# Projektergebnis - NBA Game Prediction

Gruppe C

# Inhaltsverzeichnis

# 1. Business Understanding

## Einführung

Die Vorhersage von Sportereignissen ist sowohl für Fans als auch für Akteure im Sportwettenmarkt von großem Interesse. Insbesondere die Vorhersage von Sieg oder Niederlage eines Teams in der National Basketball Association (NBA) bietet erhebliches Potenzial für strategische Analysen und Wettoptimierung. Trotz des hohen Interesses gibt es derzeit kein Modell, das die Spielausgänge zuverlässig vorhersagen kann. Bisherige Ansätze zur Vorhersage von NBA Spielen haben oft nur eine marginale oder manchmal sogar schlechtere Vorhersagegenauigkeit als es die reine Siegwahrscheinlichkeit erzielt. Ziel dieses Projektes ist es daher, ein Modell zu entwickeln, das den Spielausgang durchschnittlich mit einer höheren Genauigkeit vorhersagen kann, als die reine Gewinnwahrscheinlichkeit.

## Projektziel

Das Ziel dieses Projekts besteht darin, ein prädiktives Modell zu entwickeln und zu trainieren, das die Spielausgänge von NBA-Spielen (Sieg oder Niederlage eines bestimmten Teams) mit einer Genauigkeit vorhersagt, die im Durchschnitt betrachtet über der Baseline von 62% liegt. Das Baseline Modell bezieht sich auf die reine Siegeswahrscheinlichkeit eines Teams zu einem bestimmten Zeitpunkt innerhalb einer Saison und wird errechnet, indem die Gesamtanzahl an Spielen durch die Anzahl der gewonnenen Spiele geteilt wird. Der angegebene Wert von 62% ergibt sich durch Messung der Genauigkeit des Modells auf den in diesem Projekt verwendeten Datensatz.

## Projektplan

Um das Ziel zu erreichen, wird das Projekt anhand des Cross-Industry Standard Process for Data Mining (CRISP-DM) durchgeführt. Hierzu wird zunächst ein geeigneter Datensatz an NBA-Spielen ausgewählt. Um ein tiefes Verständnis für die Features zu entwickeln, die den Ausgang von Spielen beeinflussen könnten, ist eine anschließende Analyse der Daten erforderlich. Dabei ist es für die spätere Verwendung des Datensatzes bei der Modellierung von großer Bedeutung, die Daten vorab sorgfältig zu bereinigen und zu transformieren. Hierbei sollten fehlende Werte behandelt, Datensätze normalisiert und abgeleitete Variablen erstellt werden. Abschließend können dann geeignete Modelle bestimmt werden. Hierbei werden klassische statistische Modelle zur Klassifizierung, aber auch neuronale Netze berücksichtigt. Die Modelle werden anschließend trainiert, indem ein Teil des Datensatzes als Trainingsdaten auf die ausgewählten Modelle angewendet wird. Zusätzlich werden die Hyperparameter optimiert und die Modellleistung anhand von Evaluierungen optimiert. Die Vorhersagegenauigkeit der Modelle wird sowohl im Vergleich zur Baseline bewertet als auch im Vergleich zu den anderen eingesetzten Modellen.

Das folgende Gantt-Diagramm bildet den Projektplan ab. Nachdem wir uns als Gruppe zusammen gefunden haben, eine Projektidee entwickelt und die Ziele für unsere Projekt definiert hatten, startete dieses Data Science Projekt. Im folgenden Bericht werden die Details und Findings zu den einzelnen Schritten beschrieben. Hier ist auffällig, dass einige Schritte sich überschneiden. Dies liegt daran, dass verschiedene Parameter agil verändert haben, um die Kombination aus Datengrundlage, Feature Selectoren, Modellen und Evaluationsverfahren versucht haben, um die eine bestmögliche Spielvorhersage in der Projektlösung treffen zu können.

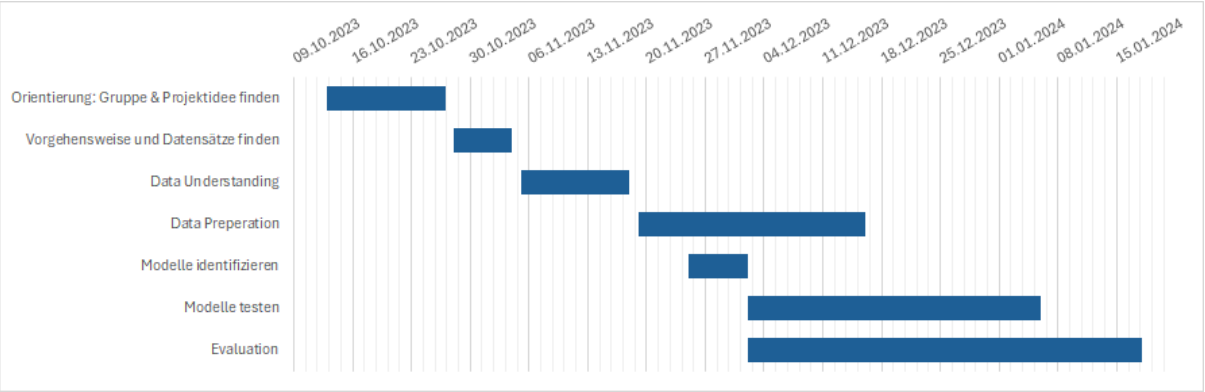



Abbildung 1: Projektplan in Gantt-Diagramm

# 2. Data Understanding

Die Webseite Basketball-Reference.com stellt umfangreiche Statistiken über NBA Spiele zur Verfügung. Hierüber war es möglich, den erforderlichen Datensatz für unser Projekt als CSV-Datei herunterzuladen. Der Datensatz umfasst 17.772 Zeilen, wobei jede zweite Zeile ein neues Spiel repräsentiert, da die Spiele einmal aus Sicht der Heimmannschaft und in der folgenden Zeile aus Sicht der Auswärtsmannschaft enthalten sind. Dies stellt 8.886 Spiele der insgesamt 30 Teams über fünf Saisons in einem Zeitraum von 2016 bis 2022 hinweg dar. Insgesamt weist der Datensatz über 150 Spalten auf, was auf eine hohe Anzahl an Features schließen lässt. Abbildung 1 zeigt die Korrelationen aller einzelnen Features des Datensatzes zueinander. Hierbei zeigt sich, welche Werte innerhalb eines NBA Spiel eine Korrelation Einfluss aufeinander haben. Um zu einem späteren Zeitpunkt die Siege vorhersagen zu können, sind jedoch insbesondere die Features von Interesse, die eine hohe Korrelation zu dem Attribut “Won” (Sieg) aufweisen. Aus nachfolgender Tabelle 1 sind diese Werte zu entnehmen, die eine höhere Korrelation als 0,4 besitzen.

| Feature | Correlated with | Correlation |
|---|---|---|
| won | ortg | 0.504125 |
| won | drtg_opp | 0.504125 |
| won | drtg_max_opp | 0.488571 |
| won | ts% | 0.474602 |
| won | efg% | 0.452853 |
| won | fg% | 0.447418 |
| won | total | 0.444858 |
| won | pts | 0.444858 |
| won | trb% | 0.406343 |
| won | trb%_opp | -0.406335 |
| won | fg%_opp | -0.446621 |
| won | pts_opp | -0.455497 |
| won | total_opp | -0.455497 |
| won | efg%_opp | -0.459670 |
| won | ts%_opp | -0.480076 |
| won | drtg_max | -0.492918 |
| won | ortg_opp | -0.509871 |
| won | drtg | -0.509871 |

Tabelle 1: Correlation to ‘won’

Ebenfalls durchgeführte Kausalitätstest wie der *ssr based F test, likelihood ratio test* oder der *parameter F test* zeigten, dass die Werte nicht nur korrelieren, sondern auch kausal sind. Dies lässt sich ebenfalls auf die Tatsache zurückführen, dass alle untersuchten Features in einem NBA Spiel auftreten und somit einen direkten Einfluss auf den Ausgang des Spiels haben.

Eine weitere wichtige Erkenntnis, die aus dem Datensatz gewonnen werden konnte, ist die Verteilung der Siege bei Heim- und Auswärtsspielen.
Wie Abbildung 2 zeigt, gewinnen Mannschaften über den gesamten Zeitraum hinweg durchschnittlich häufiger vor heimischem Publikum als zu Gast bei ihren Gegnern.

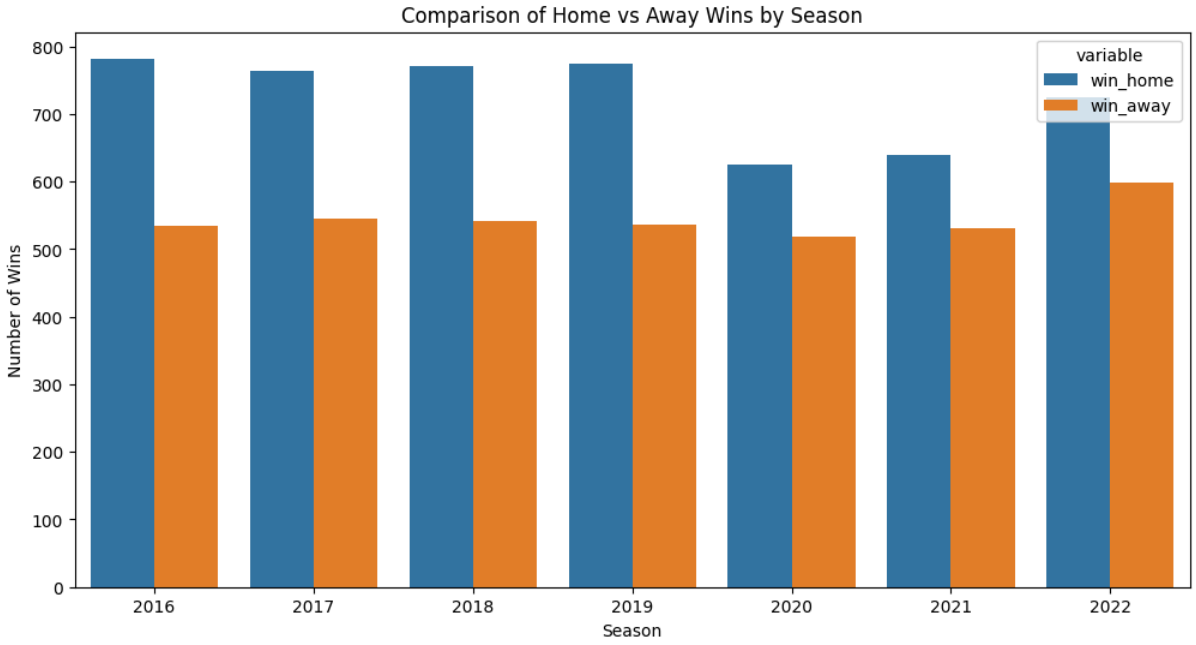



Abbildung 2: Comparison of Home vs Away Wins by Season

Ebenfalls entscheidend, um die späteren Ergebnisse der Modelle bezogen auf die Teams besser interpretieren zu können, sind die Gewinnwahrscheinlichkeiten der Teams. Hierbei werden die gespielten Spiele der Mannschaften durch deren Siege geteilt. Abbildung 3 zeigt, wie sich die Gewinnwahrscheinlichkeit der einzelnen Teams zusammensetzt.

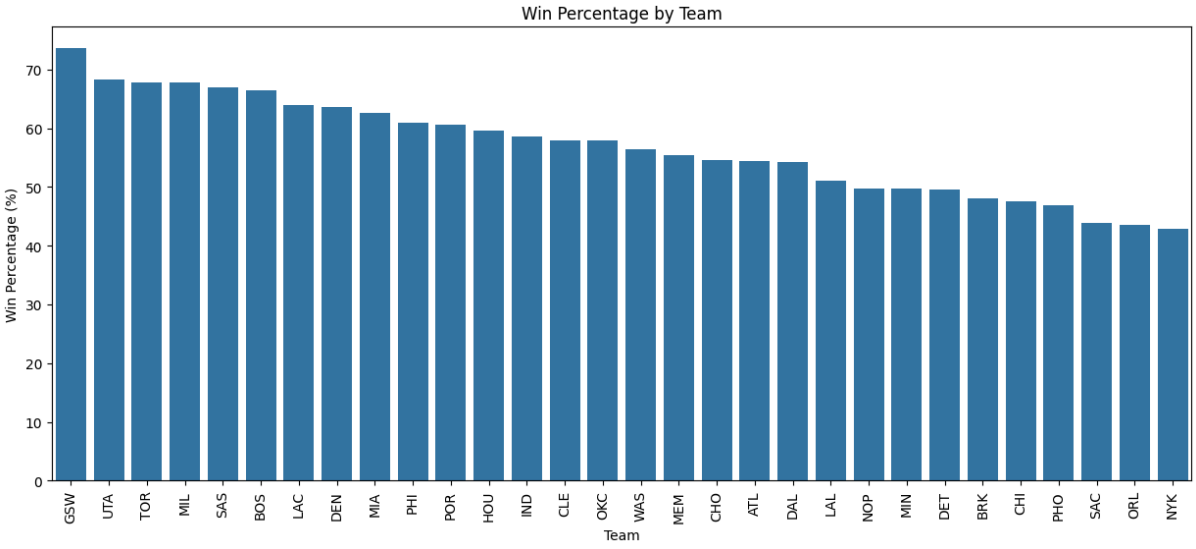



Abbildung 3: Win Percentage by Team

Wie die Grafik zeigt, haben die Golden State Warriors (GSW) die höchste durchschnittliche Gewinnwahrscheinlichkeit von 73.57% und die New York Knicks (NYK) die niedrigste mit 42.96%. Dass dieser Wert mitunter auch deutliche Schwankungen aufweisen kann, zeigt Abbildung 4. Hier ist die Gewinnerwartung der GSW in einem Monatsdurchschnitt über den gesamten Zeitraum dargestellt.

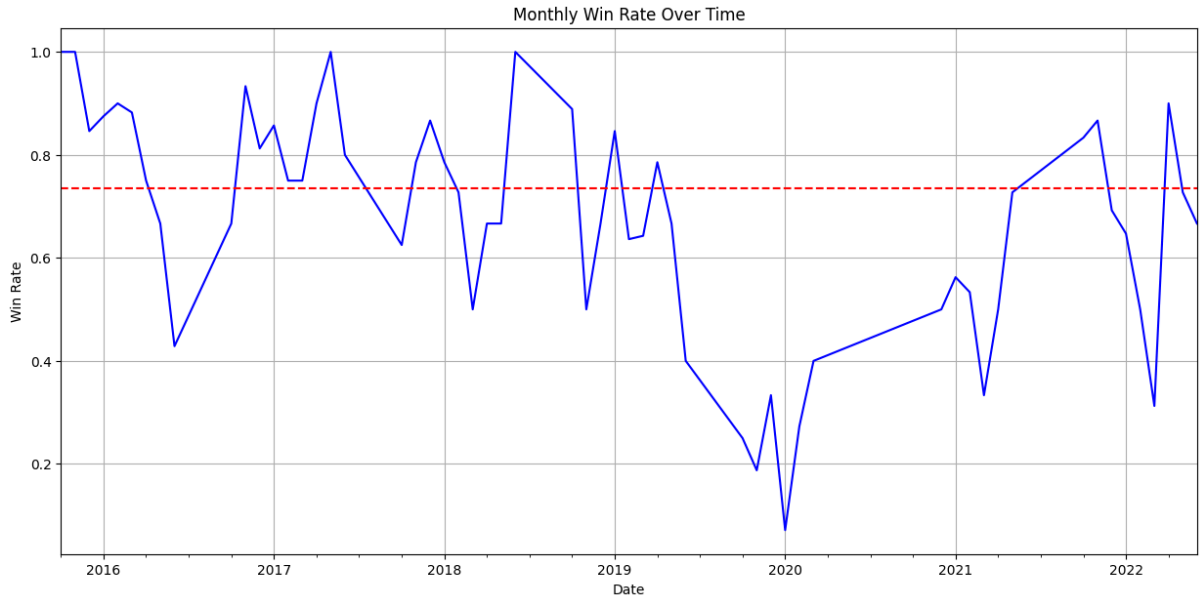



Abbildung 4: Monthly Win Rate Over Time

Diese Erkenntnisse können insbesondere bei der Auswertung der Modelle und dem Vergleich auf Teamebene eine wichtige Rolle spielen. Insgesamt ist das Verständnis der Daten daher nicht nur ein erster Schritt im Analyseprozess, sondern ein kontinuierlicher Begleiter, der die Aussagekraft der Ergebnisinterpretation maßgeblich mitbestimmt.

# 3. Data Preparation

## Create Target

Die Datenvorbereitung ist ein entscheidender Schritt im Data-Science-Prozess, der die Daten für die anschließende Analyse und Modellierung optimiert. In einem ersten Schritt gilt es, die Zielvariable zu erstellen, die die Modelle später vorhersagen sollen. Im vorliegenden Fall bezieht sich dies auf den Wert des Features ‘*won’,* eines jeweiligen Teams in dessen nächstem Spiel.

## Clean Data

Des Weiteren gilt es alle Werte zu bereinigen, um Fehler bei der Anwendung der Modelle im späteren Verlauf weitestgehend zu minimieren. Zu den zu bereinigenden Daten zählen insbesondere fehlende Werte (*missing values),* fehlerhafte Daten (*noisy data*) und ausreißer Werte (*outliers*). Gründe für fehlende Werte sind im vorliegenden Fall insbesondere Spiele zu Beginn einer jeden Saison, in der es keine Werte zur vorherigen Partie gibt. Fehlerhafte Werte oder ausreißer Werte konnten in unserem Datensatz nicht festgestellt werden. Bereinigt wurde der Datensatz, indem alle Spalten, die keine numerischen Werte (*NaN*) aufweisen, entfernt wurden.

## Normalize Data

Um sicherzustellen, dass alle Eingabevariablen in einem vergleichbaren Maßstab vorliegen, werden die Daten normalisiert. Dies ist besonders wichtig, wenn verschiedene Metriken stark unterschiedliche Wertebereiche aufweisen. Hierfür wurde die Min-Max-Normalisierung angewandt, um alle numerischen Features auf einen Bereich zwischen 0 und 1 zu skalieren. Diese Technik stellt sicher, dass Variablen mit großen Werten keinen unangemessenen Einfluss auf Variablen mit kleinen Werten ausüben und die Empfindlichkeit gegenüber der Skala der Daten zu reduzieren.

## Integrate Data

Anhand des bis dato vorliegenden Datensatzes, wäre es den Modellen lediglich möglich, auf Grundlage des letzten Spiels einer Mannschaft, deren nächstes vorherzusagen. Um die Anzahl an möglichen Features und die Informationen die den Modellen zur Verfügung stehen zu erhöhen, wurden neue Features erstellt. Zum einen wurden die Statistiken des letzten Spiels des folgenden Gegners aggregiert. Hierdurch wurde die Anzahl der Features bereits nahezu verdoppelt. Zum anderen wurde von jedem Feature der jeweilige Durchschnittswert der letzten 20 Partien (für das jeweilige Team) gebildet und dem Datensatz zusätzlich angefügt. Hierdurch erhalten die Modelle die Möglichkeit, auch Trendwerte mit in ihre Berechnung einzubeziehen. Zuvor war die Vorhersage lediglich auf Grundlage des zuletzt absolvierten Spiels möglich. Diese Anreicherung der Daten erhöhte die Anzahl an Features auf 422.

## Select Data

Da die Verwendung einer solch hohen Anzahl an Features der Leistung und Ergebnissen der Modelle nicht zuträglich ist, muss eine Vorauswahl der für die Modelle wichtigsten Features erfolgen. Ein gängiges Verfahren ist die Messung der Korrelation zwischen einem beliebigen Merkmal und dem Zielwert. Merkmale, die am stärksten mit dem Zielwert korrelieren, werden ausgewählt, wobei alle anderen verworfen werden. Tabelle 2 (Anhang) zeigt diese Features, die durch eine höhere Korrelation als 0,1 zum Zielwert für die weitere Anwendung selektiert wurden. Die niedrigen Korrelationswerte sowie Kausalität-Tests bestätigen jedoch, dass hierbei kein starker Zusammenhang oder gar eine Kausalität aufzufinden ist. Dies liegt der Tatsache zugrunde, dass diese Features Trendwerte der letzten 20 Spiele darstellen. Somit lassen sich aus den einzelnen Durchschnittswerten aus der Vergangenheit nicht zuverlässig die Spielausgänge in der Zukunft vorhersagen. Eine weitere Erkenntnis, die sich aus dem zweiten Korrelationstest ergibt, ist, dass kein Wert des letzten Spiels einen wirklichen Einfluss auf den Ausgang des nächsten Spiels hat. Durch diese sorgfältige Vorbereitung der Daten wurde der Datensatz für die Modellierung vorbereitet. Die Schritte sollen gewährleisten, dass die Modelle auf bereinigten und relevanten sowie vorselektierten Daten basieren.

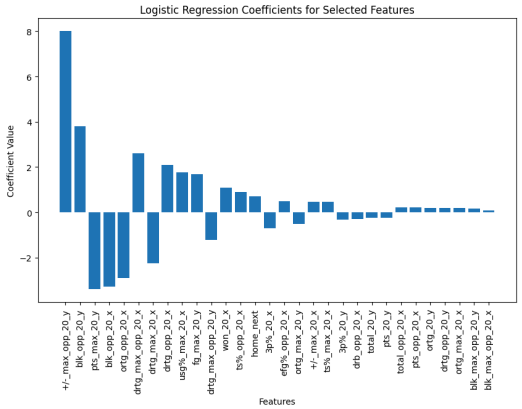
# 4. Modelling

## Pre Modellauswahl

Zu Beginn der Modellierung muss eine Entscheidung zwischen einem Regressionsmodell und einem Klassifikationsmodell getroffen werden. Die Wahl hängt vom gewünschten Ergebnis ab. In diesem Fall soll vorhergesagt werden, ob eine Mannschaft gewinnt oder verliert. Daher sind Klassifikationsmodelle die sinnvollere Wahl. Bei dieser Entscheidung ist zu überlegen, ob es für die ursprüngliche Fragestellung nicht besser wäre, die Gewinnwahrscheinlichkeit (in %) vorherzusagen. In diesem Anwendungsfall wurden Regressionsmodelle gewählt. Bei einer Vorhersage der Gewinnwahrscheinlichkeit wäre es auch schwieriger, die Genauigkeit der Modelle zu messen. In diesem Fall müsste ein Bewertungssystem entwickelt werden, um zu beurteilen, wie nahe die Vorhersage der Gewinnwahrscheinlichkeit im Durchschnitt an den tatsächlichen Ergebnissen liegt. Dies könnte z.B. so aussehen, dass ein Modell mehr Accuracy-Punkte erhält, wenn es einen Sieg zu 70% richtig vorhersagt, als wenn es einen Sieg zu 60% richtig vorhersagt. Das Gleiche würde für falsche Vorhersagen gelten. Darüber hinaus haben wir bei der Literaturrecherche festgestellt, dass in Arbeiten zu diesem Thema ähnliche Beobachtungen gemacht wurden, wenn versucht wurde, Regressionsmodelle auf eine solche Fragestellung anzuwenden. Delen et al. (2012) untersuchen, welche Modelle am besten geeignet sind, um den Ausgang von NCAA-Bowl-Spielen vorherzusagen. Dazu entwickeln sie verschiedene Modelle, sowohl Klassifikations- als auch Regressionsmodelle, um deren Vorhersagekraft zu testen. Dabei stellen sie fest, dass die Modelle, die auf Klassifikationsmodellen basieren, eine höhere Genauigkeit aufweisen. Mit den Daten von acht Jahren dieser Liga konnten sie eine Vorhersagegenauigkeit von 85% erreichen. Dies hat uns in unserer Entscheidung bestärkt, uns im weiteren Vorgehen auf die Klassifikationsmodelle zu beschränken.

Des Weiteren stellt sich die Frage, ob die Klassifikation mit supervised learning oder unsupervised learning erfolgen soll. Mit unserem Datensatz lassen sich die Modelle sehr einfach mit Supervised Learning trainieren, da sie gelabelt sind. Für jedes Spiel ist bekannt, ob eine Mannschaft gewinnt oder verliert. Damit ist auch diese Frage beantwortet und wir wenden Supervised Learning an.

Für die Feature Selection gibt es verschiedene Ansätze, die sich unterschiedlich gut auf die Modelle anwenden lassen. Die besten Ergebnisse lieferten der Sequencial Feature Selector, Random Forest Estimator (RFE) und Select KBest (abhängig von den jeweiligen Modellen). Eine unterschiedliche Gewichtung durch die verschiedenen Selektoren ergibt sich aus der Funktion der jeweiligen Modelle. Im Folgenden sind die Feature-Gewichtung für die beiden Modelle Ridge Classifier und Logistic Classifier dargestellt. Dabei wird deutlich, welche unterschiedlichen Merkmale für die beiden Modelle als relevant erachtet werden.



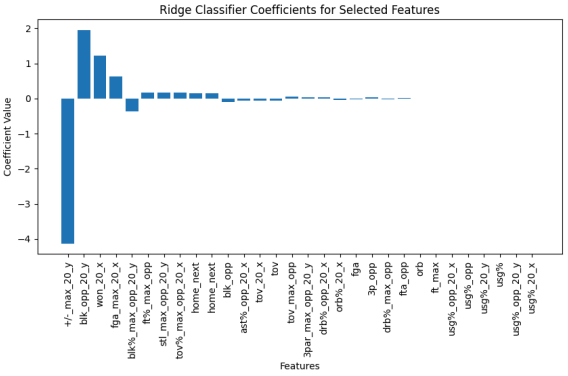




Abbildung 5: Feature Selection LR

Abbildung 4: Feature Selection RC

Der Sequential Feature Selector kann in den Sequential Forward Selector (SFS) und den Sequential Backward Selector (SBS) unterteilt werden (Abb. 6). Beide führen das Training des Modells sequentiell und mit unterschiedlichen Features durch. Beim Sequential Forward Selector beginnt der Selector mit einem Feature, wählt das beste aus, fügt das nächste Feature hinzu, durchläuft alle Features des Datensatzes und findet das beste. Dieser Vorgang wird solange wiederholt, bis die gewünschten n Features im Modell verwendet wurden und somit das nach diesem Feature Selector bestmögliche Modell erzeugt wurde. Beim Sequential Backward Selector läuft dieser Prozess rückwärts ab. Der Selektor trainiert das Modell mit allen Merkmalen, entfernt in jeder Iteration ein Features, testet, welches Merkmal das Modell am wenigsten verschlechtert, entfernt dieses Features und geht zur nächsten Iteration über, um das nächste Features zu entfernen.

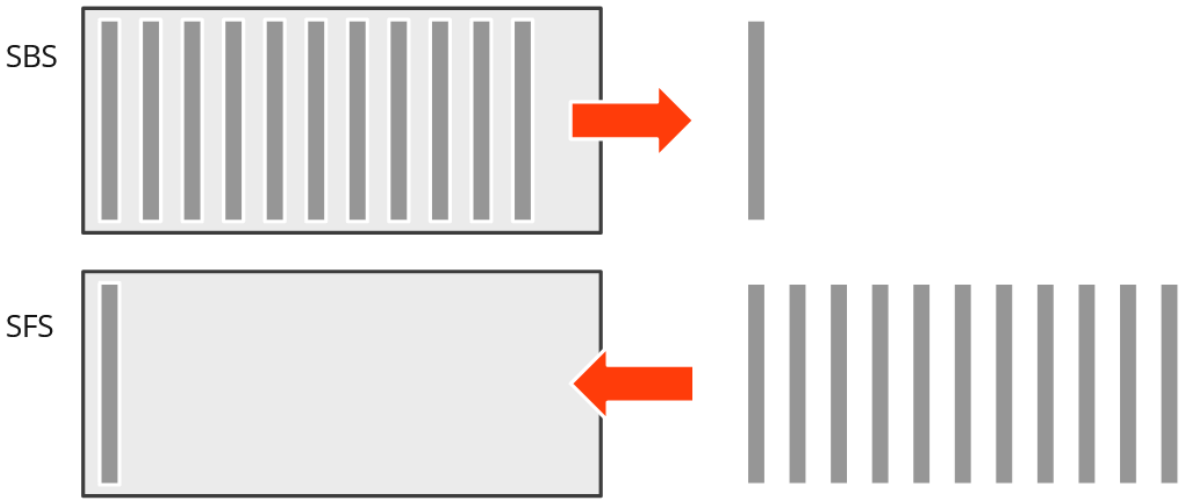



Abbildung 6: Funktionsweise Sequential Feature Selector

# Modelltests

Für die Modelltest wird sich in dieser Problemstellung, wie bereits in “Pre Modellauswahl” erwähnt, auf die Klassifikationsmodelle beschränkt. Hierfür wurden:

- Logistic Regression
- Support Vector Machine
- Random Forest Classifier
- Gradient Boosting
- Ridge Classifier
- Neural Network

verwendet. Diese wurden durch eine Literaturrecherche als geeignet für die vorliegende Fragestellung bewertet und daher auf die vorliegende Aufgabenstellung angewendet. Auf die Ergebnisse der Anwendung dieser Modelle wird im Abschnitt “Evaluation” näher eingegangen.

## Hyper Parameter Tuning

Hyper Parameter Tuning spielt eine entscheidende Rolle bei der Anpassung der Modelle an den Datensatz, um die bestmögliche Vorhersage zu erzielen. Für das Hyper Parameter Tuning von Modellen werden random search und grid search verwendet. Diese beschreiben zwei Verfahren, um mit geringem Aufwand die beste Kombination von Hyperparametern in einem Modell zu finden. Bei einigen Modellen hatten wir jedoch das Problem, dass die Feature Selection für bestimmte Modelle sehr lange dauerte. Dies hat uns dazu veranlasst, die Funktionsweise der einzelnen Hyperparameter zu untersuchen, um Entscheidungen über wichtige und unwichtige Hyperparameter zu treffen. So konnte eine Vorauswahl getroffen werden, um nicht alle Hyperparameter in den Modellen ausprobieren zu müssen. Um ein konkreteres Verständnis dafür zu entwickeln, wie sich die Anpassung bestimmter Hyperparameter auf die Leistung eines Modells auswirkt, haben wir diese teilweise separat getestet, um einen Zusammenhang zwischen Hyperparameter und der Genauigkeit unseres Modells herzustellen. Abbildung 7 zeigt diese Vorgehensweise. Zu Beginn wird der Feature Selector ausgewählt. Danach wird das Modell zunächst mit den vordefinierten Hyperparametern ausgeführt und ausgewertet. Die dabei ermittelte Genauigkeit bildet die Grundlage für die folgenden Durchläufe. Im nächsten Durchlauf werden dann einzelne Parameter verändert, um deren Auswirkung auf das Modell zu ermitteln. Diese Vorgehensweise wird in vereinfachter Form von den bereits erwähnten Methoden Random Search und Grid Search übernommen.

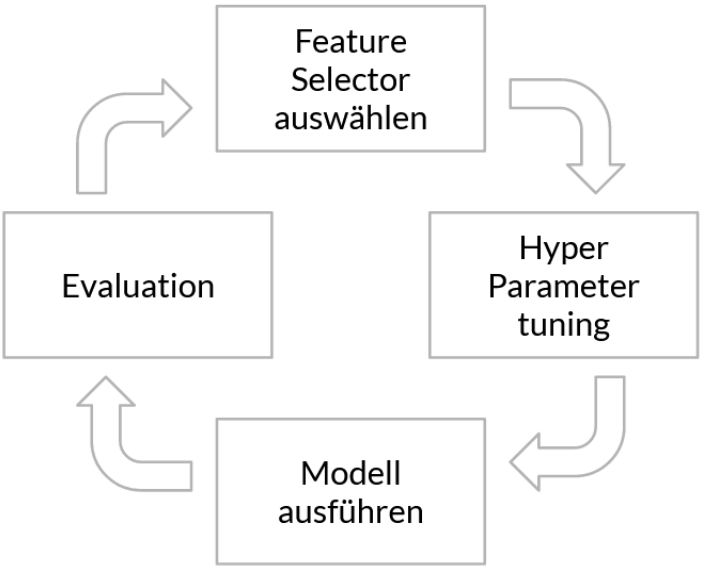



Abbildung 7: Ablauf für Modell-Erprobung und Optimierung

## Hyper Parameter Einstellungen für die Modelle

Nachfolgend sind die verwendeten Modelle und die besten Einstellungen der Hyperparameter aufgeführt. Die hier aufgeführten Werte für die Hyperparameter wurden mit dem oben beschriebenen Verfahren ermittelt. Es wurden entweder Random Search, Grid Search oder manuelle Approximationen verwendet.

**Logistic Regression**

```
LogisticRegression(C=1 , class_weight= 'balanced', penalty='l2',
solver='lbfgs')
```

**Support Vector Machine**

```
SVC(C=50, kernel='linear', gamma=0.01)
```

**Random Forest Classifier**

```
RandomForestClassifier(n_estimators=100, random_state=42,
bootstrap=False, max_features='sqrt', min_samples_leaf=4,
class_weight='balanced')
```

**Gradient Boosting**

```
GradientBoostingClassifier(n_estimators=100, learning_rate=0.2,
max_depth=2, min_samples_split=0.2, min_samples_leaf=0.01,
subsample=1.0)
```

**Ridge Classifier**

```
rc = RidgeClassifier(alpha=0.6398038489891585, copy_X=True,
fit_intercept=True, max_iter=100, positive=False, random_state=None,
solver='cholesky', tol=0.0005404909866914394)
```

**Neural Network**

```
model.fit(train[predictors], train["target"], epochs=50, batch_size=32,
verbose=0, validation_data=(val[predictors], val["target"]),
callbacks=[early_stopping])
```

# 5. Evaluation

Ziel unserer Untersuchung war es, die Effektivität verschiedener Klassifikationsmodelle anhand von Schlüsselmetriken zu messen. Zu diesem Zweck haben wir ein Evaluations-Framework mit Schlüsselmetriken entwickelt, die wir anhand von Literaturrecherchen als besonders relevant identifiziert haben: Genauigkeit, Präzision, Recall, F1-Score und AUC-ROC. Diese Metriken bieten ein breites Spektrum an Perspektiven zur Bewertung der Modellleistung.

- Genauigkeit bietet eine allgemeine Einschätzung der Modellleistung durch Messung des Anteils der insgesamt korrekten Vorhersagen.
- Präzision und Recall bewerten die Qualität der positiven Vorhersagen bzw. die Fähigkeit des Modells, alle relevanten Fälle zu erfassen.
- Der F1-Score ist ein Maß für die Balance zwischen Präzision und Recall.
- AUC-ROC bewertet die Unterscheidungsfähigkeit des Modells zwischen den Klassen.

Obwohl alle diese Metriken wichtig sind, haben wir uns auf die Genauigkeit als Hauptfokus konzentriert, da unsere Daten in Bezug auf Gewinne/Verluste perfekt ausbalanciert waren. Die Genauigkeit alleine ist daher ein aussagekräftiger Indikator für die Stärke des Modells.

Die Auswertung des Evaluations-Frameworks ergab folgende Ergebnisse:

| Modellname | Genauigkeit | Präzision | Recall | F1-Score | AUC-ROC |
|---|---|---|---|---|---|
| RidgeClassifier | 0.64 | 0.63 | 0.65 | 0.64 | 0.63 |
| GradientBoosting Classifier | 0.63 | 0.63 | 0.64 | 0.63 | 0.63 |
| Logistic Regression | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 |
| Random Forest | 0.64 | 0.63 | 0.64 | 0.64 | 0.64 |
| Neural Network | 0.66 | 0.66 | 0.66 | 0.66 | 0.66 |
| Support Vector Machine | 0.65 | 0.65 | 0.65 | 0.65 | 0.65 |

Abbildung 8: Ergebnis des Evaluation Framework

Das neuronale Netzwerk erwies sich als das leistungsstärkste Modell mit einer einheitlichen Leistung von 0,66 über alle Metriken hinweg. Wir konnten die Genauigkeit des Modells von 53% auf 66% steigern, indem wir Feature Engineering, Hyperparameter-Optimierung und Fine-Tuning durchgeführt haben.

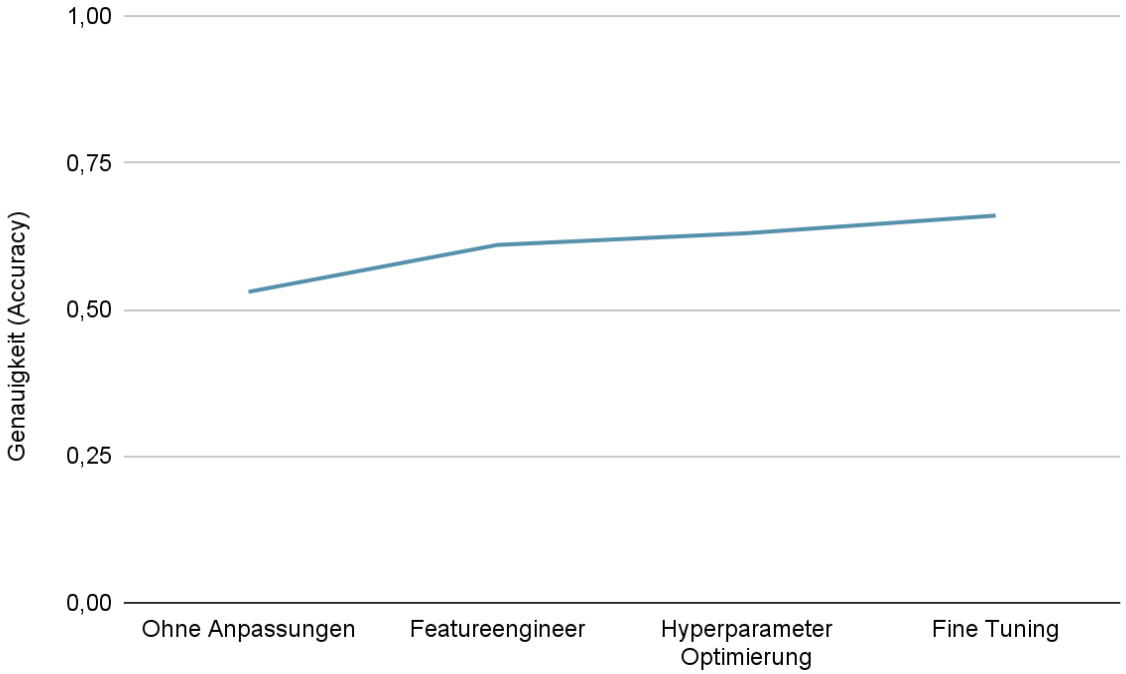



Abbildung 9: Entwicklung der Genauigkeit des Neuronalen Netzes

Die Analyse der Vorhersagegenauigkeit für bestimmte NBA-Teams ergab, dass unser Modell genauere Vorhersagen für Teams wie die Cleveland Cavaliers (CLE) und die Philadelphia 76ers (PHI) ermöglichte (70%+). Dies ist für uns im Kontext von Sportwetten von besonderem Interesse.

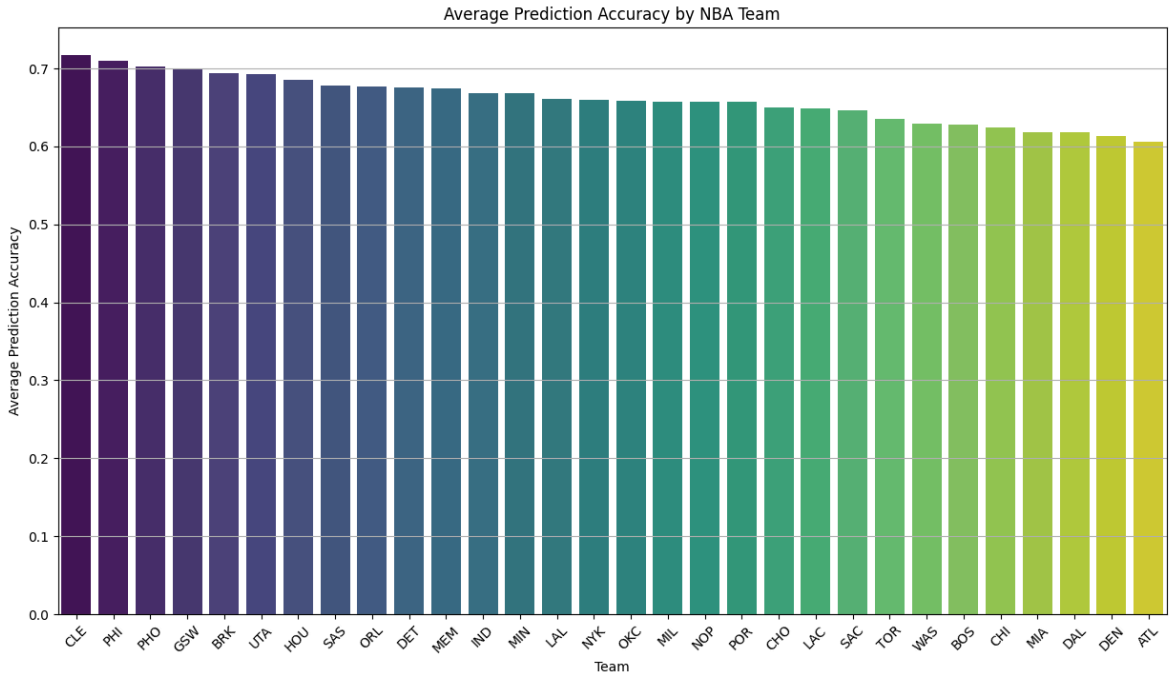



Abbildung 10: Modellgenauigkeit nach NBA-Teams

## Evaluation im Kontext von Sportwetten

Um die Anwendbarkeit unserer prädiktiven Modelle im Bereich der Sportwetten zu evaluieren, haben wir uns auf die Metrik 'Yield' konzentriert. Diese Kennzahl wird im Sportwetten-Umfeld häufig verwendet und gibt den prozentualen Gewinn oder Verlust aus den getätigten Wetten an. Sie ist entscheidend, um die Profitabilität oder Effizienz einer Wettstrategie zu messen.

Für diese tiefere Analyse haben wir relevante Daten zu Wettquoten beschafft. Mithilfe eines mit Python entwickelten Scrapers haben wir Quoten für mehr als 10.000 NBA-Spiele von 2015 bis 2022 gesammelt. Anschließend haben wir diese Wettquoten mit den Vorhersagen unseres Modells abgeglichen, um die potenzielle Profitabilität unserer Vorhersagen im Kontext realer Wettmärkte zu bewerten.

Die durchgeführte Yield-Analyse ergab einen Yield von -20,10% über alle analysierten Spiele, was einen Nettoverlust bedeutet. Auch die Analyse der einzelnen Teams ergab fast ausschließlich einen negativen Yield. Lediglich das Team der Toronto Raptors erzielte einen positiven Yield von +2,01%.

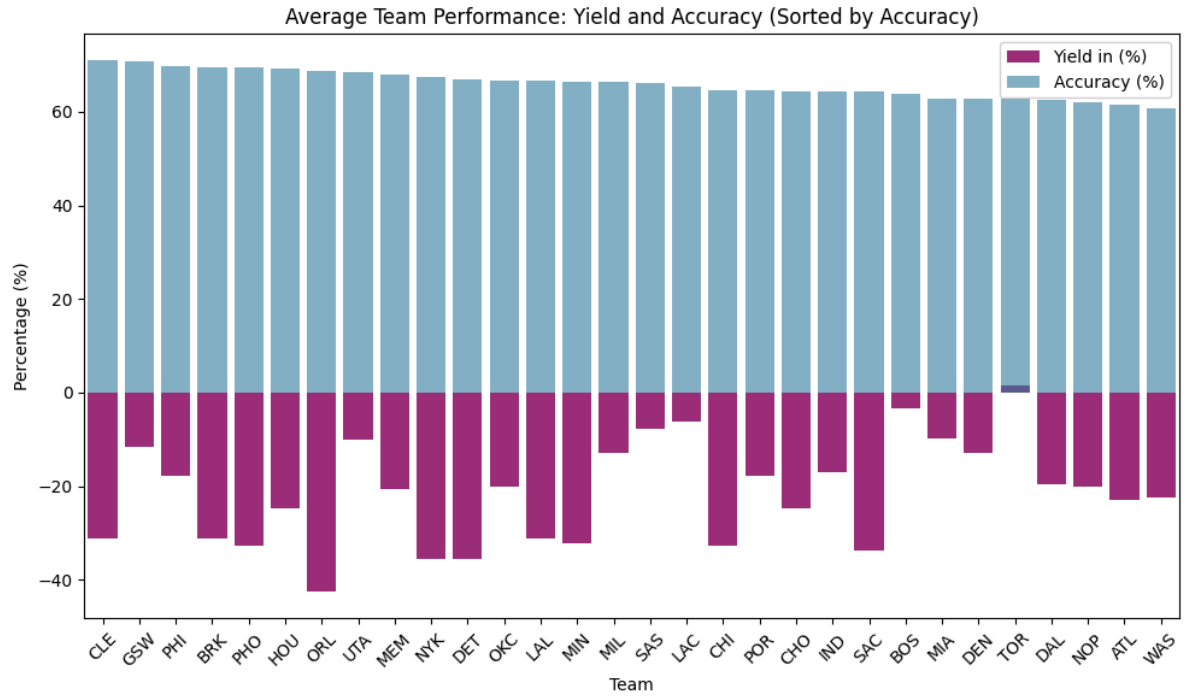



Abbildung 11: Genauigkeit und Yield-Analyse pro NBA-Team

Die Übersicht zeigt deutlich, dass Teams wie die Cleveland Cavaliers (CLE) trotz hoher Vorhersagegenauigkeit in Bezug auf den Yield weit hinten landen. Dies verdeutlicht, wie ungünstige Wettquoten die Profitabilität beeinträchtigen können, selbst wenn die Vorhersagen präziser als bei anderen Teams sind.

## Schlussfolgerungen

Die Evaluation zeigt, dass durch umfangreiche Modellierung und intensive Anpassungen das Baseline-Modell übertroffen wurde. Insbesondere das Feature Engineering und weitere Anpassungen steigerten die Genauigkeit auf 66%. Die Einbeziehung von Quoten-Daten verdeutlicht jedoch, dass Wetten auf Basis unserer Modelle mit Vorsicht zu genießen sind. Trotzdem sind wir mit dem Fortschritt zufrieden, da kein vergleichbares Modell online eine höhere Genauigkeit als 70% erzielen konnte.

# 6. Further Research

Ein vielversprechender Ansatz für zukünftige Forschung besteht in der Erweiterung unserer Modellierung durch die Integration von Spielerdaten und der Schaffung einer Spieler-Elo, die die Team-Elo beeinflusst. Diese zusätzlichen Informationen könnten als neues Feature in unser Modell eingebunden werden, um die Gesamtgenauigkeit auf 70% zu erhöhen. Eine Erweiterung des Modells hätte den Vorteil, besser auf unvorhergesehene Ereignisse wie Krankheit oder Verletzungen von Spielern reagieren zu können, die derzeit nicht berücksichtigt werden.
Regressionsmodelle könnten genutzt werden, um die Gewinnwahrscheinlichkeit in Prozent vorherzusagen. Um die Genauigkeit eines solchen Modells zu bewerten, könnte man messen, wie nahe die Vorhersage der Gewinnwahrscheinlichkeit im Durchschnitt an den tatsächlichen Ergebnissen liegt. Ein Modell würde mehr Accuracy-Punkte erhalten, wenn es einen Sieg zu 70 % richtig vorhersagt, als wenn es einen Sieg zu 60 % richtig vorhersagt. Das Gleiche gilt auch für falsche Vorhersagen.

# Quellenverzeichnis

Delen, D., Cogdell, D. & Kasap, N. (2012). A comparative analysis of data mining methods in predicting NCAA bowl outcomes. *International Journal Of Forecasting*, *28*(2), 543–552. https://doi.org/10.1016/j.ijforecast.2011.05.002

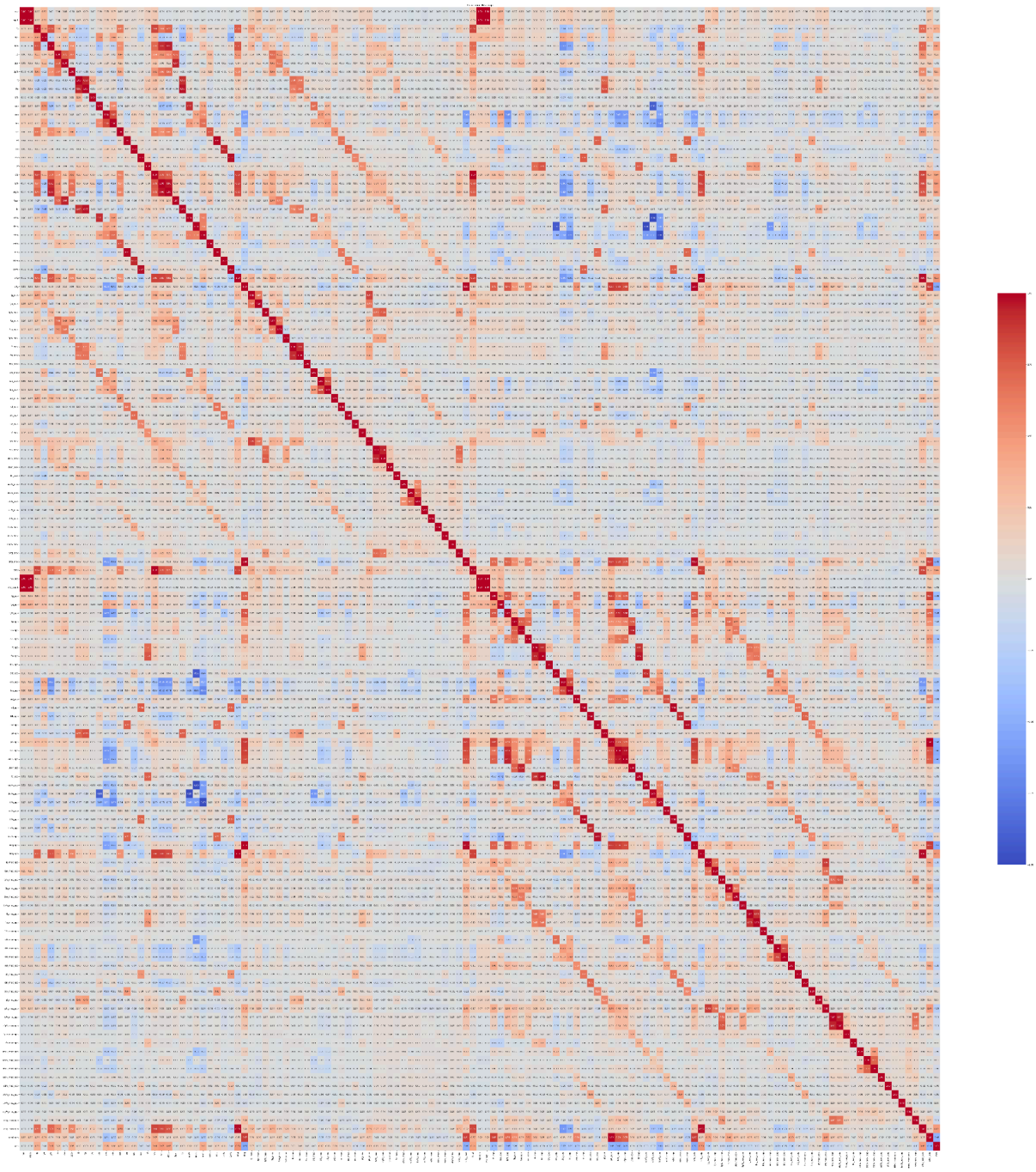
# Glossar

mp: minutes played
fg: field goals
fga: field goals attempted
fg%: field goal percentage
3p: 3-point field goals
3pa: 3-point field goals attempted
3p%: 3-point field goals percentage
ft: free throws
fta: free throws attempted
ft%: free throws percentage
orb: offensive rebounds
drb: defensive rebounds
trb: total rebounds
ast: assists
stl: steals
blk: blocks
tov: turnovers
pf: personal fouls
pts: points
+/-: plus minus (points scored minus points allowed)
ts%: true shooting percentage (a measure of shooting efficiency that takes into account 2-point field goals, 3-point field goals, and free throws)
efg%: effective field goal percentage (adjusts the fact that a 3-point field goal is worth more than a 2-point field goal)
3par: 3-point attempt rate (percentage of FG attempts from 3-point range)
ftr: free throw attempt rate (number of FT attempts per FG attempt)
orb%: offensive rebound percentage (an estimate of the percentage of available offensive rebounds a player grabbed while he was on the floor)
drb%: defensive rebound percentage (an estimate of the percentage of available defensive rebounds a player grabbed while he was on the floor)
trb%: total rebound percentage (an estimate of the percentage of available rebounds a player grabbed while he was on the floor)
ast%: assist percentage (an estimate of the percentage of teammate field goals a player assisted while he was on the floor)
stl%: steal percentage (an estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor)
blk%: block percentage (an estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor)
tov%: turnover percentage (an estimate of turnovers per 100 plays)
usg%: usage percentage (an estimate of the percentage of team plays used by a player while he was on the floor)

ortg: offensive rating (an estimate of points produced (players) or scored (teams) per 100 possessions)
drtg: defensive rating (an estimate of points allowed per 100 possessions)
mp_max: most minutes played by a player
fg_max: most field goals by a player
fga_max: most field goals attempted by a player
fg%_max: highest field goal percentage by a player
3p_max: most 3-point field goals by a player
3pa_max: most 3-point field goals attempted by a player
3p%_max: highest 3-point field goal percentage by a player
ft_max: most free throws by a player
fta_max: most free throws attempted by a player
ft%_max: highest free throw percentage by a player
orb_max: most offensive rebounds by a player
drb_max: most defensive rebounds by a player
trb_max: most total rebounds by a player
ast_max: most assists by a player
stl_max: most steals by a player
blk_max: most blocks by a player
tov_max: most turnovers by a player
pf_max: most personal fouls by a player
pts_max: most points by a player
+/-_max: highest plus minus by a player
ts%_max: highest true shooting percentage by a player
efg%_max: highest effective field goal percentage by a player
3par_max: highest 3-point attempt rate by a player
ftr_max: highest free throw attempt rate by a player
orb%_max: highest offensive rebound percentage by a player
drb%_max: highest defensive rebound percentage by a player
trb%_max: highest total rebound percentage by a player
ast%_max: highest assist percentage by a player
stl%_max: highest steal percentage by a player
blk%_max: highest block percentage by a player
tov%_max: highest turnover percentage by a player
usg%_max: highest usage percentage by a player
ortg_max: highest offensive rating by a player
drtg_max: highest defensive rating by a player
team: team name
total: total score of the home team
home: True if team is home team

# Anhang

## Abbildung 1: Correlation Heatmap



# Tabelle 2: Correlation to Target

| Feature | Correlated with | Correlation |
|---|---|---|
| target | won_20_x | 0.22776323864853598 |
| target | +/-_max_20_x | 0.22618526165457542 |
| target | +/-_max_opp_20_y | 0.198982975916889 |
| target | drtg_opp_20_x | 0.17610586060920091 |
| target | ortg_20_x | 0.17610586060920091 |
| target | drtg_max_opp_20_x | 0.17078272539150408 |
| target | ts%_20_x | 0.1667187046359275 |
| target | efg%_20_x | 0.16221645622284567 |
| target | fg%_20_x | 0.15420145611293837 |
| target | trb%_20_x | 0.14254640730710394 |
| target | home_next | 0.13886630530842425 |
| target | blk_opp_20_y | 0.13420504515003923 |
| target | 3p%_20_x | 0.12867962366636232 |
| target | ortg_max_20_x | 0.1264993410054526 |
| target | total_20_x | 0.12363391777761489 |
| target | pts_20_x | 0.12363391777761489 |
| target | pts_max_20_x | 0.12361496684324302 |
| target | drtg_20_y | 0.12192356374191819 |
| target | ortg_opp_20_y | 0.12192356374191819 |
| target | trb%_opp_20_y | 0.12175897840617965 |
| target | fg%_opp_20_y | 0.1198500578701756 |
| target | drb_opp_20_y | 0.11671860390792148 |
| target | efg%_max_20_x | 0.11587550835281304 |
| target | ts%_max_20_x | 0.11532559020114329 |
| target | drtg_max_20_y | 0.1150227360814194 |
| target | ts%_opp_20_y | 0.11456047640564014 |
| target | trb_opp_20_y | 0.1122908637421468 |
| target | blk_max_opp_20_y | 0.11089098519949747 |
| target | fg_20_x | 0.10997837533262467 |
| target | fg_max_20_x | 0.1048561007709209 |
| target | efg%_opp_20_y | 0.1044022819920562 |
| target | pts_opp_20_y | 0.10418023947501037 |
| target | total_opp_20_y | 0.10418023947501037 |
| target | drb_20_x | 0.10266743336302941 |
| target | usg%_max_20_x | 0.1010685273789391 |
| target | trb%_max_20_x | 0.10030487255351334 |
| target | fg_opp_20_x | -0.10193806866182949 |
| target | blk%_opp_20_x | -0.1023612591179989 |
| target | total_20_y | -0.10294007602237362 |
| target | pts_20_y | -0.10294007602237362 |
| target | fg_max_20_y | -0.10756174840493245 |
| target | pts_opp_20_x | -0.10960774761128532 |
| target | total_opp_20_x | -0.10960774761128532 |
| target | blk_max_opp_20_x | -0.10967965377044779 |
| target | 3p%_20_y | -0.11588328204854013 |
| target | pts_max_20_y | -0.11779254429127878 |
| target | trb_opp_20_x | -0.12028371170648619 |
| target | trb%_20_y | -0.12178234587567506 |
| target | efg%_opp_20_x | -0.12201053834517946 |
| target | drb_opp_20_x | -0.12414342601434504 |
| target | drtg_max_20_x | -0.12605005634941252 |
| target | ts%_opp_20_x | -0.1306472723032217 |
| target | drtg_20_x | -0.1357579128193597 |
| target | ortg_opp_20_x | -0.1357579128193597 |
| target | fg%_opp_20_x | -0.13652442540426077 |
| target | blk_opp_20_x | -0.13773035882296053 |
| target | fg%_20_y | -0.1398529014053923 |
| target | trb%_opp_20_x | -0.14251864071769021 |
| target | drtg_max_opp_20_y | -0.14865591397085293 |
| target | efg%_20_y | -0.15025315810185996 |
| target | ts%_20_y | -0.15051930754777487 |
| target | drtg_opp_20_y | -0.15305466412531643 |
| target | ortg_20_y | -0.15305466412531643 |
| target | +/-_max_20_y | -0.19435101422791337 |
| target | won_20_y | -0.20683192775191805 |
| target | +/-_max_opp_20_x | -0.22005924057771986 |