1389
number	18	99	96
full name	Lovro Jotić	Ball2 Ball	Mohamed el Tayar
group name	SC DHfK Leipzig	Ball	SC DHfK Leipzig
speed in m/s	0.287	1.254	0.243
ball possession (id of possessed ball)	378	NaN	380

Tabelle 4.4.: Beispiel der Positionsdaten. Dargestellt sind hier die Spalten der Datei in der ersten Spalte der Tabelle.

ts in ms	sensor id	x in m	y in m
1653220565728	45824	13.331	4.641
1653220565794	45824	13.139	4.487
1653220565860	45824	12.956	4.338

Tabelle 4.5.: Beispiel der Trägheitsdaten

Die Trägheits- und Positions-Dateien enthalten noch zusätzliche Informationen pro Eintrag, die für die anschließende Bearbeitung nicht benötigt werden und deshalb in den Tabellen nicht aufgeführt werden.

Um die Daten zu erfassen, verwendet Kinexon ein lokales Positionierungssystem. Dieses besteht aus Sensoren, die der Spieler trägt und im Ball verbaut sind. Mithilfe von Antennen, die fest um das Spielfeld aufgebaut werden, können die Standortdaten der Sensoren zentimetergenau bestimmt werden. Im Vergleich zu anderen Trackingverfahren, wie zum Beispiel Kamera-Tracking, hat dies den Vorteil, dass die Qualität der Daten besser ist und die Daten mit einer geringeren Latenz erfasst werden [2].

### 4.3. Data Preparation

Nach dem ersten Sichten der Daten, werden diese genauer analysiert und bearbeitet, um die finalen Datensätze für das Erstellen der Modelle zu erhalten. Durch die unterschiedlichen Intervalle, in der die Ereignis- aber auch die verschiedenen Arten von Positionsdaten erfasst werden, wird die erste Herausforderung sein, die einzelnen Datensätze entsprechend zuzuordnen. Zum einen müssen die Trägheitsdaten zu den dazugehörigen Einträgen in den Positionsdaten zugeordnet werden. Zum anderen müssen die zusammengeführten Positionsdaten zu den dazugehörigen Würfeln aus den Ereignisdaten zugeordnet werden. Eine weitere Herausforderung wird es sein zusätzliche Merkmale für den Wurf zu definieren, die die Wahrscheinlichkeit für den Erfolg eines Wurfes beschreiben. Hier wird sich vor allem an den in Kapitel 2 vorgestellten Arbeiten orientiert. Dies hat den Grund, dass keine Handball-Experten zur Verfügung stehen und bisher keine vergleichbaren Arbeiten aus dem Handball-Umfeld veröffentlicht wurden. Anschließend werden die Daten bereinigt. Dazu gehört auch der Umgang mit fehlenden Werten.

#### 4.3.1. Kombinieren der Datensätze

Um zusätzliche Merkmale für die Würfe berechnen zu können, müssen die Daten aus der Trägheits- und Positions-Datei einem Wurf aus den Ereignisdaten zugeordnet werden können. Da alle Dateien einen Zeitstempel enthalten, wird der Zeitstempel verwendet um die entsprechenden Einträge zuzuordnen. Die Herausforderung hierbei liegt darin, die verschiedenen Intervalle, in denen die Einträge protokolliert werden, zu vereinheitlichen. Zunächst werden die Daten aus der Trägheits- und Positions-Datei zusammengeführt. Da die Einträge der Trägheits-Datei in 66 Millisekunden und die Einträge der Positions-Datei in 50 Millisekunden erfasst werden, wird der Zeitstempel aus der Trägheits-Datei auf die nächsten 50 Millisekunden gerundet. Somit haben beide Dateien einen Zeitstempel im 50 Millisekunden Intervall. Anschließend kann mithilfe der gerundeten



Zeit und der Sensor ID die x- und y-Koordinaten aus der Trägheits-Datei den Informationen aus der Positions-Datei zugeordnet und zusammengeführt werden. Es entsteht ein kombinierter Datensatz der Positionsdaten. Durch die unterschiedlichen Intervalle kommt es vor, dass zu einem Eintrag aus der Positions-Datei kein Eintrag aus der Trägheits-Datei zugeordnet werden kann. Ein Beispiel dazu findet sich in Tabelle 4.6. Hier kann aufgrund der unterschiedlichen Erfassungsintervalle kein Eintrag aus der Trägheits-Datei einem Eintrag der Positions-Datei mit dem Zeitstempel 1635614531250 zugeordnet werden. In diesem Fall sind die x- und y-Koordinaten leer. Diese Einträge werden entfernt, da die Koordinaten essentiell sind für weitere Berechnungen. Dies kann ohne Probleme erfolgen, da es nie mehrere aufeinanderfolgende Einträge sind. Somit entsteht pro Spieler in unregelmäßigen Abständen eine Lücke von 50 Millisekunden. Da die Reaktionszeit eines Menschen um die 300 Millisekunden liegt und somit länger als 50 Millisekunden ist, kann dies vernachlässigt werden [53].

<b>ts in ms Positions-Datei</b>	<b>ts in ms Trägheits-Datei</b>	<b>ts aus Trägheits-Datei auf 50 ms gerundet</b>	<b>ts gerundet in s</b>
1635614531200	1635614531221	1635614531200	1635614531000
1635614531250			1635614531000
1635614531300	1635614531287	1635614531300	1635614531000
1635614531350	1635614531353	1635614531350	1635614531000

Tabelle 4.6.: Beispiel für die Rundung der Zeitstempel inklusive einer entstehenden Lücke durch die Rundung

Anschließend müssen pro Wurf die entsprechenden Positionsdaten ermittelt werden. Da die Einträge in der Wurf-Datei sekundengenau erfasst werden, wird für die Positionsdaten ein weiterer Zeitstempel generiert. Dabei wird aus dem Zeitstempel im 50 Millisekunden Intervall ein Zeitstempel auf Sekunden gerundet generiert. Mithilfe des sekundengenauen Zeitstempels und der Spieler ID lassen sich die Würfe und die Positionsdaten dann entsprechend zuordnen. Ein Beispiel für den entstehenden Zeitstempel ist in der Tabelle 4.6 in der Spalte *ts gerundet in s* zu sehen. Man erhält zu jedem Wurf etwa 20 Einträge aus den Positionsdaten pro Spieler und Spielball auf dem Spielfeld. Von diesen 20 Einträgen wird für jeden Spieler und Ball der letzte Eintrag verwendet, um die Positionen zum Abwurfszeitpunkt zu bestimmen.

Die Spieler ID in den Wurfdaten und in den Positionsdaten sollten eigentlich gleich sein. Über die Seasons hinweg ändern sich diese für manche Spieler. In diesen Fällen wird eine zusätzliche Tabelle angelegt, in der die IDs aus den Wurfdaten manuell zu den IDs aus den Positionsdaten zugeordnet wurden.

#### 4.3.2. Erstellen zusätzlicher Merkmale

Im Handball und in anderen Sportarten verwenden Analysten diverse Konzepte, um das Spiel zu beschreiben und zu analysieren. Um diese Konzepte durch Algorithmen zu beschreiben, werden Regeln definiert. Die Konzepte basieren auf diesen Regeln, da es der schnellste Weg ist, das Wissen in Daten zu transformieren. Bei der Festlegung der Regeln wird sich auf Arbeiten aus Kapitel 2 bezogen.

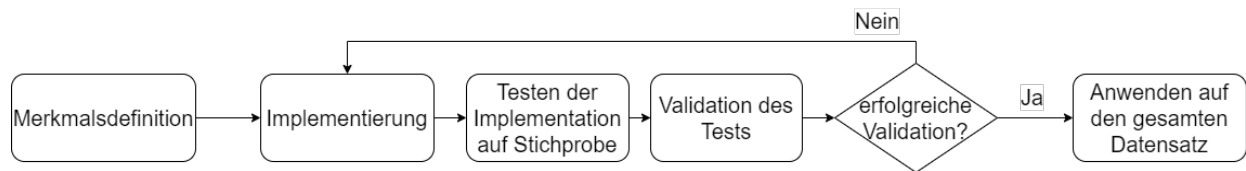


Abbildung 4.2.: Iterativer Prozess zum Erstellen von neuen Merkmalen

Der Prozess für die Erstellung jedes zusätzlichen Merkmals ist in Abbildung 4.2 gezeigt. Das Diagramm zeigt, dass es ein iterativer Prozess ist, da manche Merkmale nicht so leicht auf Anhieb durch Regeln zu definieren sind. Dies ist der allgemeine Ablauf, der für alle neuen Merkmale durchlaufen wird.

Der Prozess startet mit einer Definition der Regeln um das Merkmal in den Daten zu erkennen beziehungsweise zu berechnen. Sobald die initiale Definition der Regeln erfolgt ist, werden die Regeln als Algorithmus implementiert. Der Algorithmus erzeugt als Ergebnis einen Merkmalswert. Die Komplexität der Implementation hängt von der Komplexität der Regeln ab. Daher wird in der ersten Iteration versucht einfache Ansätze zu verwenden und diese, falls notwendig, zu erweitern. Dies hilft ebenfalls bei der Interpretation der Ergebnisse des Algorithmus. Je einfacher die Regeln zur Berechnung des Merkmals sind, desto einfacher ist es das Merkmal zu interpretieren.

Nach der Implementation der Regeln für ein Merkmal werden die Ergebnisse auf einer Probe der Daten geprüft. Die Berechnung der Ergebnisse aller Daten kann viel Zeit in Anspruch nehmen. Also wird der Algorithmus erst auf einer Teilmenge der Würfe getestet. Bei Merkmalen, die oft in den Daten vorkommen, reichen gegebenenfalls ein bis zwei Beispiele. Bei Merkmalen, die nur selten in den Daten vorkommen, benötigt man in der Regel mehrere Beispiele.

Anschließend werden die Merkmalswerte validiert. Dafür werden die Daten visualisiert. Dies hilft, falsch berechnete Merkmale zu identifizieren. Darüber hinaus unterstützt es ebenfalls, die Regeln in einem solchen Fall anzupassen.

Falls die ursprünglich definierten Regeln der Validation nicht standgehalten haben, werden die Regeln entsprechend angepasst. Somit wird versucht alle Fälle abzudecken. Die Änderungen werden anschließend in die Implementierung mit übernommen und der Prozess inklusive Test und Validation wird erneut durchlaufen bis alle Fälle richtig berechnet werden. Abschließend wird das Merkmal für alle Datensätze berechnet.

Mithilfe der Positionsdaten zu jedem Wurf, werden die zusätzlichen Merkmale berechnet, die den Wurf um zusätzlichen Kontext erweitern sollen. Um manche Merkmale zu berechnen müssen die Daten vorher grob bereinigt werden, um Fehler zu vermeiden. Diese initiale Bereinigung für die Berechnung der Merkmale ist in Kapitel 4.3.3 aufgezeigt.

Eine Liste aller Merkmale, die berechnet werden, und dazugehörige Beschreibung ist in der Tabelle 4.7 aufgelistet. Anstelle der Koordinaten für die Wurfposition zu verwenden, werden die Koordinaten in die euklidische Distanz, den sichtbaren Torwinkel und den Winkel zur Mitte der Torlinie umgewandelt. Ein visuelles Beispiel ist hierzu in Abbildung 4.3 aufgezeigt. Der Winkel zwischen den zwei roten Linien zeigt den sichtbaren Winkel des Tors. Der Winkel zwischen den zwei blauen Linien zeigt den Winkel zur Mitte. Die Vorzeichen der Koordinaten und auch das Tor, auf welches

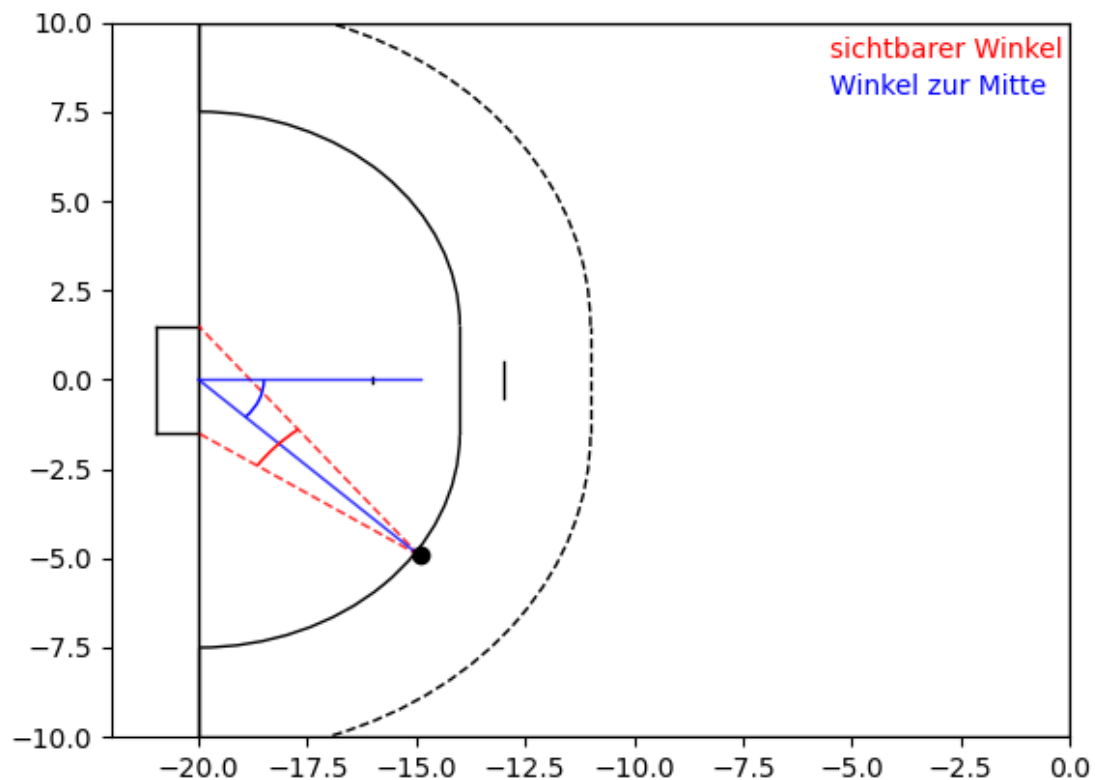


Abbildung 4.3.: Visuelle Darstellung der Winkel, die für jeden Wurf aus der Position des Werfers berechnet werden.

eine Mannschaft spielt, unterscheiden sich pro Mannschaft und pro Halbzeit. Die Repräsentation der Positionen durch die Winkel und die Distanz hat den Vorteil, dass sie einheitlich ist. Ein zusätzlicher Gedanke dahinter ist, dass das Modell durch die Vereinheitlichung der Wurfposition ähnliche Würfe besser erkennt und auch ähnlich klassifiziert. Des weiteren wird angenommen, dass je näher man am Tor ist, umso einfacher ist es, ein Tor zu erzielen. Aber auch zentral vor dem Tor zu sein, sollte hilfreich sein. Es ist bekannt, dass einige der Merkmale Korrelationen untereinander haben. Es werden dennoch alle Merkmale verwendet, da manche Klassifikatoren von den Informationen nicht linearer Beziehungen profitieren können.

Des weiteren wird die Anzahl der Spieler zwischen Werfer und Tor, der Winkel vom Wurf zum Torhüter und die Distanz vom Wurf zum Torhüter jeweils von zwei unterschiedlichen Positionen berechnet. Die erste Position ist die Position des Werfers aus den Positionsdaten. Die zweite Position ist die Position des Wurfes aus den Ereignisdaten, welche die Position des Balls darstellt. Es wird sich dazu entschieden, beide Positionen zu verwenden, da der Ball oft etwas vom Werfer entfernt geworfen wird und somit ein anderer Winkel kreiert wird. Beispielsweise kann der Werfer mit gestrecktem Arm um einen Gegenspieler herum werfen. Dies ist in den Daten zu erkennen. Ein Beispiel dazu findet sich in Abbildung 4.4. Hier kann man erkennen, dass die Position des Werfers und die Position des Wurfes aus den Ereignisdaten sich unterscheiden. Es ist zu beobachten, dass

der Werfer einen Gegenspieler vor sich hat. Aus der Position des Balls hingegen ist der Winkel etwas anders und kein Verteidiger zwischen Abwurfposition und Tor.

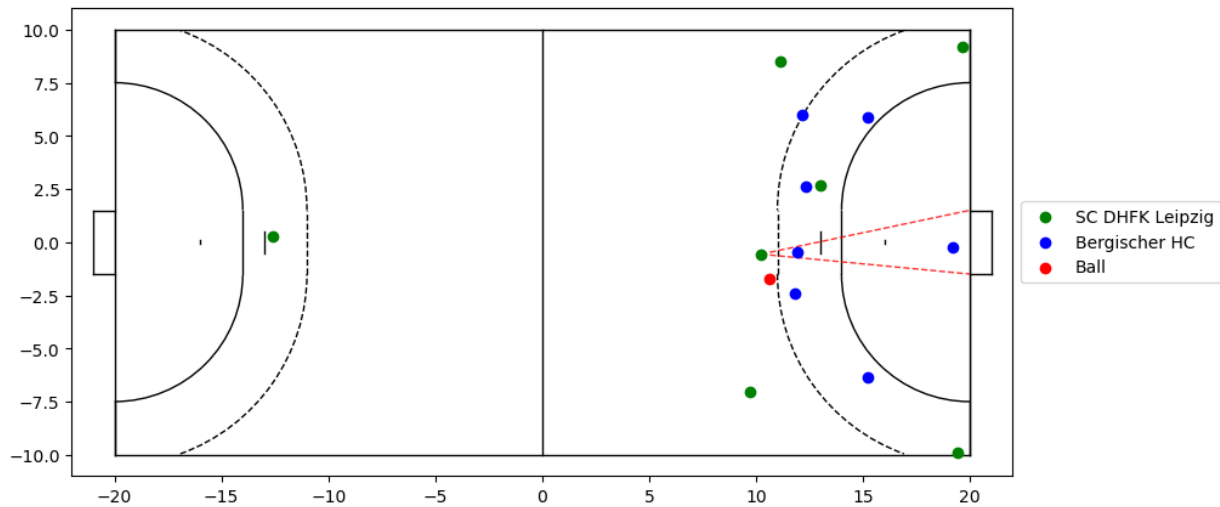


Abbildung 4.4.: Darstellung der Positionsdaten zu einem Wurf eines einzelnen Zeitpunktes. Dargestellt sind alle Spieler durch ihre Position mithilfe der aufgezeichneten x- und y-Koordinate. Die blauen und grünen Punkte stehen jeweils für einen Spieler eines Teams. Zusätzlich dazu ist die Position des Balls in rot dargestellt. Der Werfer, ist der Spieler, von dem die roten Linien ausgehen.

Die Merkmale Spielergeschwindigkeit, Ballbesitz und Anzahl der Pässe werden ausgewählt, um mögliche Konter bzw. Tempogegenstöße indirekt zu erfassen. Beispielsweise sollte bei einem abgefangenen Ball und anschließendem Konter, der Ballbesitz in Sekunden relativ gering, die Anzahl der Pässe ebenfalls gering und die Geschwindigkeit des Spielers hoch sein, da er auf das Tor zu sprintet. Die Merkmale Pressure-Index, Distance to the closest Defender sind ebenfalls Merkmale die auf der gleichen Annahme basieren. Ist der Spielzug ein Konter, sollte der nächste Verteidiger weiter entfernt sein, beziehungsweise der Pressure-Index gering sein, da weniger gegnerische Spieler um den Werfer im 1-Meter Radius stehen.

Die Merkmale Powerplay und Empty-Net werden ausgewählt um temporäre Überzahl-Situationen im Fall einer Strafe oder durch das Herausnehmen des Torwarts zu beschreiben. Wird der Torwart vom Spielfeld genommen, hat das werfende Team mehr Anspielstationen, um Spielmöglichkeiten zu schaffen. Das Merkmal Empty-Net gibt an, ob das Tor leer ist, da der Torwart vom Spielfeld genommen wurde. Bei einem Abfangen des Balls der gegnerischen Mannschaft, wäre das Tor somit bei einem Wurf frei und die Chance zu treffen könnte höher sein als mit Torwart. In Abbildung 4.5 ist für eine Powerplay- (4.52) und eine Empty-Net-Situation 4.51 ein Beispiel aufgezeigt.

Auf eine explizite Klassifizierung der Würfe nach Spielsituation, wie beispielsweise ein Konter oder ein Wurf nach einem Steckpass, wird verzichtet, da die Expertise fehlt die Situationen auf Basis der Positionsdaten allein zu beurteilen.

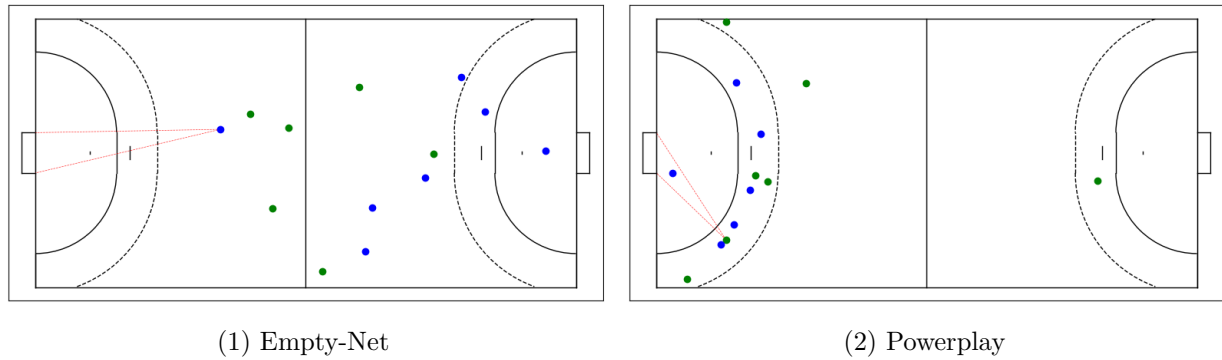


Abbildung 4.5.: Beispiel für eine Empty-Net Situation (4.51) und einer Powerplay-Situation (4.52).

Merkmal	Beschreibung
<b>Distanz</b>	Gibt die euklidische Distanz zwischen der Wurfposition aus den Ereignisdaten und der Mitte der Torlinie an.
<b>visible_angle</b>	Gibt den Winkel zwischen der Wurfposition und den beiden Torpfosten im Bogenmaß an.
<b>angle_to_mid</b>	Gibt den Winkel zwischen der Wurfposition und der Mitte der Torlinie im Bogenmaß an.
<b>time_between_shots_s</b>	Anzahl der Sekunden zwischen einem Wurf und dem vorherigen Wurf der gleichen Mannschaft. Der erste Wurf jedes Spiels jedes Teams wird mit 0 initialisiert.
<b>pressur_index</b>	Anzahl der Spieler um den Werfer in einem 1-Meter Radius
<b>dist_closest_def</b>	Die euklidische Distanz in Meter zum nächsten Spieler aus der gegnerischen Mannschaft
<b>players_in_shot_angle_from_shot</b>	Gibt die Anzahl der gegnerischen Spieler (inklusive gegnerischem Torwart) in der Fläche zwischen dem Werfer und den beiden Torpfosten an. Als Basis für die Berechnung dient die Position des Wurfes aus den Ereignisdaten.
<b>players_in_shot_angle_from_player</b>	Gibt die Anzahl der gegnerischen Spieler (inklusive gegnerischem Torwart) in der Fläche zwischen dem Werfer und den beiden Torpfosten an. Als Basis für die Berechnung dient die Position des Wurfers aus den Positionsdaten.

<b>powerplay</b>	Gibt an, ob die werfende Mannschaft mindestens einen Feld-Spieler mehr auf dem Feld hatte als die gegnerische Mannschaft.
<b>shooter_goalkeeper_dist</b>	Gibt die Distanz des Werfers zum gegnerischen Torwart an, falls dieser auf dem Feld ist. Ansonsten ist die Distanz 0. Als Basis für die Berechnung der Distanz dient die Position des Werfers aus den Positionsdaten.
<b>shot_goalkeeper_dist</b>	Gibt die Distanz des Werfers zum gegnerischen Torwart an, falls dieser auf dem Feld ist. Ansonsten ist die Distanz 0. Als Basis für die Berechnung der Distanz dient die Position des Wurfes aus den Ereignisdaten.
<b>shooter_goalkeeper_angle</b>	Gibt den Winkel des Werfers zum gegnerischen Torwart im Bogenmaß an, falls dieser auf dem Feld ist. Ansonsten ist der Winkel 0. Als Basis für die Berechnung der Distanz dient die Position des Werfers aus den Positionsdaten.
<b>shot_goalkeeper_angle</b>	Gibt den Winkel des Werfers zum gegnerischen Torwart im Bogenmaß an, falls dieser auf dem Feld ist. Ansonsten ist der Winkel 0. Als Basis für die Berechnung der Distanz dient die Position des Werfers aus den Ereignisdaten.
<b>empty_net</b>	Gibt an, ob der gegnerische Torwart im Tor stand oder nicht.
<b>player_speed</b>	Gibt die Geschwindigkeit des Spielers in Meter/-Sekunde an.
<b>number_of_passes</b>	Gibt die Anzahl der Pässe eines Teams vor dem Wurf an. Die Pässe werden nach einem Ballverlust erneut von 0 an gezählt.
<b>ball_possession</b>	Anzahl der Sekunden, die die werfende Mannschaft den Ball im Besitz hatte, bevor der Wurf erfolgt.

Tabelle 4.7.: Liste der verwendeten Merkmale inklusive Beschreibung

Für manche in Tabelle 4.7 aufgeführten Merkmale werden zusätzliche Merkmale temporär berechnet und später wieder verworfen, da diese nicht mehr benötigt werden. Dies gilt vor allem für Zeitstempel, da diese aus verschiedenen Formaten in ein einheitliches Format transformiert werden müssen.

### 4.3.3. Bereinigen der Daten

In diesem Kapitel wird beschrieben, wie mit fehlenden Werten in den Daten umgegangen wird. Die Datenbereinigung für dieses Projekt kann in zwei Teile aufgeteilt werden. Zum einen müssen Daten aus den Position- und den Ereignisdaten vor der Berechnung der neuen Merkmale bereinigt werden. Zum anderen wird nach dem Erstellen der Datensätze ( $P_t$ ) und ( $P_t, K_t$ ) mit fehlenden Werten umgegangen.

Vor dem Berechnen der zusätzlichen Merkmale werden aus den Ereignisdaten alle Würfe entfernt, die in der Halbzeitpause oder vor dem Spiel beim Aufwärmen erfolgen. Manche Würfe haben keine Zuordnung zu einem Spiel. Die fehlende Spiel ID wird über den Zeitstempel ergänzt. Wie schon in Kapitel 4.2.1 beschrieben, werden manche Würfe manuell erfasst. In solchen Fällen ist die Wurfposition nicht gegeben. Allerdings ist der werfende Spieler angegeben. In diesem Fall, wird die Wurfposition aus den Positionsdaten mithilfe der Spieler ID des Werfers und dem Zeitstempel gezogen. Auf Basis dieser Position werden anschließend alle Merkmale, die in Tabelle 4.7 gelistet sind, berechnet. Hier ist die Position des Werfers und die Position des Balles identisch. Falls keine Positionsdaten für den Wurf zugeordnet werden können, wird der Wurf aus dem Datensatz entfernt. Für diesen Wurf gibt es keinerlei Informationen zu der Position des Abwurfes. Damit ist der Wurf nicht brauchbar für die Modelle, die zumindest auf der Distanz und dem Winkel zum Tor basieren.

In den Positionsdaten kann es durch die Zuordnung der zwei verschiedenen Dateien, zu Einträgen kommen, die keine Koordinaten haben. Diese werden entfernt. Wie schon in Kapitel 4.3.1 beschrieben, entsteht so in der Zeitreihe der Sensordaten eine Lücke von 50 Millisekunden, die ignoriert werden kann. Ein weiteres Problem liegt in der Zuordnung des Werfers aus den Ereignisdaten zu dem Werfer aus den Positionsdaten. In sieben Fällen ist der Werfer aus den Ereignisdaten nicht die ballführende Person in den Positionsdaten. In diesem Fall wird der Wurf aussortiert, da keine Merkmale berechnet werden können.

Nach der Berechnung der Merkmale müssen beide Datensätze für die Berechnung der Modelle bereinigt werden. Als erstes wird das Merkmal für die Zeit zwischen den Würfen bereinigt. Da für den ersten Wurf jedes Spiels pro Team keine Dauer zu dem vorangegangenen Wurf berechnet werden kann, wird der Wert mit Null initialisiert. Für zehn Würfen steht keine Spielergeschwindigkeit zur Verfügung. In diesem Fall fehlt der Wert komplett zufällig. In einem solchen Fall gibt es verschiedene Optionen. Es wird sich dafür entschieden, die fehlenden Werte mit dem Durchschnitt aller Einträge des Datensatzes aufzufüllen. Dies hat den Nachteil, dass Werte überrepräsentiert sein können. Bei zehn Würfen wird dieses Risiko für gering erachtet, weshalb diese Option gewählt wird. Anschließend werden alle Einträge aus den Datensätzen verworfen, die für ein beliebiges Merkmal keinen Wert haben. Für den Datensatz  $P_t, K_t$  werden sieben Würfe entfernt. Es sind die sieben Würfe, bei denen kein Werfer aus den Positionsdaten zugeordnet werden kann. Für den Datensatz  $P_t$  wird eine Zeile entfernt. Alle anderen nicht vollständigen Würfe wurden durch das Entfernen der Würfe vor dem Spiel und in der Halbzeitpause entfernt.

#### 4.3.4. Datensätze

Nach dem Bereinigen der Daten gibt es zwei Datensätze  $P_t$  und  $P_t, K_t$ . In Tabelle 4.8 sind die Merkmale für  $P_t$  in der entsprechenden Spalte aufgeführt. Der Datensatz  $P_t, K_t$  enthält zusätzlich zu den Merkmalen  $P_t$  auch die aus  $K_t$ . Die Aufteilung der Merkmale wurde so gewählt, dass sich die Merkmale für  $P_t$  ausschließlich aus den Ereignisdaten erstellen lassen. Wohingegen die restlichen Merkmale auf den Positionsdaten aller Spieler basieren. Der bereinigte Datensatz  $P_t$  besteht aus 13796 Würfeln, von dem 8770 Tore sind. Der zweite Datensatz  $P_t, K_t$  enthält 430 Würfe von denen 273 Tore sind.

$P_t$	$K_t$
Distanz	pressur_index
visible_angle	dist_closest_def
angle_to_mid	players_in_shot_angle_from_shot
	players_in_shot_angle_from_player
	powerplay
	shooter_goalkeeper_dist
	shot_goalkeeper_dis
	shooter_goalkeeper_angle
	shot_goalkeeper_angle
	empty_net
	player_speed
	number_of_passes
	ball_posession
	time_between_shots_s

Tabelle 4.8.: Merkmale aufgeteilt nach Datensätzen

#### 4.4. Modellierung

Bevor der Vorgang für das Erstellen der Modelle erläutert wird, wird das vorliegende Problem formal beschrieben. Die Absicht ist es, einen Klassifikator zu trainieren, der in der Lage ist, eine Zufallsvariable  $X$  mithilfe von verschiedenen Informationen zu modellieren. Die Zufallsvariable kann die Zustände Tor oder kein Tor annehmen. Dies kann beschrieben werden durch den folgenden Ausdruck  $expectedGoals_t = E[X|P_t, K_t, A_t = Wurf]$ , wobei  $P_t$  die Position des Spielers, der im Ballbesitz ist, zum Zeitpunkt  $t$  darstellt.  $K_t$  stellt die Kontextinformationen zum Zeitpunkt  $t$  dar und  $A_t$  ist die Aktion, die der werfende Spieler zum Zeitpunkt  $t$  ausführt. Dabei ist zu beachten, dass die Formel nur gilt, wenn die ausgeführte Aktion des Spielers ein Wurf ist. Angesichts des Bernoulli-Charakters der Zufallsvariable  $X$  kann der, in der Formel ausgedrückte Erwartungswert, wenn er richtig kalibriert ist, als die Wahrscheinlichkeit interpretiert werden, dass  $X$  gleich eins beziehungsweise ein Tor ist. Die Tatsache, dass der Erwartungswert als Wahrscheinlichkeit interpretiert werden kann, ist hilfreich, da man die Wahrscheinlichkeiten der einzelnen Würfe aufsummieren kann und somit die erwartete Anzahl der Tore ermitteln kann, die eine Mannschaft oder ein Spieler in einem Spiel oder pro Saison erzielt haben könnte. Die Ergebnisse sind nicht nur auf die Wahrscheinlichkeiten von Würfeln, die zu Toren werden, limitiert. Jede Ballaktion kann man



mit dem Modell bewerten, auch wenn er nicht geworfen hat, um zu untersuchen, ob der Spieler besser geworfen hätte. Für das Training des Modells werden allerdings nur Würfe verwendet, da diese ein eindeutiges Kennzeichen besitzen, ob ein Wurf ein Tor war oder nicht.

Aus den vorangegangenen Beschreibungen kann man erkennen, dass man für das Problem einen binären Klassifikator benötigt. Neben den Merkmalen, die den Wurf beschreiben, gibt es für jeden Wurf einen Indikator, welcher angibt, ob der Wurf ein Erfolg - also ein Tor - ist oder nicht. Dies ist unsere Zielvariable für den Klassifikator. Darüber hinaus benötigen wir eine Wahrscheinlichkeit, die angibt mit welcher Wahrscheinlichkeit ein Wurf zu der Klasse „Tor“ gehört.

Da es zu Beginn schwer ist zu wissen, welcher Klassifikator das beste Ergebnis für unser Problem liefert, werden vier binäre Klassifikatoren ausgewählt. Der erste Klassifikator ist die logarithmische Regression. Dieser wird regelmäßig für ähnliche Probleme verwendet und funktioniert gut mit linearen Abhängigkeiten [12][10]. Der nächste Klassifikator ist XGBoost. Er ist ein Tree-Boost Modell, welches gut mit großen Datensätzen funktioniert und in ähnlichen Arbeiten im Fußball gute Ergebnisse erzielt [54]. Eine Alternative zu XGBoost bietet LightGBM. Dieser Klassifikator ist ebenfalls ein Tree-Boosting Modell, welches etwas effizienter im Vergleich zu XGBoost in der Berechnung des Modells ist [38]. Der vierte und letzte Klassifikator ist Random Forest. Es basiert auf einem Tree-Ensemble Modell. Es wird gewählt, da es einen anderen Ansatz zu den vorher genannten Klassifikatoren bildet und ebenfalls gute Ergebnisse in anderen Forschungsarbeiten erzielen konnte [11].

Diese vier Klassifikatoren werden jeweils für beide Datensätze aus Kapitel 4.3.4 trainiert. Dadurch sollen Unterschiede in der Leistungsfähigkeit der Klassifikatoren mit verschiedenen Parametern untersucht werden. Darüber hinaus stehen somit verschiedene Interpretationen zur Verfügung, womit unterschiedliche Ansätze analysiert werden können. Somit lässt sich ebenfalls analysieren welche Merkmale wichtig für die Klassifikation eines Tores sind.

Vor dem Training werden die Datensätze jeweils in einen Trainings- und Testdatensatz unterteilt. Somit kann untersucht werden, wie die Modelle auf Daten reagieren, mit denen sie nicht trainiert werden. Bei der Aufteilung der Datensätze wird darauf geachtet, dass beide Partitionen den gleichen Anteil der möglichen Klassen enthalten. Dies hilft dabei, einen Bias des Modells zu verhindern, welcher durch die unterschiedliche Verteilung der Tore zu Nicht-Tore entstehen kann [35, S. 203-205].

Alle numerischen Merkmale werden zusätzlich standardisiert. Das standardisieren von Werten kann dazu beitragen das die Modelle stabilere Ergebnisse liefern, da das Modell keine Merkmale bevorzugt. Wenn man die Daten nicht standardisiert können manche Merkmale bevorzugt werden, da die Skala der möglichen Werte größer ist als die von anderen Merkmalen [35, S. 182–183].

Da für die Klassifikatoren eine Vielzahl von Hyperparametern für das Training eingestellt werden können und sollten, werden diese über eine zufällige Suche ausgewählt. Dabei werden mögliche Kombinationen aus den verschiedenen Hyperparametern zufällig ausgewählt und das Modell trainiert. Das Training erfolgt auf dem Trainingsdatensatz. Mithilfe der Trainingsdaten und einer 10-fold Cross-Validation werden die Hyperparameter evaluiert. Dabei wird ebenfalls auf eine gleichmäßige Verteilung der Klassen für jede der 10 Faltung geachtet. Diese Schritte werden 20 mal wiederholt.

Die Kombination an Hyperparametern, die das beste Ergebnis erzielt hat, wird gespeichert. Das beste Ergebnis wird mithilfe des logarithmischen Verlustes ermittelt. Eine Übersicht über die möglichen Hyperparameter und die Intervalle der dazugehörigen Werte, die für die randomisierte Suche zur Verfügung stehen, sind in Tabelle 4.9 aufgelistet. Es wird eine randomisierte Suche verwendet, um die Rechenzeit für die Auswahl der Hyperparameter gering zu halten.

Klassifikator	Hyperparameter	Werte-Bereich
<b>lightgbm</b>	max_depth	[1,15)
	num_leaves	[0,200)
	learning_rate	[0,0.5]
<b>xgboost</b>	max_depth	[1,15)
	max_leaves	[0,200)
	eta	[0,1]
<b>logistic Regression</b>	C	[0,2000]
<b>Random Forest</b>	n_estimators	[1,200)
	max_leaf_nodes	[0,200)
	max_depth	[1,15)

Tabelle 4.9.: Übersicht über die Hyperparameter pro Klassifikator inklusive Wertebereich

Ist ein optimiertes Modell mit Hyperparametern gefunden, werden die Ergebnisse der Cross-Validation ausgewertet. Für die Ergebnisse dieser Klassifikation werden die Metriken aus Kapitel 3.1.2 berechnet, die gespeichert und später ausgewertet werden. Diese bilden die Werte für das Training der einzelnen Modelle. Anschließend wird das gleiche mit dem Testdatensatz für die Evaluation durchgeführt. Mithilfe der Ergebnisse aus dem Training und der Evaluation der Modelle, wird für jeden Datensatz ein Modell ausgewählt. Die beiden ausgewählten Modelle werden anschließend mit einer Kalibrierung optimiert. Für die Kalibrierung werden die Varianten ohne Kalibrierung, sigmoid und isotonic pro Modell berechnet und verglichen. Der Vergleich der Kalibrierung erfolgt über den erwarteten Kalibrierungs Fehler (ECE).

Zu Beginn des Projektes wurde über ein Over- bzw. Undersampling der Daten nachgedacht, um eine ausgeglichene Klassenverteilung innerhalb der Datensätze zu erzielen. Für das Oversampling zeigten die ersten Tests, dass dadurch schlechtere Ergebnisse erzielt werden. Der Datensatz  $P_t, K_t$  ist für ein Undersampling zu klein. Es würden zu wenige Würfe übrig bleiben. Somit werden beide Ansätze für das Training der Modelle nicht berücksichtigt und verworfen.

Ebenso wurde anfangs über eine automatische Feature Selection nachgedacht. Aber auch hier stellte sich in den ersten Tests heraus, dass die Modelle dadurch eher schlechtere Ergebnisse erzielen. Somit wird der Ansatz ebenfalls verworfen.

Des weiteren ist während der ersten Tests aufgefallen, dass die Ergebnisse des Trainings und des Tests der Modelle oft sehr abhängig vom Zufall sind. Dies ist vor allem beim  $P_t, K_t$  Datensatz der Fall. Je nachdem, welche Würfe für das Training beziehungsweise den Test verwendet werden, können sich die Ergebnisse stark unterscheiden. Das Verhalten tritt trotz gesetztem Seed für die Zufallsauswahl auf.

## 4.5. Implementation

Für die Implementation des gesamten Projekts wird Python [55] in Jupyter Notebooks verwendet. Für die Verarbeitung der Daten und das Erstellen der Modelle werden die Programmier-Frameworks Pandas, NumPy, SciPy, Scikit-Learn, XGBoost und LightGBM verwendet. Pandas ist eine Programm-Bibliothek für Python um Daten in geeigneten Datenstrukturen zu bearbeiten und zu analysieren. Es bietet verschiedene Möglichkeiten zur Bearbeitung von Daten wie beispielsweise das Zusammenführen oder das Umformen von Daten [56]. NumPy ist ein Framework für den Zugriff, für das Bearbeiten und für das Bedienen von Daten in Vektoren, Matrizen oder auch mehrdimensionale Arrays [57]. Scikit-Learn ist eine Programm-Bibliothek für Python, die Methoden für viele verschiedene Data-Mining Aufgaben bereitstellt. Darunter sind Methoden für Regression, Klassifikation, unüberwachte Lernverfahren oder die Vorverarbeitung von Daten [58]. SciPy ist eine Bibliothek für numerische Verfahren für Python und basiert auf NumPy. Es stellt Methoden für das Modellieren und lösen von wissenschaftlichen Problemen bereit [59]. XGBoost [37] und LightGBM [38] sind jeweils Implementationen für Python für die jeweiligen Algorithmen.

Für die Visualisierung der Ergebnisse kommen die Frameworks Matplot-Lib, Seaborn und SHAP zum Einsatz. Matplot-Lib ist eine Programm-Bibliothek um verschiedene zweidimensionale Diagramme von Daten aus verschiedenen Datenstrukturen zu plotten [60]. Ganz ähnlich ist das Framework Seaborn. Es basiert auf Matplot-Lib und ergänzt es um einige Diagramme [61]. Für das Plotten des Handball-Spielfeldes wird Floodlight verwendet. Floodlight ist ein Framework für die gezielte Analyse von Sport-Daten [62]. Zusätzlich dazu wird SHAP verwendet, um die erstellten Modelle zu erklären und zu interpretieren. SHAP ist eine Programm-Bibliothek für Python, die Methoden für das Erklären von Modellen bereitstellt. Es hilft dabei zu erläutern, welche Merkmale für bestimmte Schätzungen relevant sind. Darüber hinaus bietet es verschiedene Methoden, beliebige Modelle visuell zu erläutern [46].

Die Implementation dieses Projektes ist aufgeteilt in die Daten, in finalen Modelle, in Jupyter-Notebooks und in die geplotteten Diagramme und Abbildungen. Diese Aufteilung findet sich ebenfalls in der Ordnerstruktur (siehe Anhang A) wieder.

In den Daten finden sich die finalen Datensätze, die für das Training der Modelle verwendet werden, und die Liste, die die Spieler IDs aus den Ereignisdaten den Spieler IDs aus den Positionsdaten zuordnet. Die Rohdaten, die für das Erstellen der Datensätze verwendet werden, liegen in einem anderen Verzeichnis, da es rechtlich nicht gestattet ist diese zu kopieren. Dieses Projekt ist aufgeteilt in verschiedene Jupyter-Notebooks, um einzelne Projektschritte voneinander abzugrenzen. Das erste Notebook DataUnderstanding ist für das erste Verständnis der Daten notwendig. Hier werden die Ereignis- und Positionsdaten visualisiert. Dieses Notebook wird ebenfalls verwendet, um neue Merkmale zu testen und zu visualisieren. Anschließend daran ist das Notebook DataPreparation\_CreateDatasets. Dieses wird verwendet, um die Datensätze aus den Rohdaten zu erstellen und erste Schritte der Bereinigung durchzuführen. Hier werden nur die Werte bereinigt, die für die Berechnung der Merkmale nötig sind. Die Berechnung der Modelle inklusive Training, Validierung, Evaluation und Kalibrierung ist im Notebook CalculateModels-CrossValidation-preShot zu finden. Dieses beginnt mit einer Bereinigung der Daten. Alle temporären Merkmale, alle IDs oder Zeitstempel, werden entfernt. Danach werden die Modelle pro Datensatz berechnet und ausgewertet.

Zum Schluss des Notebooks werden die Ergebnisse des Trainings und der Evaluation ausgegeben und gespeichert. Die Analyse der Anwendungsfälle wird im Notebook CaseStudie durchgeführt. In dem Modell-Ordner findet man alle berechneten Modelle pro Datensatz. Der gesamte Programmcode ist in einem GitLab-Repository der Universität Leipzig verfügbar.<sup>1</sup> Die Datenvorverarbeitung, die Berechnung der Modelle und die Fallstudie erfolgen auf dem High Performance Cluster der Universität Leipzig.

---

<sup>1</sup><https://git.informatik.uni-leipzig.de/rf65isex/handball-expected-goals/-/tree/main>

## 5. Evaluation

Dieses Kapitel befasst sich mit den Ergebnissen des Trainings und der Evaluation der Klassifikatoren, die in Kapitel 4.4 ausgewählt wurden. Um die Modelle zu evaluieren, werden verschiedene Methoden verwendet. Als erstes werden die verschiedenen Bewertungsmetriken der Modelle verglichen. Anschließend wird eine Kalibrierung der Modelle vorgenommen und diskutiert. Abschließend werden die Modelle interpretiert.

Wie in Kapitel 3.1.2 beschrieben, werden für die Bewertung der Klassifikatoren der logarithmische Verlust (log loss) und der AUC-Wert berechnet. Primär wird der logarithmische Verlust für die Auswahl des besten Modells betrachtet. Der AUC-Wert dient als Ergänzung und wird betrachtet, wenn unterschiedliche Modelle einen nahezu gleichen logarithmischen Verlust haben.

### 5.1. Training der Modelle

Nach dem Training der Modelle, welches in Kapitel 4.4 beschrieben wurde, erhalten wir die Ergebnisse aus Tabelle 5.1. Wie vorher beschrieben werden vier Modelle auf zwei verschiedenen Datensätzen mit unterschiedlichen Mengen an Merkmalen trainiert. Die Metrik für die Evaluation, die in der Tabelle dargestellt ist, ist der logarithmische Verlust (log loss) und der AUC-Wert (AUC). Die Werte in der Modell-Spalte referenzieren auf die unterschiedlichen Datensätze aus Kapitel 4.3.4, die angeben, welche Merkmale und welcher Datensatz für das Erstellen des Modells verwendet werden.

Trainingsergebnisse								
Modell	log Reg		Random Forest		XGBoost		LightGBM	
	log loss	AUC	log loss	AUC	log loss	AUC	log loss	AUC
$P_t$	0.6428	0.6655	0.6075	0.6905	0.6068	0.6834	0.6075	0.6852
$P_t, K_t$	0.5457	0.8007	0.5620	0.7902	0.5333	0.8601	0.5366	0.8252

Tabelle 5.1.: Zusammenfassung der Trainingsergebnisse der Modelle

Die Trainingsergebnisse sind hilfreich, um zu prüfen, ob ein Modell beim Training zu einer Überanpassung neigt oder nicht. Ein großer Unterschied zwischen Trainings- und Testergebnissen, ein logarithmischer Verlust von 0 oder ein AUC-Wert von 1.0 beim Training wären Indizien dafür. Betrachtet man die Ergebnisse aus Tabelle 5.1, so sieht man, dass auf den ersten Blick keins der Modelle zu einer Überanpassung neigt. Ein weiterer Vergleich der Trainingsdaten mit den Testergebnissen erfolgt in Kapitel 5.2, um dies genauer zu prüfen.

Interessant ist, dass der logarithmische Verlust für alle Modelle kleiner wird, wenn mehr Merkmale für das Training verwendet werden. Ganz allgemein, sind die Ergebnisse aller Modelle relativ eng beieinander. Betrachtet man den AUC-Wert, so wird dieser für alle Modelle besser, wenn mehr Merkmale für das Training verwendet werden. Beim Training scheint XGBoost für beide Datensätze das beste Ergebnis zu erzielen. Für den Datensatz  $P_t$  folgt darauf das Random Forest- und das LightGBM-Modell mit sehr ähnlichen Ergebnissen wie das XGBoost-Modell. Das Random Forest-Modell erzielt allerdings den höchsten AUC-Wert. Das schlechteste Ergebnis liefert hier die

logistische Regression. Anders ist dies für den Datensatz  $P_t, K_t$ . Hier folgt auf das XGBoost-Modell das LightGBM-Modell. Das drittbeste Ergebnis erzielt die logistische Regression. Das Random Forest-Modell schneidet am schlechtesten ab.

## 5.2. Testen der Modelle

Die trainierten Modelle werden, wie in Kapitel 4.4 beschrieben, mithilfe des Testdatensatzes evaluiert. Auch hier werden, wie beim Training, die beiden Metriken logarithmischer Verlust und der AUC-Wert für die Evaluation berechnet. Die Ergebnisse des Tests sind in Tabelle 5.2 aufgeführt. Mithilfe dieser Werte wird für jeden Datensatz das beste Modell ausgewählt.

Evaluationsergebnisse								
Modell	log Reg		Random Forest		XGBoost		LightGBM	
	log loss	AUC	log loss	AUC	log loss	AUC	log loss	AUC
$P_t$	0.6505	0.6254	0.6230	0.6507	0.6216	0.6530	0.6234	0.6487
$P_t, K_t$	0.5744	0.7455	0.5551	0.7742	0.5380	0.7944	0.5372	0.8050

Tabelle 5.2.: Zusammenfassung der Evaluationsergebnisse der Modelle

Betrachtet man die Testergebnisse, so kann man sehen, dass die Unterschiede zu den Trainingsdaten nicht signifikant sind und somit eine Überanpassung der Modelle ausgeschlossen werden kann. Allgemein sind die Testergebnisse sehr ähnlich zu den Trainingsdaten. Es fällt auf, dass die Evaluationsergebnisse etwas schlechter ausfallen, als die Trainingsergebnisse.

Allgemein ist zu beobachten, dass die Testergebnisse für die Modelle mit mehr Merkmalen besser ausfallen. Das beste Ergebnis für den  $P_t, K_t$  Datensatz erzielen das LightGBM- und XGBoost-Modell, deren Ergebnisse sich nur minimal unterscheiden. Das LightGBM-Modell erzielt allerdings einen besseren AUC-Wert. Danach folgen das Random Forest- und logarithmische Regressions-Modell. Alle Modelle für den  $P_t, K_t$  erzielen leicht bessere Ergebnisse, als die Modelle des  $P_t$  Datensatzes. Diese Ergebnisse basieren allerdings auf sehr wenigen Trainings- und Testdaten, weshalb die Ergebnisse nicht überinterpretiert werden sollten. Dies liegt vor allem daran, dass zu wenige Würfe innerhalb des Datensatzes existieren und somit der Datenraum nicht ausreichend abgedeckt werden kann.

Für den  $P_t$  erzielt das XGBoost-Modell das beste Ergebnis, wobei das LightGBM- und Random Forest-Modell nur minimal schlechter abschneiden. Das logistische Regressions-Modell schneidet für diesen Datensatz ebenfalls am schlechtesten ab. Aufgrund der Evaluationsergebnisse wird für den Datensatz  $P_t, K_t$  das LightGBM-Modell und für den Datensatz  $P_t$  das XGBoost-Modell gewählt, welche in den folgenden Kapiteln optimiert und interpretiert werden.

Somit kann die Forschungsfrage **RQ1** tendenziell mit ja beantwortet werden. Zusätzliche Merkmale, die einen Wurf in einen Kontext setzen, können ein Modell verbessern. Dies ist nur tendenziell, da für eine klare Aussage mehr Datensätze benötigt werden.

Wenn man die Evaluationsergebnisse mit anderen Arbeiten zu erwarteten Toren aus anderen Sportarten vergleicht, so schneiden die Ergebnisse dieser Arbeit teilweise wesentlich schlechter ab. Aktuelle Arbeiten aus dem Fußball wie beispielsweise [18][19][20] erzielen bessere Ergebnisse mit einem

kontextbasiertem Modell. Vergleicht man die Ergebnisse mit den Ergebnissen im Eishockey [21], so sind die Ergebnisse auf den ersten Blick ähnlich. Allerdings ist der Datensatz in der Arbeit von Paerels et al. wesentlich größer und somit aussagekräftiger.

### 5.3. Kalibrierung

Damit man die berechneten Wahrscheinlichkeiten der Modelle auch als Wahrscheinlichkeiten interpretieren kann, muss man sicherstellen, dass diese auch der tatsächlichen Wahrscheinlichkeiten entsprechen. Anders gesagt, wenn der geschätzte xG-Wert 0.8 ist, dann erwartet man, dass dieser Wurf acht mal aus 10 Versuchen ein Tor erzielt. Da jeder Wurf nur einmal ausgeführt wird und man die tatsächlichen Wahrscheinlichkeiten nicht kennt, kann man dies nicht direkt überprüfen. Eine andere Möglichkeit dies zu verifizieren, ist, Würfe mit einem ähnlichen xG-Wert, in Körbe aufzuteilen und zu prüfen, ob der durchschnittliche xG-Wert dem prozentualen Anteil an Toren innerhalb diesen Korbes entspricht. Das ist die Idee hinter dem Kalibrierungsdiagramm. In dieser Arbeit wird versucht die Werte auf zehn Körbe der gleichen Länge aufzuteilen.

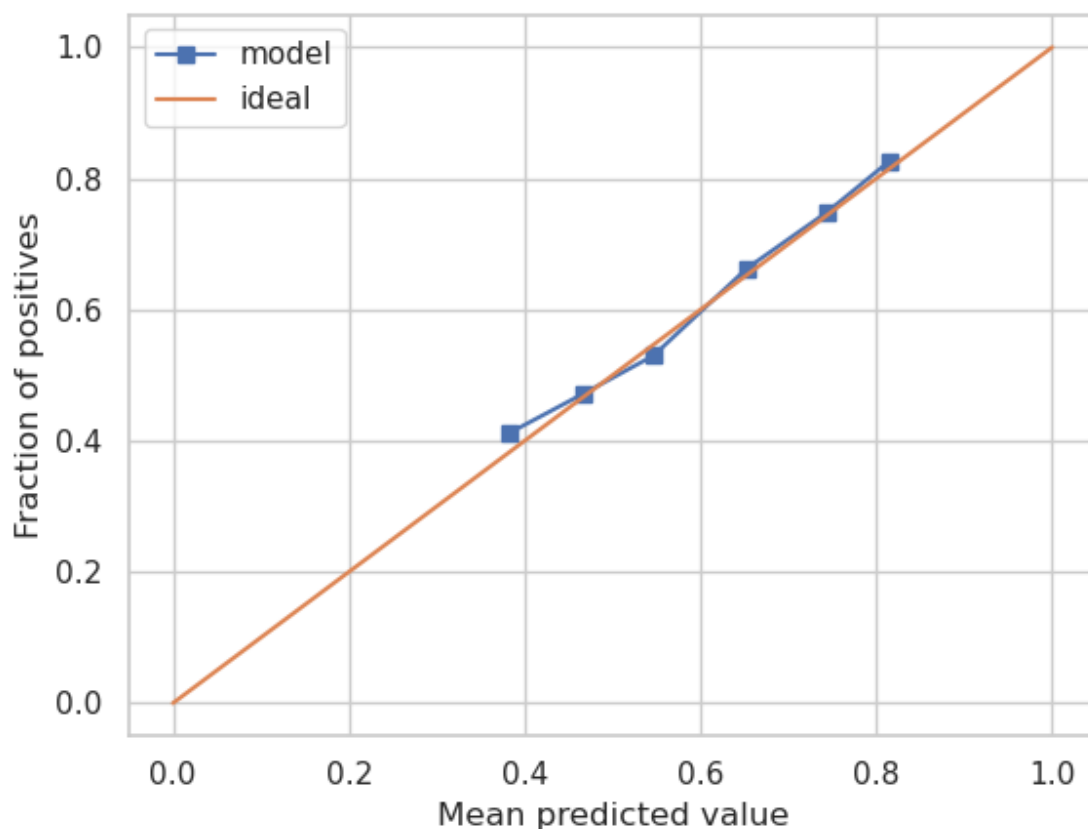


Abbildung 5.1.: Kalibrierungskurve vor einer Nachkalibrierung des XGBoost-Modells für den Datensatz  $P_t$

In Abbildung 5.1 ist das Kalibrierungsdiagramm für das XGBoost-Modell für den Test-Datensatz aus  $P_t$  abgebildet. Wenn auf der X-Achse der geschätzte mittlere xG-Wert abgebildet ist und auf

der Y-Achse der prozentuale Anteil an Toren abgebildet ist, dann würde ein perfekt kalibriertes Modell der Diagonalen folgen. Dies gilt, wenn auf beiden Achsen die Werte für alle berechneten Wahrscheinlichkeiten nahezu gleich sind. Im Fall des XGBoost-Modells für  $P_t$  folgt die Linie der Diagonalen ziemlich genau. Nur in den niedrigeren xG-Werten, die kleiner als 0.4 sind weicht sie leicht ab. Dies deckt sich auch mit einer signifikanten Verkleinerung der Stichprobengröße innerhalb der Körbe, da es weniger Würfe gibt, die keine Treffer waren, als Würfe, die zu einem Tor geführt haben. Da die Würfe allgemein eine eher hohen xG-Wert haben, gibt es für niedrige xG-Werte keine Kalibrierungskurve. Eine Abweichung der Kurve von der Diagonalen kommt daher, dass wenige Werte für die Berechnung des Durchschnitts des Korbes verwendet werden. Je mehr Werte pro Korb verwendet werden, desto eher nähert sich der Durchschnitt eines Korbes dem eigentlichen Kalibrierungswert an. Das ist eine Limitierung des Diagramms. Für die anderen Bereiche ohne einen Korb, gibt es keine Würfe im Testdatensatz mit einem entsprechenden xG-Wert.

Um zu überprüfen, ob eine Verbesserung durch eine Kalibrierung der Wahrscheinlichkeiten erzielt werden kann, werden zwei Varianten der Nachkalibrierung ausprobiert. Die erste nennt sich Platt-Skalierung, welches von [42] vorgestellt wurde. Es ist ein parametrischer Ansatz, bei dem Platt ein logistisches Regressionsmodell zur Schätzung der Zielwahrscheinlichkeiten mithilfe der nicht probabilistischen Vorhersagen des Klassifikators zur Kalibrierung verwendet. Die andere Variante heißt Isotonic-Regression und wurde erstmals von [43] vorgestellt. Es ist ein nicht-parametrischer Ansatz. Es wird eine konstante Funktion erlernt, die den quadratischen Verlust zwischen der Ausgabe und dem tatsächlichen Zielwert minimiert.

Für beide Kalibrierungsmethoden wird ein Validierungsdatensatz benötigt, der nicht für das Training des Klassifikators benutzt wird. Hier werden die gleichen Faltungen der Cross-Validation aus dem Training des ursprünglichen Modells verwendet. Diesmal wird der Trainingsdatensatz jeder Faltung verwendet, um den Klassifikator zu trainieren und der dazugehörige Validierungsdatensatz wird verwendet, um den Kalibrierungsregressor zu trainieren. Somit erhält man für jede Faltung ein Paar aus Klassifikator und Kalibrierungsregressor. In unserem Fall werden zehn Paare trainiert, da wir eine 10-Fold Cross Validation für das Training verwendet haben. Das Resultat der Kalibrierung wird mithilfe des Testdatensatzes ermittelt und errechnet sich aus dem Durchschnitt der berechneten Wahrscheinlichkeiten der zehn Paare.

Um die Kalibrierung zu bewerten, wird der erwartete Kalibrierungsfehler (Expected Calibration Error (ECE) (ECE)) verwendet. Diese Metrik wurde vorgestellt in [63] und misst die Differenz in der Erwartung zwischen der Konfidenz und der Genauigkeit. Dies wird auch Kalibrierungslücke genannt. Wie bei der Kalibrierungskurve werden die Körbe in Bereiche mit gleicher Länge der geschätzten Werte aufgeteilt. Zu jedem Korb wird die Kalibrierungslücke hinzu addiert. Die Kalibrierungslücke wird dabei gewichtet nach der Anzahl der Schätzungen pro Korb. Der perfekte Klassifikator hätte einen ECE-Wert von 0.

Für das XGBoost-Modell des  $P_t$  Datensatzes ist der ECE-Wert der Isotonic-Kalibrierung, der Platt-Kalibrierung und ohne Kalibrierung in Tabelle 5.3 aufgeführt. Man erkennt, dass die Kalibrierung den ECE-Wert vergrößert und somit den Fehler erhöht. Das Modell ohne Kalibrierung erzielt das beste Ergebnis, wobei die Isotonic- und Platt-Kalibrierung nur minimal schlechter abschneiden. Die Kalibrierungskurve nach der Isotonic-Kalibrierung und nach der Kalibrierung nach Platt ist



	ECE
ohne Kalibrierung	0.0088
Isotonic	0.0126
Platt	0.0170

Tabelle 5.3.: ECE Kalibrierungsmetrik für jede Kalibrierungsmethode für das XGBoost-Modell des Datensatzes  $P_t$

in Abbildung 5.2 aufgeführt. Hier ist zu sehen, dass bei beiden Kalibrierungsmethoden die xG-Werte auf ein etwas größeres Intervall gestreckt werden, aber die Kalibrierung etwas ungenauer um die Randwerte wird. Durch den vergrößerten Fehler, wird für das XGBoost-Modell auf eine Nachkalibrierung verzichtet.

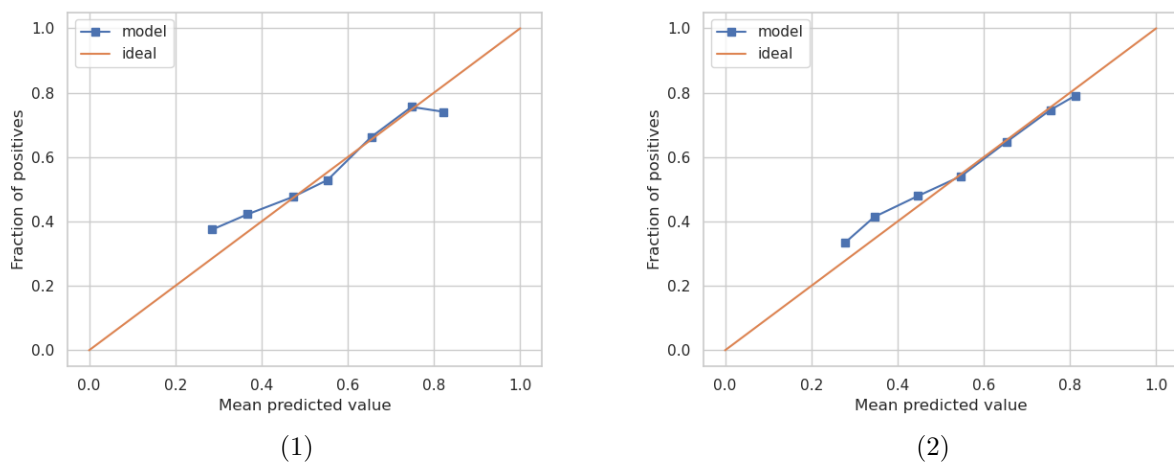


Abbildung 5.2.: Kalibrierungskurve des XGBoost-Modells für den Datensatz  $P_t$  nach der Isotonic-Kalibrierung (a) und nach der Kalibrierung nach Platt (b)

Für das LightGBM-Modell des  $P_t, K_t$  Datensatzes ist die Kalibrierungskurve der Testdaten vor einer zusätzlichen Kalibrierung in Abbildung 5.3 abgebildet. Hier ist zu beobachten, dass sehr niedrige xG-Werte, die kleiner als 0.35 sind, sehr schlecht kalibriert sind. Die übrigen Werte weichen etwas von der Diagonalen ab, folgen dieser aber recht grob. In der Tabelle 5.4 sind die ECE-Werte der Nachkalibrierung dem Modell ohne Kalibrierung gegenübergestellt. Es ist zu beobachten, dass die Isotonic-Kalibrierung den ECE-Wert minimal verbessert. Obwohl die Isotonic-Kalibrierung eine leichte Verbesserung zur Folge hat, wird sich dafür entschieden das Modell ohne eine zusätzliche Kalibrierung zu verwenden, da die ECE-Werte ohne Kalibrierung nur minimal schlechter sind als die mit Kalibrierung.

	ECE
ohne Kalibrierung	0.1053
Isotonic	0.0941
Platt	0.1156

Tabelle 5.4.: ECE Kalibrierungsmetrik für jede Kalibrierungsmethode für das LightGBM-Modell des Datensatzes  $P_t, K_t$

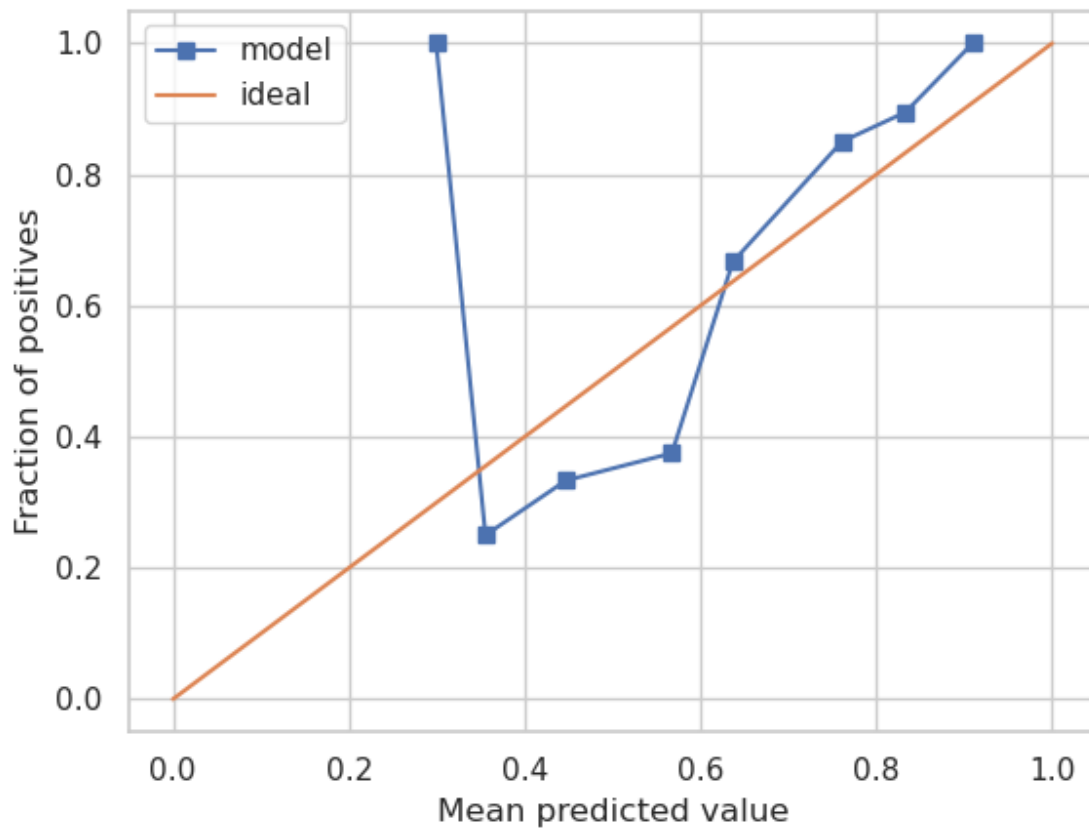


Abbildung 5.3.: Kalibrierungskurve des LightGBM-Modells für den Datensatz  $P_t, K_t$

## 5.4. Interpretation

Auf der einen Seite ist es wichtig, genaue Schätzungen mit verlässlichen Wahrscheinlichkeiten durch das Modell zu erhalten. Auf der anderen Seite ist es ebenfalls wichtig, dass das Modell interpretiert werden kann, um die Ergebnisse gut zu vermitteln. Am Ende sollen die Anwender des Modells Trainer, Spieler und Analysten sein, die das Modell als zusätzliche Quelle für Informationen nutzen. Der Nutzen des Modells ist somit mit der Interpretierbarkeit des Modells limitiert. Ein Beispiel dafür wäre, wenn das Modell bei einem Spieler bei Würfeln in seinen prägenden Jahren verwendet wird. Hier wäre es essentiell für den Spieler welche Eigenschaften des Wurfes die Wahrscheinlichkeit negativ oder positiv beeinflussen. Dies kann dazu beitragen, einem Spieler beizubringen, in welcher Situation es sich lohnt einen Wurf zu nehmen und wann eher nicht.

Ganz allgemein kann man sagen, dass eine notwendige Interpretierbarkeit die möglichen verwendbaren Modelle einschränkt. Komplexere Modelle sind in der Regel für die meisten Probleme präziser. Allerdings tendieren sie auch dazu, dass ihre Ergebnisse schwerer zu interpretieren sind und eher als Black-Box agieren. Deshalb muss man zwischen der Genauigkeit des Modells und der Interpretierbarkeit abwägen.

Um die Modelle für xG zu interpretieren und somit die Forschungsfragen **RQ2** und **RQ3** zu beantworten wird das Framework SHAP verwendet, welches von [46] vorgestellt wurde. Dieses weist

jedem Merkmal einen Wert für dessen Wichtigkeit innerhalb des Modells zu. Das Framework erstellt für einen beliebigen Klassifikator ein Erklärungsmodell. Dieses weist jedem Merkmal Punkte mithilfe von verschiedenen Methoden zu. Der Vorteil an diesem Framework ist, dass es für jeden beliebigen Klassifikator verwendet werden kann.

Zuerst wird auf das XGBoost-Modell des  $P_t$  Datensatzes eingegangen. Hierbei betrachten wir zunächst die Bedeutung, die das Modell jedem verwendeten Merkmal beimisst. Die Bedeutungen werden durch das SHAP Framework gemessen. Die Abbildungen 5.4 und 5.5 zeigen eine Zusammenfassung der Bedeutung aller Merkmale auf die Ausgabe des XGBoost-Modells.

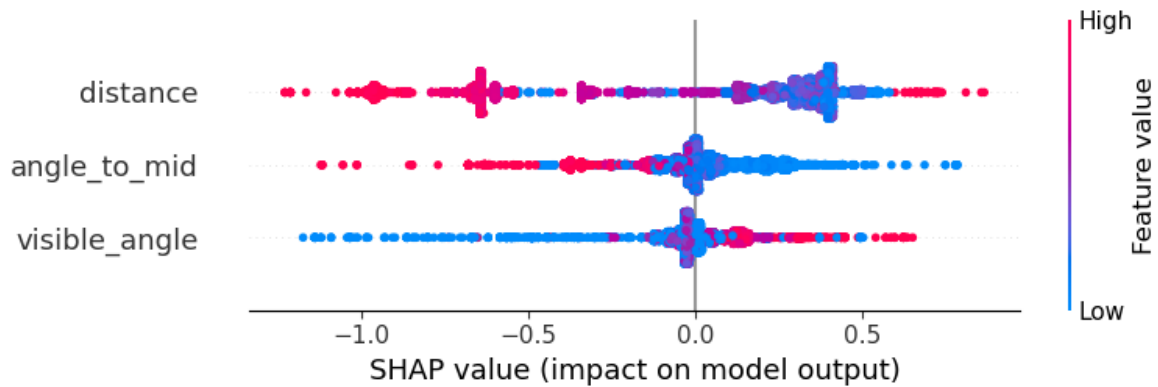


Abbildung 5.4.: Bedeutung aller Merkmale für das finale XGBoost-Modell mit den Merkmalen des  $P_t$  Datensatzes. Die Merkmale sind absteigend nach Bedeutung sortiert.

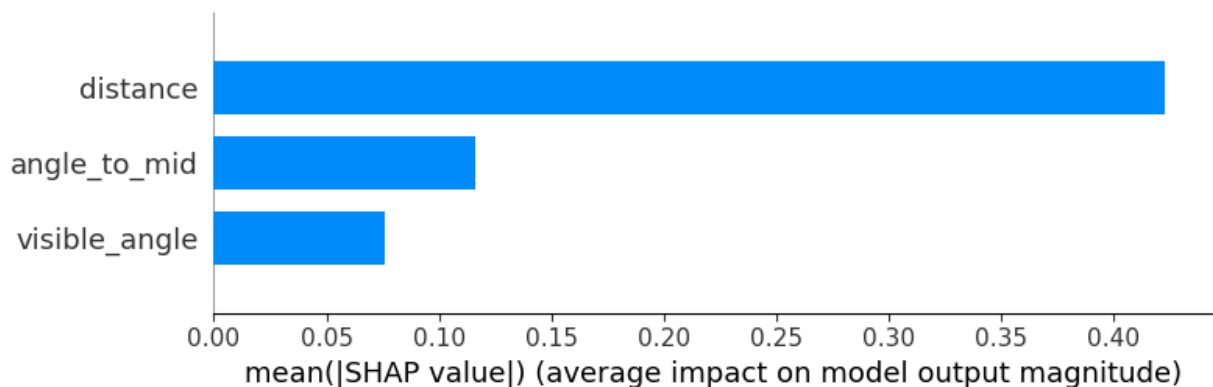


Abbildung 5.5.: vereinfachte Darstellung der Bedeutung der Merkmale für das finale XGBoost-Modell mit den Merkmalen des  $P_t$  Datensatzes. Die Merkmale sind absteigend nach Bedeutung sortiert.

menfassung der Bedeutung aller Merkmale auf die Ausgabe des XGBoost-Modells. Abbildung 5.4 listet die Merkmale absteigend nach ihrer Bedeutung auf. Für jedes Merkmal wird eine Verteilung der Merkmalswerte entlang der X-Achse der Abbildung angezeigt. Jeder Punkt in der Verteilung repräsentiert eine Gruppe ähnlicher Würfe, die durch das Framework analysiert wurden. Die Position des Punktes auf der X-Achse quantifiziert den Einfluss der Würfe auf die Ausgabe des Modells. Dabei wird zwischen positiven und negativen Einfluss unterschieden. Die Farbe gibt den Wert des Merkmals für die Würfe eines Punktes an. Die Abbildung 5.5 gibt das gleiche in vereinfachter Form an, aber es hilft, die Unterschiede der durchschnittlichen Einflüsse der Merkmale visuell zu vergleichen.

Beide Diagramme stellen ganz klar heraus, dass das wichtigste Merkmal für die Klassifikation die Distanz zum Tor ist. Dies ist nachvollziehbar, da es schwieriger wird akkurat zu zielen, je weiter man vom Tor entfernt ist. Außerdem hat der Torwart mehr Zeit um auf einen Wurf zu reagieren. Dazu kommt, dass der Ball an Geschwindigkeit verliert, wenn er lange unterwegs ist. In Abbildung 5.4 kann man darüber hinaus erkennen, dass hohe und moderate Distanzen einen starken negativen Effekt auf das Erzielen eines Tores haben. Niedrige Distanzen hingegen haben einen moderaten positiven Einfluss. Hier ist allerdings zu bemerken, dass selten auch hohe Distanzen einen positiven Effekt haben und niedrige Distanzen einen leicht negativen Einfluss. Bei dem sichtbaren Winkel des Tores ist es anders herum. Hier haben hohe Werte, also ein großer sichtbarer Winkel zum Tor, einen positiven Einfluss auf das Erzielen eines Tores. Kleine und moderate Werte haben einen neutralen bis stark negativen Einfluss auf den Erfolg eines Wurfs. Dies ist ebenfalls nachvollziehbar, da je größer der Winkel ist, desto zentraler und gegebenenfalls auch näher befindet man sich vor dem Tor. Zentral vor dem Tor hat man mehr Teile des Tors, die nicht vom Torhüter abgedeckt werden können. Somit hat man mehr Möglichkeiten für einen Abschluss. Teilweise haben niedrige Werte ebenfalls einen schwachen positiven Effekt auf den Erfolg. Dies könnte erklärbar sein durch Sprungwürfe, die von außen kommen. Diese haben einen relativ kleinen sichtbaren Winkel und sind in der Regel relativ nah zum Tor. Das Merkmal für den Winkel zur Mitte des Spielfelds verhält sich äquivalent zum sichtbaren Winkel. Niedrige Werte haben einen schwachen bis starken positiven Einfluss auf den Erfolg. Hohe Werte haben eher einen negativen Einfluss, wobei moderate Werte eher neutral für den Erfolg eines Wurfs sind. Dieses Merkmal ist kleiner je zentraler man vor dem Tor steht und wird größer je weiter man Richtung Seitenlinie steht. Auch hier werden zentrale Positionen einen ähnlich starken Einfluss beigemessen, wie bei dem sichtbaren Winkel vom Tor. Allgemein lässt sich sagen, dass Würfe, die zentral vor dem Tor und am besten mit einer geringen Distanz zum Tor erfolgen, mit einem hohen xG-Wert klassifiziert werden.

Anschließend wird das LightGBM-Modell des  $P_t, K_t$  genauer analysiert. Die Abbildungen 5.6 und 5.7 zeigen ebenfalls die Einflüsse der einzelnen Merkmale auf den Erfolg aller Würfe, die durch das LightGBM-Modell klassifiziert werden. Anders als bei dem XGBoost-Modell aus Abbildung 5.4 und 5.5 ist nicht die Distanz zum Tor das wichtigste Merkmal, sondern der sichtbare Winkel des Tors und anschließend die Distanz des Abwurfs zum Torhüter. Die Distanz an sich scheint keinen Einfluss auf das Ergebnis eines Wurfs zu haben. Der Einfluss des sichtbaren Winkels vom Tor verhält sich wie in 5.4. Allerdings ist der Einfluss der Merkmalswerte klarer in positiven und negativen Einfluss getrennt. Kleine und moderate Winkel haben einen stark negativen Einfluss auf den Erfolg eines Wurfs. Hohe Winkel hingegen haben einen stark positiven Einfluss. Die Interpretation des Merkmals ist analog zum XGBoost-Modell. Die Distanz des Wurfs zum Torhüter verhält sich ähnlich wie die Distanz im vorherigen Modell. Niedrige Distanzen zum Torhüter haben einen stark positiven Einfluss auf den Erfolg eines Wurfs. Hohe Distanzen haben einen stark negativen Einfluss. Kurze Distanzen zum Torhüter bedeuten in der Regel auch eine kurze Distanz zum Tor. Je näher der Torwart vor dem Werfer steht, umso geringer ist auch die Zeit, die der Torhüter hat zu reagieren. Somit ist es laut Modell einfacher den Torhüter zu überwinden, je näher man vor ihm steht.

Auf die beiden positionsbezogenen Merkmale folgen mehrere kontextbezogenen Merkmale. Diese haben alle einen ähnlichen durchschnittlichen Einfluss auf die Klassifikation. Das Modell geht davon aus, dass die wichtigsten kontextbezogenen Merkmale die Anzahl der Spieler zwischen Wer-

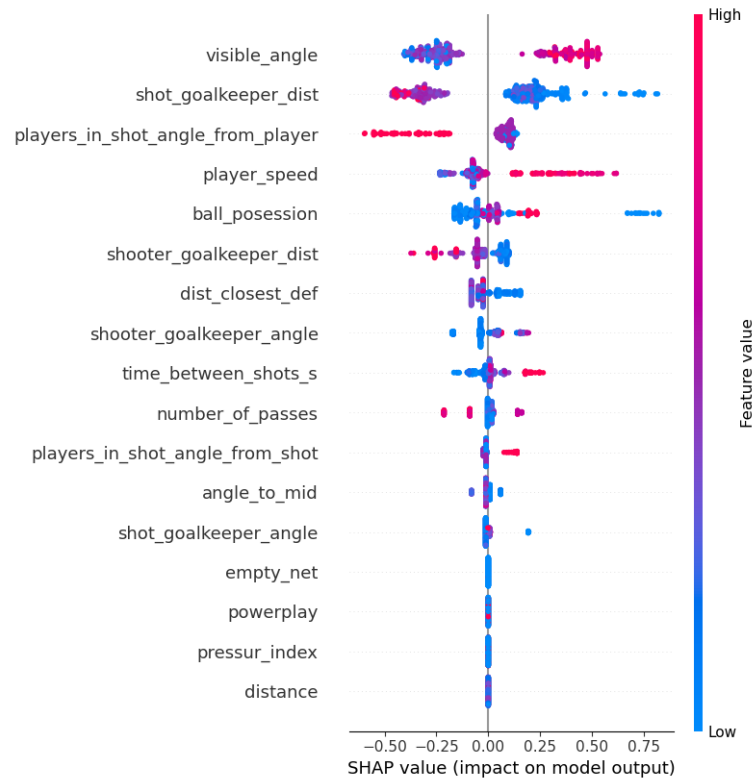


Abbildung 5.6.: Bedeutung aller Merkmale für das finale LightGBM-Modell mit den Merkmalen des  $P_t, K_t$  Datensatzes. Die Merkmale sind absteigend nach Bedeutung sortiert.

ferposition und Tor, die Geschwindigkeit des Werfers und die Dauer des Ballbesitzes der werfenden Mannschaft sind. Eine hohe Anzahl an Spielern zwischen dem Werfer und dem Tor hat einen stark negativen Einfluss, wohingegen eine geringe und moderate Anzahl an Verteidigern einen leicht positiven Einfluss haben. Ist mehr als ein Verteidiger vor dem Werfer ist es für den Werfer schwer den Ball aufs Tor zu werfen, ohne dass dieser vorher abgefangen wird. Ebenso haben hohe Spilergeschwindigkeiten einen stark positiven Einfluss auf die Klassifikation eines Tors. Dies tritt vor allem bei schnellen Umschaltsituationen auf, aber auch in Situationen, in denen der Werfer in kurzer Zeit schnell beschleunigt und somit eine Lücke in die Verteidigung reißen kann. Beim Ballbesitz ist die Interpretation etwas schwieriger. Eine sehr niedrige Dauer des Ballbesitzes hat einen stark positiven Einfluss auf den Erfolg eines Wurfs. Eine hohe Dauer des Ballbesitzes hat einen schwach positiven Einfluss. Eine moderate und niedrige Dauer hingegen haben einen neutralen bis negativen Einfluss auf die Klassifikation eines Treffer. Eine mögliche Interpretation ist, dass der Aufbau eines Wurfs nicht zu kurz und nicht zu lang sein sollte. Bei langen Zeiten zwischen Würfen, kann sich die Verteidigung gegebenenfalls gut auf den Angreifer einstellen und Wurfmöglichkeiten verhindern, da der Angreifer zu langsam agiert. Bei zu kurzen Zeiten handelt der Angreifer zu hastig und ungeduldig. Bei durchschnittlichen Ballbesitzphasen hat die werfende Mannschaft eher die Zeit einen Wurf vorzubereiten und Lücken in der Verteidigung zu erzeugen oder durchzubrechen. Ausnahmen bilden sehr niedrige Werte, wie bei einem Tempogegenstoß, wo die Ballbesitzdauer sehr gering ist und meist in einer eins zu eins Situation zwischen Werfer und dem Torwart endet. Die restlichen Merkmale haben einen eher geringen Einfluss auf die Ausgabe der Klassifikation und sind meist durch andere ähnliche Merkmale bereits abgedeckt. Die Merkmale Distanz, Powerplay, Pressur-Index und Empty-Net scheinen gar keinen Einfluss auf die Ausgabe des Modells zu haben. Die Distanz hat

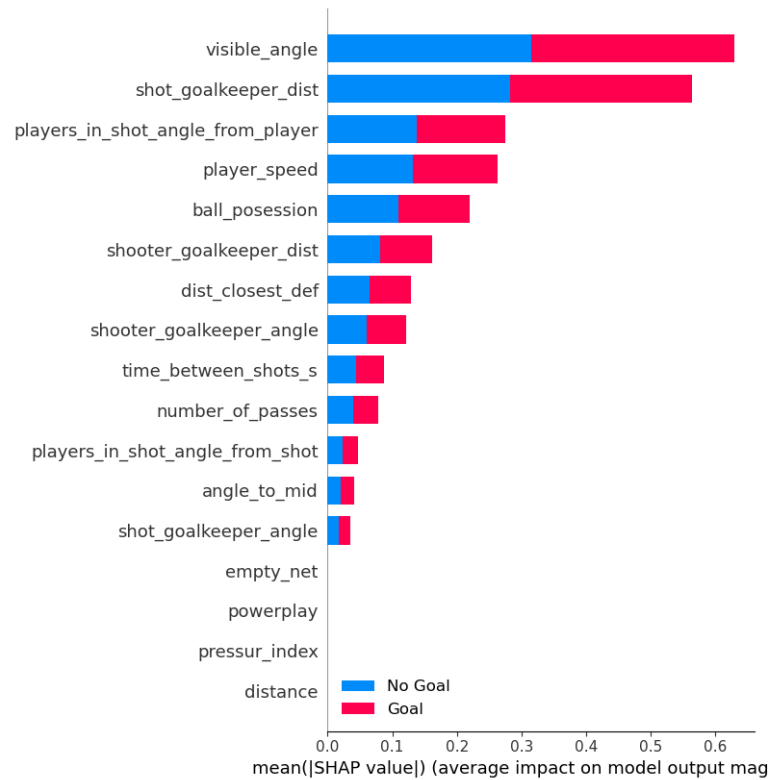


Abbildung 5.7.: vereinfachte Darstellung der Bedeutung der Merkmale für das finale LightGBM-Modell mit den Merkmalen des  $P_t$ ,  $K_t$  Datensatzes. Die Merkmale sind absteigend nach Bedeutung sortiert.

sehr wahrscheinlich keinen Einfluss, da die Werte durch das Merkmal der Distanz zum Torhüter mit abgedeckt werden. Ebenso ist der Pressur-Index (die Anzahl der Verteidiger um den Werfer in einem ein Meter Radius) im Handball nicht relevant. Spieler hinter oder neben dem Werfer können den Wurf nur schlecht verhindern. Spieler vor dem Werfer sind in dem Merkmal der Anzahl der Spieler zwischen Werfer und Tor enthalten. Interessant ist allerdings, dass Überzahlsituationen wie beim Powerplay oder bei Empty-Net für die Klassifikation eines Tors keinen oder einen nur sehr geringen Einfluss haben. Laut Modell sind solche Situationen für den Erfolg eines Wurfes nicht wirklich relevant. Das Entfernen dieser Merkmale aus dem Datensatz führte allerdings dazu, dass sich die Ergebnisse der Modelle verschlechtern. Deshalb wird sich dazu entschieden, diese beizubehalten. Zusammenfassend kann man sagen, dass das LightGBM-Modell Würfe mit einem hohen xG-Wert bewertet, wenn diese zentral vor dem Tor, mit einer geringen Distanz zum Torhüter, ohne Verteidiger zwischen Werfer und Tor und mit einer hohen Geschwindigkeit des Werfers erfolgen.

In Abbildung 5.8 sind vier verschiedene Würfe inklusive Spielerpositionen aufgezeigt. Zu jedem Wurf ist der xG-Wert der beiden Modelle mit angegeben. Wenn man sich die xG-Werte ansieht, kann man die Unterschiede in den Ergebnissen der Klassifikation in Kombination mit den verschiedenen Einflüssen der Merkmale erkennen. Die Würfe 5.81 und 5.84 werden von beiden Modellen ähnlich bewertet. Dies liegt daran, dass in diesen Fällen zum größten Teil nur die Positionen des Abwurfs für die Klassifikationen entscheidend sind. Allerdings fällt auf, dass das XGBoost-Modell die xG-Werte etwas niedriger schätzt als das LightGBM-Modell. In den Würfen 5.82 und 5.83 hingegen approximiert das LightGBM-Modell den xG-Wert viel geringer als das XGBoost-Modell. Dies liegt

vor allem an dem Merkmal der Anzahl der Spieler zwischen Werfer und Tor. Wie man in Abbildung 5.6 sieht, reduziert es die Wahrscheinlichkeit auf einen Treffer stark. Eine genauere Analyse der Würfe wird in Kapitel 6.1 durchgeführt.

Abschließend kann man zur Forschungsfrage **RQ2** sagen, dass die Abwurfposition entscheidend ist. Zusätzliche Kontextinformationen wie die Anzahl der Spieler zwischen Werfer und Tor und die Spielergeschwindigkeit haben auch einen signifikanten Einfluss auf die Klassifikation und tragen daher einen Mehrwert zum Modell bei. Für die Forschungsfrage **RQ3** wurde mithilfe des SHAP-Frameworks eine Möglichkeit gezeigt die Modelle zu interpretieren.

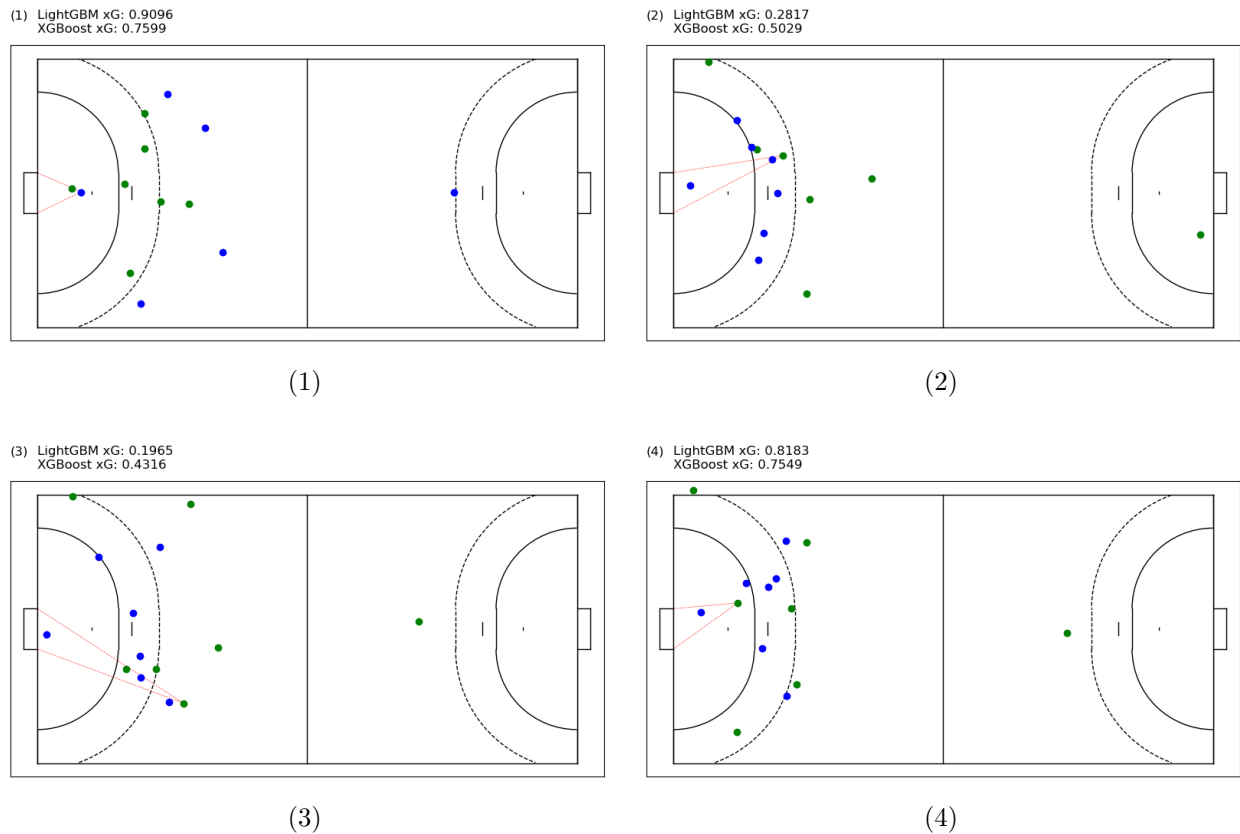


Abbildung 5.8.: Vier verschiedene Würfe inklusive berechnetem xG-Wert für das XGBoost- ( $P_t$ ) und LightGBM-Modell ( $P_t, K_t$ )

## 6. Anwendungsfälle

Da wir die beiden Modelle evaluiert und untersucht haben, ob die berechneten Wahrscheinlichkeiten auch als Wahrscheinlichkeiten interpretiert werden können, gehen wir davon aus, dass die xG-Werte für beide Modelle valide sind. Dies gilt für Daten, die das Modell noch nicht gesehen hat und für bereits gesehene Daten. In diesem Kapitel wird aufgezeigt, was man mit der xG-Metrik machen kann. Dafür werden die Ergebnisse der beiden Modelle, die in Kapitel 5 ausgewählt wurden, verwendet.

Für eine erste Analyse werden die xG-Werte beider Modelle auf dem Spielfeld visualisiert. Damit soll untersucht werden, wie sich unterschiedliche Abwurfpositionen auf den xG-Wert auswirken. Der Grund dafür ist, dass die Merkmale, die aus der Abwurfposition abgeleitet werden, für beide Modelle die wichtigsten Merkmale für die Klassifikation darstellen. Dafür wird das Feld in hexagonale Felder unterteilt. Pro Feld wird der durchschnittliche xG-Wert aller Würfe berechnet, die aus diesem Feld heraus erfolgten. Für das XGBoost-Modell werden die Daten des  $P_t$  Datensatzes verwendet. Pro Korb müssen mindestens 10 Würfe erfolgt sein, damit dieser berechnet wird. Für das LightGBM-Modell werden die Daten des  $P_t, K_t$  Datensatzes verwendet. Hier muss pro Korb mindestens ein Wurf erfolgt sein, damit dieser berechnet wird. Für das LightGBM-Modell wird die Mindestanzahl an Wurfen so gewählt, da es zu wenige Daten gibt. Die Ergebnisse sind in Abbildung 6.1 abgebildet.

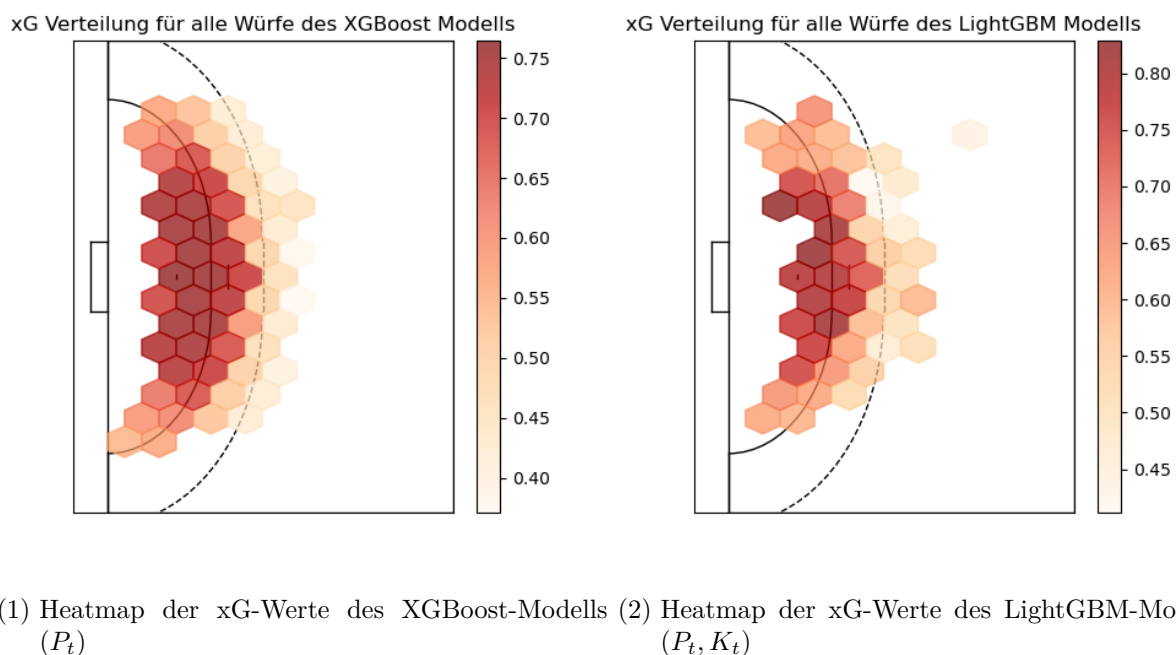


Abbildung 6.1.: Darstellung der xG-Werte als Heatmap des XGBoost-Modells 6.11 und des LightGBM-Modells 6.12.

In den Abbildungen kann man sehen, dass für beide Modelle die xG-Werte zentral vor dem Tor um den 7-Meter-Punkt am höchsten sind. Außerdem kann man sehen, dass die xG-Werte von der Mitte heraus nach Außen um die 6-Meter-Linie abnehmen. Dies deckt sich mit den Folgerungen aus der Interpretation der einzelnen Merkmale aus Kapitel 5.4. Das LightGBM-Modell bewertet die



sehr zentralen Würfe etwas höher als das XGBoost-Modell. Um dies genauer zu bewerten, fehlen an dieser Stelle die Kontextinformationen. Diese werden in Kapitel 6.1 für einzelne Würfe genauer analysiert.

Man beachte außerdem, dass einige Felder weiter außerhalb einen höheren durchschnittlichen xG-Wert haben als andere. Dies ist vor allem beim LightGBM-Modell der Fall. Das liegt daran, dass Schüsse aus solchen Positionen relativ ungewöhnlich sind. Sollte ein solcher Wurf dennoch ein Tor erzielen, kann dies dazu führen, dass das Modell die Abwurfposition überbewertet.

In den folgenden Kapiteln werden nur noch Würfe aus dem  $P_t, K_t$  Datensatz analysiert. Für jeden darin enthaltenen Wurf, wird jeweils der xG-Wert mit dem XGBoost- und mit dem LightGBM-Modell berechnet.

## 6.1. Analyse einzelner Würfe

Der erste Anwendungsfall ist eine Fortführung der Analyse des Einflusses der einzelnen Merkmale auf den Erfolg eines Wurfes aus Kapitel 5.4. Anstatt die Ergebnisse des ganzen Modells zu interpretieren, werden diesmal einzelne Würfe genauer analysiert. Dies kann dabei helfen, dass Spieler und Trainer besser verstehen können, was in einzelnen Situationen besser gemacht werden könnte.

Abbildung 6.2 zeigt den Einfluss der Merkmalsausprägungen für vier verschiedene Würfe aus dem Datensatz  $P_t, K_t$ . Jeder Wurf ist dargestellt mit der Abwurfposition im oberen Plot und wird ergänzt durch den Einfluss der Merkmale im unteren Plot. Die einzelnen Würfe werden referenziert über die Nummern. Die Würfe (1) und (2) sind Würfe, die zu einem Tor geführt haben. Die restlichen in rot markierten Würfe erzielten kein Tor.

Schaut man sich die Einflüsse der Merkmale an, so stellen rot markierte Merkmale ein Merkmal dar, welches zur Wahrscheinlichkeit eines Tors beiträgt. Blau markierte Merkmale hingegen sind Merkmale, die die Wahrscheinlichkeit eines Tores verringern. Die Länge des Balkens eines Merkmals repräsentiert die Höhe des Einflusses dieses spezifischen Merkmals auf die Ausgabe des Modells. Je länger der Balken ist, umso mehr Einfluss hat das Merkmal in diesem spezifischen Falls auf die Ausgabe des Modells. Die Zahlen direkt neben den Merkmalsnamen sind die standardisierten Merkmalsausprägungen für den entsprechenden Wurf. Diese werden deshalb im folgenden ignoriert. In Abbildung 5.8 sind die Spielsituationen zu den einzelnen Würfeln ebenfalls abgebildet. Die einzelnen Würfe sind über die Zahl in den Klammern referenziert.

Der erste Wurf ist mit 0.906, der Wurf mit dem höchsten xG-Wert des LightGBM-Modells. Fast alle Merkmale tragen zu der Wahrscheinlichkeit etwas bei. Nur wenige verringern die Wahrscheinlichkeit und dies auch nur zu einem sehr geringen Teil. Man kann sehen, dass die wichtigsten Merkmale für diesen Wurf die Gleichen sind, wie für das gesamte Modell. Da der Wurf zentral erfolgt, der Torwart nah vor dem Werfer steht und der Werfer relativ schnell ist, haben die Distanz zum Torwart, der sichtbare Winkel und die Spielergeschwindigkeit den höchsten Einfluss auf den Wurf.

Der zweite und dritte Wurf erfolgen aus ähnlichen Abwurfpositionen nur auf entgegengesetzten Seiten. Beide haben eine ähnliche Wahrscheinlichkeit für einen Treffer. Mit 0.2816 hat der zweite Wurf eine etwas höhere Wahrscheinlichkeit als der Dritte mit 0.1965. Der dritte Wurf ist etwas

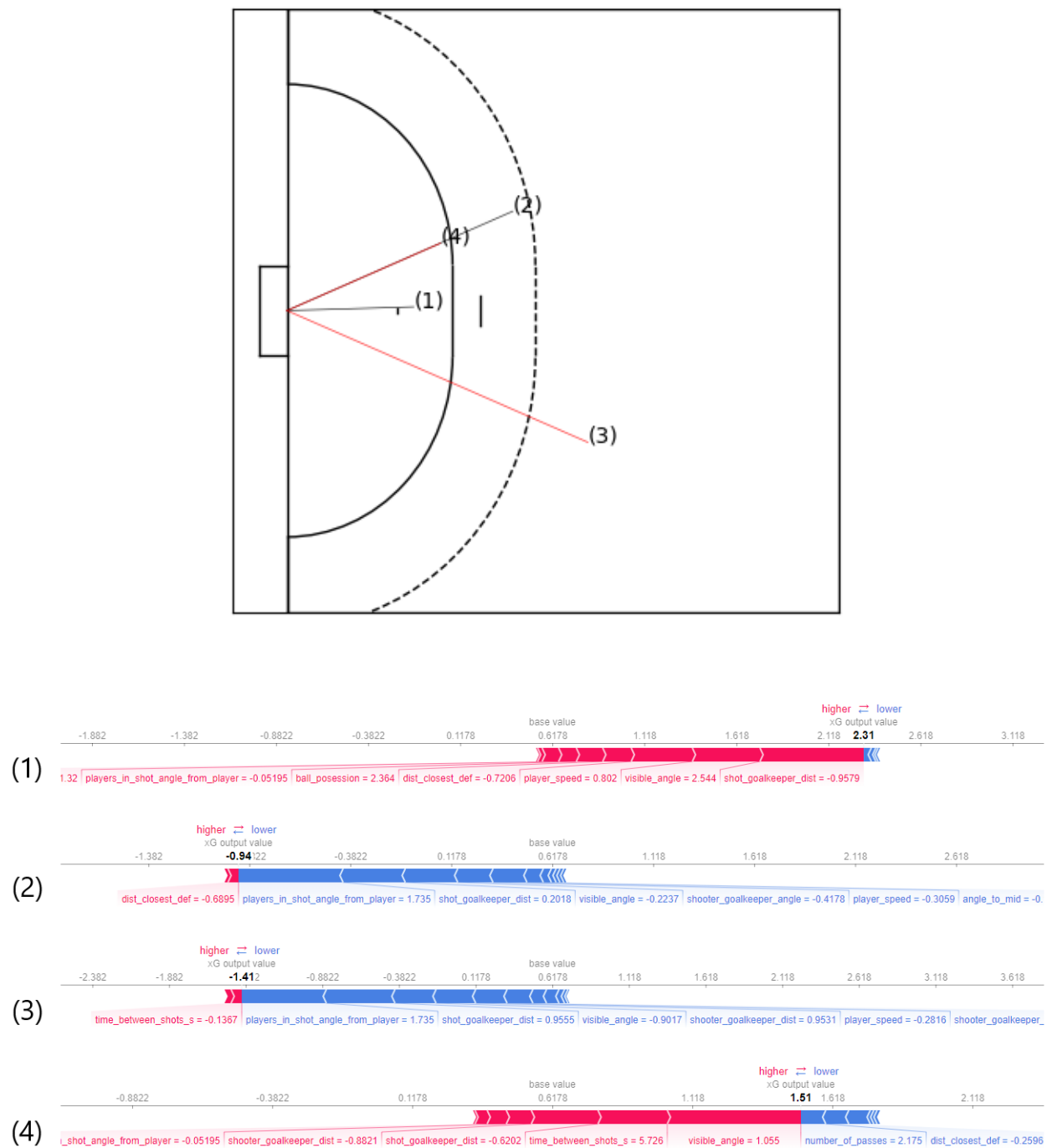


Abbildung 6.2.: Einfluss der Merkmale des LightGBM Modells für vier verschiedene Würfe. Der obere Plot zeigt die Abwurfstelle. Die Würfe sind durchnummeriert. Der untere Plot zeigt die Einflüsse der einzelnen Merkmale. Die Zahl vor jedem Diagramm spiegelt die Referenz zur Abwurfposition aus dem oberen Plot wieder. Die Würfe mit einer roten Linie sind Würfe, die zu keinem Tor geführt haben.

weiter entfernt und nicht ganz so zentral wie der zweite Wurf, was die Wahrscheinlichkeit des dritten Wurfes etwas mehr verringert als die des Zweiten. Was bei beiden ins Gewicht fällt und

die Wahrscheinlichkeit verringert, sind Verteidiger zwischen Werfer und Tor. Die Distanz zum Torwart, die beim ersten Wurf noch einen positiven Effekt auf die Wahrscheinlichkeit eines Tors gehabt hat, verringert in diesen beiden Fällen die Wahrscheinlichkeit für ein Tor. Trotz geringer Wahrscheinlichkeit, erzielt der Werfer beim zweiten Wurf ein Tor. Der dritte Wurf führte nicht zum Erfolg.

Der letzte Wurf ist ein Wurf von innerhalb des Torraums. Er hat eine berechnete Wahrscheinlichkeit für ein Tor von 0.8183. Die Distanz ist ähnlich zum ersten Wurf. Allerdings erfolgt er nicht ganz so zentral. Trotzdem ist der sichtbare Winkel des Tores das Merkmal, das die Wahrscheinlichkeit für ein Tor am meisten erhöht. Darauf folgt das Merkmal für die Sekunden zwischen den Würfen des werfenden Teams. Dieses hatte bei den anderen Würfen so gut wie keinen Einfluss. Das Merkmal, dass die Wahrscheinlichkeit am größten verringert, ist die Anzahl der Pässe vor dem Wurf. Trotz der hohen Wahrscheinlichkeit konnte der Werfer den Ball nicht aufs Tor bringen. Er warf über das Tor.

## 6.2. Leistungsanalyse eines Spiels

Wie in Kapitel 4.4 bereits erwähnt, kann man die Summe der Wahrscheinlichkeiten von mehreren Toren interpretieren als die Anzahl der zu erwartenden Tore, die aus den Würfen resultieren. Wenn wir also die Wahrscheinlichkeiten für einen Spieler für ein gegebenes Intervall addieren, dann erhält man die Anzahl der erwarteten Tore für diesen Zeitraum. Ein Zeitraum kann beispielsweise eine Saison oder ein Spiel sein. Vergleicht man die Anzahl der erwarteten Tore des Spielers mit der Anzahl der tatsächlich erzielten Tore innerhalb des betrachteten Zeitraumes, dann kann man diesen Wert als Metrik für die Bewertung der Leistung des Spielers verwenden. Da wir den xG-Wert des LightGBM-Modells nur für Würfe aus dem Datensatz  $P_t, K_t$  vorliegen haben und dieser nur einzelne Spiele aus einer Saison beinhaltet, wird als Intervall ein Spiel gewählt.

In Tabelle 6.1 sind die Wurfstatistiken von allen Spielern des 9. Spieltag der Saison 2021-2022 zwischen dem SC DHfK Leipzig und dem Bergischen HC aufgelistet, die mindestens einen Wurf abgegeben haben. Für jeden Spieler werden die Anzahl der Würfe, die Anzahl der Tore, der xG-Wert des LightGBM-Modells ( $xG_{P_t, K_t}$ ) und der xG-Wert des XGBoost-Modells  $xG_{P_t}$  angegeben. Die Anzahl der Würfe dient als Anhaltspunkt, wie viele Möglichkeiten ein Spieler zum Erzielen eines Tores hatte. Die xG-Werte beider Modelle können wie folgt interpretiert werden. Das Modell, welches nur die Positionsmerkmale verwendet ( $xG_{P_t}$ ) ist die Anzahl der zu erwartenden Tore von einem durchschnittlichen Spieler der Liga der letzten drei Saisons, aus einer ähnlichen Position. Anders gesagt ist es eine Approximation des Anteils der Tore von dieser Position aus. Der xG-Wert des Modells mit den zusätzlichen Kontextmerkmalen ( $xG_{P_t, K_t}$ ) kann interpretiert werden, als die Anzahl der zu erwartenden Tore eines durchschnittlichen Spielers in gleichen Spielsituationen des SC DHfK Leipzig oder Bergischen HC der letzten drei Saisons.

Bei der Analyse der Tabelle kann man festhalten, dass alle Spieler eine überdurchschnittliche Leistung in diesem Spiel abgeliefert haben, die mehr Tore erzielt haben als erwartet. Alle Spieler, die weniger Tore erzielt haben als erwartet, haben eine unterdurchschnittliche Leistung in diesem Spiel erzielt. Der Spieler, der laut LightGBM-Modell, die größte überdurchschnittliche Leistung in

diesem Spiel gezeigt hat, ist Sebastian Damm. Der Spieler, der die größte unterdurchschnittliche Leistung gezeigt hat, ist Lucas Krzikalla. Die gleiche Analyse könnte man mit dem XGBoost-Modell durchführen, allerdings sind die aufsummierten xG-Werte der beiden Modelle schwer miteinander vergleichbar, da die kontextbezogenen Merkmale in dem XGBoost-Modell fehlen. Als zusätzliche Metrik könnte man pro Spieler auch den durchschnittlichen xG-Wert pro Wurf berechnen, indem man den aufsummierten xG-Wert durch die Anzahl der Würfe dividiert. Dieser würde Aufschluss darüber geben, ob ein Spieler eher schwierige Würfe nimmt oder nicht. Anders ausgedrückt gibt der durchschnittliche xG-Wert pro Wurf die Torgefahr des Spielers an.

Name	Würfe	Tore	$xG_{P_t, K_t}$	$xG_{P_t}$
Linus Arnesson	10	7	6.25	7.19
Fabian Gutbrod	9	5	4.38	4.20
Tom Kare Nikolaisen	5	4	4.04	3.78
Patrick Wiesmach Larsen	6	4	4.06	4.24
Luca Witzke	7	4	4.00	4.12
Sebastian Damm	4	4	2.86	2.55
Arnor Thor Gunnarsson	3	3	2.15	1.71
Maciej Gebala	3	3	2.53	2.27
Tom Bergner	3	2	2.49	2.27
Lovro Jotić	6	2	2.98	2.99
Sime Ivic	4	2	2.33	2.41
Lukas Stutzke	4	2	2.83	2.72
Marko Mamic	5	2	2.40	2.79
Csaba Szücs	3	1	1.70	1.42
Gregor Remke	2	1	1.11	1.07
Alexander Weck	1	1	0.81	0.77
Simon Ernst	3	1	1.49	1.66
Marc Esche	3	1	2.02	2.22
David Schmidt	2	1	1.22	1.22
Lukas Binder	1	0	0.45	0.46
Simen Schønningsen	2	0	1.04	0.94
Lucas Krzikalla	2	0	1.53	1.52
Oskar Sunnefeldt	2	0	0.74	0.99

Tabelle 6.1.: Wurfstatistiken der Spieler vom SC DHfK Leipzig und dem Bergischen HC des 9. Spieltags in der Saison 2021-2022

Wenn man die xG-Werte pro Mannschaft für das vorher genannte Spiel aufsummiert, erhält man die Leistungsübersicht des gesamten Teams für dieses Spiel. In Tabelle 6.2 ist dies zu sehen. Es ist zu beobachten, dass der Bergische HC laut beiden Modellen in diesem Spiel eine überdurchschnittliche Leistung erbracht hat. Der SC DHfK Leipzig hingegen hat nach beiden Modellen eher unterdurchschnittlich Würfe abgeschlossen.

Darüber hinaus kann man den aufsummierten xG-Wert ebenfalls interpretieren als Torgefahr, die das Team innerhalb des Spiels erzeugt hat. Dies ist möglich, da der Wert die Frequenz der Würfe und die Qualität der Würfe beinhaltet. Je mehr Würfe ein Team auf das gegnerische Tor bringt, umso höher wird der xG-Wert und je klarer der Torversuch, desto höher ist der einzelne xG-Wert pro Wurf. Ein offensiv starkes Team sollte also die klareren und häufigeren Torchancen haben. Auch

hier wäre der durchschnittliche xG-Wert pro Wurf eine Möglichkeit die durchschnittliche Torgefahr einer Mannschaft pro Spiel oder auch pro Saison zu quantifizieren.

Team	Würfe	Tore	$xG_{P_t, K_t}$	$xG_{P_t}$
Bergischer HC	46	30	29.79	28.78
SC DHfK Leipzig	44	20	25.66	26.77

Tabelle 6.2.: xG-Statistik des DHfK Leipzig und des Bergischen HC des 9. Spieltags in der Saison 2021-2022

Der aufsummierte xG-Wert ist auf den ersten Blick eine Metrik für die Offensive einer Mannschaft. Allerdings kann man es auch als defensive Metrik sehen. Summiert man die xG-Werte für das Team, welches den Wurf verteidigt, auf, so erhält man einen Überblick darüber, wie viel Torgefahr eine Mannschaft zulässt. Somit könnte man die Verteidigung einer Mannschaft äquivalent zur Offensive bewerten.

## 7. Limitation

In diesem Kapitel wird darauf eingegangen, welche Limitationen die Modelle und diese Arbeit haben. Dabei wird darauf eingegangen, was man anders hätte machen können. Des weiteren wird auf die Forschungsfrage **RQ4** eingegangen, ob die erwarteten Tore für den Sport Handball eine geeignete Metrik für die Leistungsbewertung der Spieler oder Teams darstellt.

Die größte Limitation der Modelle sind die verwendeten Daten. Zum einen sind die Datensätze klein. Vor allem der Datensatz mit den zusätzlichen kontextbezogenen Merkmalen besteht nur aus Würfeln aus fünf Spielen. Dies limitiert das Training des Modells sowie auch die Interpretation des Modells. Durch die wenigen Würfe ist das Training des Modells extrem abhängig von den Würfeln, die für das Training ausgewählt werden. Die Ursache dafür ist der Fluch der Dimensionalität. Durch die wenigen Datensätze und die zusätzlichen Merkmale wird der komplette Datenraum nicht vollständig abgedeckt. Deshalb sind die Ergebnisse des LightGBM-Modells kritisch zu bewerten. Die Aussagekraft der Ergebnisse ist durch den Mangel an Daten limitiert. Auf den ersten Blick kann das Modell durch die zusätzlichen Merkmale verbessert werden. Um dies genauer sagen zu können, müsste man das Modell nochmals mit mehr Trainingsdaten trainieren und evaluieren. Diese Arbeit war ein Versuch das Konzept der erwarteten Tore auf Handball zu übertragen und ein Versuch die Modelle nicht ausschließlich auf den Positionsdaten basieren zu lassen. Um ein aussagekräftiges Modell zu erstellen, müssten die Positionsdaten aller Mannschaften der Bundesliga von Kinexon veröffentlicht werden. Aufgrund der verwendeten Trainingsdaten ist die Interpretation des LightGBM-Modells im Gegensatz zum XGBoost-Modell anders. Das XGBoost-Modell bildet ein Modell über die erwarteten Tore des durchschnittlichen Spielers aus ähnlichen Positionen der ersten Handball Bundesliga ab. Das LightGBM-Modell hingegen, ist ein Modell, welches sich nur auf die erwarteten Tore eines durchschnittlichen Spielers des Bergischen HC und des SC DHfK Leipzig bezieht. Ein eins zu eins Vergleich der beiden Modelle hinkt, da die erwarteten Tore der gesamten Bundesliga nicht gleich der erwarteten Tore der zuvor genannten beiden Mannschaften ist. Im Zuge dieser Arbeit werden beide Modelle dennoch verglichen, da das XGBoost-Modell als Vergleichsmodell dienen sollte, um das Konzept zu prüfen, ob das Modell mit weiteren Kontextinformationen genauer werden kann.

Eine weitere Limitation des LightGBM-Modells ist die Auswahl und die Berechnung der Merkmale. Diese basieren zum größten Teil auf Arbeiten aus anderen Sportarten, wie zum Beispiel Fußball oder Eishockey. Hier wäre eine Validierung, der in dieser Arbeit getroffenen Annahmen, durch einen Handball-Analysten ein erster Schritt. Des weiteren könnte man die Merkmale auch anders darstellen. Beispielsweise könnte man anstelle der Distanz zum Torhüter auch die x- und y-Koordinate des Torhüters verwenden. Baumverfahren wie XGBoost, Random Forest und LightGBM können mit den Koordinaten ebenfalls funktionieren. Eine Klassifikation der Spielsituation, aus der der Wurf entstanden ist, könnte ebenfalls das Modell verbessern. Hier wäre beispielsweise ein kategorisches Merkmal denkbar, das definiert, ob der Wurf aus einem Tempogegenstoß, nach einem Freiwurf oder einem normalen Spielaufbau heraus erfolgt ist. Dafür werden allerdings mehr Daten und ein Experte benötigt, der die Regeln für die Definition einer solchen Spielsituation festlegt. Hier wäre Videomaterial zu den Situationen von Vorteil, wenn nicht sogar notwendig. Dieses ist derzeit aber nicht frei zugänglich. Eine weitere Limitation sind die Daten, die bisher von Kinexon erfasst

werden. Hier wären zusätzliche Informationen zu dem Wurfarm oder die Art des Wurfes hilfreich. Diese könnten die Modelle weiter verbessern.

Ein weitere Limitation ist die Kombination von Merkmalen, die für das Training verwendet werden. Es wurde zu Beginn zwar mit einer automatischen Feature Selection experimentiert, aber die Idee wurde wieder verworfen, da die Evaluationsergebnisse schlechter waren als die hier vorgestellten Evaluationsergebnisse. Auch auf das manuelle Entfernen von Merkmalen, die keinen oder nur einen sehr geringen Einfluss auf die Modelle haben, wird in dieser Arbeit nicht genau eingegangen. Gegebenenfalls könnte das manuelle Eliminieren von anderen Merkmalen bessere Ergebnisse liefern. Das gleiche gilt für ein Over- beziehungsweise Undersampling der Trainingsdaten. Somit könnte man einen ausgeglichenen Trainingsdatensatz erhalten, in dem die möglichen Klassen, also Tor oder kein Tor, zu gleichen Teilen vertreten sind. Auch diese Idee wird in dieser Arbeit aus Mangel an Daten nicht weiter verfolgt.

Eine weitere Limitation der Modelle findet sich in der Vorverarbeitung der Daten. Hier wird auf eine Eliminierung von Ausnahmen innerhalb des Datensatzes verzichtet. Beispielsweise gibt es Würfe die aus der eigenen Spielhälfte erfolgen. Diese Würfe sind zwar selten, aber sie sind Teil des Spiels, weshalb diese in den Datensätzen enthalten bleiben. Diese Würfe sind allerdings sehr unterbesetzt in den Daten. Dies könnte dazu führen, dass Würfe aus ähnlichen Positionen und Spielsituationen zu hoch bewertet werden. Dies gilt vor allem dann, wenn Kontextinformationen zu diesen Würfen fehlen. Ein Beispiel hierfür ist eine sogenannte schnelle Mitte. Hierbei wird beim Anwurf direkt auf das gegnerische Tor geworfen, da die verteidigende Mannschaft sich noch nicht geordnet hat oder der gegnerische Torwart ungeeignet steht. Eine Option hierfür wäre, diese Würfe gegebenenfalls zu entfernen. Da es in dieser Arbeit nicht erfolgt ist, misst das XGBoost-Modell einer weiten Distanz einen hohen positiven Einfluss bei. Dies ist in Abbildung 5.4 zu sehen. Das führt dazu, dass Würfe aus ähnlichen Positionen auch ähnlich klassifiziert werden, auch wenn die Spielsituation eine komplett andere ist.

Eine weitere Limitation ist die Auswahl der Modellverfahren, die für das Training der Modelle ausgewählt werden. In dieser Arbeit wird sich nur auf die vier Verfahren logistische Regression, LightGBM, XGBoost und Random Forest beschränkt. Andere Modellverfahren könnten bessere Ergebnisse liefern. Beispiele, die man anstelle der hier verwendeten Verfahren verwenden könnte, wären AdaBoost, Generalized Linear Models oder auch künstliche Neuronale Netze. Dazu kommen die Hyperparameter, die pro Modell eingestellt werden können. Es wird zwar eine Optimierung der Hyperparameter durchgeführt, allerdings gibt es noch weitere pro Modellverfahren, die im Zuge dieser Arbeit nicht berücksichtigt werden. Andere Hyperparameter oder Selektionsintervalle könnten die Modelle weiter verbessern.

Zum Schluss gehen wir noch auf die Forschungsfrage **RQ4** und den Nutzen eines xG-Modells für den Handball ein. Die xG-Metrik wird in Sportarten wie Fußball oder Eishockey schon aktiv eingesetzt. Im Handball hingegen gibt es zwei verschiedene Arbeiten, die sich bisher mit dem Thema befasst haben. Hier stellt sich die Frage, warum dies der Fall ist. Ein Argument ist, dass die Natur des Spiels im Handball eine andere ist, als im Fußball oder im Eishockey. Im Handball fallen viel mehr Tore. Ein Tor ist keine Seltenheit wie in den anderen beiden Sportarten. Tore fallen im Handball viel regelmäßiger und in kürzeren Intervallen. Die Aussagekraft des xG-Werts im Handball sollte

also geringer sein als im Fußball, da die Torgefahr bei einem Wurf höher sein sollte. Allgemein sind knapp zwei Drittel aller Würfe ein Tor. Alternativen zur xG-Metrik im Handball wären der Leistungs-Index von Schwenkreis [34] oder Ansätze aus dem Basketball [64]. Die Leistungs-Index-Metriken werden bereits in der Handball Bundesliga verwendet [65]. Die Frage, ob die xG-Metrik einen Mehrwert zu den anderen Metriken im Handball hat, kann nicht final beantwortet werden. Trotzdem bietet der xG-Wert eine Möglichkeit oder eine Ergänzung für die Leistungsbewertung.



## 8. Zusammenfassung und Ausblick

In diesem Kapitel wird die Arbeit noch einmal kurz zusammengefasst, die Ergebnisse aufgezeigt und die Arbeit final abgeschlossen. Dabei wird auf die Forschungsfragen, die zum Anfang gestellt wurden, eingegangen. Abschließen wird die Arbeit mit einem Ausblick auf zukünftige Arbeiten zu erwarteten Toren im Handball.

### 8.1. Zusammenfassung

In dieser Arbeit werden zwei verschiedene Modelle für einen xG-Wert im Handball vorgestellt. Jedes Modell berechnet, mit welcher Wahrscheinlichkeit ein Wurf ein Tor erzielt. Das erste Modell wird nur mit Informationen zu den Positionen des Werfers trainiert. Dies führt zu einem Modell, welches für Würfe aus ähnlichen Positionen ähnliche Wahrscheinlichkeiten für ein Tor berechnet. Das zweite Modell wird mithilfe von den Positionsdaten des Werfers und zusätzlichen Kontextinformationen trainiert, die den Wurf und die Spielsituation genauer beschreiben. Dies führt zu einem Modell, wo angenommen wird, dass es Würfe in ähnlichen Spielsituationen von ähnlichen Positionen mit einer ähnlichen Wahrscheinlichkeit für einen Treffer approximiert.

Mithilfe der beiden Modelle, konnte gezeigt werden, dass zusätzliche Kontextinformationen die Qualität des Modells verbessern können und die Approximation der Wahrscheinlichkeiten verbessert werden kann. Zusätzlich dazu wird auf die Kalibrierung der Modelle eingegangen, die wichtig ist, damit die berechneten Wahrscheinlichkeiten auch als Wahrscheinlichkeiten interpretiert werden können. Hier konnte gezeigt werden, dass zumindest ein Modell gut kalibriert ist. Die Kalibrierung für das kontextbezogene Modell fällt schlechter aus, trotz verschiedenen Nachkalibrierungsmethoden. Die Ergebnisse der Modelle sind deshalb auch kritisch zu betrachten, da für das Training des kontextbezogenen Modells nur sehr wenig Daten verfügbar sind.

Es wird ebenfalls gezeigt, dass die Interpretierbarkeit der Modelle und der Ergebnisse ein wichtiger Faktor ist. Dafür wird ein Framework verwendet, welches zeigt, wie einzelne xG-Werte zustande kommen und welchen Einfluss einzelne Merkmale auf die Ergebnisse jedes Modells haben. Dabei stellt sich heraus, dass die Abwurfposition den größten Einfluss auf beide Modelle hat. Zusätzliche Kontextinformationen, wie die Anzahl der Verteidiger zwischen Werfer und Tor oder die Spielergeschwindigkeit, haben ebenfalls einen signifikanten Einfluss auf die Berechnung der xG-Werte.

Mit den Ergebnissen werden Beispiele gezeigt, wie diese für das Messen der Spieler- oder Mannschaftsleistung verwendet werden können. Die Leistung kann sich dabei auf ein Spiel, auf eine Situation oder auch auf eine Saison beziehen. Zusätzlich dazu werden Beispiele für weitere Metriken gegeben, die sich aus dem xG-Wert ableiten können.

Die Frage, ob die xG-Metrik für den Handball geeignet ist, kann nicht final beantwortet werden. Dennoch bietet diese eine Möglichkeit zur Leistungsbewertung der Spieler bzw. der Mannschaften.

## 8.2. Ausblick

Weitere Arbeiten zu diesem Thema könnten sich mit verschiedenen Modellverfahren inklusive Merkmalen, Datenvorverarbeitung oder Hyperparameter beschäftigen. Eine Hauptaufgabe für die Zukunft wird es sein, weitere Daten zu sammeln. Je mehr Seasons an Daten gesammelt und verfügbar gemacht werden, umso mehr kann man diese Arbeit weiterführen. Vor allem die Positionsdaten der Spieler sind hier interessant, um neue Merkmale für einen Wurf zu extrahieren. Zusätzlich wäre es denkbar, beispielsweise die letzten fünf Sekunden vor dem Wurf zu analysieren. Man könnte das Modell auch auf ein Intervall im Spiel anwenden, um sich in einem bestimmten Rhythmus den xG-Wert angeben lassen. Somit könnte man gegebenenfalls definieren, welche Aktion zu welchem Zeitpunkt die beste Entscheidung wäre.

Eine weitere Möglichkeit bieten Statistiken oder Metriken pro Spieler mit in das Modell einfließen zu lassen. Somit würde sich die Interpretation der Ergebnisse nochmals ändern. Man würde ein Modell erhalten, welches ähnliche Würfe zu ähnlichen Spielsituationen von Spielern mit gleichen Statistiken oder Fähigkeiten ähnlich klassifiziert. Damit könnte man mit der Metrik, die Spieler bewerten, beziehungsweise die Spielerbewertung nach einem gewissen Zeitraum anpassen. Dafür müssten man schauen, ob der Spieler in diesem Zeitraum eher eine über- oder unterdurchschnittliche Leistung erbracht hat.

Eine weitere Möglichkeit wäre, das Modell zu einem Post-Shot erwarteten Tor-Modell zu erweitern. Dieses ist eher dafür gedacht, die Fähigkeit eines Torhüters zu bewerten.

# Literatur

- [1] Elia Morgulev, Ofer H. Azar und Ronnie Lidor. „Sports analytics and the big-data era“. In: *International Journal of Data Science and Analytics* 5.4 (2018), S. 213–222. ISSN: 2364-4168. DOI: 10.1007/s41060-017-0093-7.
- [2] Kinexon. *Kinexon Spieler Tracking*. 2023. URL: <https://kinexon.com/de/technologie/spieler-tracking/> (besucht am 31.10.2023).
- [3] Friedemann Schwenkreis. „A Graded Concept of an Information Model for Evaluating Performance in Team Handball“. In: 2019. DOI: 10.5220/0007920001960202.
- [4] Bill James. *Baseball abstract*. 1981.
- [5] Michael Lewis. *Moneyball*. W. W. Norton & Company Incorporated, 2004. ISBN: 9780393066234.
- [6] Jim Albert u. a. *Handbook of statistical methods and analyses in sports*. Chapman und Hall/-CRC, 2017. ISBN: 9781315166070. DOI: <https://doi.org/10.1201/9781315166070>.
- [7] Richard Pollard, Jake Ensum und Samuel Taylor. „Estimating the probability of a shot resulting in a goal: The effects of distance, angle and space“. In: *International Journal of Soccer and Science* 2.1 (Jan. 2004), S. 50–55.
- [8] Craig Wright u. a. „Factors associated with goals and goal scoring opportunities in professional soccer“. In: *International Journal of Performance Analysis in Sport* 11.3 (Dez. 2011), S. 438–449. DOI: 10.1080/24748668.2011.11868563.
- [9] Brian Macdonald. „An expected goals model for evaluating NHL teams and players“. In: *Proceedings of the 2012 MIT Sloan Sports Analytics Conference* (Jan. 2012).
- [10] Michael Caley. *The Minute-by-Minute Database III: New Expected Goals and introducing Si-BoT*. 2013. URL: <https://cartilagefreecaptain.sbnation.com/2013/6/27/4463922/tottenham-hotspur-analysis-epl-expected-goals-statistics> (besucht am 12.07.2023).
- [11] Paolo Cintia u. a. „The harsh rule of the goals: Data-driven performance indicators for football teams“. In: *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Okt. 2015, S. 1–10. DOI: 10.1109/DSAA.2015.7344823.
- [12] Sam Green. *Assessing the performance of Premier League goalscorers*. OptaPro Blog, 2012. URL: <https://www.statsperform.com/resource/assessing-the-performance-of-premier-league-goalscorers/> (besucht am 03.07.2023).
- [13] Colin Trainor. *Chelsea’s Striker Options*. 2013. URL: <https://statsbomb.com/articles/soccer/chelseas-striker-options/> (besucht am 31.10.2023).
- [14] Patrick Lucey u. a. „quality vs quantity: Improved shot prediction in soccer using strategic features from spatiotemporal data“. In: *Proceedings of 8th Annual MIT Sloan Sports Analytics Conference (pp. 1-9)*. Hrsg. von SLOAN. 2015.

- [15] H. Eggels, R. van Elk und M. Pechenizkiy. „Explaining soccer match outcomes with goal scoring opportunities predictive analytics“. English. In: *Proceedings of the Workshop on Machine Learning and Data Mining for Sports Analytics 2016 co-located with the 2016 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2016)*. CEUR Workshop Proceedings. 3rd Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA 2016), MLSA 2016 ; Conference date: 19-09-2016 Through 19-09-2016. CEUR-WS.org, 2016.
- [16] William Spearman. „Beyond Expected Goals“. In: *Sloan sports analytics conference*. Hrsg. von SLOAN. 2018.
- [17] James H. Hewitt und Oktay Karakuş. „A machine learning approach for player and position adjusted expected goals in football (soccer)“. In: *Franklin Open* 4 (Sep. 2023), S. 100034. DOI: 10.1016/j.fraope.2023.100034. URL: <https://doi.org/10.1016%2Fj.fraope.2023.100034>.
- [18] James Mead, Anthony O’Hare und Paul McMenemy. „Expected goals in football: Improving model performance and demonstrating value“. In: *PLOS ONE* 18.4 (2023), e0282295. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0282295.
- [19] Gabriel Anzer und Pascal Bauer. „A Goal Scoring Probability Model for Shots Based on Synchronized Positional and Event Data in Football (Soccer)“. In: *Frontiers in sports and active living* 3 (2021), S. 624475. DOI: 10.3389/fspor.2021.624475.
- [20] Pau Madrero Pardo. „Creating a model for expected Goals in football using qualitative player information“. Diss. Universitat Politècnica de Catalunya, 18. Sep. 2020. URL: <https://upcommons.upc.edu/handle/2117/328922> (besucht am 31.10.2023).
- [21] Taylor Paerels. *Play for the Point or Go for the Win: Expected Goals in the National Hockey League*. 2020. URL: <https://scholarworks.calstate.edu/downloads/n583z1407> (besucht am 31.10.2023).
- [22] Pensare Pallamano. *A STATISTICAL MODEL FOR HANDBALL: SPAM – HANDBALL*. 2017. URL: <https://pensarepallamano.wordpress.com/2017/01/02/a-statistical-model-for-handball-spam-handball/> (besucht am 22.06.2023).
- [23] Julian Rux. *Welches Team wirft am effizientesten? Expected Goals für die HBL*. 2020. URL: <https://handballytics.de/expected-goals-wurfeffizienz/> (besucht am 22.06.2023).
- [24] European Handball Federation. *How the next Generation of Data Analysis is improving Handball*. 2021. URL: <https://www.eurohandball.com/en/news/en/how-the-next-generation-of-data-analysis-is-improving-handball/> (besucht am 09.04.2023).
- [25] Dimitris Hatzimanouil. „Throwing Effectiveness per Throwing Area and Playing Position among High Level Handball Players“. In: *JOURNAL OF PHYSICAL EDUCATION AND SPORTS MANAGEMENT* 6.1 (2019). ISSN: 23732156. DOI: 10.15640/jpesm.v6n1a2.
- [26] P. Meletakos, G. Vagenas und I. Bayios. „A multivariate assessment of offensive performance indicators in Men’s Handball: Trends and differences in the World Championships“. In: *International Journal of Performance Analysis in Sport* 11.2 (2011), S. 284–294. DOI: 10.1080/24748668.2011.11868548.

- [27] Herbert Wagner u. a. „Individual and team performance in team-handball: A review“. English. In: *Journal of Sports Science and Medicine* 13.4 (Dez. 2014), S. 808–816. ISSN: 1303-2968.
- [28] Marco Cardinale u. a. „Activity Profiles and Positional Differences of Handball Players During the World Championships in Qatar 2015“. In: *International journal of sports physiology and performance* 12.7 (2017), S. 908–915. DOI: 10.1123/ijssp.2016-0314.
- [29] Willian Roque Ferrari, Hugo Sarmiento und Vasco Vaz. „Match Analysis in Handball: A Systematic Review“. In: *Montenegrin Journal of Sports Science and Medicine* 8.2 (2019), S. 63–76. ISSN: 18008755. DOI: 10.26773/mjssm.190909.
- [30] Matthias Hinz u. a. „Differences in Decision-Making Behavior Between Elite and Amateur Team-Handball Players in a Near-Game Test Situation“. In: *Frontiers in psychology* 13 (2022), S. 854208. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2022.854208.
- [31] Christian Saal u. a. „Physical match demands of four LIQUI-MOLY Handball-Bundesliga teams from 2019-2022: effects of season, team, match outcome, playing position, and half-time“. In: *Frontiers in sports and active living* 5 (2023), S. 1183881. DOI: 10.3389/fspor.2023.1183881.
- [32] Friedemann Schwenkreis. „An Approach to Use Deep Learning to Automatically Recognize Team Tactics in Team Ball Games“. In: *DATA 2018*. Hrsg. von Jorge Bernardino und Christoph Quix. Setúbal, Portugal: SCITEPRESS - Science and Technology Publications Lda, 2018, S. 157–162. ISBN: 978-989-758-318-6. DOI: 10.5220/0006823901570162.
- [33] Friedemann Schwenkreis und Eckard Nothdurft. „Applied Data Science: An Approach to Explain a Complex Team Ball Game“. In: *DATA 2020*. Hrsg. von Slimane Hammoudi, Christoph Quix und Jorge Bernardino. Setúbal, Portugal: SCITEPRESS - Science and Technology Publications Lda, 2020, S. 135–142. ISBN: 978-989-758-440-4. DOI: 10.5220/0009561301350142.
- [34] Friedemann Schwenkreis. „Why the Concept of Shopping Baskets helps to analyze Team-Handball“. In: *2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*. 2020, S. 4–10. DOI: 10.1109/IDSTA50958.2020.9264068.
- [35] Gareth James u. a. *An Introduction to Statistical Learning: with Applications in R*. Springer, 2013. URL: <https://faculty.marshall.usc.edu/gareth-james/ISL/>.
- [36] Leo Breiman. „Random Forests“. English. In: *Machine Learning* 45.1 (2001), S. 5–32. ISSN: 0885-6125. DOI: 10.1023/A:1010933404324. URL: <http://dx.doi.org/10.1023/A%3A1010933404324>.
- [37] Tianqi Chen und Carlos Guestrin. „XGBoost: A Scalable Tree Boosting System“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: ACM, 2016, S. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: <http://doi.acm.org/10.1145/2939672.2939785>.
- [38] Guolin Ke u. a. „LightGBM: A Highly Efficient Gradient Boosting Decision Tree“. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Hrsg. von Isabelle Guyon u. a. 2017, S. 3146–3154. URL: <https://proceedings.neurips.cc/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html>.

- [39] Daniel Ramos u. a. „Deconstructing Cross-Entropy for Probabilistic Binary Classifiers“. In: *Entropy (Basel, Switzerland)* 20.3 (2018). DOI: 10.3390/e20030208.
- [40] J. A. Hanley und B. J. McNeil. „The meaning and use of the area under a receiver operating characteristic (ROC) curve“. In: *Radiology* 143.1 (1982), S. 29–36. ISSN: 0033-8419. DOI: 10.1148/radiology.143.1.7063747.
- [41] Alexandru Niculescu-Mizil und Rich Caruana. „Predicting good probabilities with supervised learning“. In: *Proceedings of the 22nd international conference on Machine learning - ICML '05*. New York, New York, USA: ACM Press, 2005. DOI: 10.1145/1102351.1102430.
- [42] John Platt. „Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods“. In: *Adv. Large Margin Classif.* 10 (Juni 2000).
- [43] Bianca Zadrozny und Charles Elkan. „Transforming Classifier Scores into Accurate Multiclass Probability Estimates“. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Aug. 2002). DOI: 10.1145/775047.775151.
- [44] Marco Tulio Ribeiro, Sameer Singh und Carlos Guestrin. „Why Should I Trust You?: Explaining the Predictions of Any Classifier“. 16. Feb. 2016. URL: <https://arxiv.org/pdf/1602.04938.pdf>.
- [45] Marco Tulio Ribeiro, Sameer Singh und Carlos Guestrin. „Anchors: High-Precision Model-Agnostic Explanations“. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018). ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.11491.
- [46] Scott M. Lundberg und Su-In Lee. „A unified approach to interpreting model predictions“. In: *CoRR* abs/1705.07874 (2017). arXiv: 1705.07874. URL: <http://arxiv.org/abs/1705.07874>.
- [47] Scott M. Lundberg u. a. „From local explanations to global understanding with explainable AI for trees“. In: *Nat. Mach. Intell.* 2.1 (2020), S. 56–67. DOI: 10.1038/s42256-019-0138-9. URL: <https://doi.org/10.1038/s42256-019-0138-9>.
- [48] Christoph Molnar. *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. 2. Aufl. 2022. URL: <https://christophm.github.io/interpretable-ml-book>.
- [49] Soumen Chakrabarti u. a. „Data mining curriculum: A proposal (version 1.0)“. In: *Intensive working group of ACM SIGKDD curriculum committee* 140 (2006), S. 1–10.
- [50] R. Wirth und Jochen Hipp. „CRISP-DM: Towards a standard process model for data mining“. In: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (Jan. 2000).
- [51] Liquimoly Handball Bundesliga. *Liquimoly Handball Bundesliga*. 2023. URL: <https://www.liquimoly-hbl.de/de/> (besucht am 09.04.2023).
- [52] Jed C. Davies. *Coaching the tiki taka style of play*. SoccerTutor.com, 2013. ISBN: 9780957670549.
- [53] Simon Thorpe, Denis Fize und Catherine Marlot. „Speed of processing in the human visual system“. In: *nature* 381.6582 (1996), S. 520–522.
- [54] Tarak Kharrat, Ian G. McHale und Javier López Peña. „Plus–minus player ratings for soccer“. In: *European Journal of Operational Research* 283.2 (2020), S. 726–736.

- [55] Guido van Rossum. *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020.
- [56] The pandas development team. *pandas-dev/pandas: Pandas*. 2023. DOI: 10.5281/ZENODO.7549438.
- [57] Charles R. Harris u. a. „Array programming with NumPy“. In: *nature* 585.7825 (2020), S. 357–362. DOI: 10.1038/s41586-020-2649-2.
- [58] F. Pedregosa u. a. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830.
- [59] Pauli Virtanen u. a. „SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python“. In: *Nature Methods* 17 (2020), S. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [60] Thomas A Caswell u. a. *matplotlib/matplotlib: REL: v3.5.3*. 2022. DOI: 10.5281/ZENODO.6982547.
- [61] Michael L. Waskom. „seaborn: statistical data visualization“. In: *Journal of Open Source Software* 6.60 (2021), S. 3021. DOI: 10.21105/joss.03021.
- [62] Dominik Raabe u. a. *floodlight – A high-level, data-driven sports analytics framework*. 3. Juni 2022. URL: <https://arxiv.org/pdf/2206.02562>.
- [63] Chuan Guo u. a. „On calibration of modern neural networks“. In: *International conference on machine learning*. 2017, S. 1321–1330.
- [64] Dan Cervone u. a. „Pointwise: Predicting points and valuing decisions in real time with nba optical tracking data“. In: *Proceedings of the 8th MIT Sloan Sports Analytics Conference, Boston, MA, USA*. Hrsg. von SLOAN. Bd. 28. 2014, S. 3. URL: [http://www.lukebornn.com/papers/cervone\\_ssac\\_2014.pdf](http://www.lukebornn.com/papers/cervone_ssac_2014.pdf).
- [65] Liquimoly Handball Bundesliga. *Liquimoly Handball Bundesliga*. 2023. URL: <https://www.liquimoly-hbl.de/de/s/handball-performance-index/2021-22/liqui-moly-hbl/> (besucht am 31.10.2023).

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit mit dem Thema:

*„Sports Analytics im Handball: Erstellen eines Modells zum Berechnen von erwarteten Toren auf Basis von Raum-Zeit Sensordaten“*

selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, den 14.11.2023

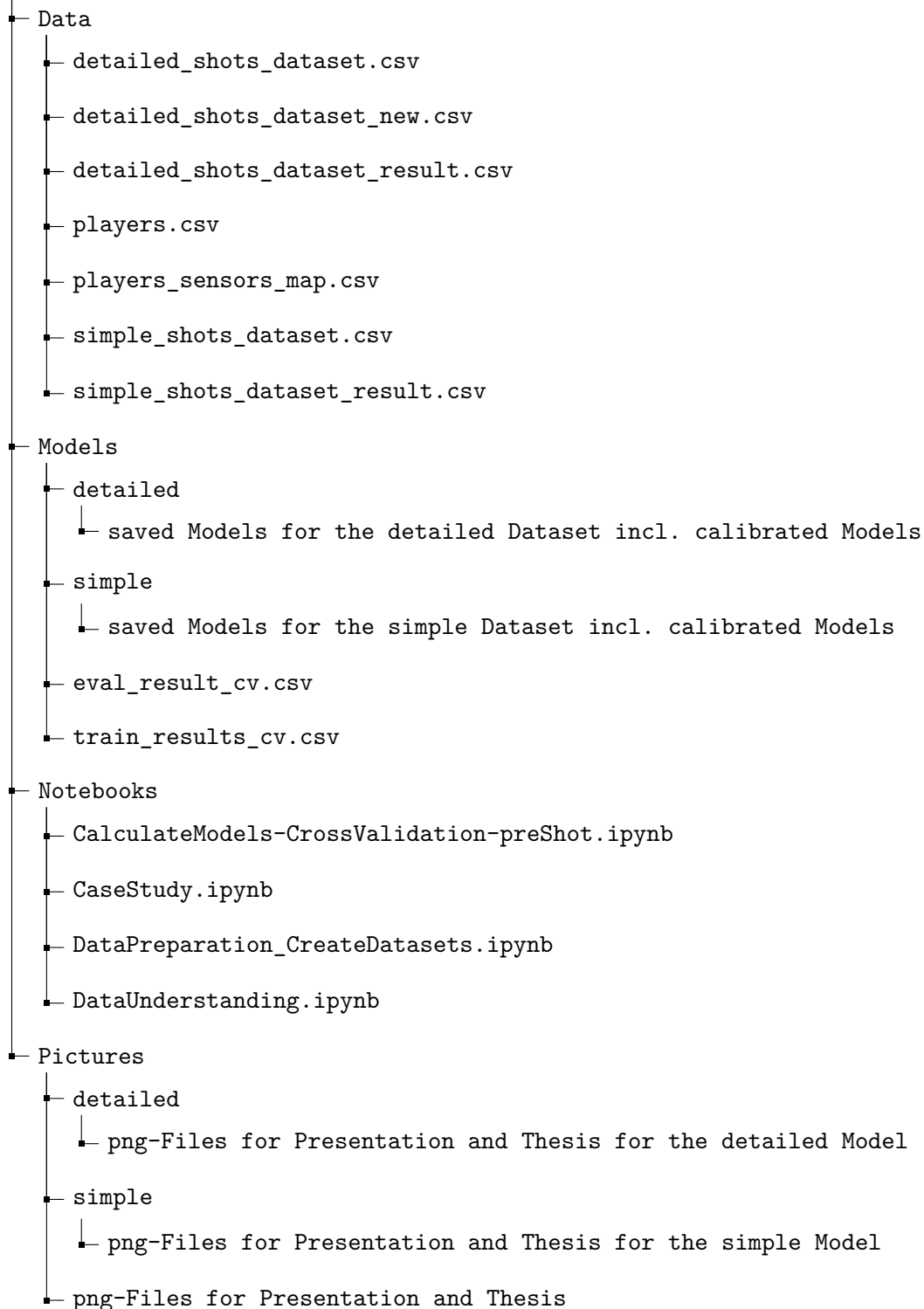
---

ANDREAS KRETSCHMER



## A. Ordnerstruktur

handball-expected-goals



## B. Dateischemata

### B.1. Schema der Spiel-Liste

games\_list.csv

- match\_id
- date
- game\_day
- home\_team
- away\_team
- saison

## B.2. Schema der Wurf-Liste

shot\_list.csv

- player\_id
- timestamp
- success
- hit\_position\_z
- hit\_position\_y
- shot\_position\_y
- shot\_position\_x
- distance
- speed\_ball
- shot\_type
- game\_clock
- validated
- goalkeeper\_id
- id
- league\_id
- goalkeeper\_league\_id
- action\_id
- team
- last\_group
- event
- updated\_at
- team\_id
- match\_id

### B.3. Schema der Positionsdaten

Match\_positions.csv

- ts in ms
- formatted local time
- sensor id
- mapped id
- number
- full name
- league id
- group id
- group name
- speed in m/s
- direction of movement in deg
- acceleration in m/s<sup>2</sup>
- total distance in m
- heart rate in bpm
- core temperature in celsius
- metabolic power in W/kg
- ball possession (id of possessed ball)
- acceleration load

## B.4. Schema der Trägheitsdaten

Match\_inertials.csv

- └ ts in ms
- └ formatted local time
- └ sensor id
- └ mapped id
- └ number
- └ full name
- └ league id
- └ group id
- └ group name
- └ x in m
- └ y in m
- └ z in m
- └ heart rate in bpm
- └ rr interval in ms as
  - └ n1
  - └ n2
  - └ ...
- └ unfiltered player orientation in deg
- └ accelerometer in g as
  - └ x1:y1:z1
  - └ x2:y2:z2
  - └ ...
- └ gyroscope in rad/s as
  - └ x1:y1:z1
  - └ x2:y2:z2
  - └ ...
- └ orientation as
  - └ quat11:quati1:quatj1:quatk1
  - └ quat12:quati2:quatj2:quatk2
  - └ ...