Projektbericht

in dem Modul

Web Mining

zu dem Thema:

Web-Scraping von Daten der ersten Bundesliga zur Vorhersage von Spielergebnissen

Vorgelegt im berufsbegleitenden Studiengang M.Sc. Data Science

von

Alfred Anselm

Matrikelnummer 30258459

Kevin Diec

Matrikelnummer 30245778

Luca Janas

Matrikelnummer 30277119

Prüfer: Prof. Dr. Christian Gawron

Im Sommersemester 2023

**Eigenständigkeitserklärung**

Ich erkläre hiermit, dass die vorgelegte Arbeit mein eigenes Werk ist. Alle direkt oder indirekt verwendeten Quellen sind als Referenzen angegeben. Die Arbeit wurde bisher nicht vor einem anderen Prüfungsausschuss vorgelegt und nicht veröffentlicht.

Mir ist bekannt, dass die Arbeit in digitaler Form auf die Verwendung unerlaubter Hilfsmittel überprüft werden kann, um festzustellen, ob die Arbeit als Ganzes oder darin enthaltene Teile als Plagiat zu werten sind. Für den Vergleich meiner Arbeit mit vorhandenen Quellen erkläre ich mich damit einverstanden, dass sie in eine Datenbank aufgenommen wird und dort auch nach der Prüfung verbleibt, um einen Vergleich mit künftigen eingereichten Arbeiten zu ermöglichen.

Münster, 30. September 2023.

**Alfred Anselm Kevin Diec Luca Janas**

1. Inhaltsverzeichnis

[I Inhaltsverzeichnis I](#_Toc145184229)

[II Abbildungsverzeichnis II](#_Toc145184230)

[III Tabellenverzeichnis III](#_Toc145184231)

[1 Projektbeschreibung 1](#_Toc145184232)

[2 Web-Scraping 2](#_Toc145184233)

[3 Datenaufbereitung und Datenanalyse 4](#_Toc145184234)

[4 Feature Engineering 11](#_Toc145184235)

[5 Modelltraining und -test 11](#_Toc145184236)

[6 Ergebnisanalyse 11](#_Toc145184237)

1. Abbildungsverzeichnis

[Abbildung 1 - Projektvorgehen 1](#_Toc145184210)

[Abbildung 2 - Scatterplot: Bundesliga-Platzierung vs. Gesamtmarktwert 5](#_Toc145184211)

[Abbildung 3 - Durchschnittliche Einnahmen und Ausgaben pro Saison 6](#_Toc145184212)

[Abbildung 4 - Anzahl der verschiedenen Spielausgänge je Saison 7](#_Toc145184213)

[Abbildung 5 - Confusion Matrix basierend auf Tipprundentendenz 8](#_Toc145184214)

[Abbildung 6 - Confusion Matrix basierend auf Tipprundentendenz mit 90%-Quote 9](#_Toc145184215)

[Abbildung 7 - Siegesquote des FC Bayern München pro Schiedsrichter 10](#_Toc145184216)

[Abbildung 8 - Siegesquote des 1. FC Köln pro Schiedsrichter 10](#_Toc145184217)

1. Tabellenverzeichnis

[Tabelle 1 - Spieltagsdaten 4](#_Toc145184222)

[Tabelle 2 - Attribute der finalen Bundesligatabelle 4](#_Toc145184223)

# Projektbeschreibung

In dem hier vorgelegten Projektbericht werden Inhalte und der Aufbau des Projektes im Modul Web Mining im berufsbegleitenden M.Sc. Data Science an der Fachhochschule Südwestfalen beschrieben. Ziel des Projektes ist es, unter Verwendung von Daten, die auf der Webseite <https://transfermarkt.de/> zur Verfügung gestellt werden, Spielergebnisse in der ersten deutschen Bundesliga vorherzusagen. Dazu werden historische Daten zu Spielen und Vereinen von transfermarkt.de durch Web-Scraping gesammelt und anschließend aufbereitet und analysiert, um einen Datensatz zu erstellen, der zur Vorhersage der Spielergebnisse verwendet werden kann. Dafür bietet Transfermarkt Informationen zu Fußballspielern, Vereinen, Marktwerten und Statistiken für verschiedene Ligen weltweit.

Das Projekt kann in folgende 5 Einzelteile aufgeteilt werden.

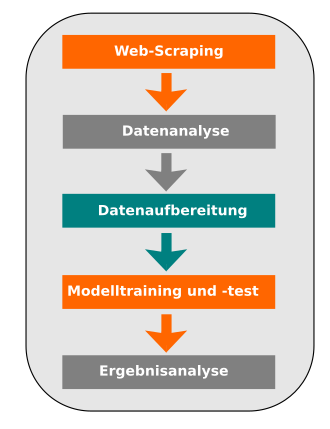


Abbildung 1 - Projektvorgehen

# Web-Scraping

Für die Selektion der Daten zu den Mannschaften aus der ersten Bundesliga und den jeweiligen Spielergebnissen pro Spieltag und Saison wird das Verfahren des Web-Scrapings auf die Internetseite <https://transfermarkt.de/> als Basis-URL angewendet. Für die jeweilige Datenselektion werden weitere URL-Pfade untersucht, die unter anderem Parameter beinhalten, welche die ausgewählte Saison und Spieltage enthalten. Über eine Loop-Funktion können alle Kombinationsmöglichkeiten daraus extrahiert werden.

Die Datenselektionen erfolgten in der Regel mit den Python-Bibliotheken *requests*, *BeautifulSoup* und *lxml.* Mit dem Modul *requests* ist es möglich, eine Anfrage an die vorgegebene URL zu senden und die HTML-Daten der Webseite auszulesen und zu extrahieren.

Um die extrahierten Daten aus dem HTML-Quellcode zu analysieren und zu filtern, wird das Modul *BeautifulSoup* verwendet. Es handelt sich dabei um eine Python-Bibliothek, die beim Web-Scraping eine wichtige Rolle spielt. Sie dient dazu, HTML-Dokumente zu parsen und sie in einer aufbereiteten Struktur darzustellen. In dieser Ausarbeitung wird *BeautifulSoup* in Verbindung mit der „lxml“-Bibliothek als Parser verwendet. Durch die Verwendung von *lxml* kann *BeautifulSoup* von den Funktionen und Vorteilen dieser Bibliothek profitieren, um die Analyse und Manipulation von Webseiteninhalten zu optimieren. Dabei bietet sie eine Reihe von Funktionen und Methoden an, um den Inhalt von Webseiten zu analysieren, die Datenstruktur zu verstehen und spezifische Elemente wie Tabellen, Überschriften, Links oder Absätze zu identifizieren.

Um die Daten weiterzuverarbeiten und zu analysieren, wird in gesonderten Fällen die Methode *etree* aus dem Modul *lxml* verwendet. Diese Methode ermöglicht es, die HTML-Struktur der Webseite genauer zu untersuchen und die Informationen gezielt auszuwählen. Insbesondere wird hierbei auf die Struktur der XPATH-Logik mit *etree* zurückgegriffen, um Schwierigkeiten bei der eindeutigen Identifizierung der Struktur zu überwinden. Das XPATH-Format erlaubt es, bestimmte Elemente in einem HTML-Dokument basierend auf ihrer Position und Hierarchie in einer Tabelle zu identifizieren und über eine Schleife über die Reihen und Spalten gezielt auszuwählen.

Für dieses Projekt wurden die folgenden Daten von Transfermarkt für die erste deutsche Bundesliga gescraped. Der Code dazu steht unter [web-mining/01\_Crawler.ipynb at main · lucajanas/web-mining (github.com)](https://github.com/lucajanas/web-mining/blob/main/01_Crawler.ipynb) zur Verfügung und ist so aufgebaut, dass auch die Daten anderer Ligen, wie der englischen Premier League, ohne Weiteres abgerufen und verarbeitet werden können.

*Vereinsüberblick pro Saison*

Transfermarkt bietet eine umfassende Übersicht über sämtliche Mannschaften der ersten Bundesliga pro Saison an. Hierbei ist zu beachten, dass sich die Zusammensetzung der Mannschaften in der Liga von Saison zu Saison ändert, da Teams auf- und absteigen können. Mithilfe der folgenden URL kann der Parameter für die jeweilige Saison übergeben werden:

[https://www.transfermarkt.de/bundesliga/startseite/wettbewerb/L1/plus/?saison\_id=<VALUE](https://www.transfermarkt.de/bundesliga/startseite/wettbewerb/L1/plus/?saison_id=%3cVALUE)>.

In diesem Kontext kann <VALUE> durch die gewünschte Saison ersetzt werden. Beispielsweise kann für die Saison 2022/2023 der Wert 2022 für <VALUE> eingesetzt werden. Auf diese Weise gelangt man zur entsprechenden Seite, die die Informationen zur Bundesliga-Saison 2022/2023 auf Transfermarkt präsentiert. Diese Herangehensweise ermöglicht es, beim Web-Scraping systematisch mittels einer For-Schleife alle verfügbaren Saisons zu durchlaufen. Die Informationen, die daraus selektiert werden können, sind in einem Dataframe zusammengefasst und beinhalten die folgenden Spalteninformationen.

|  |  |
| --- | --- |
| **Tabellenspalte** | **Erklärung** |
| CLUB\_NAME | Vereinsname aus der ersten Bundesliga |
| PLAYERS\_COUNT | Entspricht dem Kader (Die Anzahl der Spieler pro Verein innerhalb der Saison) |
| PLAYERS\_AVG\_AGE | Durchschnittliches Alter des Kaders |
| LEGIONARIES\_COUNT | Anzahl der Legionäre (Spieler außerhalb seines Heimatlandes) |
| AVG\_MARKET\_VALUE | Durchschnittlicher Marktwert des Vereins zum Zeitpunkt der Saison |
| TOTAL\_MARKET\_VALUE | Gesamtmarktwert des Vereins zum Zeitpunkt der Saison |
| season | Das Jahr, in dem die Saison begann |

Tabelle 1: Vereinsübersicht pro Saison

Normalerweise sind in jeder Bundesliga-Saison 18 Vereine vertreten. In den Jahren 1963 und 1964 gab es jedoch lediglich 16 Vereine, während es im Jahr 1991 sogar 20 Vereine pro Saison gab. Um diese besonderen Fälle in unserem Skript angemessen zu berücksichtigen, wurde eine IF-Bedingung eingeführt, die es ermöglicht, diese Ausnahmen korrekt abzubilden.

Es ist erwähnenswert, dass Marktwerte von Spielern erst seit der Saison 2002 erfasst und aufgelistet werden.

*Transferdaten pro Saison*

Über die folgende URL-Adresse

[https://www.transfermarkt.de/bundesliga/transfers/wettbewerb/L1/plus/?saison\_id=<SAISON>&s\_w=&leihe=1&intern=0&intern=1](https://www.transfermarkt.de/bundesliga/transfers/wettbewerb/L1/plus/?saison_id=%3cSAISON%3e&s_w=&leihe=1&intern=0&intern=1)

kann eine umfassende Untersuchung der Zu- und Abgänge innerhalb der Vereine innerhalb einer Saison durchgeführt werden. Hierbei könnten die entsprechenden Transfers auf Spielerbasis erfasst werden. Da diese detaillierte Erfassung auf Spielerbasis für unsere Analyse irrelevant und zu granular ist, konzentrieren wir uns auf aggregierte Informationen auf Vereinsebene. In diesem Zusammenhang sind die nachfolgenden Informationen extrahiert worden.

|  |  |
| --- | --- |
| **Tabellenspalte** | **Erklärung** |
| CLUB\_NAME | Vereinsname aus der ersten Bundesliga |
| AVG\_AGE\_JOINING | Durchschnittsalter aller Zugänge in der Saison |
| AVG\_AGE\_LEAVING | Durchschnittsalter aller Abgänge in der Saison |
| TOTAL\_VALUE\_JOINING\_MIO | Gesamtmarktwert der Zugänge in Mio. |
| TOTAL\_VALUE\_LEAVING\_MIO | Gesamtmarktwert der Abgänge in Mio. |
| EXPENSES\_JOINING\_MIO | Ausgaben durch Zugänge in Mio. |
| REVENUE\_LEAVING\_MIO | Einnahmen durch Abgänge in Mio. |
| season | Das Jahr, in dem die Saison begann |

Tabelle 2: Transferdaten pro Saison

Es ist zu beachten, dass die Angaben bezüglich der Marktwerte und finanziellen Transaktionen hier nicht in einer einheitlichen Währungseinheit („Mio. €“) präsentiert werden, sondern auch in Tausender Schritten („Tsd. €“) vorkommen können. Daher war eine Transformation notwendig, um sämtliche Angaben in einer einheitlichen Struktur darzustellen. Dies gewährleistet, dass die Daten für unsere Analysezwecke bereinigt und konsistent sind.

Es ist anzumerken, dass die Angaben zum Marktwert erst ab dem Jahr 2004 verfügbar sind.

*Spielinformationen zu allen Spieltagen pro Saison*

Transfermarkt stellt die Ergebnisse der Spiele der ersten Bundesliga für pro Spieltag zur Verfügung, so z. B. unter [Bundesliga - Spieltagsübersicht - 2. Spieltag 23/24 | Transfermarkt](https://www.transfermarkt.de/bundesliga/spieltag/wettbewerb/L1/plus/?saison_id=2023&spieltag=2). Neben den Mannschaften, die gegeneinander spielen, und dem Spielergebnis stehen das Spieldatum, der Schiedsrichter der Partie sowie die Tipprunden-Tendenzen zur Verfügung. Die Tipprunden-Tendenzen stellen die Ergebnisse von Umfragen in der Transfermarkt-Community dar. Es stehen drei Prozentwerte zur Verfügung, die angeben, welche Wahrscheinlichkeit die Transfermarkt den folgenden Ergebnissen zuordnet: Heimsieg, Unentschieden oder Auswärtssieg.

Für das Scraping und Parsing der Informationen werden requests und BeautifulSoup verwendet. Requests wird dabei für das Ausführen des http-Requests verwendet, um den Inhalt der Transfermarkt-Seite abzufragen und BeautifulSoup für das Parsen des Inhalts. Gezeigt wird dies im Folgenden am Beispiel der Heimmannschaften eines Spieltags, die ermittelt werden.

Nach Ausführung des http-Requests wird ein BeautifulSoup-Objekt erstellt. Nach Sichtung des HTML-Inhalts der abgerufenen Transfermarkt-Website steht fest, dass die benötigten Heimmannschaften in einer Tabelle mit dem HTML-Tag *td* und dem CSS-Klassennamen *„rechts hauptlink no-border-rechts hide-for-small spieltagsansicht-vereinsname“* stehen. Diese Tabellenelemente können über das erstellte BeautifulSoup-Objekt gefiltert werden. In diesem Fall müssen die tatsächlichen Vereinsnamen anschließend per regulärem Ausdruck aus den vorgefilterten Tabelleneinträgen ermittelt werden.

Auf diese Art und Weise werden die weiteren genannten Spieltagsdaten ebenfalls ermittelt. Letztendlich entsteht folgende Tabellenstruktur:

|  |  |
| --- | --- |
| **Tabellenspalte** | **Erklärung** |
| {HOME, AWAY}\_TEAM | Name der Heim- und Auswärtsmannschaftmannschaft |
| {HOME, AWAY}\_GOALS | Anzahl der Tore der Heim- und Auswärtsmannschaft |
| PLACE\_{HOME, AWAY}\_TEAM | Platzierung der Heim- und Auswärtsmannschaft vor Beginn des Spieltags |
| REFEREE | Schiedsrichter der Partie |
| DATE | Datum, an dem das Spiel stattfand |
| WEEKDAY | Wochentag des Spiels |
| MONTH | Monat, in dem das Spiel stattfand |
| SEASON | Saison, in der das Spiel stattfand |
| MATCHDAY | Spieltag |
| WIN\_PERC\_{HOME, AWAY} | Tipprundentendenzen der Transfermarkt-Community zu „Sieg Heim- bzw. Auswärtsmannschaft“ |
| REMIS\_PERC | Tipprundentendenzen der Transfermarkt-Community zum Ergebnis „Unentschieden“ |

Tabelle 3 - Spieltagsdaten

Die Notation {HOME, AWAY} bedeutet, dass es das Attribut sowohl für die Heim- als auch für die Auswärtsmannschaft gibt. Die geschriebene Funktion zum Scrapen der Spieltagsdaten wird für je einen Spieltag einer Saison aufgerufen. Dies passiert in einer Schleife für alle benötigten Spieltage.

*Finale Bundesliga-Tabelle pro Saison*

Um den Zusammenhang zwischen Attributen wie dem Marktwert eines Vereins und seiner finalen Platzierung in der Bundesliga-Tabelle untersuchen zu können, werden die finale Bundesliga-Tabellen jeder Saison von Transfermarkt abgerufen. Eine Beispiel-Tabelle findet sich auf der folgenden Seite: [Bundesliga - Tabelle | Transfermarkt](https://www.transfermarkt.de/bundesliga/tabelle/wettbewerb/L1?saison_id=2022)

Es werden die Tabellenelemente aus der CSS-Klasse *"no-border-links hauptlink"* benötigt. Diese werden über das erstellte BeautifulSoup ermittelt. Dies resultiert in folgendem Tabellenaufbau:

|  |  |
| --- | --- |
| **Tabellenspalte** | **Erklärung** |
| SEASON | Saison |
| CLUB\_NAME | Vereinsname |
| PLACE | Finale Platzierung am Ende der Saison |

Tabelle 4 - Attribute der finalen Bundesligatabelle

Die erstellten Datensätze stellen die Grundlage für die nachfolgenden Schritte dar.

*Wetterdaten:*

Um weitere Einflussfaktoren für die Analyse der Spielergebnisse zu berücksichtigen, haben wir zusätzliche Wetterdaten herangezogen, welche wir durch Web-Scraping von www.wetterkontor.de gewinnen konnten. Auf dieser Plattform sind tägliche Wetterdaten ab dem 01.01.2011 verfügbar. Über die nachfolgende URL können tageweise alle Wetterinformation herausselektiert werden, wenn das Datum im Parameter eingepflegt wird:

[https://www.wetterkontor.de/de/wetter/deutschland/extremwerte.asp?id=<yyyymmdd](https://www.wetterkontor.de/de/wetter/deutschland/extremwerte.asp?id=%3cyyyymmdd)>

Dadurch stehen folgende Wetterdaten zur Verfügung:

|  |  |
| --- | --- |
| **Tabellenspalte** | **Erklärung** |
| WEATHER\_STATION | Standort der Wetterstation |
| MIN\_TEMP\_C | Minimale Temperatur in °C des Tages |
| MAX\_TEMP\_C | Maximale Temperatur in °C des Tages |
| SNOW\_HEIGHT\_cm | Schneehöhe in cm |
| RAIN\_l/m2 | Niederschlagsmenge l/m² |
| SUNSHINE\_DURATION\_h | Sonnenscheindauer in Stunden pro Tag |
| HOME\_TEAM | Zu der Wetterstation naheliegender Standort der Heimmannschaft |
| MATCHDAY | Spieltag |

Tabelle 5: Wetterdaten pro Tag pro Wetterstation

Es ist jedoch anzumerken, dass es mitunter schwierig ist, die genauen Standorte der Fußballvereine den verfügbaren Wetterstationen eindeutig zuzuordnen. Dies ergibt sich aus der Tatsache, dass nicht in allen Städten Wetterstationen vorhanden sind und einige Städte über mehrere Wetterstationen verfügen. Die Zuordnung erfolgte daher manuell unter Berücksichtigung des bestmöglichen Wissens und Gewissens. Darüber hinaus unterliegen die Wetterstationen im Laufe der Zeit Veränderungen, da neue Stationen hinzukommen und einige wieder entfernt worden sind.

Bedauerlicherweise führt diese Situation dazu, dass eine erhebliche Anzahl von Datensätzen an den verschiedenen Spieltagen verloren gehen, wie dies in der unten dargestellten Heatmap deutlich wird. Die Flächen, die in roter Farbe markiert sind, repräsentieren die fehlenden Werte. Es sollte beachtet werden, dass die Heatmap so strukturiert ist, dass jede Zeile einem Spieltag entspricht. Obwohl es herausfordernd sein kann, die Daten zeilenweise zu interpretieren, erweist sich die Heatmap als äußerst hilfreich, um einen umfassenden Überblick über die fehlenden Informationen zu erhalten.

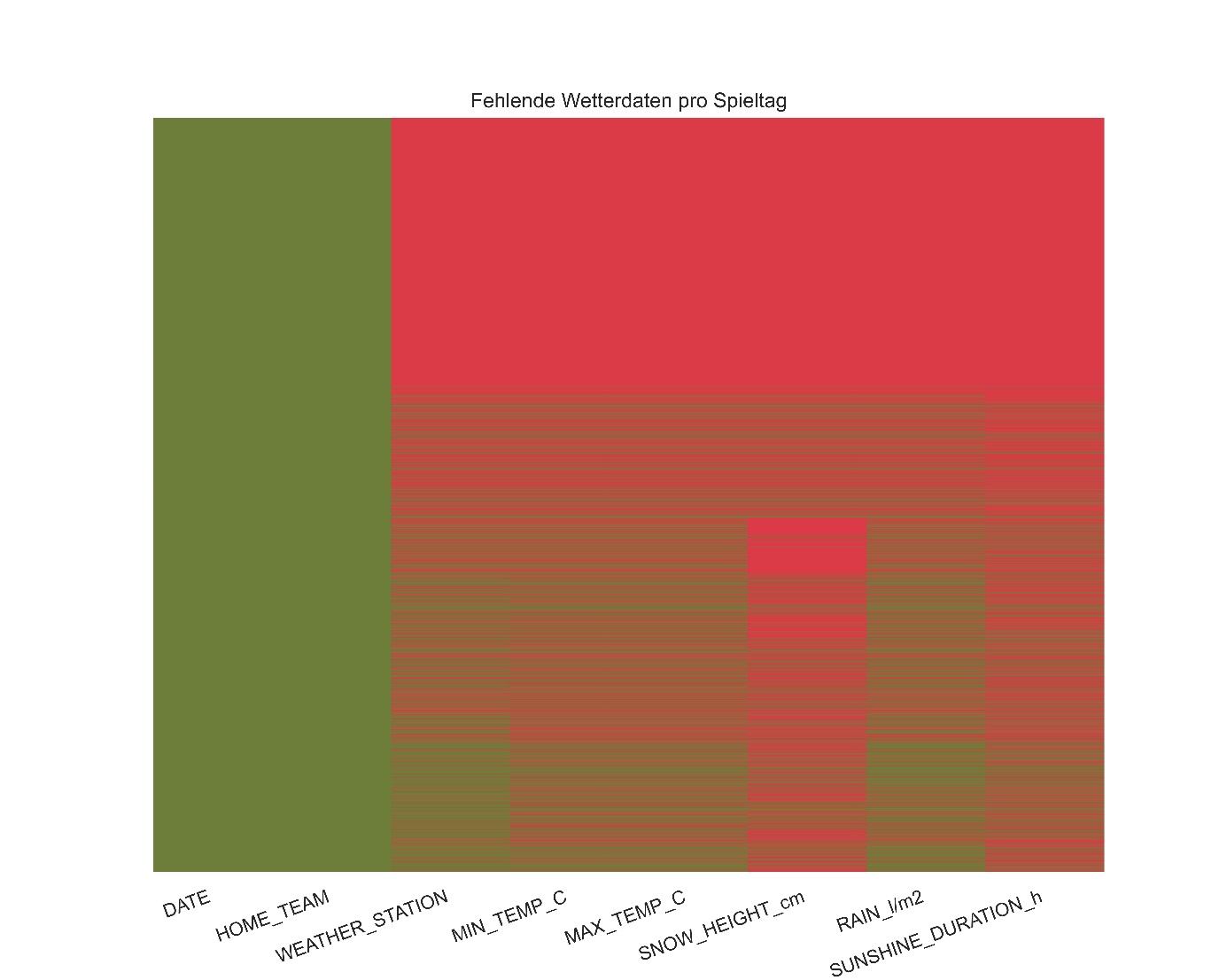


Abbildung 2: Heatmap über fehlende Wetterdaten pro Spieltag

In Zahlen ausgedrückt bedeutet dies, dass wir von 5814 Spieltagen nur 2251 Wetterstation zuordnen konnten, welche wiederum nur 1980 verwertbare Temperaturdaten, 1198 Schneedaten, 2170 Regendaten und 1425 Sonnenscheindaten aufweist.

Zusammengefasst ist dies aus der Python-Funktion df.info() zu entnehmen:

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 DATE 5814 non-null datetime64[ns]

1 HOME\_TEAM 5814 non-null object

2 WEATHER\_STATION 2251 non-null object

3 MIN\_TEMP\_C 1980 non-null object

4 MAX\_TEMP\_C 1980 non-null object

5 SNOW\_HEIGHT\_cm 1193 non-null object

6 RAIN\_l/m2 2170 non-null object

7 SUNSHINE\_DURATION\_h 1425 non-null object

# Datenaufbereitung und Datenanalyse

Das Ziel besteht darin, basierend auf einer Datenzeile das Ergebnis eines Spiels vorherzusagen. Die erstellten Datensätze weisen eine unterschiedliche Granularität auf. Sowohl die allgemeinen Vereinsdaten als auch die Transferdaten und die finale Bundesliga-Platzierung stehen pro Verein pro Saison zur Verfügung. Im Gegensatz dazu stehen die Spieltagsdaten pro Spieltag pro Saison zur Verfügung. Dieses Format stellt das Zielformat dar, da der Ausgang dieser Spiele geschätzt werden soll. Die Daten, die ausschließlich pro Saison zur Verfügung stehen, müssen demnach, um für die Schätzung des Spielausgangs verwendet werden zu können, auf die Spieltage der jeweils dazugehörigen Saison verteilt werden. Der Code dazu findet sich in [web-mining/02\_Merging.ipynb at main · lucajanas/web-mining (github.com)](https://github.com/lucajanas/web-mining/blob/main/02_Merging.ipynb)

Als erstes müssen die Vereinsnamen in dem Spieltagsdatensatz angepasst werden, da dort Kurzversionen der verschiedenen Vereinsnamen verwendet werden. Diese Kurzversionen werden auf die ausgeschriebenen Varianten gemapped, um im nächsten Schritt über einen Left-Join mit pandas jedem Spieltag die vereins- und transferbezogenen Informationen für Heim- und Auswärtsteam für die jeweilige Saison zuzuordnen. Der Join wird daher auf den Attributen Saison und Vereinsname durchgeführt.

Der erstellte Datensatz wird im Folgenden zur Visualisierung und Analyse der Spieltagsdaten verwendet. Der Code zur Erstellung der Abbildungen sowie die Abbildungen selbst sind unter [web-mining/04\_DataAnalysis.ipynb at main · lucajanas/web-mining (github.com)](https://github.com/lucajanas/web-mining/blob/main/04_DataAnalysis.ipynb) abgelegt.

Abbildung 2 zeigt die Bundesliga-Platzierung sowie den Marktwert einer Mannschaft dargestellt in einem Scatterplot.

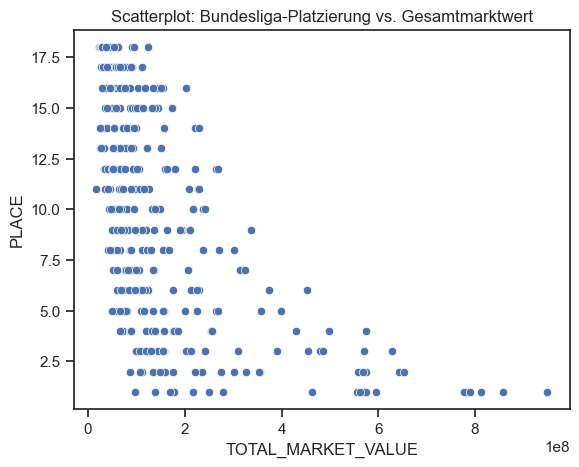


Abbildung 3 - Scatterplot: Bundesliga-Platzierung vs. Gesamtmarktwert

Abbildung 2 zeigt, wie zu vermuten war, dass Mannschaften mit einem höheren Marktwert tendenziell eine bessere Platzierung am Saison-Ende erreichen. Deswegen wird der Marktwert einer Mannschaft beim Training eines Klassifikationsmodells zur Vorhersage des Spielausgangs vermutlich ein wichtiges Feature darstellen.

Abbildung 3 zeigt die durchschnittlichen Einnahmen und Ausgaben in der Bundesliga pro Saison.

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 4 - Durchschnittliche Einnahmen und Ausgaben pro Saison

Abbildung 3 zeigt, dass die Ausgaben der Bundesliga für Spielereinkäufe üblicherweise über den Einnahmen durch Spielerverkäufe liegen, mit einigen Ausnahmen, wo die Einnahmen die Ausgaben leicht überstiegen, so z. B. im Jahr 2022. Da sich die Einnahmen und Ausgaben pro Verein zum Teil deutlich unterscheiden, werden diese Angaben ebenfalls als Input für das Klassifikationsmodell verwendet. Insgesamt steigen sowohl die Einnahmen als auch die Ausgaben von einem mittleren einstelligen Millionenbetrag im Jahr 2004 auf mittlere zweistellige Millionenbeträge bis hin zu Ausgaben von über 50 Millionen Euro im Jahr 2019. 2020 wurden deutlich niedrigere Einnahmen und Ausgaben verzeichnet, was auf die durch die Corona-Pandemie zu dem Zeitpunkt unsichere wirtschaftliche Situation vieler Vereine sowie des Gesamtprofifußballs zurückzuführen ist.

Abbildung 4 zeigt die Anzahl der verschiedenen Spielausgänge – Heimsieg, Auswärtssieg oder Unentschieden, pro Saison.

Ein Bild, das Reihe, Diagramm, Text enthält.

Automatisch generierte Beschreibung

Abbildung 5 - Anzahl der verschiedenen Spielausgänge je Saison

Abbildung 4 lässt vermuten, dass der oft vermutete Heimvorteil tatsächlich zu existieren scheint. Die Anzahl der Heimsiege ist stets größer als die Anzahl der Auswärtssiege. Einzig im Jahr 2019 sind die Anzahl der Heimsiege sowie die Anzahl der Auswärtssiege beide rund um 120, da in dieser Saison unüblich viele Auswärtssiege und unüblich wenige Heimsiege auftraten. Üblicherweise treten zwischen 120 und 150 Heimsiege auf und zwischen 80 und 100 Auswärtssiege. Die Anzahl der Unentschieden schwankt zwischen 60 und 100, wobei in einem Großteil der Saisons zwischen 60 und 80 Unentschieden auftreten. Abbildung 4 lässt vermuten, dass die Unterscheidung zwischen Heim- und Auswärtsteam einen Einfluss auf die Genauigkeit des Klassifikationsmodells haben wird. Deswegen wird diese Unterscheidung in den Features vorgenommen.

Abbildung 5 zeigt die Confusion Matrix, die entsteht, wenn die Tipprundentendenzen der Transfermarkt-Community verwendet werden, um die Spielergebnisse vorherzusagen.

Ein Bild, das Text, Screenshot, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 6 - Confusion Matrix basierend auf Tipprundentendenz

Abbildung 5 zeigt, dass Tipprundentendenzen allein keine verlässlichen Indikatoren zur Vorhersage des Spielergebnisses sind. Es werden 50,6% der Spielergebnisse richtig getippt. Während ca. 75% der Heimsiege richtig erkannt werden, werden lediglich ca. 51% der Auswärtssiege und 5% der Unentschieden richtig getippt. Abbildung 6 zeigt die Confusion Matrix, wenn ausschließlich Spiele berücksichtigt werden, bei denen die Transfermarkt-Community mit mindestens 90% auf den jeweiligen Spielausgang getippt hat. Das trifft auf 23% der ursprünglich im Datensatz enthaltenen Spiele zu. In diesen 23% sind keine Unentschieden enthalten, d. h. die Transfermarkt-Community setzt nie zu 90% oder mehr auf ein Unentschieden als Spielausgang. Das Filtern erhöht die Gesamtgenauigkeit auf circa 69%. Es werden 94,04% der Heimsiege erkannt, 64,54% der Auswärtssiege und kein einziges Unentschieden.

Ein Bild, das Text, Screenshot, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 7 - Confusion Matrix basierend auf Tipprundentendenz mit 90%-Quote

Um zu untersuchen, welchen Einfluss der Schiedsrichter der Partie auf den Spielausgang für die einzelnen Vereine hat, werden die Siegquoten der Vereine für die einzelnen Schiedsrichter ermittelt. Abbildung 7 zeigt die Siegquoten pro Schiedsrichter für den FC Bayern München. Dabei wurden ausschließlich Schiedsrichter berücksichtigt, die mehr als 15 Spiele des FC Bayern München geleitet haben. Die durchschnittliche Siegesquote des FC Bayern München liegt bei ca. 85%. Auffällig ist, dass kein einziges Spiel, das von Bastian Dankert geleitet wurde, verloren wurde. Unter Sascha Stegemann konnten nur 65% der Spiele gewonnen werden. Auch wenn die Stichproben sicherlich nicht groß genug sind, um einen eindeutigen Zusammenhang zwischen Schiedsrichter und Spielausgang herzustellen und weitere Faktoren weitaus mehr Einfluss haben werden, wird der Schiedsrichter der Partie im Klassifikationsmodell berücksichtigt.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 8 - Siegesquote des FC Bayern München pro Schiedsrichter

Abbildung 8 zeigt die Siegesquote des 1. FC Köln pro Schiedsrichter.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Abbildung 9 - Siegesquote des 1. FC Köln pro Schiedsrichter

Es wurden Schiedsrichter berücksichtigt, die mehr als zehn Partien des 1. FC Köln geleitet haben. Die durchschnittliche Siegesquote des 1. FC Köln unter diesen Schiedsrichtern liegt bei ca. 45%. Auffällig ist, dass mit 73 bzw. 75% Siegesquote unter Peter Sippel bzw. Felix Zwayer deutlich mehr Spiele gewonnen werden als unter den anderen Schiedsrichtern.

# Feature Engineering

# Modelltraining und -test

# Ergebnisanalyse

# Hypothesentests

Unsere Zielsetzung besteht darin, statistische Hypothesentests zu konzipieren, um die statistische Signifikanz der Leistung unserer Modelle im Vergleich zum reinen Zufall oder einer reinen Rategenauigkeit bei der Vorhersage von Spielergebnissen zu ermitteln. Um dies erfolgreich durchzuführen, ist es notwendig, im Voraus Informationen über die Apriori-Wahrscheinlichkeit zu sammeln, die uns Aufschluss darüber gibt, wie die Verteilung der Spielergebnisse aussieht. Konkret interessiert uns der Anteil, der angibt, wie oft die Heimmannschaft gewinnt, wie oft es zu einem Unentschieden kommt und wie oft die Heimmannschaft verliert. Dies erfordert eine eingehende Analyse der historischen Verteilung von Spielausgängen, wobei wir auf einen Datensatz von insgesamt 5814 Spielen zurückgreifen.

|  |  |  |
| --- | --- | --- |
| **Heimmannschaft gewinnt** | **Unentschieden** | **Heimmannschaft verliert** |
| 45% | 25% | 30% |

Im Rahmen dieses Projekts können wir mithilfe von Python eine gewichtete, pseudo-zufällige Auswahl der Spielergebnisse durchführen. Die Gewichtung orientiert sich an die oben erzeugte Verteilung. Anschließend vergleichen wir diese Ergebnisse mit unseren Prognosemodellen.

Für die Analyse von Unterschieden zwischen zwei Stichproben bietet sich der McNemar-Test als geeignetes Verfahren an. Dieser Test eignet sich insbesondere für binäre Ausgabewerte, wie korrekte Vorhersagen gegenüber falschen Vorhersagen.

Die Nullhypthose zu diesem Test lautet:

: Es gibt keinen Unterschied zwischen Stichprobe1 (Prognosemodell) und Stichprobe2 (zufälliges Spielergebnis)

Bevor wir den McNemar Test auswerten, ist es hilfreich vorab eine Kontingenztafel aufzustellen, die die Anzahl der richtig prognostizierten und richtig zufällig erzeugten Spielergebnisse gegenüber der abweichenden Spielergebnisvorhersagen macht, stellt. Relevant für den Test sind allerdings nur die abweichenden Ergebnisse zwischen dem Zufallsmodell und dem Prognosemodell (b und c)

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Zufälliges Raten** | |
|  |  | Richtig | Falsch |
| **Prognosemodell** | Richtig | a | b |
| Falsch | c | d |

Auf dieser Basis können wir die Teststatistik errechnen und dem kritischen Wert der -Verteilung gegenüberstellen.

Die Teststatistik für die Anwendung des McNemar-Tests ergibt sich ausgehend von der Kontingenztafel durch den Wert .

Demgegenüber wird der kritische Wert der -Verteilung mit 1 Freiheitsgrad und dem 95%-Quantil = 3.84 gestellt, wenn von einem Signifikanzniveau von 5% ausgegangen wird. Der Wert von 3.84 lässt sich von der Tabelle der -Verteilung ablesen.

Die Nullhypothese wird abgelehnt, falls und die Alternativhypothese wird angenommen, dass das Prognosemodell vom Zufallsmodell statistisch signifikant abweicht.

**Ergebnisse der Auswertung:**

**Random Forest** vs. Zufallsmodell:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Zufälliges Raten** | |
|  |  | Richtig | Falsch |
| **Prognosemodell** | Richtig | 252 | 337 |
| Falsch | 186 | 388 |

Resultat: Die Nullhypothese wird abgelehnt.

**K-Neighbor** vs. Zufallsmodell:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Zufälliges Raten** | |
|  |  | Richtig | Falsch |
| **Prognosemodell** | Richtig | 253 | 338 |
| Falsch | 185 | 387 |

Resultat: Die Nullhypothese wird abgelehnt.

**Multinominal Naive Bayes** vs. Zufallsmodell:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Zufälliges Raten** | |
|  |  | Richtig | Falsch |
| **Prognosemodell** | Richtig | 233 | 332 |
| Falsch | 205 | 393 |

Resultat: Die Nullhypothese wird abgelehnt.

**Gaussian Naive Bayes** vs. Zufallsmodell:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Zufälliges Raten** | |
|  |  | Richtig | Falsch |
| **Prognosemodell** | Richtig | 250 | 339 |
| Falsch | 188 | 386 |

Resultat: Die Nullhypothese wird abgelehnt.

**Quadratische Diskriminanzanalyse** vs. Zufallsmodell:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Zufälliges Raten** | |
|  |  | Richtig | Falsch |
| **Prognosemodell** | Richtig | 244 | 340 |
| Falsch | 194 | 385 |

Resultat: Die Nullhypothese wird abgelehnt.

In allen untersuchten Fällen wurden demnach die Nullhypothesen abgelehnt. Das heißt, die Prognosemodell unterscheiden sich statistisch signifikant von dem Zufallsmodell. Da der Test selbst keine Aussage darüber trifft, wie stark und in welche Richtung sich die Abweichungen bewegen, kann man sich an der Kontingenztafel orientieren und feststellen, dass stets das Prognosemodell mehr richtige Vorhersagen getroffen hat, während das Zufallsmodell falsch lag, als umgekehrt. Also der -Wert ist stets größer als der -Wert. Es kann daher angenommen werden, dass die hier untersuchten Prognosemodelle stets signifikant besser sind als das Zufallsmodell.