

Analyse und Interpretation von Big Data im Fußball

Kooperationsprojekt mit SAP Sports One

Abschlussbericht der Kooperationsphase 2016/2017

Timo Kratz

Betreuer: Herr McCormick-Smith (SAP SE)

Inhaltsverzeichnis

1. Big Data im Sport	3
2. Datenerfassung	4
2.1 Event.xml	6
2.2 Tracking.xml	6
3. Datenauswertung	7
3.1 Parser	8
3.2 Programmablauf: Event_Heatmap.py	9
3.3 Programmablauf: Tracking_Heatmap.py	12
3.4 Herausforderungen	15
4. Interpretation der Daten	16
5. Danksagung	21
6. Literaturverzeichnis	22

Abbildungsverzeichnis

Abb. 1: Event-Heatmap mit 21 auf 12 Feldern und Koordinaten	10
Abb. 2: Ablaufdiagramm Event_Heatmap.py	12
Abb. 3: Tracking-Heatmap mit 21 auf 12 Feldern und Koordinaten	13
Abb. 4: Ablaufdiagramm Tracking_Heatmap.py	14
Abb. 5: Tracking Heatmap Niklas Süle (Verteidiger)	17
Abb. 6: Event Heatmap Eduardo Vargas (Stürmer)	18
Abb. 7: Tracking Heatmap Eduardo Vargas (Stürmer)	18
Abb. 8: Event Heatmap Pierre-Emerick Aubameyang (Stürmer)	19

1. Big Data im Sport

Fußball ist ein sehr einfacher Sport. Wie Sepp Herberger schon sagte: „Das Runde muss ins Eckige.“ Wer nach neunzig Minuten mehr Tore erzielt hat, gewinnt. Im Profisport geht es neben dem Sportlichen aber auch um wirtschaftliche Interessen. Beim populärsten Sport in Deutschland sind zweistellige Millionenbeträge durch Sponsoren oder bei Spielertransfers keine Seltenheit mehr. Deshalb hat in den letzten Jahren eine neue Entwicklung im Profifußball Einzug gehalten. Die großen Vereine vertrauen immer mehr auf Hightech und Innovation. Diese Entwicklung stammt aus dem amerikanischen Sport, wo in den großen Profiligen des Basketballs, Footballs oder Baseballs alles mit Werten, Noten und Statistiken dokumentiert wird.

Um solche Bewertungen erstellen zu können, werden heutzutage während eines Fußballspiels Millionen von Daten gesammelt. Diese Daten werden von außerhalb des Feldes erhoben, also mit Kameras und Software. Diese Daten haben alleine für sich jedoch keine besonders hohe Aussagekraft. Deshalb sind bei den Profivereinen in der deutschen Bundesliga bis zu zwölf Analysten angestellt, die aus den Big Data verlässliche Ergebnisse herausfiltern sollen. Dazu sind die Ursprungsdaten ungeeignet, es werden Algorithmen zur Berechnung von aussagekräftigen Kennzahlen entwickelt, die den Analysten, Trainern und Scouts verlässliche Spielerbewertungen und Trainingsunterstützung liefern sollen. Im Rahmen meines BOGY-Praktikums hatte ich die Möglichkeit, beim 1. FC Köln hinter die Kulissen eines Profifußballvereins zu schauen und Einblicke in die Vorgänge des Scoutings und der Spielanalyse zu erlangen.

Die Analyse von Daten ist ein schwieriger Prozess, deswegen benutzen alle Profivereine Softwaretools zur Unterstützung. Ein Softwareanbieter ist SAP SE mit ihrer sportspezifischen Cloud-Lösung SAP Sports One.¹ SAP Sports One bietet eine einheitliche Plattform, um individuelle Teams und Spieler effizient zu verwalten, sowie analytische Einblicke für die Leistungssteigerung zu erhalten. Die Lösung umfasst die

¹ Vgl. SAP SE. Sport und Entertainment

Module Team-Management, Trainingsplanung, Fitness, Performance-Analysen sowie Scouting und Talentfindung.

Im Rahmen der diesjährigen Hector-Kooperationsphase bekam ich Gelegenheit, mit SAP Sports One zu kooperieren und meine Leidenschaft zum Fußball mit der Informatik verbinden. Zum einen konnte ich das Produkt SAP Sports One kennenlernen, zum anderen konnte ich selber programmieren und einen Algorithmus für die Auswertung von Spieldaten entwickeln. Ich habe zwei Heatmaps entwickelt, in denen Aktionen und Positionen eines Spielers auf dem Feld in einer Übersicht dargestellt werden. Das Spielfeld wird in Quadrate aufgeteilt und je nach Häufigkeit eines Ereignisses werden die Felder verschieden eingefärbt. Da ich bereits Erfahrung mit der Programmiersprache Python sammeln konnte, habe ich meine Programme in Python implementiert.

2. Datenerfassung

Während der neunzig Minuten eines Bundesligaspiels werden bis zu sechzig Millionen Daten gesammelt. Es werden spieterspezifische Daten, wie Koordinaten, Laufstrecke, Zweikampfbilanz oder Ballkontakte gezählt, außerdem werden Mannschaftsdaten wie zum Beispiel die Aufstellung oder die Passwege dokumentiert. Diese Daten werden alle in Echtzeit erfasst. Dadurch können die Daten dem Fernsehzuschauer in der Halbzeitpause präsentiert werden, und das Trainerteam kann den Spielern nach ersten Analysen in der Pause neue Informationen und Taktiken vermitteln.

Opta ist der weltweit führende Anbieter von Live-Sportdaten und gehört zu der digitalen Sportmedien-Gruppe PERFORM. SAP Sports One verwendet die Daten von Opta. Diese werden in XML-Dateien zur Verfügung gestellt. XML (Extensible Markup Language) ist eine Auszeichnungssprache zur Darstellung von Daten in einer Textdatei, wobei die Daten hierarchisch in sogenannten Tags geordnet werden. Ein Tag kann mehrfach verwendet werden.

Für meine Arbeit konnte ich auf die Daten des Fußballspiels Hoffenheim gegen Dortmund aus dem Jahre 2015 zugreifen. Dies ist zum einen eine Event-Datei, die alle Events des Spiels auflistet und eine Tracking-Datei, die die Position aller Spieler beinhaltet.

Die **Event-Daten** werden von Opta live teils per Software und teils händisch erfasst und in Sekundenschnelle an die Kunden herausgegeben. Alle Daten werden nach einem einheitlichen System erhoben und an die 300 Kunden in 40 verschiedenen Länder verkauft. So soll für alle verschiedenen Anforderungen das passende Paket an Daten geliefert werden, egal ob es um einen Profiverein oder den übertragenden TV-Sender geht. Seit dem Jahre 2013 ist Opta der offizielle Partner der Deutschen Fußball Liga und erhebt die Daten für alle Spiele der ersten und zweiten Bundesliga. In Deutschland arbeiten etwa 25 feste Mitarbeiter für Opta in der Niederlassung in Unterföhring bei München. Opta erstellt Leistungsdaten von Fußballspielern. Diese sind reine Aufzählungen von verschiedenen Parametern. Eine Bewertung oder Interpretation der Daten wird von Opta nicht vorgenommen. Bei jedem Spiel der ersten und zweiten Bundesliga werden von Opta-Mitarbeitern die Spielgeschehnisse live aufgezeichnet. Dazu gibt es spezielle Tastenkombinationen, die eingegeben werden, um möglichst schnell sehr viele Daten zu sammeln. Die gesammelten Daten werden nach dem Spiel überprüft und in einer großen XML-Datei an die Kunden weitergegeben. In dieser XML-Datei stehen alle relevanten Fakten des Fußballspiels. Opta dokumentiert alle Aktionen, die etwas mit dem Ball oder besonderen Aktionen einzelner Spieler zu tun haben.² Opta selbst erfasst die Bewegungen der Spieler während des Spiels jedoch nicht lückenlos. Um diese weitaus umfangreicheren Daten zu erhalten, erfasst das Tracking System TRACAB automatisch mithilfe von speziellen Kameras die Position jedes Spielers und des Balls. Kamerasystem und Software werden von der Firma ChyronHego betrieben und an Opta weitergegeben. Die so gewonnenen Tracking-Daten werden dann ebenfalls in Echtzeit zur Verfügung gestellt.

² Vgl. Opta (2017). Football Data Collection Process

Der Aufbau der XML-Dateien ist nicht selbsterklärend, da viele Abkürzungen und Codes verwendet werden. Der Inhalt kann nur anhand der umfangreichen Dokumentationen verstanden werden. Die für meine Arbeit relevanten Einträge der beiden Dateien werde ich im Folgenden kurz erläutern.

2.1 Event.xml

```
<Event id="1105269413" event_id="643" type_id="5" period_id="2"
min="47" sec="2" player_id="39476" team_id="157" outcome="0" x="14.5" y="-1.1"
timestamp="2015-09-23T20:04:58.433" last_modified="2015-09-23T20:05:16" version="1443035116495">
  <Q id="823571005" qualifier_id="56" value="Back" />
  <Q id="460096570" qualifier_id="233" value="446" />
</Event>
```

Die Event-Datei listet alle Spielevents auf, darunter fallen Zweikämpfe, Pässe, Schüsse oder Tore. Diese Events werden einer Position auf dem Spielfeld zugeordnet und es werden weitere Informationen wie zum Beispiel die beteiligten Spieler ergänzt. Die Events werden chronologisch untereinander mit dem Tag Event geordnet. Dieser Tag enthält erneut verschiedene Attribute: event-id zählt alle Events chronologisch hoch. type-id gibt den Typ des Events an, also ob es ein Pass, ein Zweikampf oder ein anderes Event ist. period_id gibt die Hälfte des Spiels an. Mit min und sec wird das Event einem bestimmten Zeitpunkt zugeordnet. Die player_id gibt den Spieler an, die team_id die Mannschaft. X und Y geben die Koordinaten an, timestamp die Ortszeit. Unterhalb der Ebene Event gibt es noch eine weitere Ebene, die qualifier_id. Diese gibt weitere Informationen über ein bestimmtes Event, beispielsweise mit welchem Fuß ein Tor erzielt oder in welchen Bereich des Spielfelds ein Pass hineingespielt wurde. Ein Event hat oft etwa fünf bis zehn qualifier_ids, die das Event dann sehr genau beschreiben. Die Event-Datei eines Spiels enthält etwa 800 Events.

2.2 Tracking.xml

```
<FrameSet GameSection="firstHalf" MatchId="DFL-MAT-0025EV" TeamId="DFL-CLU-000002" PersonId="DFL-OBJ-0026PE">
  <Frame N="10000" T="2015-09-23T20:01:07.000+02:00" X="-0.01" Y="-0.65" S="0.00" M="1"/>
  <Frame N="10001" T="2015-09-23T20:01:07.040+02:00" X="-0.02" Y="-0.65" S="0.23" M="1"/>
```

Die Tracking-Datei gibt die Positionen der Spieler und des Balls zu jedem Zeitpunkt des Spiels an. Die Position des Spielers wird 25-mal in der Sekunde gemessen und als ein Eintrag in der XML-Datei abgespeichert. Jeder Eintrag ist gleich aufgebaut. Die gesammelten Einträge eines Spielers werden untereinander aufgelistet. Sie werden unter den Tag `FrameSet` gestellt. Innerhalb dieses Tags gibt es noch mehrere Attribute, die weitere Informationen enthalten. `GameSection` gibt die Hälfte des Spiels an, `MatchId` weist die Einträge einem bestimmten Spiel zu, `TeamId` weist das Team zu und `PersonId` den ausgewählten Spieler. Die einzelnen Positionen des Spielers sind immer als Attribut unter dem Tag `Frame` gespeichert. `Frame` enthält auch einige Attribute: `N` ist ein Zähler für die Anzahl der Einträge, `T` ist ein Zeitstempel, der die genaue Ortszeit angibt, `X` gibt die X-Koordinate des Spielers an, `Y` gibt die Y-Koordinate an, `S` enthält die aktuelle Geschwindigkeit und `M` beschreibt die Minute des Spiels.

Insgesamt enthält die Tracking-Datei knapp drei Millionen Einträge und ist mit etwa 280 MB sehr groß.

3. Datenauswertung

Es gibt im Sport verschiedene Ansätze, Statistiken auszuwerten. Das große Problem dabei ist, dass eine einfache Statistik oftmals nur wenig über das eigentliche Spiel aussagt. Es muss eine Analyse der Statistiken vorgenommen werden, die wirklich wichtigen Erkenntnisse sind oft nur schwer zu finden. Der Fußball lässt sich nicht mit wenigen Zahlenwerten beschreiben, aber Statistiken liefern Anhaltspunkte und bestätigen Eindrücke, bei denen Statistiken helfen können. Eine Möglichkeit, Fußballdaten auszuwerten, ist das Erstellen einer Heatmap. Dazu werden die Dateien über das Fußballspiel verwendet, also die Event-Datei und die Tracking-Datei. Auf Grundlage der Koordinaten, die den einzelnen Ereignissen angefügt sind, wird eine Karte des Spielfeldes mit einem Netz aus rechteckigen Feldern erstellt, die unterschiedlich eingefärbt werden. Je öfter ein Spieler ein Feld auf dem Spielfeld betreten hat, desto stärker wird dieses Feld eingefärbt. Ich habe im Laufe des Projekts

zwei Programme geschrieben, die jeweils eine Heatmap erstellen. Ein Programm wertet die Daten der Event-Datei aus, das andere die Daten der Tracking-Datei. Bevor die Heatmaps erstellt werden können, müssen die beiden Dateien eingelesen werden. Dies wird mit einem sogenannten Parser gemacht.

3.1 Parser

Ein Parser ist ein Computerprogramm, das dafür zuständig ist, eine Datei oder Eingabe zu zerlegen und umzuwandeln. Er wird häufig dazu genutzt, eine Datei sinnvoll einzulesen, um danach mit den Informationen weiterarbeiten zu können. Ein Parser kann auch die Struktur eines Dokuments beschreiben und ausgeben. Verschiedene Parser haben unterschiedliche Eigenschaften und Möglichkeiten. In meinen beiden Programmen habe ich zwei verschiedene Parser verwendet, den DOM Parser und den Iterparse Parser.

Der DOM Parser (Document Object Model) ist ein Parser zum Einlesen einer XML-Datei. Er kann XML oder HTML Code aus einem String in ein DOM Dokument parsen. Der Vorteil des DOM Parsers ist, dass nach dem Parsen des Dokuments ein Baum erstellt wird, auf den man gezielt zugreifen kann. Deshalb kann man beim DOM Parser sehr differenziert und gezielt mit der geparsen Datei interagieren.

Der Iterparse Parser ist ebenfalls ein Parser zum Einlesen von XML-Dateien. Im Gegensatz zum DOM Parser kann er jedoch keinen komplexen Baum erstellen, er erzeugt eine Liste der verschiedenen Einträge. Deshalb eignet er sich besser zum Einlesen von großen Dateien, da er weniger Rechenleistung benötigt.

Beim Erstellen meiner Programme habe ich zunächst den DOM Parser verwendet. Dieser hat mir einen einfachen Zugriff auf die Struktur des Dokumentes gegeben. Da ich mit der weniger umfangreichen Event XML begonnen habe, war auch die geringe Performance des Parsers kein Problem. Als ich dann jedoch das zweite, größere Tracking Dokument einlesen wollte, musste ich feststellen, dass die Datei von etwa 280 MB und knapp 3 Millionen Einträgen meinen Rechner überfordert hat. Versuche auf

einem leistungsfähigen SAP Rechner haben ergeben, dass der DOM Parser im Prinzip funktioniert, das Einlesen der Daten aber etwa 40 Minuten dauert. Ich habe einen Alternativparser gesucht und mich mit dem Konzept des iterativen Lesens beschäftigt. Mit dem Iterparse Parser dauert das Einlesen nur noch eine Minute, einen Algorithmus für diesen Parser zu finden war jedoch deutlich komplizierter.

3.2 Programmablauf: Event_Heatmap.py

In der Event.xml Datei werden alle wichtigen Ereignisse eines Spiels aufgelistet. Dabei gibt es auch immer die beiden Attribute X und Y, die dem Event eine Position auf dem Spielfeld zuweisen. Aus diesen Positionen habe ich eine Heatmap erstellt. Diese Heatmap teilt das Spielfeld in rechteckige Felder ein, die je nach Anzahl der Events in diesem Feld eingefärbt werden. So kann man sehen, in welchen Bereichen des Spielfeldes sich ein Spieler besonders oft aufgehalten hat. Mithilfe der Heatmap kann man Aussagen treffen, wie sich ein Spieler bewegt hat und ob er zum Beispiel seine Position eher offensiv oder eher defensiv interpretiert hat. Ebenfalls kann man aus der Heatmap herauslesen, ob eine Mannschaft im entsprechenden Spiel eine offensive Ausrichtung hatte oder in die Verteidigung gedrängt wurde.

Das Programm zum Einlesen der XML-Datei und Erstellen der Heatmap wurde in der Programmiersprache Python geschrieben. Die komplette XML-Datei mit allen Einträgen wird nach dem Parsen in einer Liste abgespeichert, mit der dann weitergearbeitet wird.

Das Ziel des Programms ist es, alle Events eines Spielers auf einer Heatmap zu lokalisieren. In der Liste mit allen Einträgen stehen allerdings die Events aller Spieler. Daher muss diese Liste zunächst nach dem gewünschten Spieler gefiltert werden. Jeder Spieler besitzt eine persönliche Spieler_Id, mit der jeder Event dem richtigen Spieler zugeordnet wird. Da der Nutzer des Programms diese Spieler_Id nicht kennt, wurde ein Input erstellt, in den man den Spielernamen eingeben kann. Dabei wird nur der vollständige Nachname eingegeben. Dieser Name wird dann über einen einfachen Vergleich der jeweiligen Spieler_Id zugeordnet. Die Größe der Heatmap hat einen entscheidenden Einfluss auf die Genauigkeit. Wenn die Heatmap in mehr Felder

unterteilt wird, wird sie genauer. Jedoch bedeuten mehr Felder auch mehr Rechenaufwand und bei relativ wenigen Einträgen sieht man keine differenzierte Verteilung über das komplette Spielfeld. Nach Versuchen mit verschiedenen Größen, habe ich entschieden, die Heatmap 20 auf 10 Felder groß zu machen, um einen optimalen Kompromiss zwischen Rechenaufwand und Genauigkeit zu finden. Diese 200 Felder entsprechen einem Fußballfeld. Dazu kommt noch eine Reihe von Feldern um das Spielfeld herum, um auch alle Events, die außerhalb des Spielfeldes (hinter beiden Toren und vor den Auswechselbänken) stattfinden, abzudecken. Deshalb kommt man auf eine endgültige Größe von 21 auf 12 Feldern.

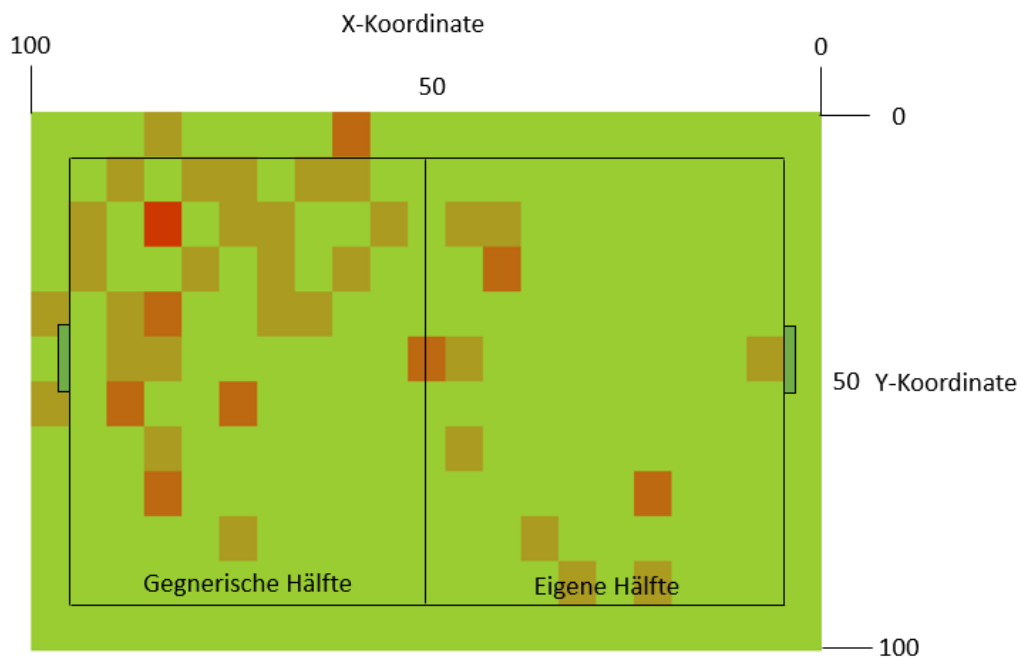


Abb. 1: Event-Heatmap mit 21 auf 12 Feldern und eingezeichneten Koordinaten

Eine Heatmap kann nur schwer mit der Python Standardfunktionalität erstellt werden. Zur grafischen Darstellung habe ich daher die Python Bibliothek Plotly verwendet. Plotly ist eine kostenlose Software zum Erstellen von mathematischen und analytischen Graphen in Python. Das Interface von Plotly verlangt die Übergabe der Daten in Form einer Matrix. Deshalb muss vor dem Aufruf von Plotly die Eventliste in eine Matrix

umgewandelt werden. Dazu wird eine Abfrage über die gesamte Eventliste gestartet. Bei jedem Eintrag wird geprüft, ob es ein Element ist, welches einen Spieler betrifft und nicht die ganze Mannschaft. Zusätzlich wird noch geprüft, ob das Event noch aktuell ist, d.h. die `type_Id` ist ungleich 43, was einem gelöschten Element entspricht. Danach wird mit einer Abfrage geklärt, ob das Event zum gewünschten Spieler gehört. Wenn der gewünschte Spieler zum abgefragten Event gehört, wird aus den Attributen die X-Koordinate und die Y-Koordinate ausgelesen. Das Spielfeld ist in der Event-Datei in Koordinaten eingeteilt, eine X-Koordinate für die Länge des Spielfelds und eine Y-Koordinate für die Breite mit Werten von jeweils 0 bis 100 und einem Koordinatenursprung in der linken oberen Ecke des Spielfeldes. Da das Fußballspielfeld nicht quadratisch ist (circa 100 m auf 60 m) und die Heatmap 20 auf 10 Felder groß ist, sind die Felder der fertigen Heatmap nicht quadratisch, sondern haben die Abmessungen von circa 5m Länge und 6m Breite. Nach dem Auslesen der Koordinaten erhält man zwei Variablen für die X und Y Koordinaten. Jetzt muss geprüft werden, in welchem Feld die Position liegt, die durch die Koordinaten beschrieben wird. In der linken oberen Ecke des Spielfelds wird begonnen, es wird ein Feld nach dem anderen abgefragt. Parallel dazu läuft ein Zähler hoch. Die Event Liste wird insgesamt 12-mal durchlaufen, für jede Zeile werden die Werte in die Matrix eingetragen, bevor die nächste Zeile der Heatmap abgefragt wird. Am Ende enthält die Matrix alle Einträge des Spielers. Diese Matrix folgt der Anordnung des Fußballfeldes von Tor zu Tor. Damit plotly das Fußballfeld im Querformat darstellt, muss die Matrix noch transponiert werden, d.h. Zeilen werden zu Spalten und Spalten entsprechend zu Zeilen. Ich habe einen Farbbereich vom kleinsten zum größten Wert definiert, mit Farben von grün bei niedrigen Zahlen über orange zu rot bei sehr großen Zahlen als Farbskala zu wählen. Die Heatmap wird immer im Browser geöffnet. Das Durchlaufen des Programms dauert etwa 40 Sekunden, die Anzahl der Events eines Spielers hat keinen Einfluss auf die Performance, da immer alle Einträge durchlaufen werden.

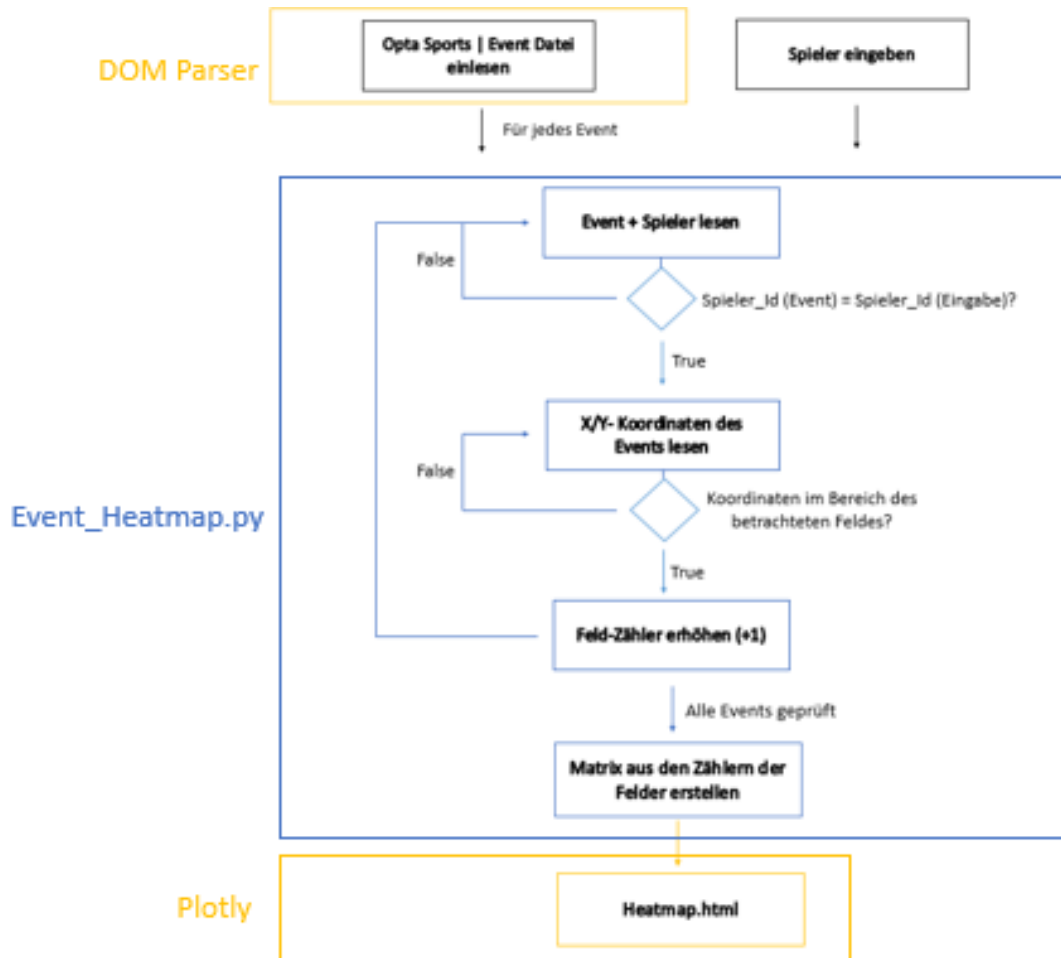


Abb. 2: Ablaufdiagramm Event_Heatmap.py

3.3 Programmablauf: Tracking_Heatmap.py

Das Programm Tracking_Heatmap.py erstellt eine Heatmap aus den Einträgen der Tracking Datei. Dabei ist die größte Schwierigkeit, dass die Datei mit 280 MB sehr groß ist und sehr viele Einträge enthält.

Zunächst erfolgt wieder die Eingabe eines Spielernamens, für den die Heatmap erstellt wird. Dieser Input wird in eine spezifische Spieler_Id umgewandelt, wobei beachtet werden muss, dass die Datenstrukturen in den XML Dateien nicht einheitlich ist, d.h. die Spieler_Ids für einen Spieler unterscheiden sich in den Dateien. Nach dem Eingeben des Spielernamens wird die Datei Tracking.xml mithilfe des Iterparse Parsers

eingelassen. Iterparse hat den Vorteil, dass eingelesene Elemente, die nicht dem Spieler zugeordnet werden können und somit nicht gebraucht werden, direkt wieder gelöscht werden. Wenn man große Dateien einlesen will, kann man sie mit vergleichsweise wenig Rechenaufwand lesen und verarbeiten. Im Gegensatz zum DOM-Parser werden nicht alle 3 Millionen Elemente in einer Liste gespeichert, es wird auf die einzelnen Teile zugegriffen, wenn sie gebraucht werden. Daher wird bereits während dem Einlesen der Datei eine Matrix erstellt, die vom Aufbau und der Struktur der Matrix aus dem Event Programm entspricht. Die Trackingdatei hat eine sehr einfache Struktur. Jeder Spieler hat pro Halbzeit einen Block, in dem in chronologischer Reihenfolge seine Position etwa 25-mal pro Sekunde aufgelistet ist. Pro Block gibt es für einen Spieler ein FrameSet. Eine Ebene darunter werden die X- und Y-Werte innerhalb des Tags Frame abgelegt. Im Gegensatz zur Eventdatei liegt der Koordinatenursprung in der Mitte des Spielfeldes, daher gibt es auch negative Koordinatenwerte. Einem Spieler sind pro Halbzeit 67.500 Frames zugeordnet, aus denen mithilfe einer for-Schleife die Werte die X- und Y-Koordinaten ausgelesen werden und in je einer Liste gespeichert werden.

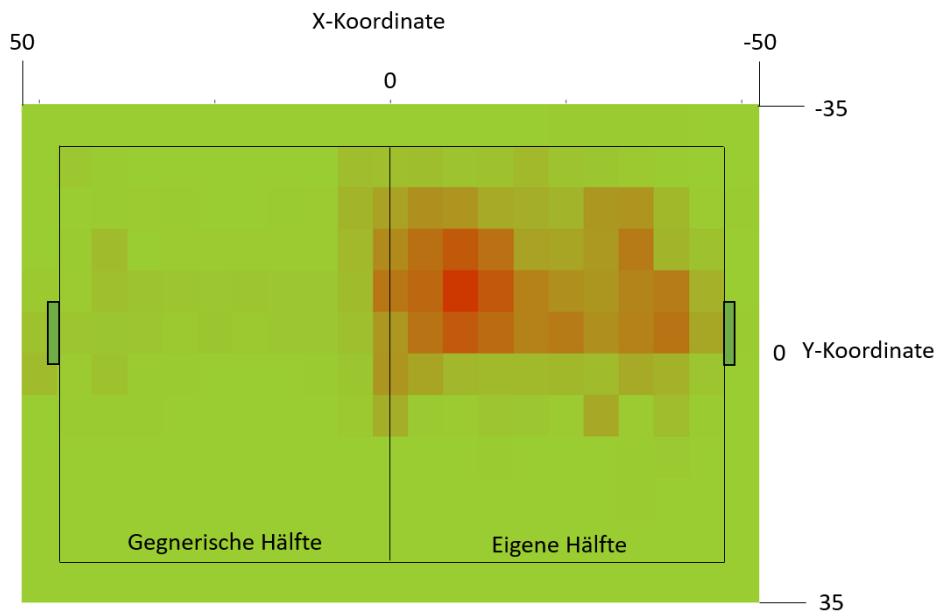


Abb. 3: Tracking-Heatmap mit 21 auf 12 Feldern und eingezeichneten Koordinaten

Die Listen werden wieder zu einer Matrix verarbeitet. Das Erstellen der Matrix funktioniert analog zum Erstellen der Matrix im ersten Programm. Das Durchlaufen dieses Programms dauert etwa zwei Minuten.

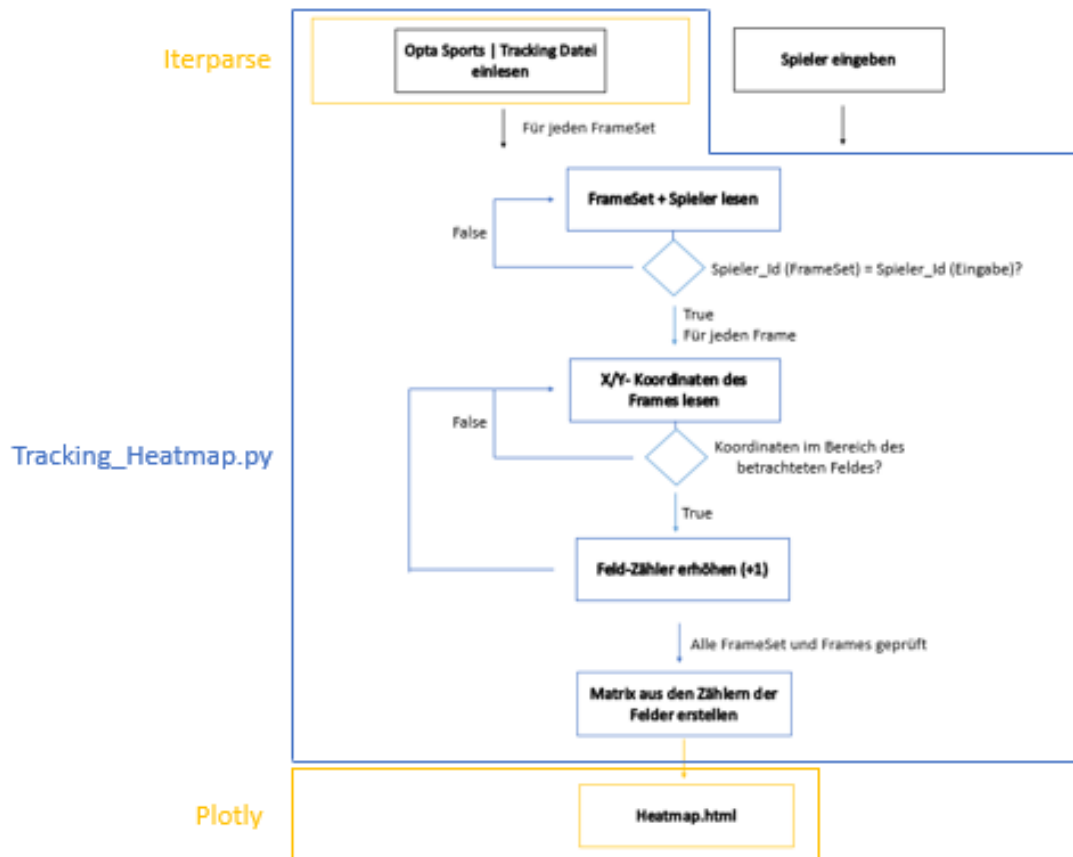


Abb. 4: Ablaufdiagramm Tracking_Heatmap.py

3.4 Herausforderungen

Die größte Herausforderung in diesem Projekt war die performante Umwandlung von Big Data in Smart Data, das heißt aussagekräftige Daten. Um die Tracking-Informationen eines Spielers auswerten zu können, müssen seine 135.000 Datensätze aus dem Gesamtvolumen von 3 Millionen Datensätzen herausgefiltert werden. Der geplante Einsatz von Sensoren, die zusätzlich Gesundheitsdaten und Bewegungen erfassen, wird die Menge an Daten noch einmal drastisch erhöhen. Entscheidend für die Entwicklung von neuen Algorithmen ist daher eine Big-Data-Plattform mit einer hohen Verarbeitungsgeschwindigkeit, die ein verteiltes Ausführen der Algorithmen über große Datenmengen erlaubt.

Eine weitere Herausforderung war das Verständnis der erfassten Daten. Pro Spiel werden mehr als 2000 Events erfasst. Ballkontakte und Pässe machen den größten Teil dieser Events aus. Um nur sinnvolle Events in der Heatmap darzustellen, mussten alle Events verstanden werden, um einige Events wie zum Beispiel nachträglich korrigierte Werte herausfiltern zu können.

In einer ersten Version meiner Tracking Heatmap bekam ich eine falsche Raumaufteilung dargestellt. Im Fußball werden in der Halbzeit die Seiten gewechselt. Die Koordinaten bleiben aber gleich. Deshalb müssen die Abfragebereiche für die einzelnen Felder an die beiden Halbzeiten angepasst werden. Nach der Halbzeit werden alle Koordinaten invertiert, bevor das richtige Feld ermittelt wird. Diese Vorgehensweise ist nur bei der Tracking-Datei notwendig. In der Event-Datei werden die Koordinaten für die zweite Hälfte direkt in den Daten invertiert, sodass man zu jedem Zeitpunkt des Spiels für die gleiche Position die gleiche Koordinate erhält.

Beim Testen des Event Programms hat sich herausgestellt, dass es Events gibt, die außerhalb des Feldes liegen. Diese Events haben entweder außerhalb des Spielfeldes stattgefunden, oder sie sind fehlerhaft. SAP Sports One hat von Opta Daten bekommen, die direkt nach Spielende zur Verfügung gestellt wurden. Diese sind oftmals sehr fehlerhaft und werden normalerweise später noch überarbeitet. Um auch Events

außerhalb des Spielfeldes darstellen zu können, wurde der Matrix eine weitere Zeile und zwei weitere Spalten an Feldern hinzugefügt.

Auch mit einigen Tücken der Programmiersprache musste ich kämpfen. In der XML-Datei sind alle Einträge als string, das heißt als Kette von Zeichen gespeichert. Auch die Zahlenwerte bei der X-Koordinate und der Y-Koordinate sind als String und nicht als Integer, also als Zahlenwert abgespeichert. Das führt dazu, dass Python Fehlermeldungen ausgegeben hat, als ich versucht habe, diese Werte miteinander oder mit Zahlen zu vergleichen, um sie einem bestimmten Feld zuzuordnen. Deshalb habe ich erst die Werte, die als String abgespeichert wurden, mithilfe des Befehls zur Typumwandlung `int(float)` in einen Zahlenwert umgewandelt. Dadurch konnte ich diese Werte mit verschiedenen Zahlen vergleichen und sie einem bestimmten Feld zuordnen. Auch ist Python sehr restriktiv bei der Benennung von Variablen. Eine Variable mit dem Namen `List` führte zu einem Programmabbruch, da `List` ein Python Befehl ist und deshalb nicht als Variablenname verwendet werden darf.

4. Interpretation der Daten

Aus den erstellten Heatmaps lassen sich einfach Rückschlüsse über die Raumaufteilung und die Leistung einzelner Spieler ablesen. Jedoch muss man dabei immer beachten, welche Position dieser Spieler im betrachteten Spiel ausgefüllt hat und wie die taktische Ausrichtung des Teams war. Es macht wenig Sinn, die Heatmap eines Mittelfeld Spielers mit der eines Torwarts zu vergleichen. Dagegen kann man aus dem Vergleich zweier Mittelfeldspieler herauslesen, welcher der beiden Spieler seine Rolle offensiver interpretiert hat.



Abb. 5: Tracking Heatmap Niklas Süle (Verteidiger)

Bei Verteidigern kann man beispielsweise sehen, ob sie sich an ihre vorgegebene Position gehalten haben oder öfters die Seite gewechselt haben. Dadurch kann der Trainer Erkenntnisse gewinnen, ob seine Anweisungen befolgt wurden. In der Tracking-Datei von Niklas Süle kann man sehen, dass er auf der rechten Innenverteidiger Position gespielt hat, er hat sich kaum von seiner vorgegebenen Position wegbewegt.

Aus dem Vergleich beider Heatmaps kann man noch differenziertere Interpretationen ablesen. Wenn man die Tracking-Heatmap eines Spielers mit seiner Event-Heatmap vergleicht, kann man seine Effektivität beurteilen.

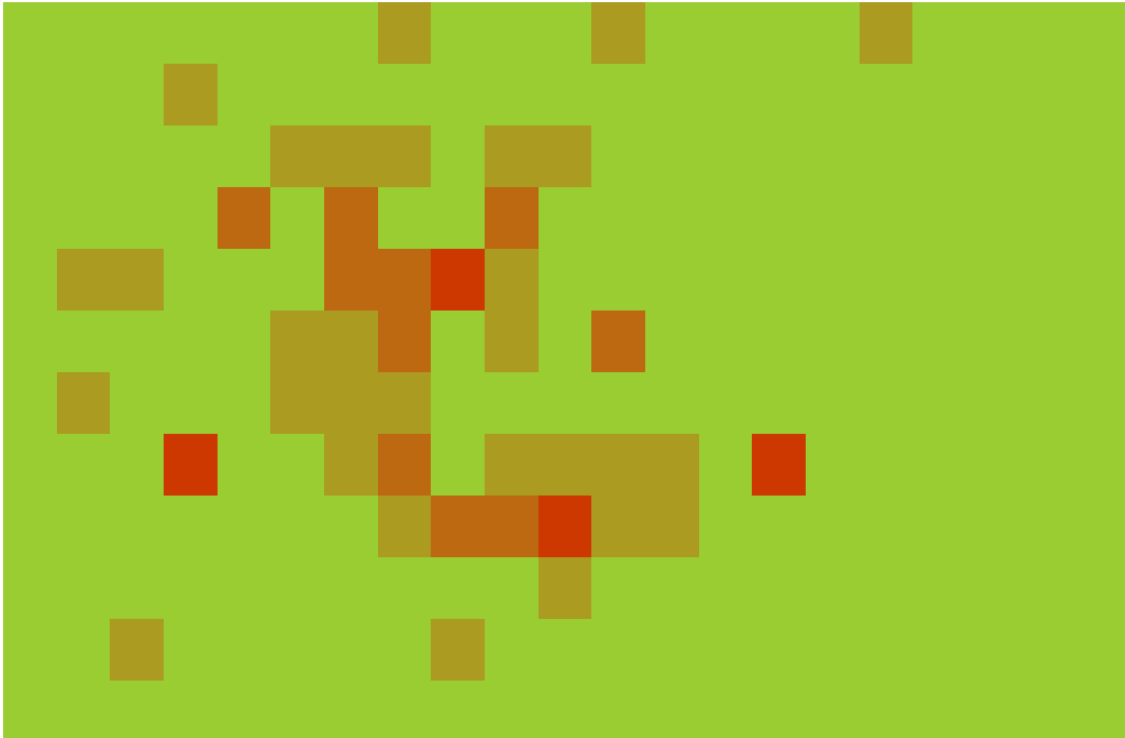


Abb .6: Event Heatmap Eduardo Vargas (Stürmer)

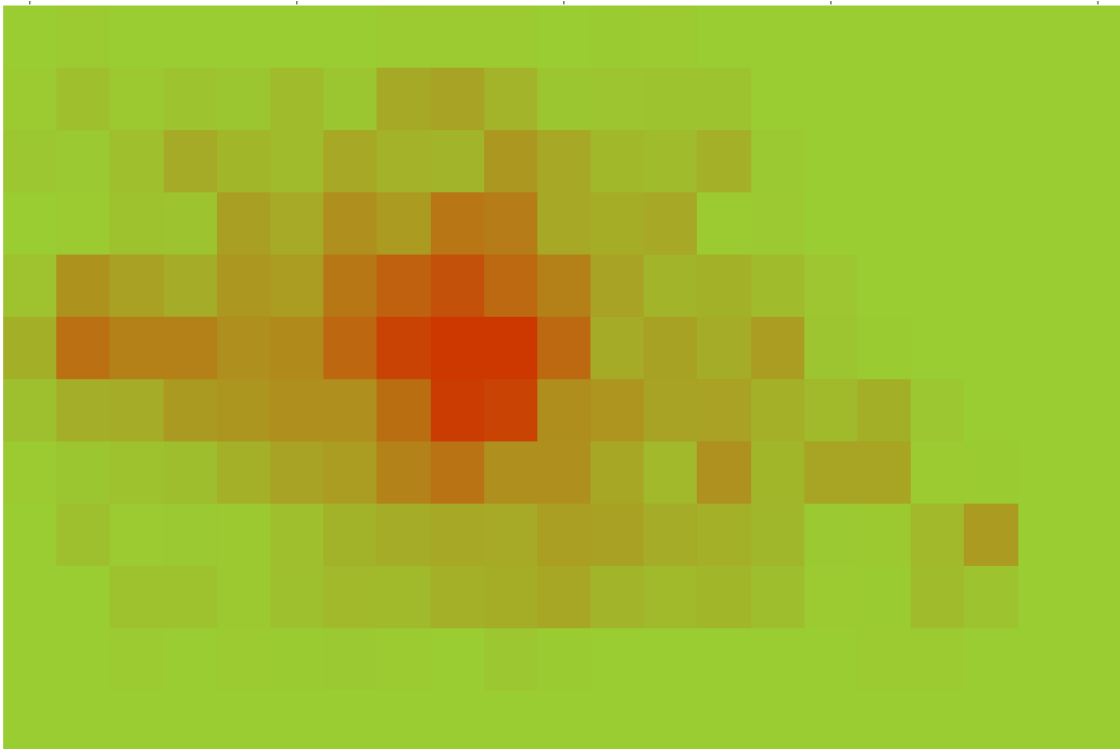


Abb. 7: Tracking Heatmap Eduardo Vargas (Stürmer)

Beim Hoffenheimer Stürmer Vargas sieht man, dass er zwar viel in der gegnerischen Hälfte gelaufen ist, dabei aber nur wenige Events hatte. Er ist also hauptsächlich dem Ball hinterhergelaufen, anstatt am Ball zu sein oder Zweikämpfe zu bestreiten.

Betrachtet man hingegen die Event Heatmap vom Dortmunder Stürmer Aubameyang, der auch das Dortmunder Tor erzielt hat, sieht man, dass er deutlich mehr Aktionen im gegnerischen Strafraum hatte.

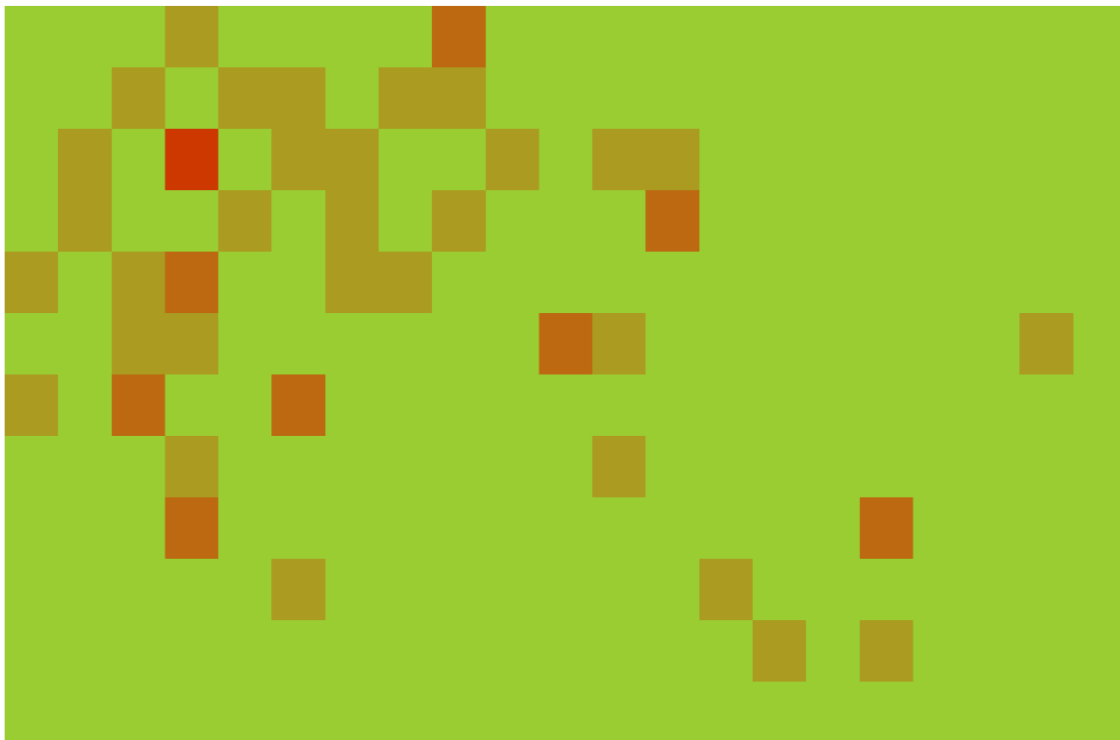


Abb. 8: Event Heatmap Pierre-Emerick Aubameyang (Stürmer)

Der alleinige Einsatz von Heatmap-Analysen macht keinen Sinn. Man muss sie in Kombination mit anderen Kennzahlen betrachten und immer in den Kontext des Spiels setzen. Alle gesammelten Statistikwerte im Saisonverlauf ermöglichen den Trainern und Analysten, erfolgreiche Taktiken herauszufiltern und neue taktische Ausrichtungen zu entwickeln. Daher erstaunt es nicht, dass Statistiken im Fußball immer beliebter werden. Doch was für eine Aussagekraft haben diese Statistiken? Ist Big Data wirklich

so wertvoll wie von den Experten beschrieben? Durch neue technische Möglichkeiten wird es immer einfacher, viele Daten zu erheben. Quasi für jede Spielsituation gibt es eine Fülle von Datenpunkten, das Problem ist es aber, die richtigen Algorithmen zu ermitteln und anzuwenden. In der Theorie ist Fußball zwar ein sehr einfacher Sport, in der Praxis ist er unberechenbar, irrational und chaotisch. Fußball ist sehr komplex und wird sich nie ganz erklären lassen. Glück und Pech werden immer ein Teil sein, Fehler und Emotion kann man nie gänzlich herausfiltern. Es ist fast unmöglich, mit Statistiken alles richtig zu analysieren. Den Analysten geht es nicht darum, alles zu erkennen und zu erklären. Selbst wenn man neunzig Prozent richtig analysiert, gibt es oft die zehn Prozent Zufall, die am Ende entscheiden. Wenn Mario Götze in der 114. Minute mit seinem Tor das WM Finale entscheidet, dann zählt plötzlich nur noch die individuelle Leistung in dieser Minute und nicht die durchschnittlichen Statistiken Götzes in den Minuten davor.

Ich bin der Meinung, dass man durchaus Statistiken verwenden sollte, es mit ihnen aber nicht übertreiben darf. Ein schlechter Fußballer kann sich mit viel Analyse und Statistik verbessern, diese Entwicklung hat jedoch ihre Grenzen. Den Traum der Analysten, in Zukunft den Trainer durch ein Computerprogramm zu ersetzen, wird sich meiner Meinung nach nicht erfüllen. Dafür ist der Fußball zu differenziert, ein Trainer, der auch Entscheidungen aus dem Bauch heraus trifft, wird immer gebraucht werden. Ich denke aber, dass der Trend in Richtung noch mehr Big Data sich verstärken wird.³ Im Moment ist es verboten, im Spiel Sensoren an den Spielern anzubringen, dies könnte in Zukunft jedoch erlaubt werden. Das könnte die Menge an Daten noch einmal drastisch erhöhen und noch detailliertere und aussagekräftigere Analysen über Fitness und Trainingszustand der Spieler liefern. Generell kann man sagen, dass die Menge an Statistiken und das Vertrauen auf Big Data Vorteile haben kann und es interessant ist, diese Entwicklung weiter zu verfolgen. Die Grundidee des Fußballs, das Runde muss ins Eckige, wird sich dadurch aber nicht verändern.

³ Vgl. SAP SE. Changing the Game with Wearable Sensors

5. Danksagung

Ich möchte mich an dieser Stelle bei all denjenigen bedanken, die mich während der Anfertigung dieser Seminar-Arbeit unterstützt haben.

Ganz besonders gilt dieser Dank Herrn Dr. Andrew McCormick-Smith, der zum ersten Mal eine Hector Kooperation mit der SAP SE ermöglicht hat. Er war sofort bereit, mir ein Projekt im Bereich SAP Sport One anzubieten und mich während meiner Arbeit zu betreuen. Dabei hat er mich ermutigt ein Thema auszuwählen, bei dem ich schon vorhandene Programmierkenntnisse vertiefen konnte, aber auch meinen Fußballsavverstand optimal einbringen konnte.

Mit dem Abschluss des Kooperationsprojektes endet meine Zeit im Hector Seminar Heidelberg. Ich möchte mich ganz herzlich bei Herrn Dr. h.c. Hans-Werner und Frau Josephine Hector bedanken, die diese erlebnisreiche Zeit ermöglicht haben.

Vielen Dank auch an Herrn Dr. Karl-Friedrich Raqué und Herrn Dr. Christoph Gölz, die unseren Kurs die letzten sechs Jahre betreut haben und uns auf vielen unvergesslichen Exkursionen begleitet haben.

6. Literaturverzeichnis

1. Deutschlandfunk (2015): Der Plan vom perfekt programmierten Fußball-Spiel.
URL: http://www.deutschlandfunk.de/die-fussball-matrix-der-plan-vom-perfekt-programmierten.740.de.html?dram:article_id=320642http://www.deutschlandfunk.de/die-fussball-matrix-der-plan-vom-perfekt-programmierten.740.de.html?dram:article_id=320642 (02.07.2017).
2. Opta (2017): Football Data Collection Process.
URL: <http://www.optasports.com/about/how-we-do-it/the-data-collection-process.aspx> (18.06.2017)
3. Stein, T (2014): Die Vermessung der Fußballwelt.
URL: <http://cicero.de/kultur/statistikwahn-die-vermessung-der-fussballwelt/57437> (10.07.17).
4. SAP SE: Sport und Entertainment.
URL: <https://www.sap.com/germany/solution/industry/sports-entertainment.html> (08.07.2017)
5. SAP SE: Changing the Game with Wearable Sensors.
URL: <https://www.sap.com/about/customer-testimonials/sports-entertainment/tsg-1899-hoffenheim.html> (08.07.2017)
6. SAP SE (2016): SAP Sports One.
URL: <https://www.sap.com/germany/products/sports-one.html>
NatSchG+BW&psml=bsbawueprod.psml&max=true&aiz=true#jlr (29.06.2017)