FREIE UNIVERSITÄT BERLIN

FACHBEREICH WIRTSCHAFTSWISSENSCHAFT

Projektdokumentation im Modul Business Intelligence
des Fachbereichs Wirtschaftswissenschaft
der Freien Universität Berlin

Projektdokumentation:
Kobe Bryant Shot Selection

Vorgelegt von

[redacted]

Betreuer: Jun.-Prof. Dr. Bastian Amberg

[redacted]

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

1. EINLEITUNG

Kobe Bryant spielte zehn Jahre bei den LA Lakers und gilt als einer der erfolgreichsten Basketballer seiner Zeit. Während seiner Spiele wurde jeder Kobes Wurfversuche dokumentiert und in einem Datensatz aggregiert. In diesem Projekt soll ein Vorhersagemodell entwickelt werden, dass anhand der Ausgangssituation eines Wurfes voraussagt, ob er gelingt oder fehlschlägt. Das Vorgehen dieser Arbeit orientiert sich an dem CRISIP-DM Prozess, der sich in insgesamt sechs Schritte gliedert. Am Anfang steht das Project Understanding mit dem Ziel das Projekt inhaltlich zu präzisieren und das Ziel zu definieren. Im anschließenden Data Understanding wird untersucht, welche Daten vorhanden sind und wie sie sich zur Zielerfüllung eignen. Nachfolgend werden die vorhandenen Daten im Schritt Data Preparation aufbereitet. Dazu gehören etwa das Reinigen und Prüfen der Daten hinsichtlich fehlerhafter Werte sowie die Restrukturierung der Daten.

Auf Basis der bearbeiteten Daten findet die Modellierung statt, welche die Entwicklung des Vorhersagemodells umfasst. Die Ergebnisse des Modells werden anschließend hinsichtlich des initial definierten Projektziels evaluiert. Den Abschluss der Arbeit stellt eine Reflexion der erreichten Ergebnisse dar und zeigt einen Ausblick sowie Grenzen des gewählten Vorgehens und Projekts.

2. CRIPS-DM: PROJECT UNDERSTANDING

2.1 Content

Die National Basketball Association (NBA) ist die Basketball-Profiliga von Nordamerika und gilt als die mit Abstand stärkste und bekannteste Basketball-Liga der Welt. 30 Mannschaften, aufgeteilt in eine East- und eine West-Conference, kämpfen jährlich um den Einzug in die Play-Offs, um dann am Ende um die Meisterschaft zu spielen. Wer in dieser Liga als Spieler auflaufen möchte, muss im jährlichen, sogenannten Entry-Draft von einer der Mannschaften ausgewählt werden. Es lässt sich leicht erkennen, dass es nicht nur für Spieler einen harten Weg in die NBA bedeutet, sondern auch für die einzelnen Mannschaften ein hart umkämpfter Weg bis an die Spitze der Liga ist. Einer der erfolgreichsten Spieler der NBA ist Kobe Bryant. Er ist 198 cm groß und spielte von 1996 bis 2016 für die Los Angeles Lakers auf der Position des Shooting Guards. Kobe Bryant wurde seit 1998 ununterbrochen in die Auswahl der besten NBA-Spieler (AllStar-Team) der jeweiligen Saison gewählt und gewann 2008 die Wahl zum MVP (Most Valuable Player), die Auszeichnung zum wertvollsten NBA-Spieler der regulären Saison. Er galt als einer der besten Scorer der NBA zu seiner Zeit.

Um als Spieler in der NBA aufgenommen zu werden, muss man nicht nur Talent aufweisen, sondern auch den Willen an seinen Schwächen zu arbeiten. Hierfür werden Statistiken genutzt, um diese herauszufiltern. Kobe Bryant hat während seiner Karriere als Basketballspieler immer wieder an seinen Schwächen gearbeitet und versucht diese zu verbessern. So ist er zu einem der größten Spieler seiner Zeit geworden. Natürlich entwickelt ein langjähriger Sportler ein Gefühl für seine Fähigkeiten und kann einschätzen, welche dieser noch ausbaufähig sind. Nichts desto trotz sind statistische Auswertungen eine wichtige Grundlage für einen Sportler, um Gewissheit darüber zu bekommen.

Im Basketball wollen aber nicht nur die Spieler selbst Informationen über ihre Trefferquoten und den damit zusammenhängenden Einflussfaktoren bekommen. Eine wichtige Aufgabe vor jedem Spiel ist das vorbereitende Scouting der gegnerischen Mannschaften. So können Trainer die gegnerischen Spieler besser einschätzen und eine geeignete Taktik für das Spiel entwickeln. Um einen Vorteil daraus zu generieren, werden die eigenen Spieler vor einem Spiel darüber instruiert, welche Stärken und Schwächen die Gegner besitzen. Eine statistische Analyse verspricht also nicht nur dem Spieler selbst eine Verbesserung, sondern hilft auch Teams sich auf kommende Spiele besser vorbereiten zu können.

2.2 Project Goal

Ziel dieses Projekts ist es, anhand der Daten Vorhersagen darüber treffen zu können, ob in gewissen Spielsituationen ein bestimmter Wurf durch Kobe getroffen wird oder nicht. Dabei sollte die Vorhersage besser als die des Major Classifiers mit 55,1% sein (vgl. *3.4 Data Visualization*). In den Daten befinden sich neben der Ausprägung eines (Nicht-)Treffers verschiedene Spezifikationen eines Wurfs. Dazu gehört die Art des Wurfs, die Distanz zum Korb und die verbleibenden Minuten des Spiels. Analysiert werden diese Daten mit den Tools *Pandas* und *sklearn* in der Programmiersprache Python. Das Vorgehen dabei wird nach dem CRISP-DM gegliedert.

Die genutzten Methoden, um eine möglichst genaue Vorhersage zu treffen, können Spieler sowie Mannschaften in der Zukunft für ihre eigenen Bedürfnisse nutzen. Hieraus können bspw. Spieler Schlussfolgerungen für ihre weitere Karriere und ihre Trainingsschwerpunkte treffen, als auch Mannschaften Spielstrategien aufstellen, um sich bestmöglich auf das nächste Spiel vorzubereiten.

2.3 Domain Knowledge

Ob ein Wurf im Basketball getroffen wird oder nicht, wird von verschiedenen Faktoren beeinflusst. In den Daten werden Faktoren abgebildet, die durch einen Beobachter objektiv messbar sind. Dazu gehört beispielsweise die Art des Wurfs, die Distanz zum Korb und die verbleibende Spielzeit. So ist durch den aufkommenden Druck und die Verausgabung der Spieler zum Ende des Spiels die Trefferquote oft geringer als zu Beginn. Zudem sorgt die 24-Sekunden-Uhr, die die verbleibende Zeit eines Teams für einen Angriff angibt, für eine erhöhte Spielgeschwindigkeit. Diese objektiv beobachtbaren Faktoren hängen allerdings von den Fähigkeiten des Spielers, wie seiner Kondition oder seinem Wurfgeschick, ab. Aber auch die persönliche und körperliche Verfassung des Spielers stellt durchaus einen wichtigen Einflussfaktor dar.

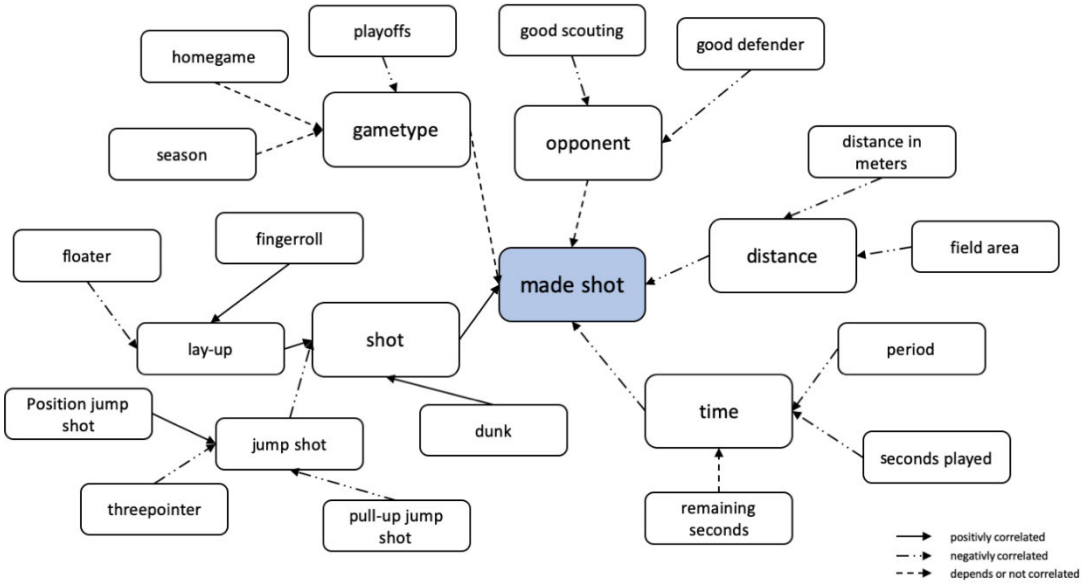

Abbildung 1: Cognitive Map

Um die einzelnen Abhängigkeiten darzustellen, wird zur Veranschaulichung der verschiedenen Einflussfaktoren eine Cognitive Map (*Abbildung 1*) genutzt. Es wird eine Vermutung illustriert, in welchem Zusammenhang die einzelnen Daten mit dem Projektziel stehen und welchen Einfluss diese haben. So steht im Mittelpunkt der *Abbildung 1* der erfolgreiche Wurf und es findet eine Klassifizierung der Korrelation einzelner Faktoren statt. So wird beispielsweise vermutet, dass die Distanz zum Korb negativ mit einem erfolgreichen Wurf korreliert oder eine positive Korrelation von Wurftypen zu einem erfolgreichen Korb besteht.

3. CRIPS-DM: DATA UNDERSTANDING

Um eine möglichst präzise Treffer-Vorhersage über Kobes Würfe tätigen zu können, wird eine große Anzahl an relevanten Daten benötigt. Zur Verfügung steht eine CSV-Datei mit Einträgen zu circa 30.500 Würfen. Die vorhandenen Daten werden nachfolgend beschrieben, veranschaulicht und auf ihre Qualität untersucht.

3.1 Data Description

Die für dieses Projekt zu Verfügung gestellte Datei umfasst Daten zu Würfen aus allen Spielen an denen Kobe Bryant in seiner Karriere jemals teilgenommen hat. Dabei stehen wurf- und spielspezifizierende Informationen zu Verfügung. Kobes Wurf wird durch einen Wurftyp (z.B. *Jump Bank Shot, Driving Dunking Shot*), die Distanz zum Korb, seiner Position auf dem Spielfeld, die verbleibenden Spielzeit und ob der Wurf getroffen hat beschrieben. Die Wurfdaten werden in der Datei weiter aggregiert und so die Wurftypen in generischen Kategorien zusammengefasst dargestellt. So werden beispielsweise übergeordnete Kategorien wie *Jump Shot* für alle einzelnen Jump Shot Typen erstellt oder die Zeit in Perioden aufgeteilt. Weiterhin werden seine Positionsdaten in Zonen (z.B. *Mid-Range, Left-Side*) und Distanzen kategorisiert (*less than 8ft, 8-16 ft*, etc.) und in zwei oder drei Punkte Würfe unterteilt. Neben den Wurf-Daten, stehen auch Informationen zu dem aktuellen Spiel zu Verfügung. Dazu gehört der Gegner, das Datum, ob es sich um ein Auswärtsspiel handelt und ob es ein Playoff Spiel ist. Da die Datei zahlreiche für den Wurf objektiv relevante Informationen enthält, ist davon auszugehen, dass sie sich gut für Kobes Wurfanalyse eignet.

3.2 Attribute Understanding

Für das weitere Vorgehen ist es maßgeblich die Dimension und Bedeutung der zu Verfügung gestellten Daten herauszuarbeiten. Die nachfolgende *Tabelle 1* stellt alle verfügbaren Attribute, deren Dimension und eine Beschreibung davon dar.

Attribut	Dimension	Beschreibung
shot_id	Integer	Aufsteigende eindeutige ID des Wurfs
action_type	Kategorisch: 57 verschiedene Typen	Eine detaillierte Wurfart. Z.B. Driving Dunk Shot oder Driving Finger Roll Shot
combined_shot_type	Kategorisch: Jump Shot, Dunk, Layup, Tip Shot, Hook Shot, Bank Shot	Kategorisierung des Attributs *action_type*
shot_distance	Integer ϵ (0,94)	Distanz zum Korb in Fuß
shot_zone_range	Less Than 8 ft., 8-16 ft., 16-24 ft., 24+ ft., Back Court Shot	Kategorisierung der Distanzen
shot_type	Kategorisch: 2PT Field Goal, 3PT Field Goal	Wie viele Punkte der Wurf gebracht hat / hätte
shot_zone_area	Kategorisch: Right Side, Left Side, Left Side Center, Right Side Center, Center, Back Court	Horizontale Kategorisierung der Position auf dem Feld
shot_zone_basic	Kategorisch: Mid-Range, Restricted Area, In The Paint, Above the Break, Right Corner, Backcourt, Left Corner	Vertikale Kategorisierung der Position auf dem Feld
period	Integer	Um welche Periode es sich handelt. Bei Verlängerungen ist ein Wert größer 4 möglich
minutes_remaining	Integer ϵ (0,11)	Verblendende Minuten in einem Viertel / einer Periode
seconds_remaining	Integer ϵ (0,59)	Wie viele Sekunden verbleiben in dem Attribut *minutes_remaining*
lat	Integer	Geokoordinate (Breitengrad)
lon	Integer	Geokoordinate (Längengrad)
loc_x	Integer	x-Koordinate auf dem Spielfeld
loc_y	Integer	y-Koordinate auf dem Spielfeld
playoffs	Boolean	Ob es sich um ein Playoff Spiel handelt
shot_made_flag	Boolean	Ob er getroffen hat
game_event_id	Integer	Eindeutige ID für die Art des Events (z.B. eine ID für regular Season oder Playoffs)
game_id	Integer	Aufsteigende eindeutige ID des Spiels
season	Datum: YYYY-Quartal	Um welches Jahr und Saison es sich handelt (4 Saisons möglich)
team_id	Integer	ID der LA Lakers (immer 1610612747)
team_name	String	Name des Teams (immer Los Angeles Lakers)
game_date	Date: MM/DD/YYYY	Datum des Spiels
matchup	String	Beschreibt wer gegen wen spielt und ob es ein Heimspiel ist
opponent	String	Name des Gegners

Tabelle 1: Attributbeschreibung

Ziel ist es, aus allen vorhandenen Attributen der Tabelle in *Abbildung 2* die relevantesten herauszufiltern und anhand deren das Feld *shot_made_flag* vorherzusagen. Trivial erscheint beispielsweise eine Korrelation zwischen der Distanz zum Korb und der Trefferquote. Wie es

sich jedoch mit anderen Attributen verhält und ob anhand einer ganzheitlichen Betrachtung oder mit bestimmten Permutationen von Attributen eine präzisere Aussage bzgl. des *shot_made_flags* getroffen werden kann, gilt es zu erarbeiten. Auffällig bei den Daten ist die Redundanz mancher Informationen. So kann aus den Koordinaten des Wurfs (*loc_x* und *loc_y*) die *shot_distance*, *shot_zone_range*, *shot_zone_are*, *shot_zone_basic* und der *shot_type* errechnet werden. Alle genannten Attribute beruhen auf der Positionierung von Kobe. So sollte bei dem weiteren Vorgehen darauf geachtet werden, redundante Informationen nicht doppelt einzubeziehen.

3.3 Data Quality

Zur Beurteilung der Datenqualität sind drei Kriterien entscheidend: die Vollständigkeit der Daten, die Richtigkeit der Daten und ob sie zielgerichtet sind.

Die Vollständigkeit der Daten kann anhand von öffentlichen externen Informationen grob abgeschätzt werden. Kobe Bryant hat in seinem Leben 1.346 Spiele in der regulären Saison und zusätzlich 220 Playoff Spiele für die LA Lakers bestritten. Dabei hat er insgesamt 33.643 Punkte erzielt. Geht man nun grob davon aus, dass die Quote von zwei- zu dreipunkte Würfen zwei zu eins ist, ergibt das ungefähr 2,3 Punkte pro Wurf. Daraus ergibt sich $\frac{33.643\ Punkte}{2{,}3\ \text{Punkte/Wurf}} = 14.672$ Würfe mit Treffern. Mit einer Trefferquote von etwa 40% ergibt das circa 36.500 Würfe. Zu Verfügung stehen Daten von ~30.500 Würfen. In den 33.643 Karriere Punkten, sind jedoch auch Freiwürfe enthalten (in den betrachteten Daten nicht). Bei einer Reduzierung um diese Punkte, scheint die grobe Schätzung etwa den verfügbaren Daten zu entsprechen. Es scheinen also keine bzw. keine maßgebliche Anzahl von Würfen zu fehlen oder zu viel zu sein. Nun gilt es zu prüfen, ob die einzelnen Tupel vollständig sind. Nach einer Analyse der Daten ergibt sich, dass jedes Attribut in jeder Zeile einen Eintrag in der korrekten Dimension hat. Lediglich bei dem zu untersuchenden Wert *shot_mad_flag* fehlen 5.000 Einträge. Wie damit umgegangen wurde, wird in *4.2 Data Cleaning* dargestellt.

Die Richtigkeit eines großen Anteils der Daten kann derivativ ermittelt werden. Geht man davon aus, dass die *loc_x* und *loc_y* korrekt ist, kann wie in *4.2 Data Cleaning* erläutert die *shot_distance*, *shot_zone_range*, *shot_zone_are*, *shot_zone_basic* und der *shot_type* validiert werden. Dabei konnte ermittelt werden, dass nur 516 der Einträge falsch klassifiziert wurden. Diese Fehler werden wie in *4.2 Data Cleaning* beschrieben berücksichtigt. Weitere Daten können nur insofern überprüft werden, ob sie der korrekten Domäne angehöhren. Also ob es

beispielsweise *seconds_remaining* > 59 oder eine *shot_distance* von mehr als 94 Fuß gibt. Bei der Untersuchung wurden keine auffälligen oder fehlerhaften Daten identifiziert.

Ziel ist es herauszufinden, ob Kobe Bryant einen Wurf trifft oder nicht. Also ist die Entscheidungsvariable *shot_made_flag* essenziell für die Analyse ob die Daten zielgerichtet sind. Die verfügbaren Daten umfassen einen Großteil objektiv beobachtbarer realitätsnaher Daten. Weiterhin kann die verteidigende Person implizit über die Betrachtung des gegnerischen Teams in Kombination mit der Saison / dem Datum berücksichtigt werden. Auch weitere Informationen, wie der Winkel des Wurfs, können anhand der Position auf dem Spielfeld ermittelt werden. Es ist also davon auszugehen, dass entweder alle objektiv relevanten Daten vorliegen oder ermittelt werden können. Möglich wäre es noch weitere Daten wie ein Momentum (Wurden die letzten n Spiele gewonnen? Hat er die letzten n Würfe getroffen?) herzuleiten. Subjektive Daten, wie Kobe Bryants Stimmung vor einem Spiel oder dem Respekt vor einem Gegner, sind schwer messbar und auch aus den Daten nicht zu ermitteln. Diese Einflüsse auf einen Wurf sollten jedoch durch die Vielzahl von Spielen und Würfen einen geringen Einfluss haben. Es ist also davon auszugehen, dass die Daten zielgerichtet und geeignet für eine Analyse und Vorhersage sind.

3.4 Data Visualization

Um einen ersten Eindruck über die Auswirkung verschiedener Parameter auf die Trefferquote von Kobe Bryant zu bekommen, sind Visualisierungen dienlich. Nachfolgend werden verschiedene Abbildungen bezüglich Kobes Würfen dargestellt und erläutert. In der ersten *Abbildung 3* wird Kobes generelle Trefferanzahl und -quote über seine Karriere hinweg dargestellt.

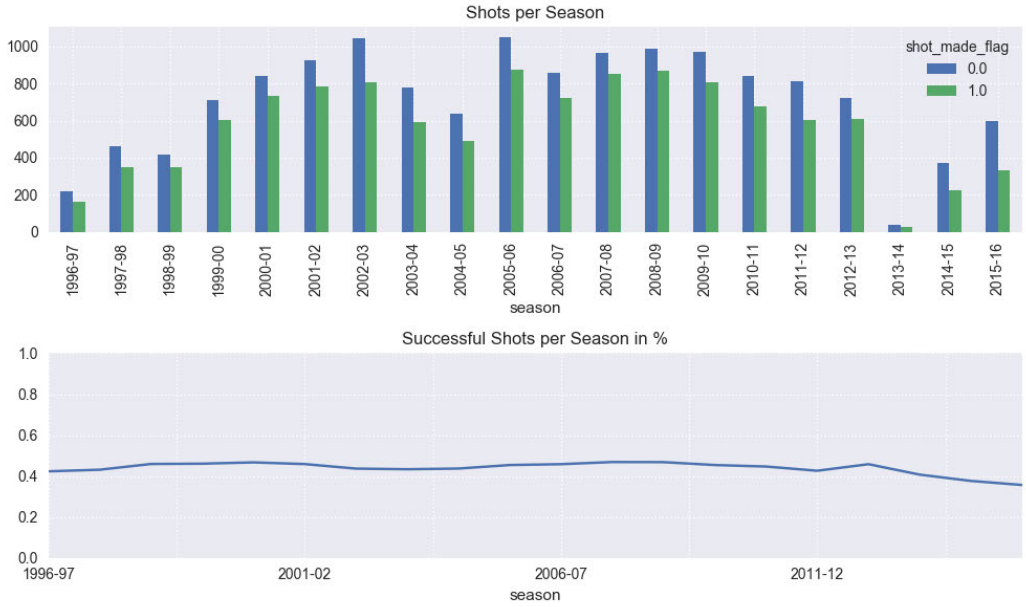

Abbildung 2: Shots per Season and Ratio

In der oberen Hälfte von *Abbildung 3* ist die gesamte Anzahl von seinen getroffenen (grün) und nicht getroffenen Würfen (blau) zu sehen. Der Verlauf der absoluten Zahlen ähnelt wie bei vielen Basketballern einer Kurve und repräsentiert mehr die gespielte Zeit als die versuchten Würfe. Zu Beginn seiner Karriere hatte Kobe eine geringere Spielzeit pro Spiel. Diese stieg jedoch an und flachte zum Ende seiner Karriere ab. Zum Ende der Saison 2004 hatte Kobe eine Verletzung, woraufhin er sowohl in der Saison 03/04 und 04/05 weniger spielen konnte. Aus gleichem Grund hat er auch in der Saison 13/14 wenig gespielt. Interessanter stellt sich der Verlauf seiner Trefferquote im Laufe seiner Karriere dar. Diese startet mit ~41% und verläuft relativ konstant in einem Intervall zwischen ~42 und ~47%. Lediglich zum Ende seiner Karriere fällt er unter die 40%-ige Trefferquote. Stellt man sich also bereits an dieser Stelle die Frage, ob Kobe einen Wurf trifft, könnte man anhand der Saison Voraussagen, dass er zu circa 45% trifft (bzw. zu 55% nicht). Dieser Wert wird auch als Major Classifier genutzt. Es gilt also die Prognose anhand von weiteren Faktoren präziser als 55% vorzunehmen.

Betrachtet man wie in *Abbildung 4* dargestellt die Erfolgsquote einzelner *Action_Types* oder deren Aggregation in verschiedene *Shot_Types*, lässt sich eine präzisere Voraussage treffen. Die Wahrscheinlichkeit, dass Kobe einen Dunk trifft, liegt bei über 90% (Maximum), jedoch bei einem Tip Shot bei lediglich ~38% (Minimum).

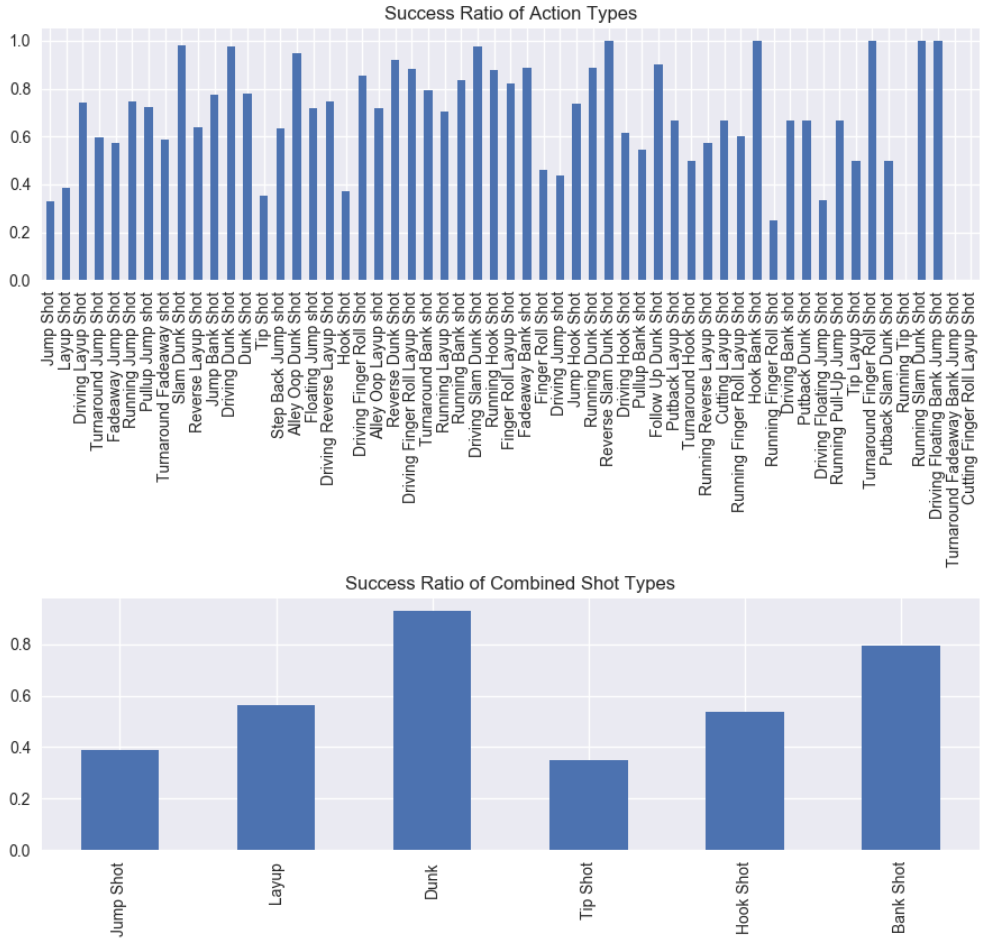

Abbildung 3: Erfolgsquote Shot Types

Absolut gesehen (*Abbildung 5*) haben jedoch Jump Shots und Layup shots die höchste Bedeutung für die Analyse und Vorhersage. Um eine präzise Vorhersage der meisten Würfe (Jump Shot) vorzunehmen, sollte also nach weiteren Kriterien unterteilt werden.

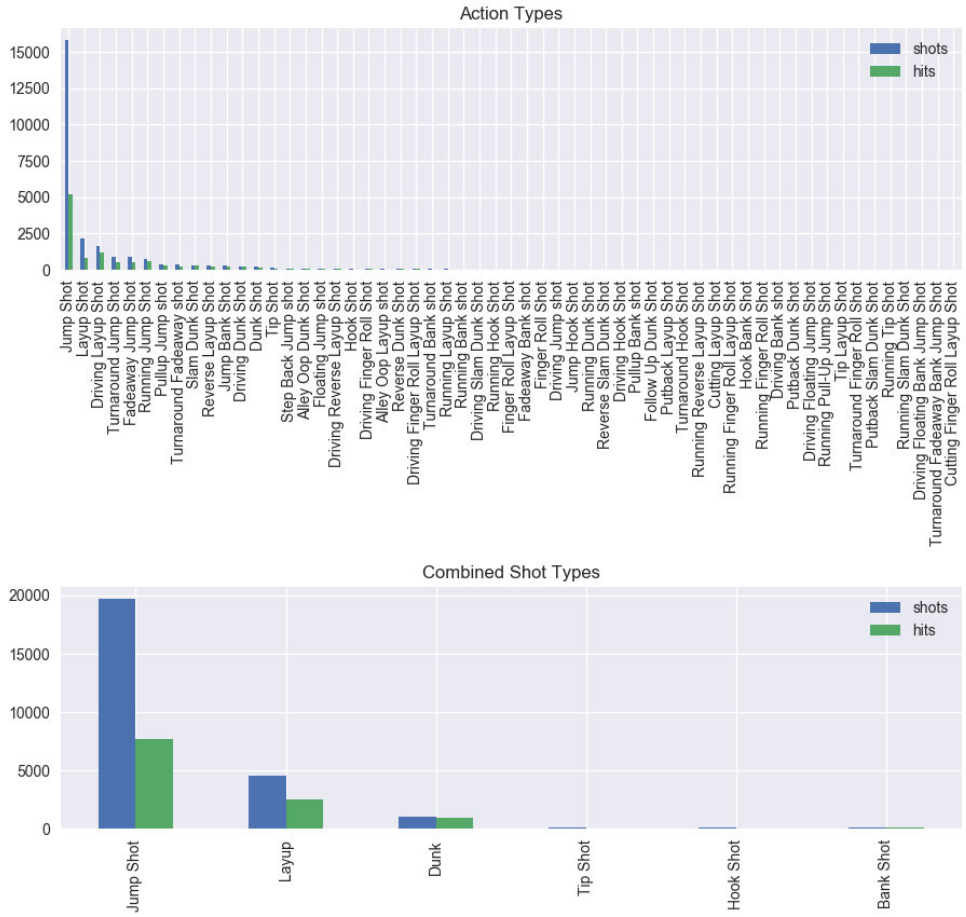

Abbildung 4: Absolute Anzahl der Würfe

So kann beispielsweise anhand der Distanz von Kobe zum Korb viel Informationen über seine Trefferwahrscheinlichkeit gewonnen werden.

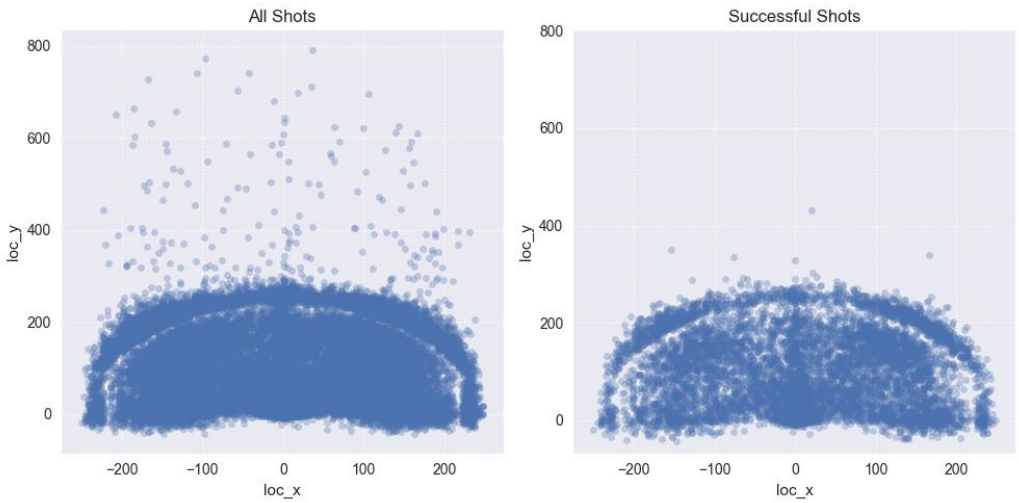

Abbildung 5: Distanz zum Korb

Auf der linken Seite der *Abbildung 6* sind alle Positionen seiner Würfe zum Korb zu sehen. Auf der rechten Seite sind seine erfolgreichen Würfe zu sehen. Hier wird schnell klar, dass unabhängig anderer Faktoren, die Distanz zum Korb eine maßgebliche Rolle für seinen Erfolg

spielt. Interessant zu sehen ist auch, dass in *Abbildung 6* die drei Punkte Linie erkennbar ist. Basketballspieler versuchen selbstverständlich möglichst viele Punkte mit einem Wurf zu erzielen. Stehen sie also knapp in dem zwei Punkte Feld, werden sie noch einen Schritt zurückgehen, um die Chance zu haben, drei Punkte zu erzielen. Diese These bestätigt sich auch in *Abbildung 7*.

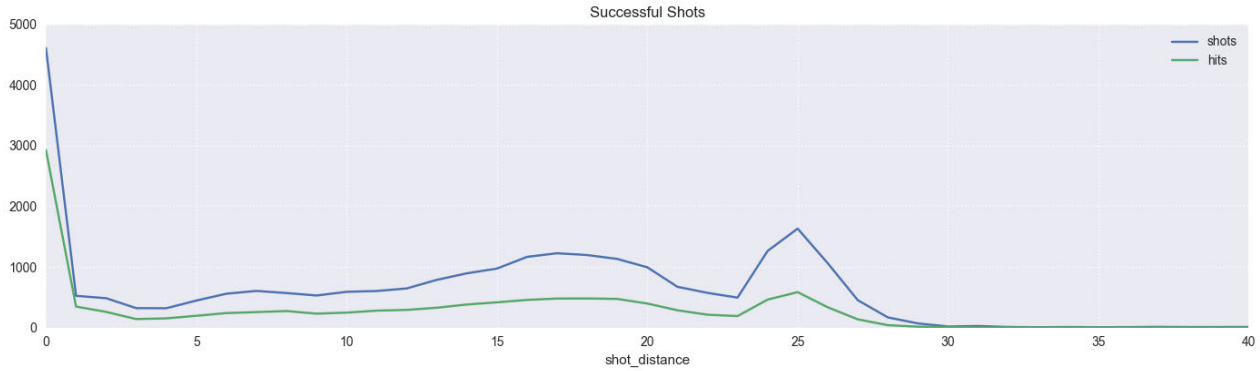

Abbildung 6: Erfolgsquote Distanz

Hier ist Kobe Bryants Trefferquote in Abhängigkeit zu seiner Distanz zum Korb zu sehen. Auf der Y-Achse ist dabei die Anzahl seiner Würfe im Verhältnis zur Distanz (X-Achse) abgebildet. In Blau ist dabei die gesamte Anzahl an Würfen und in Grün die erfolgreichen Würfe dargestellt. Auffällig dabei ist, dass er bei einer Distanz von circa einem Fuß nahezu alle Würfe trifft (Graphen tangieren sich fast) und die Anzahl der Versuche am Ende der zwei Punkte Zone wieder zurückgehen. Die Würfe direkt am Anfang der drei Punkte Zone haben jedoch die höchste Diskrepanz zwischen Würfen und Erfolg. Anhand dieser Grafik könnte man ggf. das Fazit ziehen, dass Kobe auf den zusätzlichen Schritt zurück verzichten sollte, da er bei einer höheren Trefferquote von 2-Punkten Würfen absolut gesehen mehr Punkte erzielt. Abschließend soll die Korrelation aller Attribute zu einem erfolgreichen Wurf betrachtet werden.

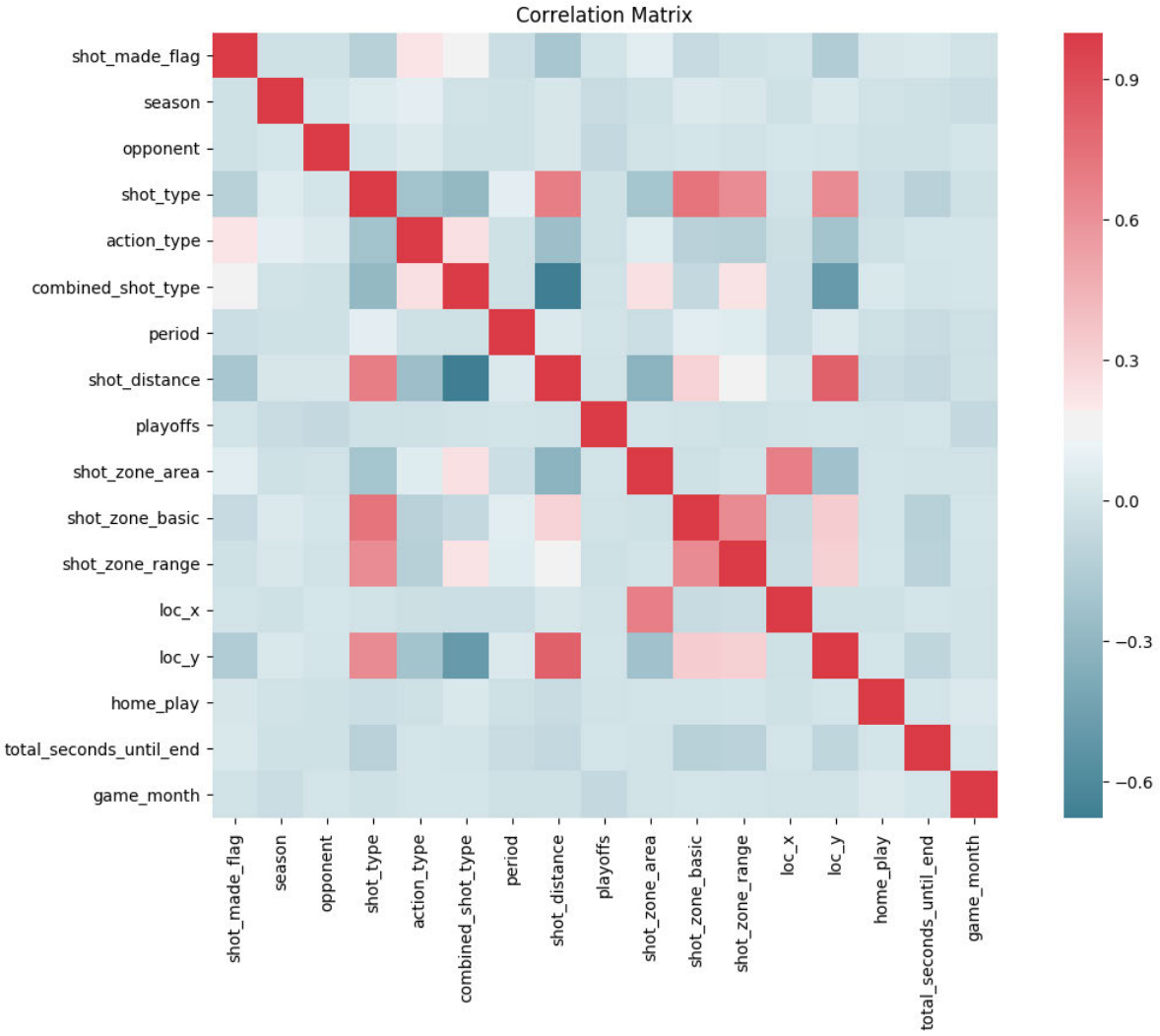

Abbildung 7: Korrelationsmatrix

Die Korrelationsmatrix in *Abbildung 8* stellt die Korrelation zwischen den einzelnen Attributen dar. Die Korrelation reicht dabei von -1 (dunkelblau) bis +1 (rot). Die Diagonale zeigt die Korrelation eines Wertes mit sich selbst und ist daher immer gleich eins. Wie bereits erläutert bestätigt sich hier erneut der Zusammenhang der Distanz zu bestimmten Kategorien. Um die einzelnen Korrelationen mit dem *shot_made_flag* herauszufinden, liegt das größte Interesse der *Abbildung 8* in der ersten Zeile. Deutlich hierbei wird, dass die aktuelle Saison, der Gegner, die Periode im Spiel, seine horizontale Position (*loc_x*), ob es sich um ein Heimspiel handelt, in welchem Monat das Spiel stattfindet, ob es ein Playoff Spiel ist und wie viel Zeit in der Periode noch ist, kaum oder gar nicht mit der Trefferquote korrelieren. Die Art des Wurfs (*shot_type, action_type, Combined_shot_type*) und die Distanz zum Korb (*loc_y, shot_distance*) scheinen dafür starker mit dem Erfolg seines Wurfs zu korrelieren.

4. CRIPS-DM: DATA PREPARATION

Data Preperation umfasst die Selektion, Bereinigung, Konstruktion und Integration von Daten und dient als Fundament für einen validen Data Mining Prozess. Es spielt eine maßgebliche Rolle, welche Daten ausgewählt und wie diese aufbereitet werden. Jeder dieser Schritte sollte stets vor dem Hintergrund der in *2.2 Project Goal* dargestellten Zielstellung sein und jederzeit reevaluiert werden können.

4.1 Data Selection

Zur Auswahl der Daten stehen zahlreiche Tupel und Attribute zu Verfügung. Offen ist die Frage, welche Attribute in den Data Mining Prozess eingehen sollen und ob diese gefiltert werden (z.B. nur die Jahre in der Mitte von Kobes Karriere). Dabei sollte bei der Selektion von Attributen stets versucht werden ein Overfitting zu vermeiden. Nachfolgende *Tabelle 2* stellt den zugrundeliegenden Evaluationsprozess dar. Ist eine Alternative für ein Attribut gegeben, sollte nur das Attribut selbst oder die Alternative gewählt werden.

Attribut	Auswählen	Begründung	Risiko Overfitting?	Alternative
shot_id	nein	irrelevant - interne Kennzahl	-	-
action_type	möglich	gibt präzise an, um was für einen Wurf es sich handelt	ja	combined_shot_type
combined_shot_type	ja	Überkategorie. Kann statt action_type genutzt werden um ein Overfitting zu vermeiden	nein	action_type
shot_distance	möglich	hohe Korrelation	ja	loc_y, shot_zone_range
shot_zone_range	ja	Kategoriesierung von shot_distance	nein	shot_distance, loc_y
shot_type	nein	zwei und 3 Punkte wurf zu generisch und implizit betrachtet	nein	shot_distance, shot_zone_range, loc_y
shot_zone_area	ja	gute Kategorisierung	nein	Loc_x
shot_zone_basic	ja	gute Kategorisierung	nein	Loc_y
period	möglich	keine Korrelation, vllt jedoch hilfreich	nein	-
minutes_remaining	möglich	ggf. bei wenig verbleibender Zeit höheres Druck und mehr Erschöpfung	ja	-
seconds_remaining	nein	es sollte egal sein wann ein Wurf innerhalb einer Minute geworfen wird	ja	-
lat	nein	Kein Mehrwert gegenüber loc_x	ja	Loc_x
lon	nein	Kein Mehrwert gegenüber loc_y	ja	Loc_y
loc_x	möglich	Korrelation vorhanden	ja	Shot_zone_basic
loc_y	möglich	Korrelation vorhanden	ja	shot_distance, shot_zone_range, shot_type, shot_zone_area
playoffs	möglich	keine starke Korrelation	nein	-
shot_made_flag	-	Zielvariable	-	-
game_event_id	nein	irrelevant - interne Kennzahl	-	-
game_id	nein	irrelevant - interne Kennzahl	-	-
season	möglich	Ggf. unterschiedliche Performance unt Entwicklung	nein	game_date
team_id	nein	irrelevant - immer identisch	-	-
team_name	nein	irrelevant - immer identisch	-	-
game_date	möglich	Das Jahr könnte von Relevanz sein. Das ganze Datum bieten ein Risiko zum Overfitting	ja	season
matchup	möglich	muss untersucht werden	möglich	-
opponent	möglich	muss untersucht werden	möglich	-

Tabelle 2: Auswahl der Attribute

Wie in *Tabelle 2* zu sehen ist, gibt es zahlreiche Möglichkeiten Attribute zu selektieren. Alle mit „ja“ gekennzeichneten Attribute werden in der in *5.CRISP-DM: Modeling* dargestellten Modellierung vorerst genutzt und iterativ mit Alternativen ausgetauscht und/oder weitere mit „möglich“ gekennzeichnete Felder ergänzt. Welche Kombination von Attributen final gewählt wird, kann an dieser Stelle noch nicht antizipiert werden. Es wird jedoch eine Aufteilung zwischen Meta- und Detaildatensatz vorgenommen. Der Detaildatensatz umfasst die präzisesten Daten (z.B. loc_x) wohingegen der Metadatensatz die kategorischen Attribute beinhaltet. Es besteht derzeit noch kein Verdacht auf einen analytischen Mehrwert durch das Einschränken mancher Attribute auf bestimmte Dimensionen. Es wäre beispielsweise denkbar den Anfang oder das Ende von Kobes Karriere auszuschließen, da es möglich ist, dass er zu seiner besten Zeit auch am konstantesten warf und somit die Vorhersagen am präzisesten sind. Mögliche Einschränkungen können jedoch iterativ erarbeitet werden.

4.2 Data Cleaning

Für die Datenbereinigung werden fehlende und falsch klassifizierte Daten betrachtet. Es fehlen insgesamt 5.000 Einträge des *shot_made_flags.* Jedoch konnte keine Korrelation mit dem Fehlen von Daten und anderen Attributen identifiziert werden. Es wird davon ausgegangen, dass die Daten Missing Completely at Random (MCAR) sind. Möglich ist es also die Daten auszuschließen oder mit einem treffenden Durchschnittswert zu befüllen. Aufgrund der Größe des Datensatzes und der unzureichenden Signifikanz der fehlenden Daten, wurden die Tupel mit einem fehlenden *shot_made_flag* ausgeschlossen.

Die Überprüfung der Datenklassifikation erfolgte in zwei Schritten. Im ersten Schritt wurde die Korrektheit der Attributdomänen überprüft. So dürfen beispielsweise die *shot_types* nur die Werte *2PT Field Goal* oder *3PT Field Goal* annehmen, *minutes_remaining* immer kleiner als 11 sein oder die *shot_distance* kleiner gleich 94ft. sein. Sollte einem Attribut ein Wert außerhalb seiner Domäne zugewiesen sein, wird der Inhalt vorerst gelöscht. Im zweiten Schritt wird der Wert von Attributen validiert. So ist es beispielsweise aufgrund der Distanz nicht möglich, dass ein Dunk Shot als *3PT Field Goal* klassifiziert wird. In diesem Schritt wurde insbesondere die Klassifizierung von Distanzfeldern und Wurftypen überprüft. Anhand von implementierten Tests konnte überprüft werden, dass die *loc_x* und *loc_y* vermutlich korrekt ermittelt wurden. Somit konnten auf dieser Grundlage alle distanzbezogenen Attribute (*shot_distance, lat, lon, shot_zone_range, shot_zone_basic, shot_zone_area, shot_type)* überprüft werden. Sollte dabei einem Attribut ein falscher Wert zugewiesen sein, wurde das durch einen errechneten Wert korrigiert. Gab es nach dem zweiten Schritt nach wie vor noch relevante Attribute ohne Eintrag,

wurde dieses Tupel gelöscht. Es konnte so insgesamt 516 Tupel korrigiert werden und es musste kein zusätzliches Tupel gelöscht werden.

4.3 Data Transformation and Integration

Bevor das eigentliche Modell zur Vorhersage von Kobes Würfen erstellt und implementiert werden kann, gilt es noch den letzten Schritt der Datentransformation und -integration zu vollziehen. Da die Daten nur in einer Datei vorliegen, muss kein gesonderter Schritt für die Integration vorgenommen werden. Die Daten müssen lediglich eingelesen und transformiert werden. Im ersten Schritt wurden dafür die Datentypen der einzelnen Attribute in: *String, Integer, Date* und *Boolean* gemäß *Tabelle 1* umgewandelt. Alle kategorischen Werte wurden vorerst als *String* eingelesen. Nachfolgend wurden für eine Steigerung der Effizienz die kategorischen Werte binär konvertiert. So wurde beispielsweise das Metaset *combined_shot_type* in dummy Variablen aufgeteilt. Anstelle von *combined_shot_type* existieren dann im Datenset die booleschen Spalten: *Jump Shot, Dunk, Layup, TipShot, Hook Shot und Bank Shot*. Abhängig von der originären Ausprägung eines Attributs existiert dann exakt ein *True* je binärem Split. Neben dieser Konversion wurden noch weitere Attribute abgeleitet. So wurden zusätzlich die Felder *absolute_time_remaining, angle* und *home_game_flag* erstellt. *absolute_time_remaining* gibt die totale restliche Spielzeit an und ist somit die Kombination von *minutes_remaining, seconds_remaining* und *period.* Der Winkel (*angle*) kann über die *loc_x* und *loc_y* berechnet werden. Hier besteht der Verdacht, dass der Winkel des Wurfs einen Einfluss auf den Erfolg hat. Ob es sich um ein Heimspiel handelt, kann aus *matchup* abgeleitet werden und könnte auch einen Einfluss auf die Trefferquote haben. Mit den selektierten, bereinigten und transformierten Daten kann nun die Modellierung beginnen.

5. CRISP-DM: MODELING

5.1 Model Selection

Die Modellauswahl richtet sich nach der Vorhersage, die getroffen werden muss. In diesem Fall liegt ein Klassifikationsproblem vor, das als *Supervised Learning* deklariert werden kann. Das Resultat einer Vorhersage muss die nominalen Ausprägungen „Treffer“ oder „kein Treffer“ annehmen.

Als erstes Modell zur Klassifikation der Datensätze soll eine logistische Regression dienen. Diese berechnet für einen Datensatz mit welcher Wahrscheinlichkeit dieser zu einer Klasse gehört. Die Klassifikation entspricht der Klasse, für welche die höchste Wahrscheinlichkeit errechnet wurde. Die logistische Regression ist insbesondere durch die Recheneffizienz weit verbreitet. Zu beachten ist, dass das Verfahren jedoch relativ anfällig für Ausreißer ist, welche die Genauigkeit des Modells verschlechtern.

Als zweites Modell wird ein Entscheidungsbaum herangezogen. Dieser klassifiziert Daten iterativ anhand mehrerer Attribute und deren Ausprägungen. Für die vorliegende Problemstellung eignet sich dieses Modell insbesondere, da der Datensatz eine Vielzahl an Attributen vorweist, nach dem sog. Splits vorgenommen werden können. Zudem ist ein Entscheidungsbaum vergleichsweiße robust gegen Ausreißer, wobei aber die Gefahr des Overfittings bei zunehmendem Training höher ist, als beispielsweise bei der logistischen Regression. Als Ergänzung zum Entscheidungsbaum wird die Verwendung des Random Forest Modells aufgezeigt. Das Modell nutzt mehrere Entscheidungsbäume, die individuell über eine Klassifikation entscheiden. Dabei werden die Entscheidungen aller Bäume ausgezählt – die Klasse mit den häufigsten Nennungen wird anschließend gewählt.

Über die hier genannten Modelle hinaus gibt es viele weitere, die sich zur Klassifikation eignen. Zu nennen sind etwa Support Vector Mashines, k-nearest-neighbour oder neuronale Netze. Aufgrund des Umfangs dieser Arbeit, sollen jedoch nur die zuvor erwähnten betrachtet werden.

Die Modelle werden mehreren Tests unterzogen – da die Modellierung auf einer bestimmten Zufälligkeit beruht, sind die Ergebnisse jedoch nur bedingt reproduzierbar. Insbesondere beim Entscheidungsbaum soll aufgezeigt werden, wie sich die Parametrisierung positiv auf die Genauigkeit auswirkt. Die Experimente können dabei beliebig umfangreich gestaltet werden und sollen hier daher vielmehr das grundsätzliche Vorgehen aufzeigen.

5.1 Logistic Regression

Aufbauend auf dem Schritt Data Preparation, wurde die logistische Regression mit verschiedenen Datensätzen durchgeführt und die Ergebnisse dokumentiert. Um ein Overfitting zu vermeiden bzw. zu identifizieren, wurden zwei Ansätze verfolgt. Zunächst wird das Modell wird mit 70% der Daten trainiert und anschließend validiert. Dazu wird die Cross-Validation verwendet, die den Trainingsdatensatz in fünf Teile trennt und ein mögliches Overfitting erkennen lässt. Anschließend wird das Modell mit einem Testdatensatz (die verbleibenden 30%) getestet, die bei der Erstellung nicht herangezogen wurden, um die Performance des Modells zu evaluieren. Im Folgenden können die Ergebnisse für verschiedene Konstellationen entnommen werden.

	Validierung	Test
Meta-Datensatz	0.61 (+/- 0.01)	0.6212
Detail-Datensatz	0.67 (+/- 0.03)	0.6808

Tabelle 3: Ergebnisse der logistischen Regression

Wie der geringen Abweichung in der Validierung entnehmen kann, findet kein signifikantes Overfitting statt. Vergleicht man die beiden Trainingssätze miteinander ist der detailliertere Datensatz ca. sechs Prozent performanter.

5.2 Decision Tree und Random Forest

Der Entscheidungsbaum wird auf derselben Datenbasis wie die logistische Regression trainiert. Auch wird der Datensatz zum Trainieren und Testen in einem Verhältnis von 70:30 geteilt. Zusätzlich können bei einem Entscheidungsbaum mehrere Parameter justiert werden, die ein Overfitting vorbeugen. Dazu wurden im Projekt folgende Einstellungen betrachtet:

- method: Der Parameter legt fest, ob der Entscheidungsbaum nach dem Gini-Koeffizienten oder der Entropie aufgebaut wird.
- max_depth: Es kann die maximale Tiefe des Baumes bestimmt werden.
- min_samples_leafs: Eine bestimme Anzahl von Dateneinträgen muss vorliegen, sodass für diese ein Blatt erzeugt wird.
- max_features: Der Entscheidungsbaum verwendet nur eine festgelegte Zahl an Attributen bzw. Spalten aus dem Datensatz.

Im Nachfolgenden werden die Ergebnisse der Tests aufgeführt. Zur Erstellung der Bäume wurde die Entropie verwendet, da bei Tests keine Performance-Unterschiede zum Gini-Koeffizienten festgestellt werden konnte.

Meta-Set				Detail-Set			
Entropie		Gini-Koeffizient		Entropie		Gini-Koeffizient	
Val.	Test	Val.	Test	Val.	Test	Val.	Test
0.60 (+/- 0.01)	0.6043	0.60 (+/- 0.02)	0.6052	0.57 (+/- 0.05)	0.5821	0.57 (+/- 0.04)	0.5747

Tabelle 4: Ergebnisvergleich nach Methoden im Entscheidungsbaum

Ausgehend von diesen Werten, wurden die Parameter sukzessive angepasst. Zunächst wurde die maximale Tiefe festgelegt. Anschließend wird die minimale Anzahl von Instanzen zur Blattgenerierung bestimmt und abschließend die maximale Anzahl von Features festgelegt. Dabei wurde der beste Parameterwert vom vorherigen Test herangezogen. Das Vorgehen und die Ergebnisse werden in *Anhang 1*weiter ausgeführt.

	max_depth	min_samples_leafs	max_features	Validierung	Test
Meta-Set	default	10	6	0.61 (+/- 0.02)	0.6233
Detail-Set	default	5	10	0.67 (+/- 0.04)	0.6813

Tabelle 5: Ergebnisse des parametrisierten Entscheidungsbaums

Insbesondere beim Entscheidungsbaum, welcher das detaillierte Datenset nutzt, erkennt man eine deutliche Verbesserung im Vergleich zum unparametrisierten Test aus *Tabelle 5* (+ 0.092). Die ideale Tiefe entsprach in beiden Fällen die der Features. Weiterhin hat es sich ausgezahlt eine Mindestanzahl an Instanzen zur Blattgenerierung zu nutzen. Auch waren die Modelle am genausten, welche alle vorhandenen Features genutzt haben.

Aufbauend auf den Ergebnissen des Entscheidungsbaumes wurde weiterhin das Random-Forest-Modell herangezogen, um einen Vergleich zu schaffen, wie mehrere Bäume in Summe die Genauigkeit des Modells verbessern. Dazu wurde die Baumtiefe übernommen und jeweils zwei Tests durchgeführt, die jeweils 100 Bäume nutzen. Das Meta-Datensez schaffte eine Genauigkeit von 0.6804 und das Detail-Datenset 0.6629. Eine signifikante Verbesserung zum Entscheidungsbaum ist demnach nicht erkennbar.

6. CRIPS-DM: EVALUATION

6.1 Confusion-Matrix

Zur dedizierten Evaluation der Ergebnisse können verschiedene Konzepte herangezogen werden, wobei hier eine Confusion-Matrix verwendet wird. Diese zeigt auf, welche Treffer richtig vorhergesagt wurden, wobei auch die dargestellt werden, die nicht korrekt waren. Nachfolgende wird exemplarisch das beste Modell der vorherigen Phase evaluiert – dies war der Entscheidungsbaum mit dem Detail-Set und einer Genauigkeit von 0.6813. Die Confusion-Matrix zeigt für tatsächliche Treffer ein relativ ausgewogenes Bild. Leicht auffällig ist, dass verfehlte Würfe tendenziell besser als getroffene vorausgesagt werden.

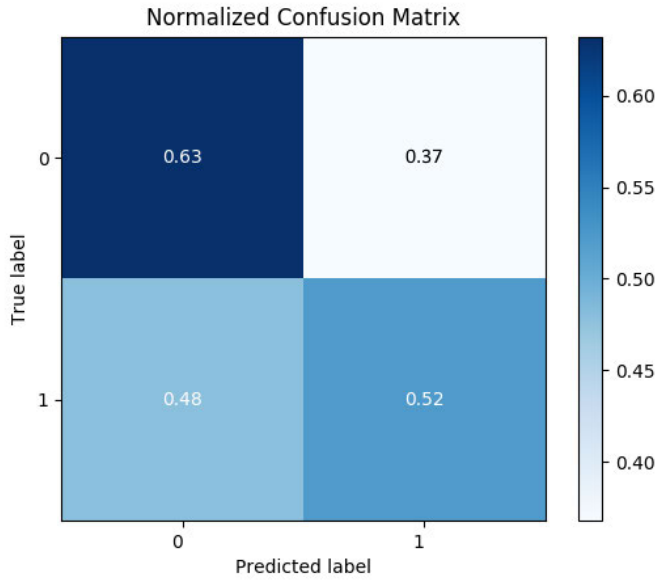

Abbildung 8: Confusion-Matrix des Entscheidungsbaums

6.2 Comparison and Assessment

Zwischen den verwendeten Modellen gibt es hinsichtlich der Genauigkeit kaum signifikante Unterschiede. Auffällig jedoch war die deutliche Verbesserung des Entscheidungsbaumes durch die Parametrisierung. Auch die verschiedenen Datensets haben sich deutlich auf die Performance der Modelle widergespiegelt. Es liegt die Vermutung nahe, dass durch die genauere Betrachtung dieser Faktoren ein noch besseres Ergebnis erzielt werden könnte. Mit dem aktuellen Stand konnte die Genauigkeit gegenüber der Baseline-Performance um 12% verbessert werden.

7. Abschluss und Fazit

7.1 Reflexion des Vorgehens

Einen Großteil des Projektes hat die Aufbereitung der Daten (4. *Data Preparation*) eingenommen. Abgesehen von fehlenden Werten, ist die Datenbasis weitestgehend fehlerfrei. Die Datenbasis beschränkt sich jedoch auf grundsätzlich beobachtbare Attribute, die teils redundant sind. Beispiele für weitere relevante Attribute zur Verbesserung der Vorhersagegenauigkeit ist die Verfassung des Spielers, gespielte Minuten, Fouls oder die verbleibende. Auch könnte ein umfangreicheres Feature Engineering zur qualitativen Aufwertung der Datenbasis beitragen.

Bei der Modellierung hat sich gezeigt, dass insbesondere die Parametrisierung der Modelle eine deutliche Verbesserung erzielt hat. So kann sich ein noch systematischeres Vorgehen und umfangreicheres Testen zusätzlich positiv auswirken. Gemäß den gewonnenen Erkenntnissen während des Projekts, wird abhängig von der aktuellen Datenbasis und der verwendeten Modelle eine weitere signifikante Verbesserung schwer umsetzbar.

7.2 Ausblick

Anhand der Limitationen, welche sich aus der Datenbasis und den verwendeten Modellen ergeben, bestehen Potenziale zur Weiterentwicklung des Projekts. So kann man die Datenbasis durch Feature Engineering weiter präzisieren oder durch externe Daten ergänzen. Auch könnten weitere Daten zum Spielverhalten der Gegner, wie etwa Fouls, oder den Trainingseinheiten hilfreich sein. Neben der Datenbasis können auch andere Vorhersagemodelle Anwendung finden. Die hier aufgeführten Modelle stellen nur eine Auswahl dar. Beispielsweise können komplexere Modelle wie neuronale Netze angewandt werden, sobald umfangreichere Daten vorhanden sind.

Anhang I

max_depth	Meta-Set	
	Validierung	Test
50	0.60 (+/- 0.01)	0.6012
40	0.60 (+/- 0.01)	0.6082
30	0.60 (+/- 0.01)	0.6113
20	0.61 (+/- 0.02)	0.6165
default	0.61 (+/- 0.02)	0.6210

min_ sample_ leaf	Meta-Set	
	Validierung	Test
1	0.61 (+/- 0.02)	0.6183
5	0.61 (+/- 0.02)	0.6210
10	0.61 (+/- 0.01)	0.6249
15	0.61 (+/- 0.02)	0.6111
20	0.61 (+/- 0.02)	0.6242
30	0.61 (+/- 0.02)	0.6222

max_ features	Meta-Set	
	Validierung	Test
10	-	-
9	-	-
8	-	-
7	-	-
6	0.61 (+/- 0.02)	0.6233
5	0.61 (+/- 0.01)	0.6210
4	0.61 (+/- 0.01)	0.6131
3	0.61 (+/- 0.01)	0.6168
2	0.61 (+/- 0.02)	0.6132
1	0.61 (+/- 0.02)	0.6119

max_depth	Detail-Set	
	Validierung	Test
50	0.57 (+/- 0.04)	0.5715
40	0.58 (+/- 0.04)	0.5899
30	0.60 (+/- 0.04)	0.6110
20	0.63 (+/- 0.04)	0.6263
default	0.65 (+/- 0.07)	0.6714

min_ sample_ leaf	Detail-Set	
	Validierung	Test
1	0.65 (+/- 0.07)	0.6705
5	0.67 (+/- 0.04)	0.6810
10	0.66 (+/- 0.04)	0.6723
15	0.64 (+/- 0.07)	0.6554
20	0.66 (+/- 0.04)	0.6761
30	0.66 (+/- 0.05)	0.6684

max_ features	Detail-Set	
	Validierung	Test
10	0.67 (+/- 0.04)	0.6813
9	0.65 (+/- 0.05)	0.6452
8	0.64 (+/- 0.07)	0.6112
7	0.62 (+/- 0.08)	0.6482
6	0.64 (+/- 0.04)	0.6486
5	0.63 (+/- 0.08)	0.6318
4	0.64 (+/- 0.03)	0.6546
3	0.63 (+/- 0.05)	0.6395
2	0.60 (+/- 0.03)	0.6138
1	0.58 (+/- 0.04)	0.5935