Ludwig-Maximilians-Universität München
Institut für Statistik

„Defense wins championships“ ?
-
Analyse und Prognose der National Basketball Association

Nico Hahn
Betreuer: Prof. Dr. Christian Heumann

18. Juli 2018

Zusammenfassung

Das Ziel der vorliegenden Bachelorarbeit ist es, Veränderungen in der Spielweise des professionellen Basketballs in den USA zu identifizieren, das Abschneiden von Mannschaften zu prognostizieren und den Ausgang von einzelnen Spielen korrekt vorherzusagen. Hierfür werden Daten von der National Basketball Association und der Website `www.basketball-reference.com` verwendet. Für die Regressionen werden, neben linearen und logistischen Modellen, regularisierte Modelle und random forest Modelle verwendet. Des weiteren wird eine Hauptkomponentenregression durchgeführt. Aufgrund der hohen Anzahl an Kovariablen (133) in dem Datensatz ist eine Variablenselektion notwendig, damit das resultierende Modell nicht zu komplex und interpretierbar ist. Dies wird durch Lasso und elastic net Regression erreicht. Der random forest ist eine Erweiterung des Prinzips der Entscheidungsbäume. Bei dieser Methode werden randomisiert verschiedene Bäume „gepflanzt“ und sich dann für den besten entschieden. Dabei handelt es sich um eine effiziente Methode für die Evaluierung großer Datensätze. Des weiteren können mit einem random forest die wichtigsten Variablen anhand verschiedener Messwerte festgemacht werden, wodurch er auch für die Variablenselektion geeignet ist. Eine genauere Erläuterung der Funktionsweise der einzelnen Verfahren erfolgt in den entsprechenden Kapiteln.
Alle Graphiken und Modelle werden mit Hilfe der statistischen Software R (R Development Core Team, 2008) erstellt. Für die Erstellung der meisten Graphiken in dieser Arbeit wird das Paket **plotly** (Sievert, 2018) verwendet. Die penalisierten Modelle werden mit Hilfe des Pakets **glmnet** (Friedman, Hastie & Tibshirani, 2010) berechnet, für den random forest wird das gleichnamigen Pakets **randomForest** (Liaw & Wiener, 2002) verwendet. Die Regressionsmodelle werden mit Hilfe der Pakete **caret** (from Jed Wing et al., 2018) und **mlr** (Bischl et al., 2016) optimiert. Des weiteren wird für die Erstellung mancher Datensätze der Webcrawler aus dem Paket **rvest** (Wickham, 2016) verwendet und für die Datenmanipulation kommt das Paket **tidyverse** (Wickham, 2017) zum Einsatz. Das beste Modell für die Vorhersage einer komplette Saison ist der random forest.
Für die Prognose der Spielausgänge wird mit einem logistischen Modell 70% der Spielausgänge in den letzten fünf Saisons korrekt prognostiziert, mit Vorhersagegenaugikeiten von bis zu 71.4% in den einzelnen Saisons.
Mit einer Hauptkomponentenregression wird in der Saison 2015-16, nach nur einem Zehntel der Saison, 67.7% der restlichen Spiele korrekt prognostiziert. Insgesamt werden mit dieser Methode 63.8% der Spiele in den letzten drei Saisons korrekt simuliert.
Zusätzlich zu der Arbeit wird mit Hilfe der Pakete **shiny** (Chang, Cheng, Allaire, Xie & McPherson, 2017) und **shinydashboard** (Chang & Borges Ribeiro, 2018) eine interaktive Applikation entwickelt, mit der zusätzliche Graphiken ausgegeben werden können und individuelle Prognosen von Spielen oder einer Saison getätigt werden können.

Gliederung

1 Einführung

In den meisten Sportarten diskutieren Fans leidenschaftlich darüber, wer die beste Mannschaft oder das beste Team ist. Sei es Lionel Messi gegen Cristiano Ronaldo oder FC Barcelona gegen Real Madrid, Fans auf beiden Seiten argumentieren oft mit Statistiken, um zu zeigen, dass ihre Mannschaft oder ihr Spieler besser als die gegnerische Mannschaft beziehungsweise der gegnerische Spieler ist. Auch im Basketball gibt es diese Art von Diskussionen, von der Frage des besten Spielers aller Zeiten (LeBron James gegen Michael Jordan), bis zur Frage des besten Teams. Bei letzterer Frage geht es auch häufig darum, was eine Mannschaft gut macht. Ein altes Sprichwort lautet „Offense wins games...defense wins championships“. Das Zitat ist von dem Football Trainer Bear Bryant, fällt aber auch immer wieder im Zusammenhang mit der National Basketball Association. Besonders in den 1960er Jahren traf dieses Motto zu, als die Boston Celtics acht Meisterschaften in Folge holen konnten. Das Team wurde angeführt von dem Spieler, der in vielen Augen der beste Verteidiger aller Zeiten ist, Bill Russell. Wie viele andere Sportarten, entwickelte sich auch Basketball weiter, in erster Linie durch die Einführung der Dreipunktelinie. Seit dem neuen Jahrtausend wird ein erhöhtes Augenmerk auf die statistische Analyse im Sport gelegt, zum Beispiel durch die Entwicklung statistischer Kennzahlen, mit denen die Kontribution einzelner Spieler zum Erfolg der Mannschaft gemessen werden kann. Durch das größere Analysepotential kann genauer gesagt werden, was wirklich zu dem Erfolg einer Mannschaft beiträgt. Ein weiteres Gebiet, in dem statistische Modellierung auf Sport trifft, sind Sportwetten. Dabei geht es um die Entwicklung von Quoten, die bestimmen, wie viel Geld mit Wetten auf den richtigen Sieger gewonnen werden kann.
Diese Arbeit beschäftigt sich mit der Frage danach, was eine Mannschaft heutzutage gut macht und was für die korrekte Prognostizierung einzelner Spiele von Relevanz ist. Nach einem kurzen geschichtlichen Überblick der National Basketball Association in Kapitel 2 und der Beschaffung der Daten in Kapitel 3 widmet sich Kapitel 4 einer deskriptiven Analyse der Veränderung des Spielstils. Dabei wird vor allem auf den Einfluss des Spieltempos eingegangen, auf die Art der Würfe von Spielern und Mannschaft und es wird begutachtet, von welcher Position die besten Spieler resultieren.
Kapitel 5 beschäftigt sich zunächst in mit der Frage nach den wichtigsten Einflussgrößen für den Erfolg einer Mannschaft. Dabei geht es vorrangig darum, was von Bedeutung ist, damit eine Mannschaft viele Siege holt. Dafür wird eine Regressionsanalyse, basierend auf den letzten fünf Saisons, durchgeführt. Neben der klassischen linearen Regression werden auch Modelle mit Hilfe der Lasso Regression und der elastic net Regression berechnet. Für diese Modelle wird jeweils auf Kreuzvalidierung zurückgegriffen. Ein abschließendes Modell stellt ein random forest für alle Daten der letzten fünf Jahre dar.
Im darauffolgenden Unterkapitel 5.2 werden Modelle für die Vorhersage einzelner Spiele berechnet. Dabei werden die realen Daten vor jedem Spieltag genutzt, um den Sieger der Begegnung zu prognostizieren. Mit Hilfe einer bivariaten linearen Regression und einer logistischen Regression werden entsprechende Modelle für die letzten fünf Saisons aufgestellt. Die Resultate der jeweiligen Modelle werden daraufhin mit den realen Ergebnissen verglichen, um den Anteil der korrekt prognostizierten Spiele zu erhalten.

Im abschließenden Unterkapitel 5.3 geht es um die Simulation der restlichen Saison nach den circa ersten zehn Spielen. Wie bereits im letzten Kapitel werden die prognostizierten Spielausgänge mit den realen verglichen.
In dem letzten Kapitel der Arbeit, Kapitel 6, wird die in R entwickelte Shiny App vorgestellt, mit der Statistiken graphisch dargestellt und Prognosen einzelner Spiele oder kompletter Saisons berechnet werden können.
Abschließend werden die gewonnen Erkenntnisse diskutiert. Im Anhang befinden sich zusätzliche Graphiken, Tabellen und die verwendeten R Pakete.

2 Geschichtlicher Überblick

Heutzutage ist die NBA zusammen mit der *Major League Baseball* (MLB), *National Football League* (NFL) und der *National Hockey League* (NHL) Teil der sogenannten *„Big Four“* in den Vereinigten Staaten von Amerika. Dadurch hat die Liga ebenfalls eine hohe mediale Aufmerksamkeit in den USA, was sich vor allem in Spielstärke der NBA im Vergleich zu anderen Ligen bemerkbar macht. Sie hat das Monopol an Superstars und nur wenige Teams außerhalb des Landes hätten eine Chance gegen das schlechteste NBA-Team. Insbesondere in den frühen Jahren der Sportart ergab sich jedoch ein anderes Bild.

2.1 Die ersten Ligen

Die erste bekannte Basketball Liga war die „National Basketball League“, die 1898 gegründet wurde. Die Liga setzte sich aus sechs verschiedenen Mannschaften aus Philadelphia und New Jersey zusammen. Mit fünf Jahren hatte die Liga jedoch eine kurze Lebensdauer. Andere Ligen, die in diesem Zeitraum gegründet wurden, zeichneten sich durch ihre Kurzlebigkeit und Lokalisierung aus. Es dauert fast 40 Jahre bis eine Liga gegründet wurde, die sich über zehn Jahre halten konnte. (hoopedia.com, o. J.)

2.2 National Basketball League

1935 wurde die „Midwest Basketball Conference“ gegründet, nach zwei Jahren wurde sie jedoch in „National Basketball League“ umbenannt. Die Liga gewann rapide an Popularität, obwohl vor allem in kleineren Städten und Stadien gespielt wurde. Mit George Mikan spielte ab 1946 auch der für viele erste Superstar in der Liga. Trotzdem wurde die Liga 1949 durch den Zusammenschluss mit der BAA aufgelöst. Somit war die NBL einer der beiden Vorläufer der heutigen NBA. (hoopedia.com, o. J.)

2.3 Basketball Association of America

Der zweite Vorläufer der NBA war die Basketball Association of America. Die 1946 gegründete Liga war in größeren Städten vertreten als die NBL und die Spiele fanden auch in größeren Arenen statt, wie zum Beispiel im Madison Square Garden in New York. Dadurch gelang es der BAA Spieler wie George Mikan oder auch diverse Teams aus der NBL für sich zu gewinnen. 1949 schloss sich die BAA mit der NBL zusammen, um gemeinsam die NBA zu gründen. (history.com, 2009b)

2.4 National Basketball Association vor dem Zusammenschluss

Nach ihrer Gründung bestand die NBA aus 17 Teams, doch innerhalb der ersten Jahre wurde die Anzahl der Teams aufgrund von schwindendem Interesse der Fans mehr als halbiert. So bestand die NBA 1954 lediglich aus acht Mannschaften. Durch die Kreierung der 24-Sekunden Wurfuhr im selbigen Jahr wurde das Spieltempo deutlich erhöht und Fans kamen zurück. Dadurch konnte die Liga in andere Städte expandieren und stieg zur primären Basketballliga in den USA auf. Von 1959 bis 1966 etablierten sich die Boston Celtics als eines der dominantesten Teams aller Zeiten, als sie acht Meisterschaften in Folge gewinnen konnten. Ab 1967 bekam die NBA

jedoch Konkurrenz durch eine zweite Liga, die American Basketball Association. (history.com, 2009b)

2.5 American Basketball Association

Die ABA wurde 1967 gegründet und beinhaltete elf Teams. Sie grenzte sich klar zur NBA ab durch eine 30-Sekunden Wurfuhr, rot-weiß-blaue Basketbälle, auffällige Frisuren, Trash-Talk und eine höhere Schmerzgrenze für Fouls ab, wodurch es öfters zu Rudelbildungen auf dem Feld kam. Die größte Errungenschaft der Liga war jedoch die Einführung des 3-Punkte Wurfs. Obwohl einige Stars in der ABA spielten, konnte sich die Liga aufgrund fehlender Fernsehverträge nicht als durabeler Konkurrent zur NBA etablieren. 1976 verblieben noch neun Teams in der Liga und am Ende der Saison entschied man sich dazu, sich mit der National Basketball Association zusammenzuschließen. (history.com, 2009a)

2.6 National Basketball Association nach dem Zusammenschluss

Nach dem Zusammenschluss bestand die NBA aus 22 Mannschaften, davon vier Teams, die aus der ABA verblieben. Die restlichen Spieler der ABA konnten von den Teams verpflichtet werden. 1980 wurde der 3-Punkte Wurf inkorporiert, jedoch versuchten zu Beginn wenige Spieler diese Würfe. Die 80er Jahre wurden vor allem von den Boston Celtics und den Los Angeles Lakers dominiert, den beiden erfolgreichsten Franchises aller Zeiten. 1984 debütierte Michael Jordan, der von vielen als bester Spieler aller Zeiten angesehen wird. Er prägte vor allem die 90er Jahre, als er mit den Chicago Bulls sechs Titel in acht Jahren holte. Zu Beginn des neuen Jahrtausends wurde die Liga vor allem von neuen Superstars wie Kobe Bryant und LeBron James dominiert. (basketball reference.com, 2018)
Über den Verlauf erzielte die NBA einen größeren Umsatz, zuletzt 7.37 Milliarden U.S. Dollar in der Saison 2016/17. (statista.com, 2018)

3 Beschaffung der Daten

Bevor die Analyse beginnt, wird noch kurz vorgestellt, woher die Daten kommen. Der Großteil der Daten kommt von der Website `www.basketball-reference.com`. Die Daten werden teils per Hand heruntergeladen und teils per Webcrawler. So werden für alle Team basierten Statistiken manuell csv-Dateien erstellt, beispielsweise von folgender Website:

`https://www.basketball-reference.com/leagues/NBA_2018.html`

Das Problem bei dem Webcrawler ist, dass dieser nicht alle Tabellen einer Website herunterlädt, sondern nur die Erste. Der Webcrawler wird eingesetzt, um die Statistiken der einzelnen Spieler zu erhalten. Eine Website aus dieser Kategorie ist zum Beispiel:

`https://www.basketball-reference.com/leagues/NBA_2018_advanced.html`

Des weiteren kommt der Crawler bei der shiny App zum Einsatz, darauf wird aber später eingegangen. Zusätzlich zu der Website wird noch die offizielle Statistik API der NBA verwendet. Auf dieser können zum Beispiel Wurfdaten oder Spieler IDs gefunden werden. Diese sind vor allem für die Wurfcharts von Bedeutung. Auch hier wird die genaue Methodik in Kapitel 6 vorgestellt.

4 Veränderung des Spielstils

Seit dem Entstehen der BAA 1946 wurden immer wieder neue Regeln eingeführt, um das Spiel attraktiver oder auch fairer zu gestalten. Die beiden bekanntesten Regeln sind die Einführung der Wurfuhr und des Dreipunktewurf. In dem nachfolgenden Kapitel soll dargestellt werden, wie sich verschiedene Regeländerungen auf das Spiel ausgewirkt haben.

4.1 Die Veränderung des Spieltempos

Bevor die Wurfuhr 1954 eingeführt wurde, konnte der Ballbesitz durch Freiwürfe, Körbe, Steals oder Blocks wechseln. Das Resultat davon war, dass das führende Team sich Zeit nahm, einen Korb zu werfen. Somit waren frühe Spiele vor allem von einer langsamen Pace geprägt. Nach der Einführung ging die Pace rapide nach oben und erreichte in der Saison 1960-61 ihren Höhepunkt.

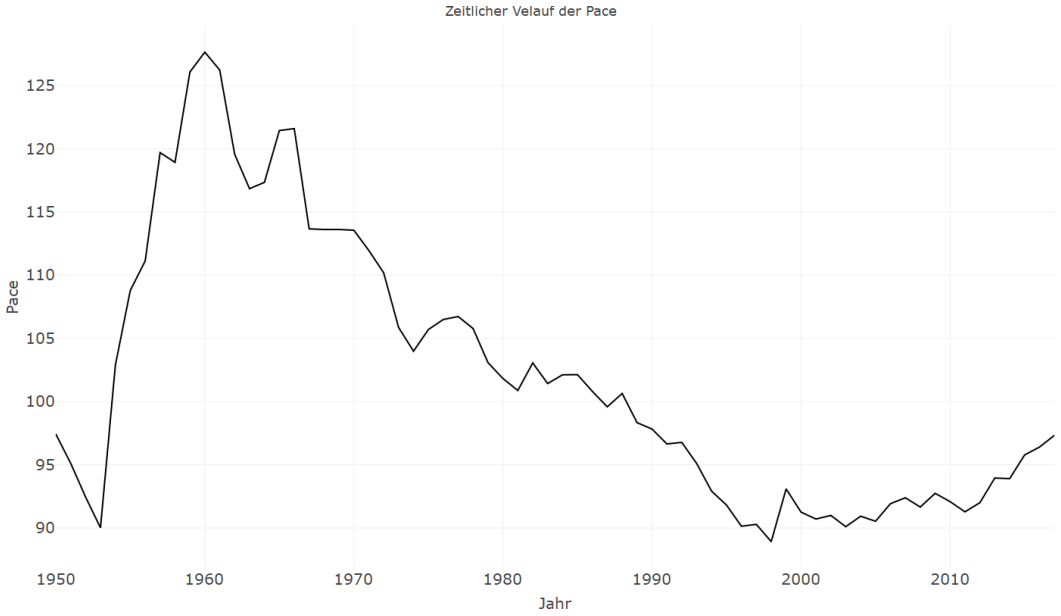


Abbildung 1: Verlauf der Pace

Das hohe Tempo in den 60ern Jahren ist vor allem auf zwei beziehungsweise drei Teams zurückzuführen: Die Boston Celtics und die San Francisco/Philadelphia Warriors. Die Teams um die beiden Star-Center Bill Russell (Boston) und Wilt Chamberlain (San Francisco/Philadelphia) konnten dank der überdurchschnittlichen Athletik der beiden Spieler mehr Offensivrebounds verzeichnen. Dementsprechend erhöhten sie die Anzahl der Ballbesitze (Possessions) und neuen Wurfchancen.

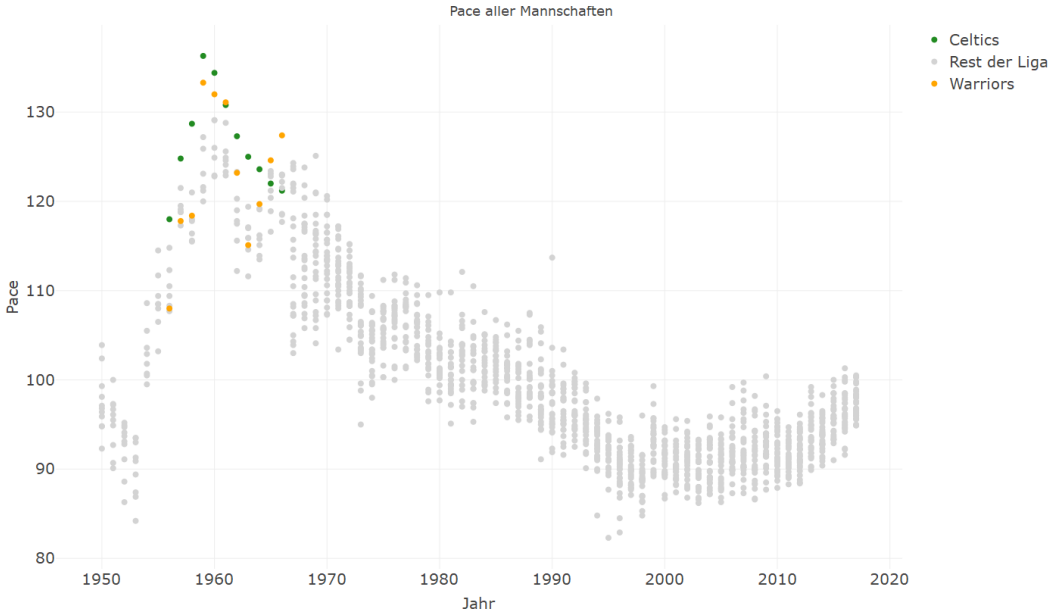


Abbildung 2: Die hohe Pace der Celtics und Warriors

Nach zehn Meisterschaften innerhalb von elf Saisons für die Boston Celtics folgten in den 70er Jahren acht verschiedene Meister. Die Pace fiel drastisch ab, während die Wurfquoten aus dem Feld weiter anstiegen. Durch das langsamer gewordene Spiel und die höhere Trefferquote fiel auch die Anzahl der Rebounds weiter ab, ein Trend der 1968 begann.

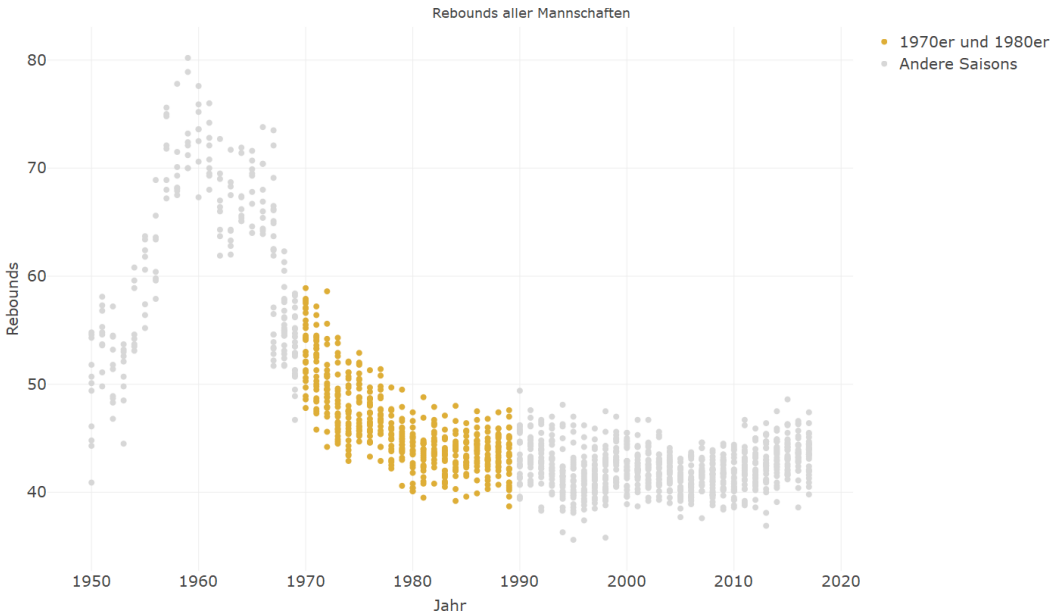


Abbildung 3: Sinkende Anzahl der Rebounds

Einer der Hauptgründe für das Abfallen der Pace nach dem Ende der Boston Dynastie ist das Coaching. Es wurde versucht gute, offene Würfe zu kreieren, anstatt schnelle

Würfe zu nehmen, bei denen um den Offensivrebound gekämpft werden muss. 1971 wurden Offensiv- und Defensivrebounds offiziell in den Boxscore aufgenommen. Seit diesem Zeitpunkt holten Teams prozentual gesehen weniger Offensivrebounds. 1971 noch über 34%, heute noch knapp über 22%.

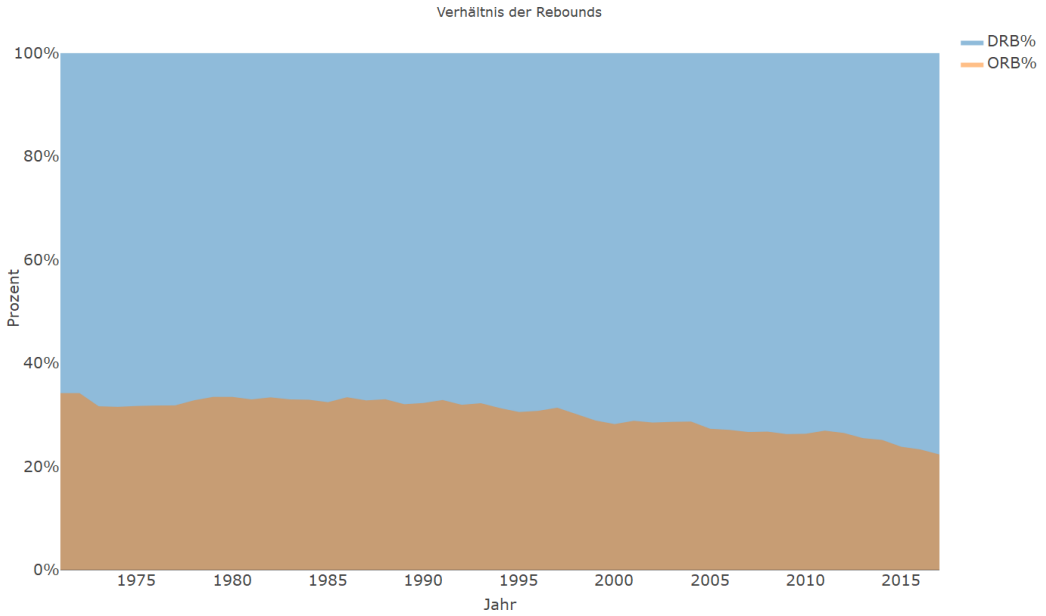


Abbildung 4: Teams holen weniger Offensivrebounds

Bemerkbar machte sich das zum Beispiel in den Assist Zahlen, die nach oben gingen. Gleichzeitig stiegen auch die Anzahl der Assists pro Field Goal bemerkbar an, was ein Indiz für besseres Teamplay ist. (Graphiken siehe Anhang)
Das langsamere Tempo macht sich ebenfalls bei den Punkten pro Spiel bemerkbar. In den 60ern wurden pro Spiel noch 114.5 Punkte erzielt, ehe in den 70ern und den 80ern jeweils 108.7 beziehungsweise 109 eher mit den 90ern (99.9) und 2000ern (97.2) die beiden Dekaden mit den wenigsten Punkten pro Spiel folgten.

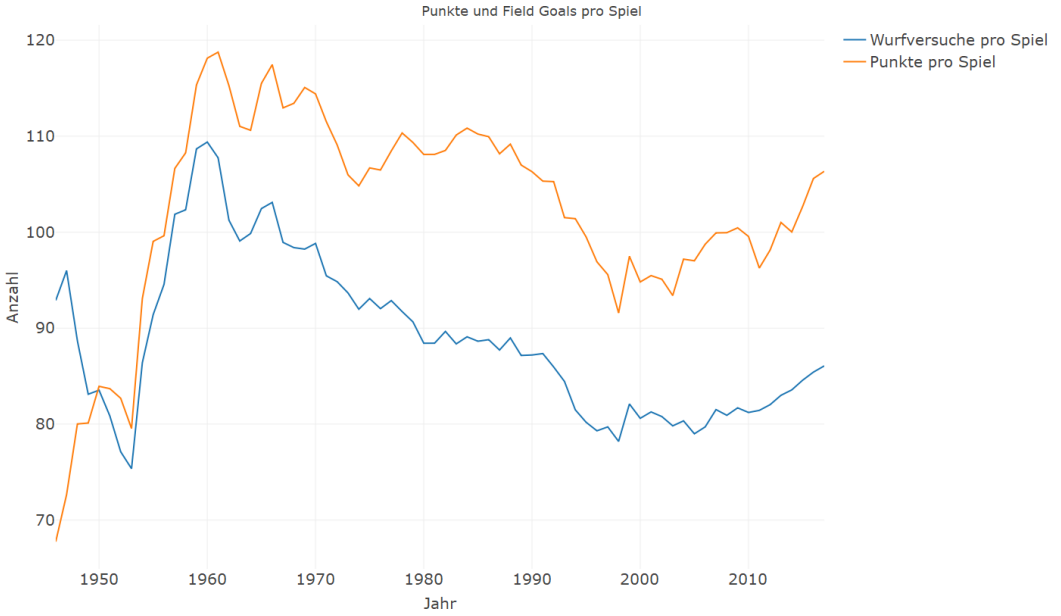


Abbildung 5: Sinkende Anzahl an Punkten und Wurfversuche pro Spiel

Durch das gesunkene Spieltempo und diszipliniertere Spiel sanken auch die Anzahl der Turnover pro Spiel und des Verhältnis der Assists zu den Turnover verbesserte sich. Da Turnover erst ab 1967 aufgezeichnet wurden, gibt es keine Daten für die Jahre davor. (Graphiken siehe Anhang)
Ab 1973 wurden Steals und Blocks im Boxscore erfasst. Die Anzahl der Steals pro Spiel ist seit 1976 von 9.6 auf 7.7 heute gefallen. Blocks pro Spiel erreichten 1982 mit 5.6 ihren Höhepunkt, heute werden im Schnitt 4.8 Würfe pro Spiel geblockt.

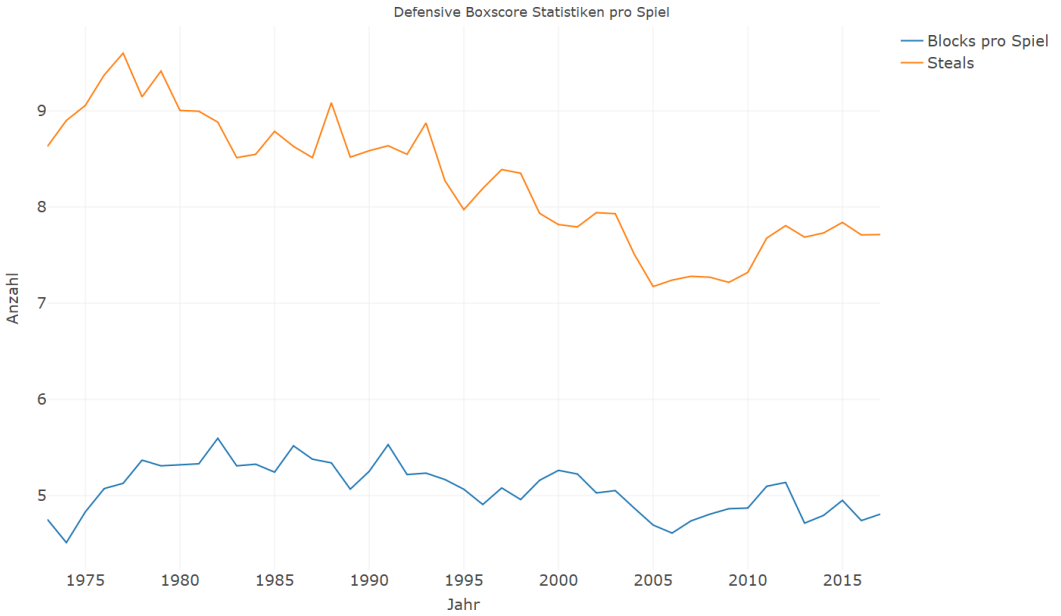


Abbildung 6: Sinkende Anzahl an Steals und Blocks pro Spiel

4.2 Veränderung der Wurfquoten

Eine gewaltige Änderung im Basketball war die Einführung der Dreipunktelinie. In der ABA wurde diese bereits zu Beginn verwendet, in der NBA erst ab 1979. Der Einfluss der Dreipunktelinie kann zum Beispiel an der Anzahl an Freiwürfen pro Spiel verdeutlicht werden. Beispielsweise betrug das Verhältnis Freiwurf/Feldwurf 1967 0.37. Heute beträgt es 0.25. Dadurch dass mehr Würfe weiter weg vom Korb genommen werden, sinkt die Anzahl der Fouls, da beim Zug zum Korb oder korbnahen Würfen die Foulwahrscheinlichkeit steigt. Es ist nicht überraschend, dass die Freiwurfrate seit Einführung des Dreipunktewurfs gefallen ist, während die Dreipunktewurfrate gestiegen ist.

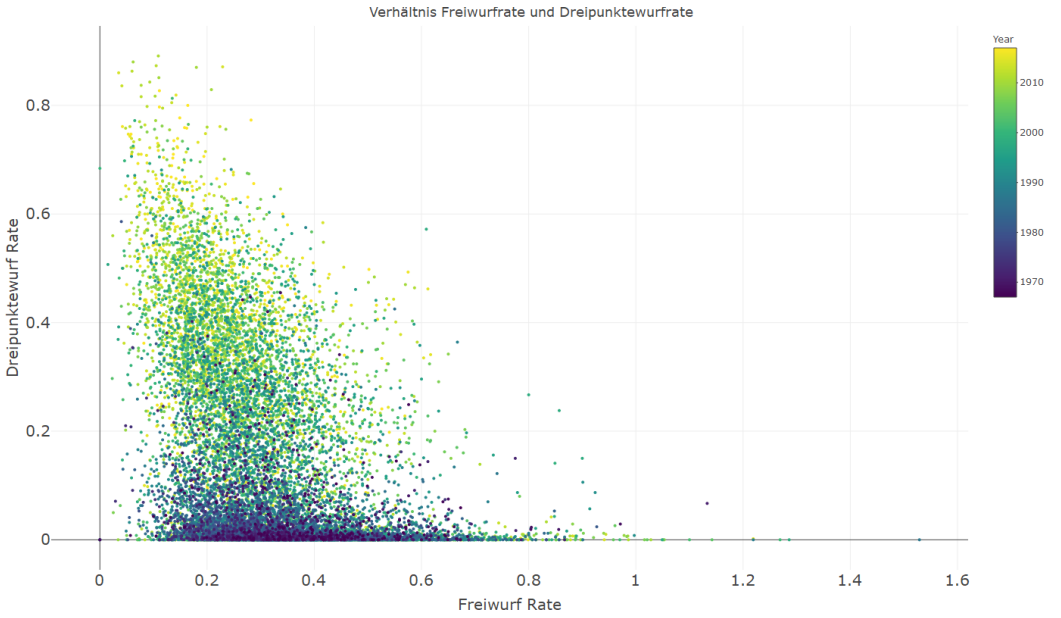


Abbildung 7: Freiwurf- und Dreipunkterate aller Spieler

Aus der erhöhten Anzahl an Dreipunktewürfen ergibt sich auch eine niedrigere Anzahl an Zweipunktewürfen. Ein Grund hierfür ist die Verbesserung des Dreipunktewurfes der Spieler, da mit einem erfolgreichem Dreipunktewurf mehr Punkte erzielt werden, als mit einem Zweipunktewurf. Bei einer graphischen Darstellung der Anzahl an genommenen Zweipunkte- und Dreipunktewürfe aller Spieler, wird deutlich, dass Spieler heutzutage deutlich mehr Dreipunktewürfe versuchen, als in dem letzten Jahrtausend.

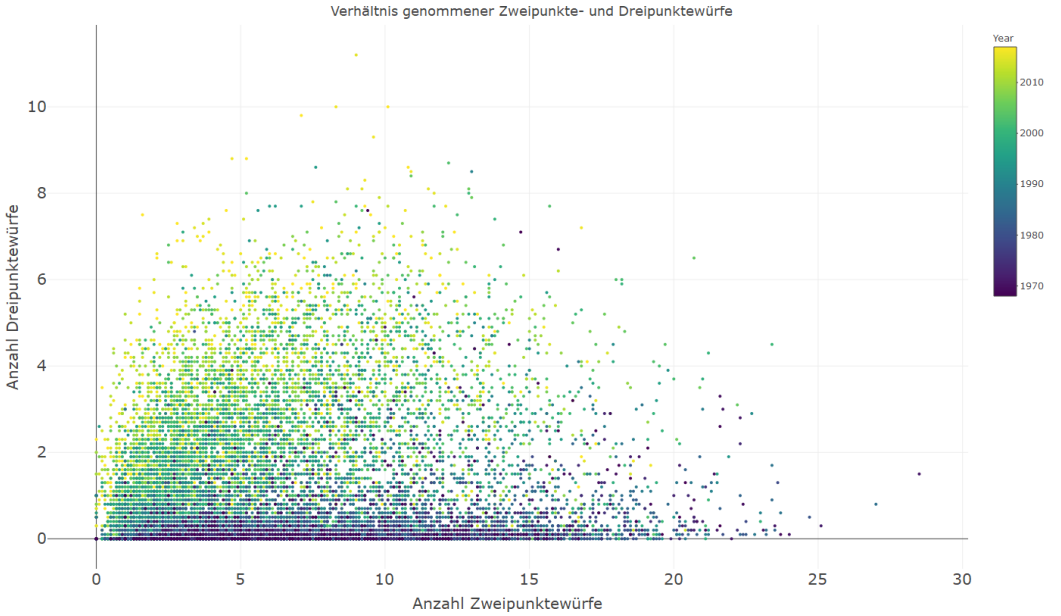


Abbildung 8: Spieler heutzutage versuchen mehr Dreier

Seit 2000 wird in der NBA Shot-Tracking verwendet, das heißt jeder Wurfversuch wird mit Koordinaten und weiteren Infos festgehalten. Dies eignet sich, um herauszufinden, wie viele Würfe ein Team aus einem bestimmten Bereich auf dem Spielfeld nimmt. Zusätzlich kann damit analysiert werden, ob erfolgreiche Teams heutzutage mehr Dreipunktewürfe nehmen, als zum Beispiel vor zehn Jahren. Eine Paradebeispiel für diesen Vergleich sind die Los Angeles Lakers aus der Saison 2000-01, die in dieser Saison Meister wurden, und die Houston Rockets, die in der aktuellen Saison die meisten Siege erringen konnten.

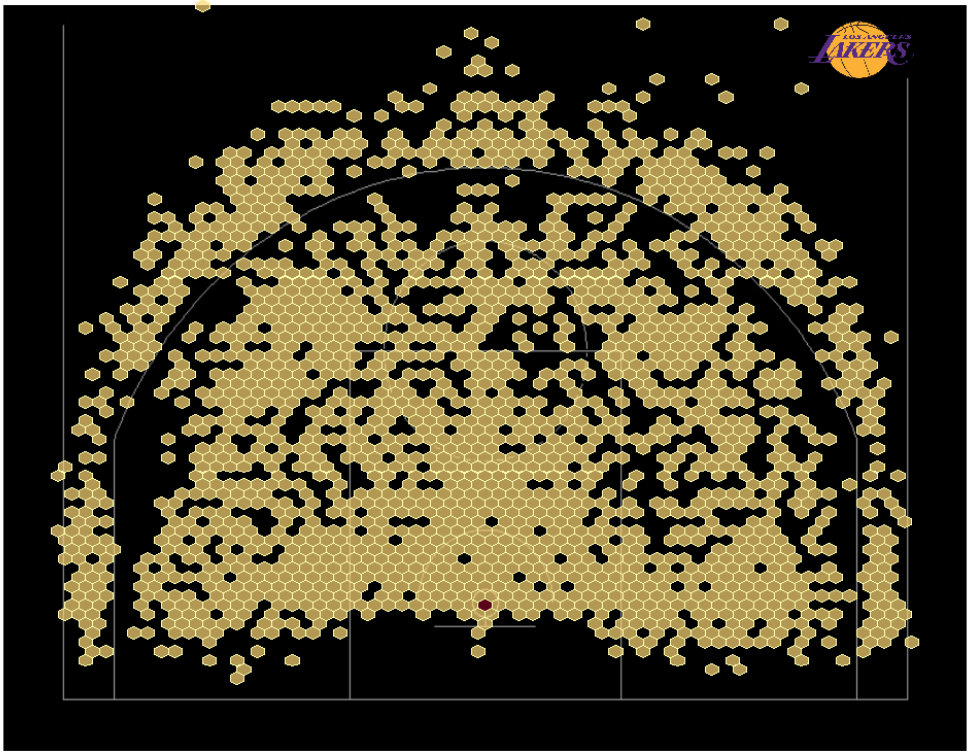


Abbildung 9: Wurfchart der Los Angeles Lakers aus der Saison 2000-01

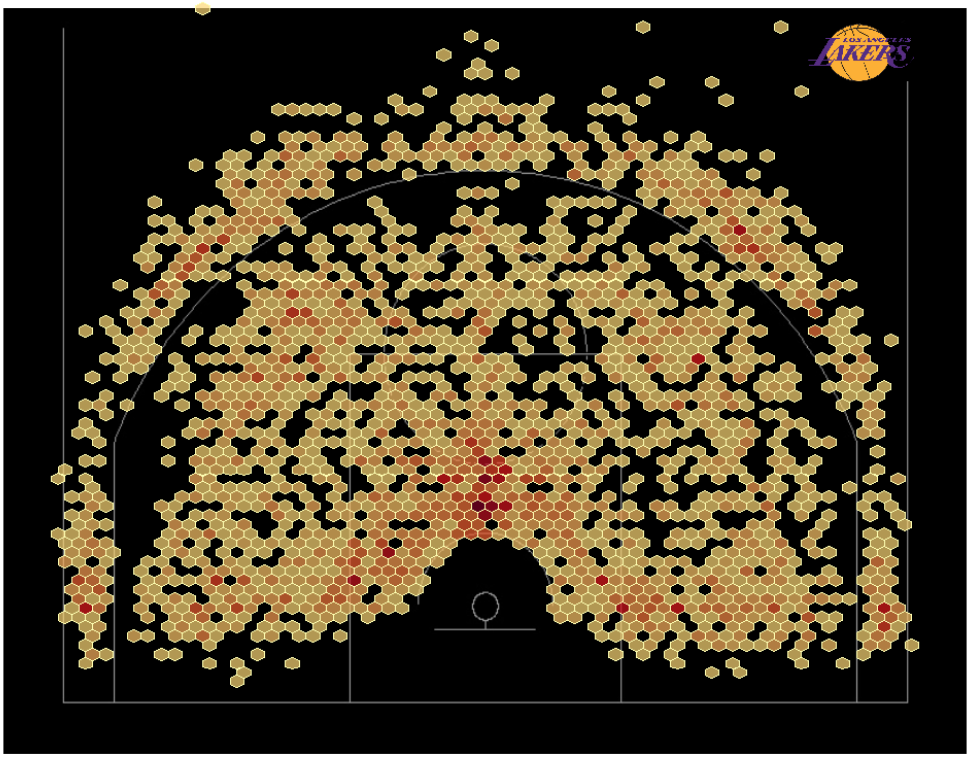


Abbildung 10: Wurfchart der Los Angeles Lakers aus der Saison 2000-01 ohne Würfe unter dem Korb

Die obere Wurfgraphik zeigt die Ausgewogenheit des Angriffs der Lakers. Es gibt wenige Bereiche auf dem Feld, aus denen keine Würfe kamen. Werden die Würfe direkt unter dem Korb herausgefiltert, ist eine Präferenz für Würfe aus der Mitteldistanz und aus der painted area zu erkennen.
Wird ein Vergleich zu den Houston Rockets aus der aktuellen Saison gezogen, fällt zunächst auf, dass das Team die meisten Würfe direkt unter dem Korb versucht hat. Außerdem gibt es weniger Würfe aus der Mitteldistanz. Nach dem Herausfiltern der Würfe unter dem Korb ist zu erkennen, dass vor allem Dreipunktewürfe in der Angriffsstrategie des Teams auftauchen und Würfe aus der Mitteldistanz fast non existent sind. Der tabellarische Vergleich der beiden Mannschaften bestätigt die Diskrepanz zwischen den beiden Teams. Zwei Mannschaften sind eine kleine Stichprobe und die Houston Rockets sind das Extrembeispiel für den Fokus auf Dreipunktewürfe und Würfe unter dem Korb. Aber auch bei der Betrachtung anderer Mannschaften aus den verschiedenen Äras ergibt sich ein identisches Bild. (siehe Shiny App)

2000-01 Los Angeles Lakers

Wurfart		Anzahl Würfe	
Zweipunkte	Paint	1164	19.51%
	Restricted Area	1882	31.55%
	Mitteldistanz	1878	31.48%
	Summe	4924	82.54%
Dreipunkte	Linke Ecke	95	1.59%
	Rechte Ecke	119	1.99%
	Zentral	816	13.68%
	Rückraum	12	0.20%
	Summe	1042	17.46%

Tabelle 1: Wurfaufteilung der Los Angeles Lakers

2017-18 Houston Rockets

Wurfart		Anzahl Würfe	
Zweipunkte	Paint	672	9.65%
	Restricted Area	2214	31.79%
	Mitteldistanz	569	8.17%
	Summe	3455	49.61%
Dreipunkte	Linke Ecke	387	5.56%
	Rechte Ecke	434	6.23%
	Zentral	2671	38.36%
	Rückraum	17	0.24%
	Summe	3509	50.39%

Tabelle 2: Wurfaufteilung der Houston Rockets

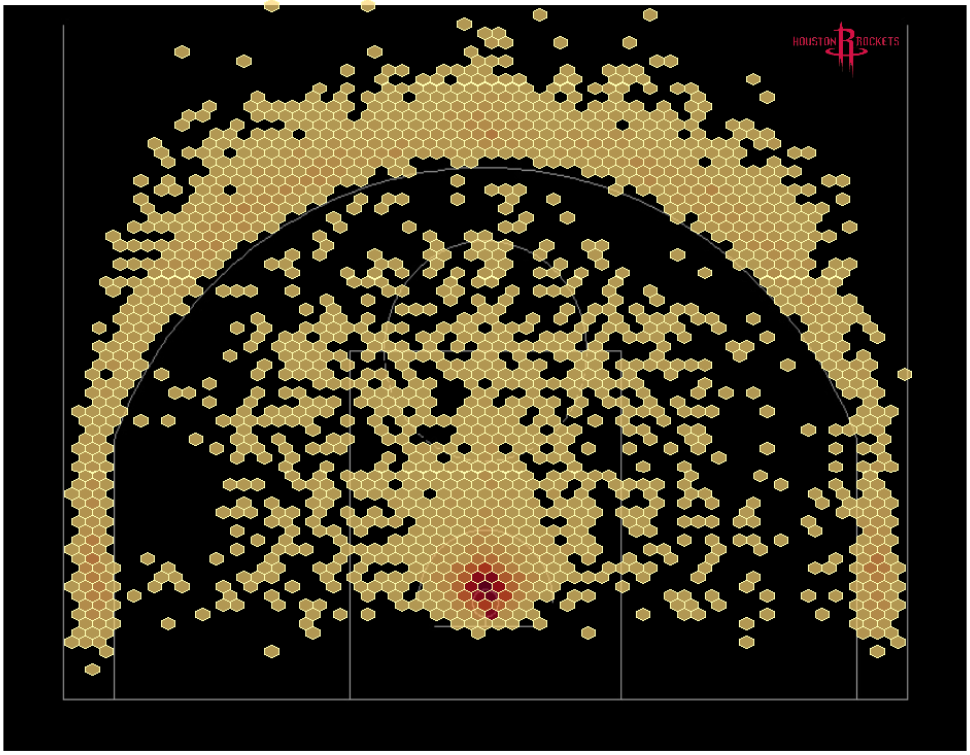


Abbildung 11: Wurfchart der Houston Rockets aus der Saison 2017-18

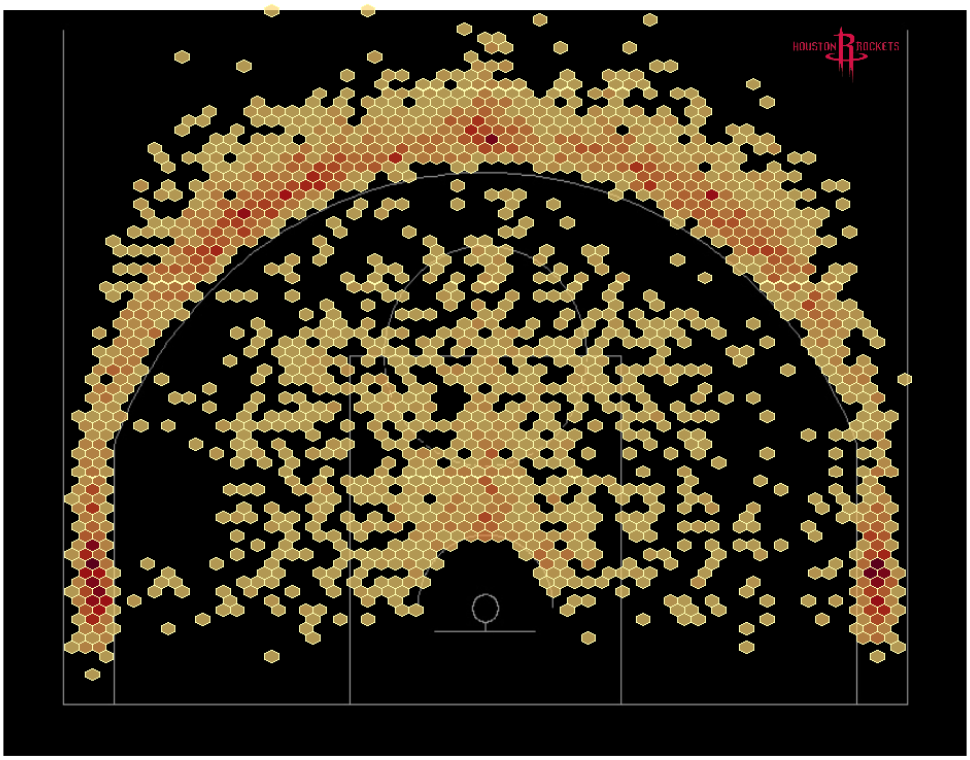


Abbildung 12: Wurfchart der Houston Rockets aus der Saison 2017-18 ohne Würfe unter dem Korb

4.3 Die beste Position

Eine immer wiederkehrende Frage im Basketball ist die der „besten Position“. In diesem Zusammenhang geht es vor allem um folgende Frage:

Welche Position beherbergt die besten Spieler?

In diesem Kontext ist es von Interesse, ob sich über die Jahre diese Position geändert hat. Zusätzlich kann untersucht werden, ob ein Zusammenhang zwischen der Anzahl der Siege einer Mannschaft und der Stärke ihrer Besetzung der einzelnen Positionen existiert.
Als erstes braucht es eine Maßzahl, mit der eine Entscheidung darüber gefällt werden kann, wie gut die Spieler in einer einzelnen Saison waren. Dafür stehen grundsätzlich zwei Statistiken zur Diskussion:

1. *Player efficiency rating*
2. *Value over replacement player*

Beides sind Maßzahlen, die versuchen die Stärke eines Spielers beziehungsweise seiner Saison festzuhalten. Eine genaue Definition der Statistiken ist in Kapitel 7 gegeben. Da PER vor allem offensive Spieler bevorzugt, ist VORP der bessere Index für die Qualität eines Spielers. Folglich wird zweiteres für die Veranschaulichung der Fragestellung verwendet.
Um die zeitliche Änderung darzustellen, werden die einzelnen Jahrzehnte betrachtet. Value over replacement player kann rückwirkend für die Spielzeiten ab 1973 berechnet werden, weswegen die Spielzeiten vor dieser Saison wegfallen. Für die jeweiligen Jahrzehnte werden die 100 besten individuellen Saisons betrachtet, um mögliche Unterschiede ausfindig zu machen. Eine graphische Darstellung der Tabellen in diesen Kapiteln befindet sich im Anhang.
In den 70er Jahren waren 46 der 100 besten Saisons von Centern produziert worden. Für 24 dieser 46 Saisons waren lediglich vier Spieler verantwortlich, darunter mit Kareem Abdul-Jabbar der Spieler, der mit 38.387 erzielten Punkte bis zum heutigen Zeitpunkt der Topscorer der NBA ist. Abdul-Jabbar war einer von zwei Spielern, der mit allen sieben Saisons in den Top 100 landen konnte. Der andere Spieler war Julius Erving, der für sieben der 23 besten Saisons eines Small Forwards verantwortlich war. Auffällig ist, wie selten Shooting Guards in der unteren Tabelle auftauchen. Lediglich fünf der 100 besten Saisons der 70er Jahren wurde von einem Shooting Guard produziert.

Position	Anzahl
Point Guard	11
Shooting Guard	5
Small Forward	23
Power Forward	15
Center	46

Tabelle 3: Die besten Saisons der 1970er

In der nächsten Dekade ergibt sich bereits ein ausgeglicheneres Bild. Die Anzahl der Center ist zurückgegangen, während vor allem bei den Guards ein starker Anstieg zu vermelden ist. Die beiden dominantesten Spieler dieser Jahre waren Larry Bird

(Small Forward) und Magic Johnson (Point Guard), die mit jeweils neun Saisons in den Top 100 vertreten sind. Weitere bekannte Gesichter sind Hakeem Olajuwon (Center), Charles Barkley (Power Forward) und Michael Jordan (Shooting Guard) mit je fünf Saisons. All drei haben jeweils fünf Saisons in den 80ern gespielt und sind somit mit jeder ihrer Saisons in den Top 100 vertreten.

Position	Anzahl
Point Guard	18
Shooting Guard	25
Small Forward	24
Power Forward	14
Center	19

Tabelle 4: Die besten Saisons der 1980er

In dem nächsten Jahrzehnt fällt Shooting von Platz 1 zurück auf den letzten Platz und die meisten Saisons in den Top 100 wurden von Power Forwards aufgestellt. Michael Jordan ist mit sechs seiner sieben Saisons vertreten, sein Teamkamerad Scottie Pippen (Small Forward) gelang dies in sieben Saisons, Center David Robinson achtmal und Karl Malone (Power Forward) ist mit neun Saisons am häufigsten vertreten.

Position	Anzahl
Point Guard	24
Shooting Guard	14
Small Forward	16
Power Forward	27
Center	19

Tabelle 5: Die besten Saisons der 1990er

Zu Beginn des neuen Jahrtausends hielt, dank Spielern wie Tim Duncan, Dirk Nowitzki und Kevin Garnett, die Dominanz der Power Forwards an. Die Relevanz der Center rutschte weiter in den Hintergrund, während mehr Shooting Guards bei den besten Saisons vertreten waren. Mit acht Saisons schaffte es Kevin Garnett am häufigsten in die Top 100, gefolgt von den beiden Shooting Guards Tracy McGrady und Kobe Bryant, welche wie Dirk Nowitzki (ebenfalls Power Forward) und LeBron James (Small Forward), sechsmal in den Top 100 verzeichnet sind.

Position	Anzahl
Point Guard	20
Shooting Guard	24
Small Forward	16
Power Forward	29
Center	11

Tabelle 6: Die besten Saisons der 2000er

Im aktuellen Jahrzehnt dominiert eine Position. Nachdem in den 70er Jahren mit den Centern die größten Spieler die besten Saisons produzierten, sind es heute mit den Point Guards die kleinsten Spieler. Vor allem Chris Paul, der mit jeder seiner acht Saisons vertreten ist, Stephen Curry und Russell Westbrook, die mit je sechs Saisons

vertreten sind, haben einen großen Anteil daran. LeBron James hat es ebenfalls geschafft, mit jeder Saison vertreten zu sein, während mit James Harden (Shooting Guard) und Kevin Durant (Small Forward) zwei weitere Spieler mit je sechs Saisons präsent sind. Shooting Guards sind am seltensten vertreten, Center befinden sich auf dem aufsteigenden Ast und die Dominanz der Power Forwards scheint vorüber zu sein.

Position	Anzahl
Point Guard	35
Shooting Guard	11
Small Forward	19
Power Forward	16
Center	19

Tabelle 7: Die besten Saisons der 2010er

5 Regressionsmodelle im Basketball

In vielen Sportarten existieren Regressionsmodelle, um die Anzahl der Siege einer Mannschaft oder den Ausgang einzelner Spiele vorherzusagen. Diese Modelle finden bei Wettanbietern eine Verwendung. Ein Ziel dieser Arbeit ist es, ein möglichst genaues Modell für die Vorhersage einer einzelnen Saison und für die Vorhersage einzelner Spiele zu finden. Des weiteren soll untersucht werden, welche Variablen einen Einfluss auf die Anzahl der Siege haben und ob diese eher offensiver oder defensiver Natur sind. Unter den untersuchten Variablen befinden sich auch das Offensiv- und Defensivrating, zwei Maßzahlen, die die Leistung einer Mannschaft in der Offensive beziehungsweise Defensive messen. Je höher das offensive Rating einer Mannschaft ist, desto besser ist das Team in der Offensive. Bei dem defensiven Rating ist das Gegenteil der Fall. Wird der Zusammenhang zwischen den Ratings und dem Siegesanteil einer Mannschaft betrachtet, haben Mannschaften mit einem sehr hohen Offensivrating einen sehr hohen Siegesanteil. Im defensiven Fall haben Mannschaften mit einem sehr niedrigen Rating zwar auch einen hohen Siegesanteil, aber auch Mannschaften mit einem hohen Defensivrating haben teilweise einen hohen Anteil an Siegen. Damit kann vermutet werden, dass die Offensive einen höheren Einfluss als die Defensive hat.

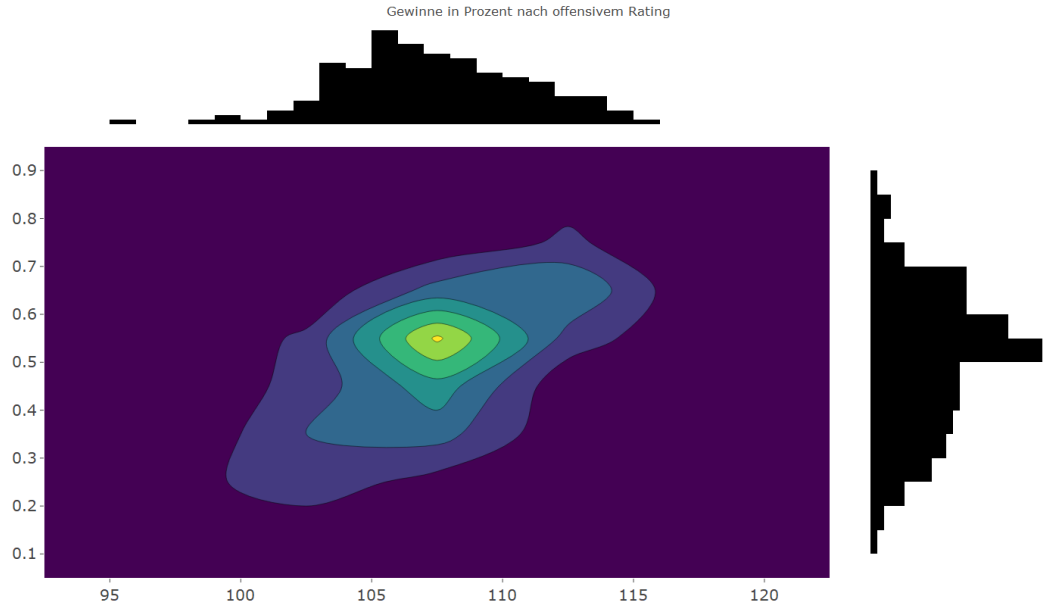


Abbildung 13: Zusammenhang zwischen ORtg und Siegesanteil in den letzten fünf Saisons

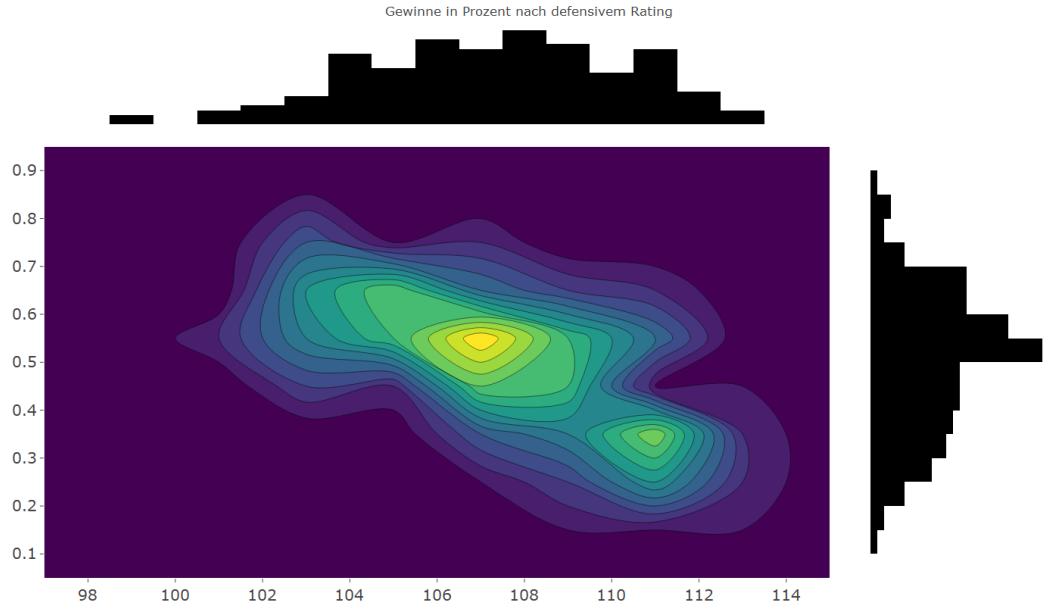


Abbildung 14: Zusammenhang zwischen DRtg und Siegesanteil in den letzten fünf Saisons

5.1 Modelle für die Vorhersage einer kompletten Saison

Für die Vorhersage einer kompletten Saison werden zusätzlich zu den Standard und fortgeschrittenen Statistiken die sogenannten „*Advanced tracking stats*“ verwendet. Diese werden seit der Saison 2013-14 durch Kameras in den Stadien erfasst. Durch diese ist es möglich, die Spielweise eines Teams möglichst genau zu erfassen. So kann aus ihnen abgeleitet werden, wie oft ein Team den Ball passt, wo auf dem Basketballplatz die Spieler häufig den Ball bekommen oder auch wie viele Freiwürfe, Fouls, Turnover usw. sich aus bestimmten Situationen ergeben. Durch die Aufnahme dieser Statistiken in das Modell liegen zunächst über 200 Kovariablen vor. Vor einer ersten Schätzung werden Kovariablen entfernt, die durch ihren direkten Zusammenhang mit der Anzahl an Siegen einen zu hohen Einfluss haben. In diese Kategorie fallen Statistiken wie zum Beispiel **Pythagorean wins** (PW) oder **Margin of victory** (MOV). Andere Variablen, die einen beinahe perfekten Zusammenhang haben, beispielsweise gespielte Pässe und angekommene Pässe, werden ebenfalls entfernt. Nach der Bereinigung verbleiben noch ca. 130 Kovariablen in dem Modell. Mit dem daraus resultierenden Datensatz werden die Modelle geschätzt und getestet. Es sollte noch erwähnt werden, dass die Vorhersagen gerundet sind, da eine Mannschaft nicht 43.25 Siege erreichen kann. Aufgrund dessen kann es sein, dass die gesamte Anzahl an Siegen bei bestimmten Modellen nicht aufgeht.

5.1.1 Lineare Regression

Bei der linearen Regression wird die Relation zwischen einer Variable y und den Kovariablen $x_1, ..., x_k$ durch eine Funktion $f(x_1, ..., x_k)$ modelliert. Dazu kommt noch ein Störterm ε. Daraus ergibt sich:

$$y = f(x_1, ..., x_k) + \varepsilon \tag{1}$$

Zur Schätzung der einzelnen β werden Daten erhoben, aus denen für jede Beobachtung die Gleichung

$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + \varepsilon_i = x_i^T \beta + \varepsilon_i \tag{2}$$

abgeleitet werden kann.
Das Regressionsmodell kann auch in Matrixform geschrieben werden, mit

$$y = \begin{pmatrix} y_i \\ \vdots \\ y_n \end{pmatrix}, \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix},$$
$$X = \begin{pmatrix} 1 & x_{11} & \dots & x_{1k} \\ \vdots & \vdots & & \vdots \\ 1 & x_{n1} & \dots & x_{nk} \end{pmatrix} = \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix}$$

Daraus folgt die Gleichung in Matrixnotation:

$$y = X\beta + \varepsilon$$

Zusätzlich gelten folgende Annahmen:

1. $\mathrm{E}(\varepsilon) = 0$
2. $\mathrm{Cov}(\varepsilon) = \mathrm{E}\left(\varepsilon\varepsilon^T\right) = \sigma^2 I$
3. Die Designmatrix X besitzt vollen Spaltenrang $\Rightarrow \mathrm{rg}(X) = k + 1 = p$

(Fahrmeir, Kneib & Lang, 2009)

Die Modelle
Da ein lineares Regressionsmodell mit 150 Kovariablen schwer zu interpretieren ist, werden drei Einflussvariablen gewählt. Diese sind:

1. TEAM_PER $= \sum_{i=1}^{\min\left(n_{Spieler}, 12\right)}$ W_PER
2. TEAM_DBPM $= \sum_{i=1}^{\min\left(n_{Spieler}, 12\right)}$ W_DBPM
3. TEAM_OBPM $= \sum_{i=1}^{\min\left(n_{Spieler}, 12\right)}$ W_OBPM
4. TEAM_VORP $= \sum_{i=1}^{\min\left(n_{Spieler}, 12\right)}$ W_VORP

Da PER, BPM und VORP nicht in Betracht zieht, wie viele Spiele ein einzelner Spieler in der Saison gemacht hat, ist das Bilden einer Summe aus den Einzelwerten nicht sinnvoll. Das Aggregieren der Daten würde dazu führen, dass der Beitrag mancher Spieler viel zu hoch gewertet werden würde. Aufgrund dessen werden neue gewichtete Statistiken berechnet. Die Formel für das weighted VORP ergibt sich durch:

$$\text{W\_VORP} = \frac{\text{MP}}{\text{MT}} * \text{VORP} \tag{3}$$

Hierbei steht *MP* für die Anzahl an Minuten, die ein Spieler in der Saison spielte. *MT* steht für die totale Anzahl an Minuten, die pro Saison gespielt wurden. (Anmerkung: Da 5 Spieler gleichzeitig auf dem Feld stehen, ist hier die totale Anzahl an Minuten geteilt durch 5 gemeint). *VORP* steht für den VORP wert einen Spielers. Für die anderen beiden Variablen ist die Berechnung analog.
Für die Modelle werden die zwölf am häufigsten eingesetzten Spieler einer Mannschaft betrachtet, beziehungsweise weniger, falls ein Team unter zwölf Spieler einsetzte.
Die Anzahl der Siege stellt die Zielvariable dar. Wird diese betrachtet, lässt sich eine Normalverteilung vermuten.

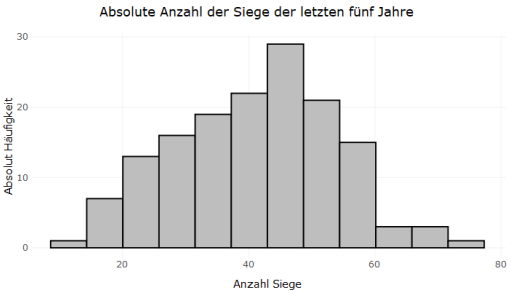


Abbildung 15: Histogramm der Zielva- riable

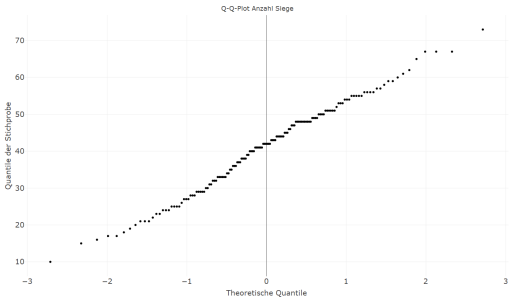


Abbildung 16: Q-Q-Plot der Zielvariable

Wird mit einem Shapiro-Wilk Test auf Normalität getestet, ergibt sich ein p-Wert von 0.3402. Es kann nicht abgelehnt werden, dass die Daten normalverteilt sind.
Für die Modelle wird eine zehnfache Kreuzvalidierung mit Hilfe des R-Pakets *caret* durchgeführt. Außerdem werden die Modelle auf Interaktionen untereinander untersucht, wodurch sich insgesamt 24 verschiedene Modelle ergaben. Interessant dabei ist, dass vor allem Modelle schlecht abschneiden, bei denen die Variable TEAM_VORP nicht enthalten ist. Diese Modelle zeichnen sich vor allem durch R^2 Werte unter 0.8 oder distinkt höhere Werte für die Informationskriterien nach Akaike und Bayes aus. Der Hauptgrund dafür ist die einseitige Richtung der Variablen. OBPM und DBPM sind jeweils Maßzahlen dafür, wie gut ein Spieler in der Offensive oder Defensive ist und PER bevorzugt offensiv orientierte Spieler, die eine hohe Anzahl an Wurfeversuchen haben, unabhängig von deren Effizienz.
Die restlichen Modelle erreichen allesamt R^2-Werte zwischen 0.93 und 0.94 und niedrigere Werte für die Informationskriterien. Die Entscheidung fiel auf das folgende Modell:

$$\begin{aligned} &\text{Anzahl Siege} = \beta_0 + \beta_1 x_{TEAM\_VORP} \Leftrightarrow \\ &\text{Anzahl Siege} = 38.726 + 2.115 x_{TEAM\_VORP} \end{aligned}$$

Dieses hat einen R^2 Wert von 0.9356 und die niedrigsten Werte für die beiden Informationskriterien. (AIC = 634.3, BIC = 642.7)

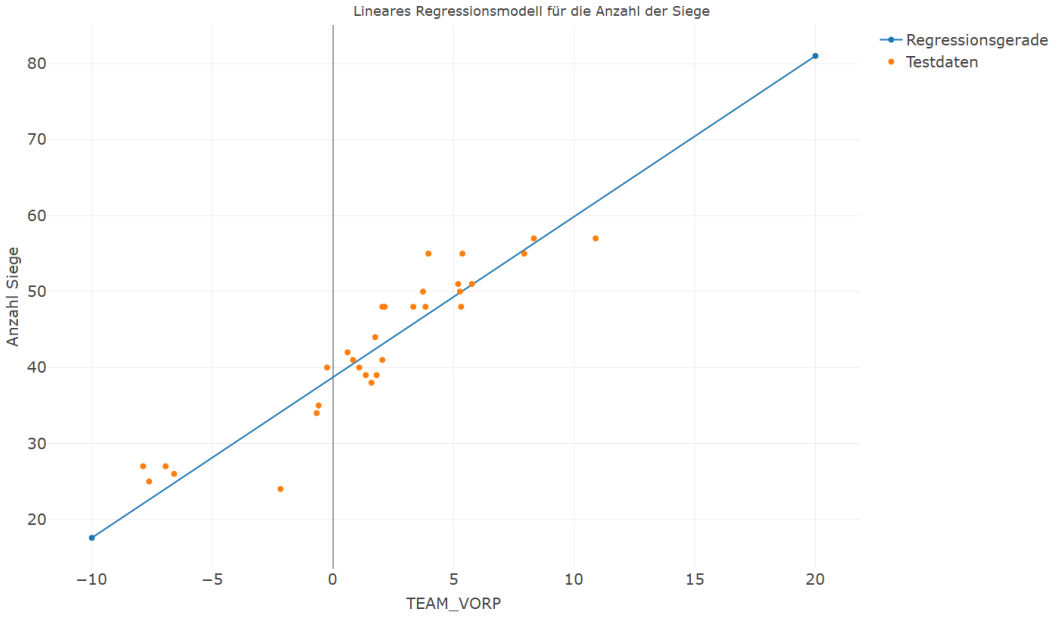
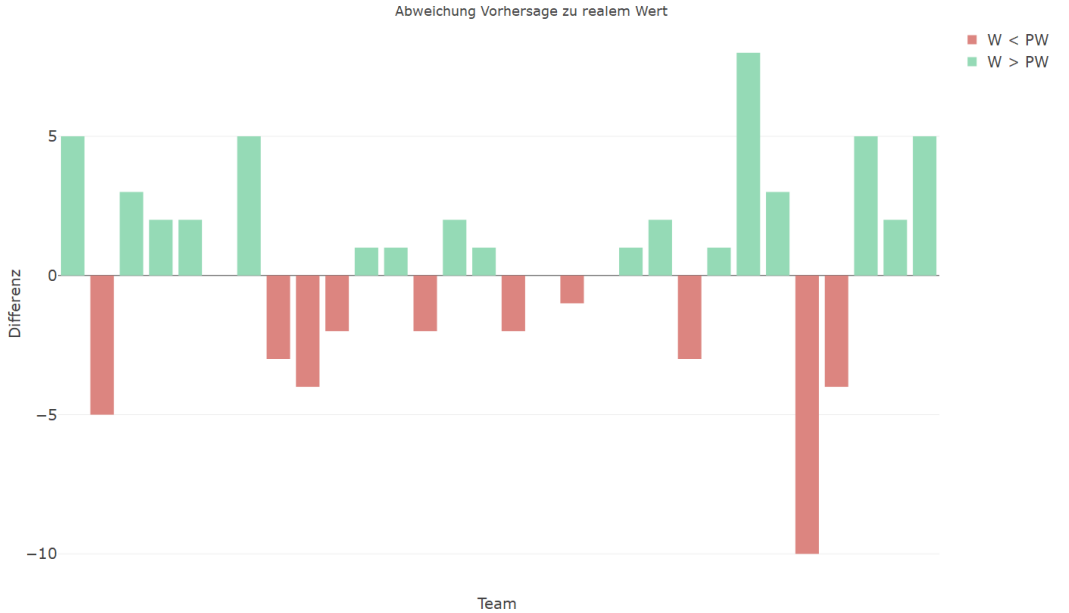


Abbildung 17: Regressionsgerade und Daten des Testdatensatzes

Im Durchschnitt beträgt der Abstand zwischen der Anzahl der Siege und der vorhergesagten Anzahl der Siege 0.43. Dieser Wert kann suggerieren, dass die vorhergesagten Werte nahe bei den tatsächlichen Werten liegen. Tatsächlich liegen jedoch neun der vorhergesagten Werte um $\pm$ 1 neben der Anzahl der Siege. Bei der Frage, warum teilweise so große Differenzen existieren, ist ein Blick auf die Einflussgröße hilfreich. $TEAM\_VORP$ ermöglicht es, zu untersuchen, wie viel Talent ein einzelnes Team hat. Eine Schlussfolgerung ist, dass talentierte Teams öfter gewinnen. Hat ein Team explizit weniger Siege errungen, als vorausgesagt wird, ist dies ein Indiz dafür, dass das Talent des Teams nicht optimal genutzt wurde. Das beste Beispiel dafür sind die Dallas Mavericks. Diese haben 24 anstelle der vorhergesagt 34 Siege erreicht. Der Grund liegt darin, dass Teambesitzer Mark Cuban seinen eigenen Spielern sagte, verlieren sei die beste Option. Damit wollte er erreichen, dass sein Team in dem nächsten NBA-Draft einen möglichst hohen Pick erhält. (nba.com, 2018)
Hat im Vergleich dazu eine Mannschaft viel mehr Siege errungen als vorhergesagt, ist es dem Trainerteam gelungen, das Optimum aus den Spielern rauszuholen. Das Paradebeispiel hierfür sind die Boston Celtics. Die Mannschaft musste die ganze Saison ohne Gordon Hayward, den wohl zweitbesten Spieler des Teams, auskommen. Dazu kamen noch kleinere Verletzungen ihres besten Spielers Kyrie Irving, der in 75% der Spiele auf dem Platz stand. Trotz dieser Verletzungen gelang es dem Team, acht Siege mehr zu holen, als vorausgesagt wird. Die Anerkennung hierfür gehört vor allem dem Trainer der Boston Celtics, Brad Stevens. Dieser hat es geschafft, durch seine guten Schemen in der Offensive und der Defensive sein Team zu mehr Siegen als erwartet zu coachen. Zu Recht gilt Stevens als einer der besten Coaches in der NBA im Moment.



Absolute Abweichung	0	1	2	3	4	5	8	10	abs(AVG)	AVG	RMSE
Häufigkeit	3	6	8	4	2	5	1	1	2.83	0.43	3.629

Abbildung 18: Abweichungen von Vorhersage zu echtem Wert

5.1.2 Lasso-Regression

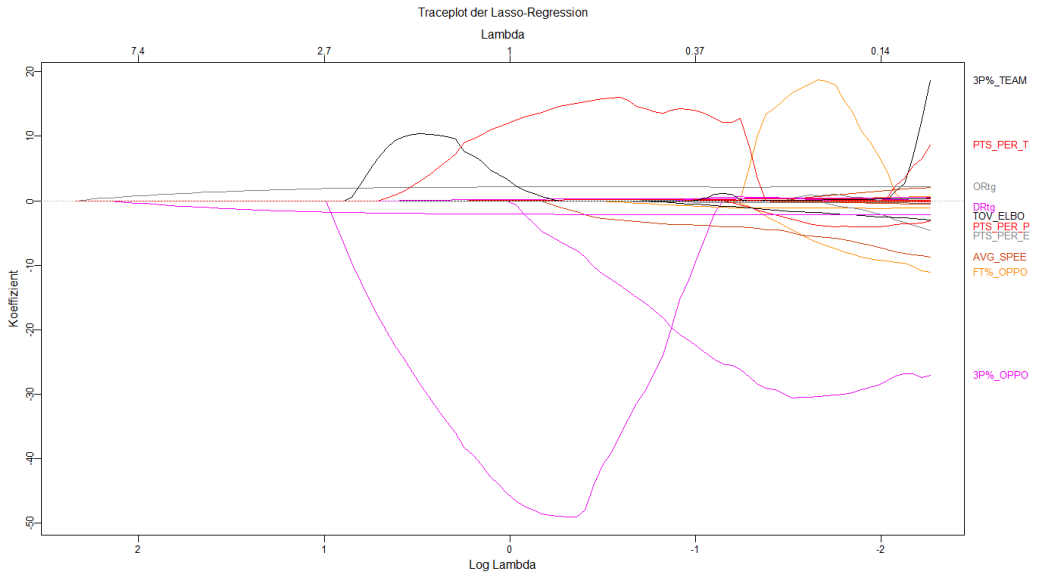
Eine Möglichkeit, um mit vielen Kovariablen in einem Modell umzugehen, ist durch Einsatz eines Lasso. Hierbei bedeutet Lasso **L**east **a**bsolute **s**hrinkage and **s**election **o**perator.
Kurz gesagt:
Eine formelle Definition des Lasso nach Tibshirani (2011):
Gegeben einer linearen Regression mit standardisierten Prädikatoren x_{ij} und Responsewerten y_i $\forall i = 1, 2, ..., N,\ j = 1, 2, ..., p$ löst das Lasso das L_1-Penalisierung Problem um $\beta = \{\beta_j\}$ zu und gleichzeitig

$$\min \left\{ \sum_{i=1}^{N} \left(y_i - \sum_j x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{4}$$

zu finden. Das Lasso führt eine Variablen Selektion und Schrumpfung durch. Im Vergleich dazu wird bei der Ridge Regression eine Schrumpfung durchgeführt. Kurz gesagt: Bei Lasso werden bestimmte Variablen ausgewählt und die anderen auf den Wert null gesetzt. Dies kann zu Problemen führen, wenn ein Zusammenhang zwischen einzelnen Variablen vorliegt, was wiederum zu einem Informationsverlust und gesunkener Genauigkeit führen kann. Eine Möglichkeit dagegen vorzugehen, wird in dem nächsten Kapitel besprochen.(Tibshirani, 2011)
Für die Berechnung der Modelle wird das R-Paket *glmnet* verwendet. Die Regression wird auf zwei verschiedenen Datensätzen durchgeführt:

1. Datensatz, der die Teamvariablen enthält
2. Kompletter Datensatz, das heißt alle Teamvariablen und aggregierten Spielervariablen (TEAM_VORP usw.)

Datensatz 1
Zunächst lohnt sich ein Blick auf einen Traceplot, um einen ersten Gedanken zu haben, was die Variablen mit dem größten Einfluss sind.



Negative Einflüsse)
Variable
3P%_OPPONENT
FT%_OPPONENT
AVG_SPEED_OFF
PTS_PER_ELBOW_TOUCHE
PTS_PER_PAINT_TOUCH
TOV_ELBOW_TOUCHES
DRtg

Positive Einflüsse
Variable
3P%_TEAM
PTS_PER_TOUCH
ORtg

Abbildung 19: Einflussreichsten Variablen der Regression

Es sollte darauf geachtet werden, dass mehrere dieser Einflussvariablen prozentual sind und deren Ausprägungen nicht den Wert von eins übersteigen können.
Für ein möglichst gutes Modell muss noch ein Parameter für λ gewählt werden. Eine Möglichkeit für dessen Wahl ist die Funktion *cv.glmnet*, eine andere ergibt sich aus dem Package *caret*. Dieses erlaubt es, Parameter zu tunen, um ein möglichst genaues Vorhersagemodell zu bekommen. Die Implementierung in R ist simpel:

```
set.seed(1234)
lasso = train(W ~., data = datasetLasso, method = "glmnet",
        tuneGrid = expand.grid(.alpha = 1,
        .lambda = seq(0, 5, by = 0.01)),
        trControl = trainControl(method = "cv",
                                 number = 10))
```

Der Wert 1 für alpha sorgt dafür, dass eine Lasso Regression durchgeführt wird. Für λ werden alle Werte zwischen 0 und 5 in Schritten der Größe 0.01 eingesetzt und

es wird eine 10-fache Kreuzvalidierung durchgeführt. Am Ende werden für die ca. 500 Modelle mit einem Testdatensatz Prognosen erstellt und das Modell mit dem geringsten RMSE Wert gewählt.
Der Datensatz wird zufällig im Verhältnis 80:20 in Trainingsdaten und Testdaten aufgeteilt. Das beste Modell wird mit dem Wert 0.59 für λ erhalten und hat einen RMSE in Höhe von 3.05505. Es ergibt sich aus der folgenden Formel:

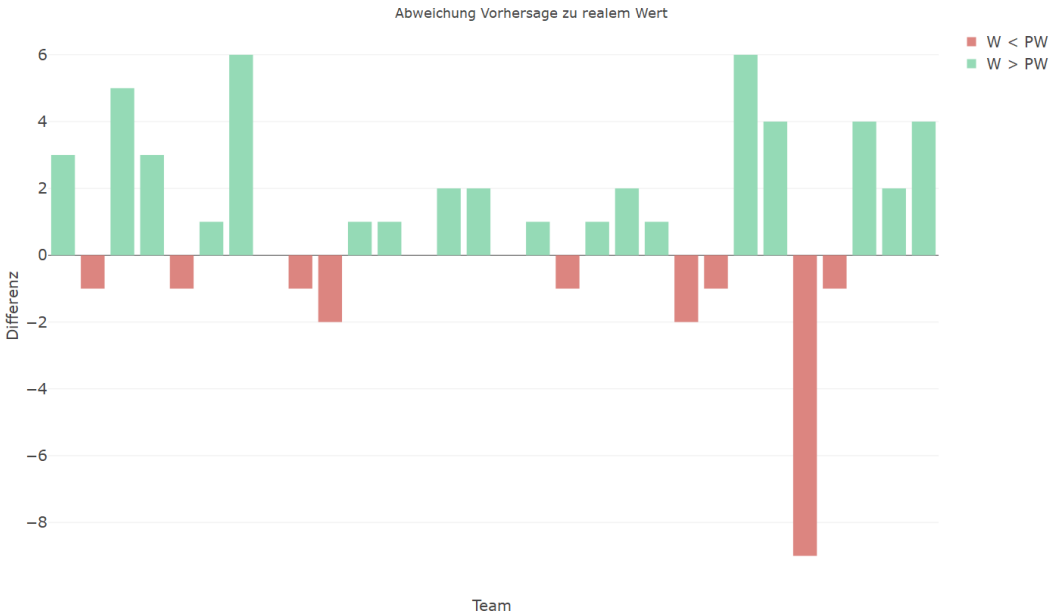
$$\begin{aligned}\text{Anzahl Siege } &= \beta_0 + \beta_1 x_{FG\%\_DRIVES} + \beta_2 x_{FG\%\_OPPONENT} \\ &+ \beta_3 x_{DFG\%} + \beta_4 x_{eFG\%} + \beta_5 x_{Age} + \beta_6 x_{ORtg} + \beta_7 x_{DRtg} \\ &+ \beta_8 x_{3P\%\_OPPONENT} + \beta_9 x_{AVG\_SPEED\_DEF}\end{aligned}$$

Für die gerundeten Werte der einzelnen Koeffizienten ergibt sich:

Werte Koeffizienten	
Variable	Wert
Intercept	45.18
FG%_DRIVESS	0.1574
FG%_OPPONENT	-47.98
DFG%	-0.01073
eFG%	15.33

Werte Koeffizienten	
Variable	Wert
Age	0.2613
ORtg	2.177
DRtg	-2.108
3P%_OPPONENT	-8.739
AVG_SPEED_DEF	-2.111

Wird der Abstand zwischen dem vorhergesagten und dem aktuellen Wert betrachtet, gibt es 15 Werte, die um ±1 neben dem wahren Wert liegen. Im Vergleich dazu lagen bei der linearen Regressions zwölf Werte um ±2 neben dem wahren Wert. Auffällig sind hier die Dallas Mavericks aus der aktuellen Saison. Die Vorhersage für diese Mannschaft lag um neun neben dem wahren Wert.

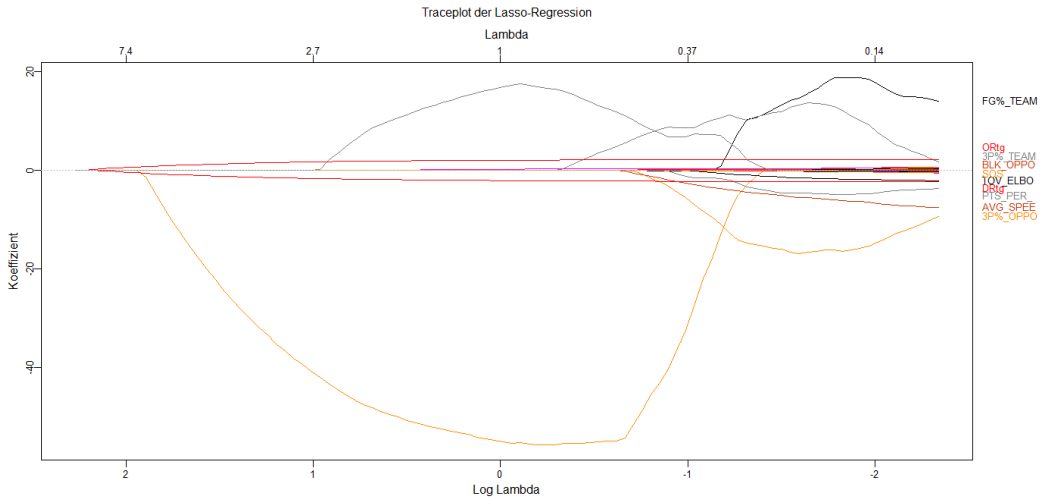


Absolute Abweichung	0	1	2	3	4	5	6	9	abs(AVG)	AVG	RMSE
Häufigkeit	3	12	6	2	3	1	2	1	2.27	1	3.055

Abbildung 20: Abweichungen von Vorhersage zu echtem Wert

Datensatz 2

Zu dem eben verwendeten Datensatz wird die Variable TEAM_VORP hinzugefügt, um zu überprüfen, ob durch diese Variable eine genauere Prognose möglich ist. Erneut wird der Traceplot der einflussreichsten Variablen betrachtet:



Negative Einflüsse	Positive Einflüsse
Variable	Variable
FG%_OPPONENT	AST_PAINT_TOUCHES
3P%_OPPONENT	PF%_DRIVES
2P%_OPPONENT	AVG_SPEED_DEF
DIST_MILES_DEF	TEAM_VORP
2P%_TEAM	
TOV_ELBOW_TOUCHES	

Abbildung 21: Einflussreichsten Variablen der Regression

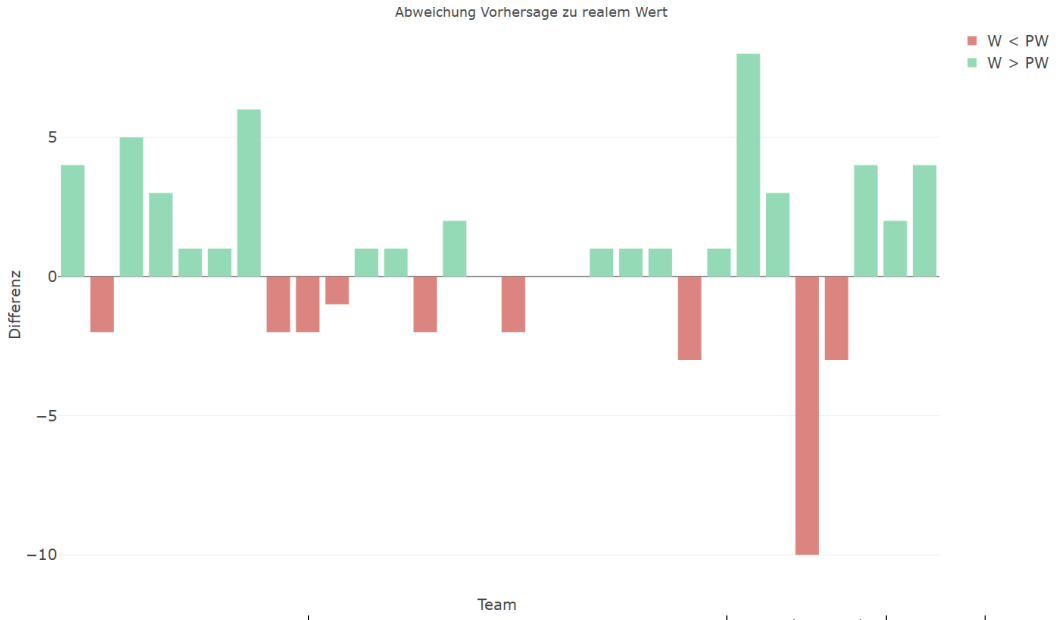
Es werden dieselben Modelle berechnet wie bei Datensatz 1 und mit dem RMSE verglichen. Der beste Wert für λ ist erneut 0.59. Durch die Aufnahme der Variable TEAM_VORP ergibt sich eine vereinfachte Regressionsgleichung aber auch ein erhöhter RMSE Wert. Die Regressionsgleichung lautet:

$$\begin{aligned}\text{Anzahl Siege} = \beta_0 + \beta_1 x_{FG\%\_DRIVES} + \beta_2 x_{Age} + \beta_3 x_{SOS} \\ + \beta_4 x_{TEAM\_VORP} + \beta_5 x_{FG\%\_OPPONENT}\end{aligned}$$

Die gerundeten Werte für die Koeffizienten lauten:

Werte Koeffizienten	
Variable	Wert
Intercept	31.82
FG%_DRIVES	0.1586
Age	0.1237
SOS	-1.622
TEAM_VORP	1.897
FG%_OPPONENT	-7.211

Wird die absolute Abweichung der vorhergesagten Anzahl der Siege zu der tatsächlichen Anzahl der Siege betrachtet, fällt diese schlechter aus als bei dem obigen Modell. In diesem Fall würde es sich somit lohnen, TEAM_VORP nicht als Kovariable in das Modell aufzunehmen. Der größte Ausreißer in der Graphik ist das Team der Dallas Mavericks. Aber auch für die Boston Celtics der aktuellen Spielzeit werden deutlich weniger Siege prognostiziert, als tatsächlich erreicht wurden. Der Grund hierfür wurde im letzten Kapitel bereits thematisiert.
Wird der Datensatz bewusst aufgeteilt, ergeben sich erneut ungenauere Modelle, die nicht vorgestellt werden.



Absolute Abweichung	0	1	2	3	4	5	6	8	9	abs(AVG)	AVG	RMSE
Häufigkeit	3	9	7	4	3	1	1	1	1	2.53	0.733	3.396

Abbildung 22: Abweichungen von Vorhersage zu echtem Wert

5.1.3 Elastic net Regression

Um elastic net Regression zu besprechen, muss zunächst die Ridge Regression vorgestellt werden, da es sich bei der EL-Regression um eine Mixtur der Lasso und Ridge Regression handelt. Wie bereits erwähnt, gibt es Ähnlichkeiten zwischen den beiden Verfahren. Hier wird das L_2-Penalisierung Problem gelöst um β und

$$\min \left\{ \sum_{i=1}^{N} \left(y_i - \sum_j x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^2 \right\} \tag{5}$$

zu finden.
In der ER-Regression werden beide Penalisierungsterme mit einbezogen und getuned, um ein möglichst gutes Mittelmaß zu finden. Der Penalisierungsterm sieht somit wie folgt aus:

$$\min \left\{ \sum_{i=1}^{N} \left(y_i - \sum_j x_{ij}\beta_j \right)^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} |\beta_j|^2 \right\} \tag{6}$$

Existieren in dem Datensatz miteinander korrelierte Variablen, werden diese in einer Gruppe zusammengefasst. Liegt in dieser Gruppe ein starker Prädiktor vor, wird die ganze Gruppe in das Modell mit aufgenommen. Im Vergleich zu dem Lasso Verfahren liegt hier ein geringerer Informationsverlust vor.
Um dieses Regressionsverfahren anzuwenden, kann in R das Paket *glmnet* verwendet werden. Der Funktion *glmnet()* wird wieder der Parameter alpha übergeben. Wird ein Wert zwischen 0 und 1 gewählt, wird eine Elastic net Regression durchgeführt. Somit muss lediglich der ideale Wert für alpha gefunden werden. Zur Bestimmung der optimalen Parameter wird die train Funktion aus dem *caret* Package verwendet. Für alpha wird der Wert 0.95 gewählt und für λ 0.59. Das resultierende Modell hat einen geringeren RMSE Wert und eine zusätzliche Kovariable:

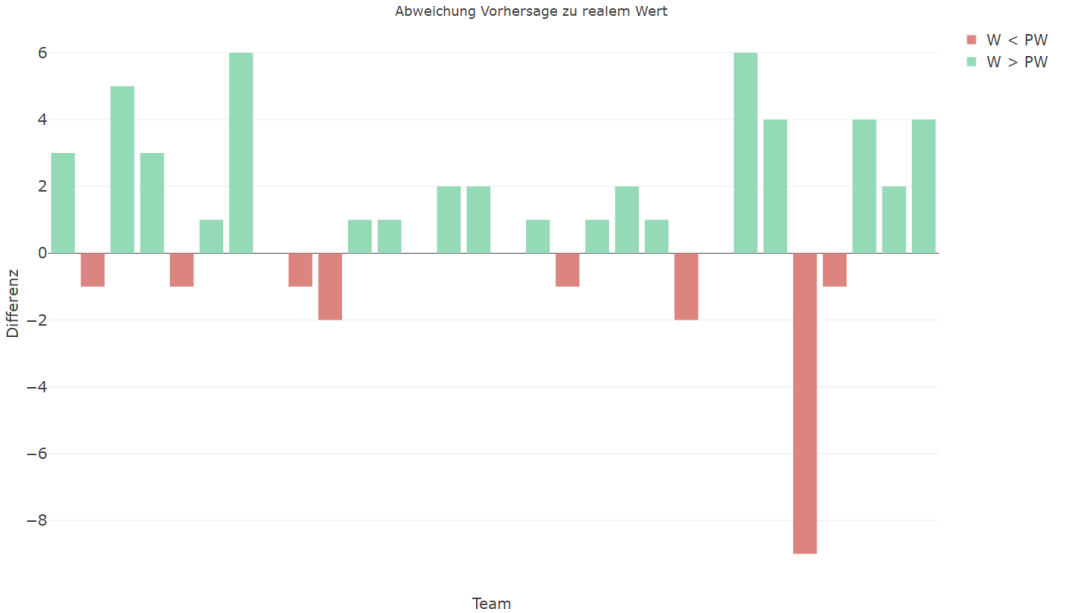
$$\begin{aligned}\text{Anzahl Siege} = {} & \beta_0 + \beta_1 x_{FG\%\_DRIVES} + \beta_2 x_{Age} + \beta_3 x_{ORtg} + \beta_4 x_{TS\%} + \beta_5 x_{eFG\%} \\ & + \beta_6 x_{FG\%\_OPPONENT} + \beta_7 x_{AVG\_SPEED\_DEF} + \beta_8 x_{DRtg} + \beta_9 x_{DFG\%} \\ & + \beta_{10} x_{3P\%\_OPPONENT}\end{aligned}$$

Für die gerundeten Werte der einzelnen Koeffizienten ergibt sich:

Werte Koeffizienten	
Variable	Wert
Intercept	45.84
FG%_DRIVES	0.1595
Age	0.2757
ORtg	2.152
TS%	2.794
eFG%	16.10

Werte Koeffizienten	
Variable	Wert
FG%_OPPONENT	-49.41
AVG_SPEED_DEF	-2.272
DRtg	-2.087
3P%_OPPONENT	-10.12
DFG%	-0.01792

Werden die absoluten Abweichungen der prognostizierten und tatsächlichen Siege betrachtet, fallen diese ähnlich zu den bisherigen Modellen aus. 15 Vorhersagen waren um ± 1 neben dem tatsächlichen Wert, die gesamte Abweichung ist identisch zu der des Lasso Modell.



Absolute Abweichung	0	1	2	3	4	5	6	9	abs(AVG)	AVG	RMSE
Häufigkeit	4	11	6	2	3	1	2	1	2.23	1.03	3.04959

Abbildung 23: Abweichungen von Vorhersage zu echtem Wert

Würde ein kleinerer Wert für alpha gewählt werden beziehungsweise ein größerer Anteil für den Ridge-Teil des Regressionsmodells verwendet werden, würde der RMSE klar höher ausfallen, wie aus der folgenden Graphik hervorgeht.

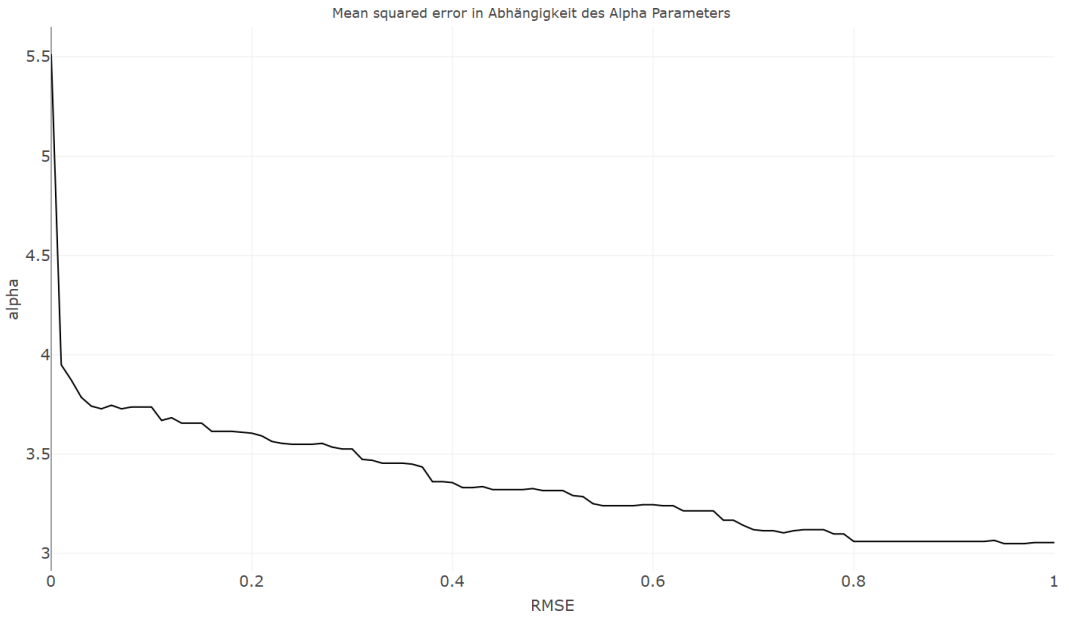


Abbildung 24: Mean squared error für verschiedene alpha Werte

5.1.4 Grundlagen random forest Regression

Der random forest ist eine Weiterentwicklung des Bagging und verwendet Klassifizierungsbeziehungsweise Regressionsbäume. Diese beiden Grundlagen werden zunächst eingeführt.

Bagging

Bagging, kurz für Bootstrap Aggregating, ist ein Verfahren, das verschiedene Regressionsbeziehungsweise Klassifizierungsmodelle mittelt und daraus ein gleich gewichtetes Modell erstellt. Dabei werden B Stichproben mit gleichbleibendem Umfang n gezogen. Für jede Stichprobe wird ein eigenes Modell berechnet, mit Vorhersagewerten $m_i(x)$. Der endgültige Vorhersagewert ergibt sich durch:

$$m^b(x) = \frac{1}{B}m_1(x) + ... + \frac{1}{B}m_b(x) \tag{7}$$

(Breiman, 1996)

CART

CART, kurz für „classification and regression trees“ , ist ein Ansatz für Klassifizierungsoder Regressionsprobleme. Für die Klassifizierung der Daten erstellt der Algorithmus einen Baum, der die Daten je nach Variable in zwei verschiedene Kategorien einteilt. Dabei wird bei jedem Knoten eine Variable gesplittet. Je nach Anzahl der Variablen und Varianz innerhalb der Daten, ergeben sich sowohl unkomplizierte als auch komplexe Bäume. Bei jedem Knoten sollte eine Variable gewählt werden, aus der eine möglichst homogene Aufteilung hervorgeht. Bei Regressionsbäumen wird meistens die quadratische Abweichung von Y zum entsprechenden Mittelwert des Splits minimiert. (Küchenhoff, 2018)

Ein möglicher Baum könnte wie folgt aussehen:

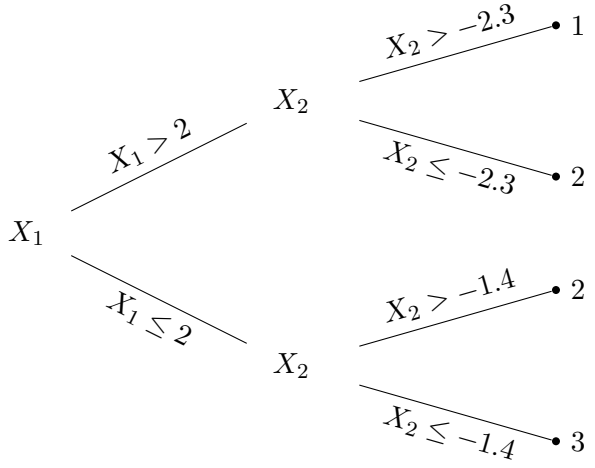


Abbildung 25: Klassifizierungsbaum für zwei Variablen X_1 und X_2 mit drei Klassen

Ein Problem der Bäume, ist ihre hohe Instabilität. Eine kleine Änderung der Daten kann große Veränderungen des Baumes hervorrufen, was dazu führen kann, dass ein neuer Baum berechnet werden muss. Des Weiteren können die Bäume eine sehr hohe Komplexität aufweisen, was einen hohen Aufwand zur Folge haben kann. Außerdem

kann sich die Schätzung von Fehlern schwer gestalten und es kann zu overfitting kommen.

5.1.5 Random forest Regression

Aufbauend auf diesen beiden Methoden ist der random forest. Bei diesem werden mehrere Klassifizierungsbäume „gepflanzt“. Jeder einzelne erhält einen Input Vektor und führt danach eine Klassifizierung durch. Alle dabei entstehenden Klassifizierungen stehen für eine „Stimme“. Am Ende wählt der Wald den Baum, für den die meisten Stimmen existieren.
Der Prozess für das Pflanzen eines Baums sieht wie folgt aus:

1. Befinden sich N Beobachtungen in dem Trainingsdatensatz, ziehe N mal mit Zurücklegen. Die Stichprobe ist der Trainingsdatensatz auf dem basierend der Baum gepflanzt wird.

2. Existieren M Kovariablen, wähle eine Nummer $m << M$. An jedem Knoten werden m Variablen zufällig ausgewählt und der beste Split wird bestimmt, um den Knoten aufzuteilen.

3. Jeder Baum wird so groß wie möglich gezüchtet.

Die Wahl von m ist instrumental für die Fehler Rate des random forest. Wird ein niedriger Wert für m gewählt, so sinkt die Korrelation zwischen zwei Bäumen wodurch auch die Fehler Rate des Waldes sinkt. Analog dazu steigt die Anzahl der Bäume mit schlechter Klassifizierung. Hier muss die „goldene Mitte“ gefunden werden.
Vorteile von random forests sind unter anderem:

- Ist auch bei großen Datensätzen effizient.
- Es werden relevante Variablen innerhalb der Klassifizierung angegeben.
- Missing data kann effektiv geschätzt werden.
- Es findet kein Overfitting statt.

Ein random forest kommt ohne Kreuzvalidierung und Aufteilung in Training und Test aus. Der Grund dafür ist, dass jeder einzelne Baum durch Bootstrapping generiert wird. Dabei wird ca. ein Drittel der Daten aus der Bootstrap Stichprobe ausgelassen und nicht in der Konstruktion des k-ten Baums verwendet. (Breiman & Cutler, 2004)
Um eine random forest Regression in R durchzuführen, kann das Paket *randomForest* benutzt werden. Neben der Formel und dem Datensatz können der Funktion drei Parameter übergeben werden:

- *ntree*: Steht für die Anzahl der gepflanzten Bäume, sollte keine kleine Nummer sein, damit jede input Reihe mehrmals vorhergesagt wird.
- *mtry*: Steht für die Anzahl der Variablen, die an jedem split zufällig gezogen werden.
- *nodesize*: Gibt die minimale Größe der untersten Knoten an. Ein größerer Wert sorgt dafür, dass kleiner Bäume wachsen und die Berechnung schneller ist.

Den größten Einfluss auf das Ergebnis hat *mtry*, die anderen beiden Parameter beeinflussen vor allem die Rechenzeit. Um den Wald zu tunen, kann das *caret* Package benutzt werden. Dieses erlaubt es jedoch lediglich *mtry*, zu tunen. Ein weiteres Paket, mit dem ein random forest getuned werden kann, ist *mlr* (Bischl et al., 2016). Dies ist ein weiteres Paket um Regressions- und Klassifizierungsprobleme zu lösen. Die Funktionsweise ist unterschiedlich zu der von *caret*. Da bei einem random forest zufällig Bäume gepflanzt werden, werden insgesamt 500 Regressionen durchgeführt und dann der Mittelwert der einzelnen RMSE Werte gebildet. Wie das Paket funktioniert, kann dem folgenden Code entnommen werden:

```
#500 Wiederholungen
for (i in 1:500)
{
 #Task erstellen
 regr.task = makeRegrTask(id = "rf", data = dataset,
                          target = "W")
 #Learner erstellen
 lrn = makeLearner("regr.randomForest")
 #Modelle berechnen, alle 150 Daten verwenden
 model = train(lrn, regr.task, subset = 1:150)
 #Vorhersagen
 preds = predict(model, regr.task, 1:150)
 true = preds$data$truth
 pred = round(preds$data$response)
 rmse = c(rmse, sqrt(mean((true-pred)^2)))
}
```

Während der Schleife laufen noch zusätzliche Funktionen zur Messung der Wichtigkeit einzelner Variablen, diese sind der Einfachheit halber nicht im Codebeispiel enthalten.
Für den Parameter *ntry* wird der Wert 44 verwendet und für *ntree* wird 500 verwendet. Für die Berechnung des RMSE werden die durchschnittliche Anzahl an Siegen einer Mannschaft berechnet und dann mit der realen verglichen. Daraus ergibt sich der Wert 1.699. Bevor eine tiefergehende Analyse der Regressions folgt, sei zunächst die Höhe des Modellfehlers in Abhängigkeit Anzahl der Bäume gegeben. Vor allem für eine geringe Anzahl an Bäumen ist der Modellfehler hoch. Der Fehler sinkt mit einer erhöhten Anzahl der Bäume und ab einem Wert von 100 findet lediglich eine geringere Verbesserung des Fehlers statt.

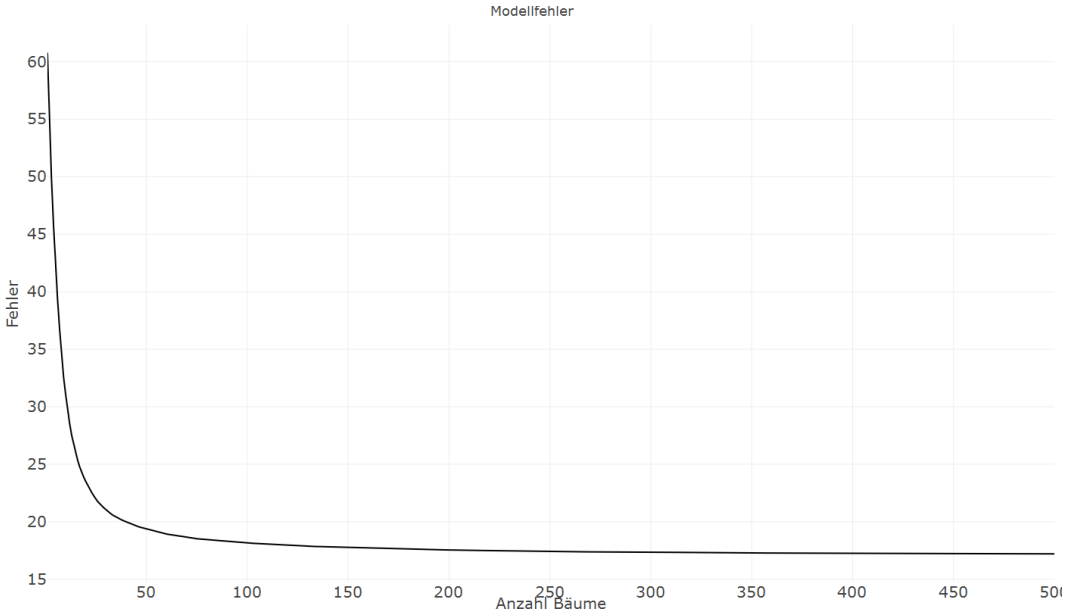


Abbildung 26: Modellfehler des random forest

Bei einem random forest können unter anderem die Verteilung der minimalen Tiefe, die Bedeutsamkeit verschiedener Variablen, Interaktionen zwischen Variablen und Vorhersagen in Abhängigkeit von Variablen untersucht werden. Dabei ist vor allem das R-Paket *randomForestExplainer* zu empfehlen. Um zu verstehen, was die Tiefe eines Baumes ist, ist ein Blick auf einen Baum hilfreich:

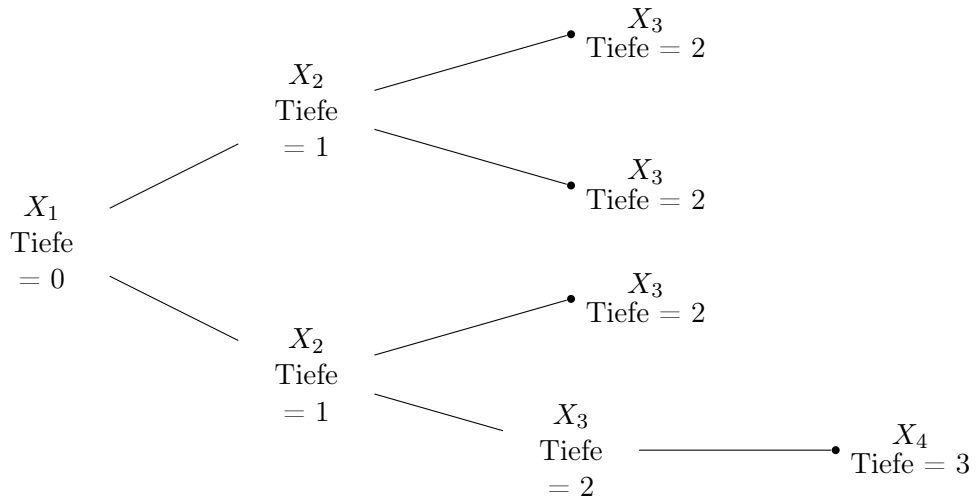


Abbildung 27: Die Tiefe von Baumknoten

Die Tiefe eines Knoten ist also ein Maß dafür, wie viele Knoten sich über diesem befinden. Mit der Verteilung der minimalen Tiefe kann betrachtet werden, wann eine Variable für einen Split verwendet wird. Auf der x-Achse ist die Anzahl der Bäume aufgetragen. In dem folgenden Graph werden die zehn Variablen, die am häufigs-

ten für eine Teilung verwendet werden, dargestellt. TEAM_VORP wird jedes Mal benutzt und ist bei über 70.000 Bäumen der initiale Knoten und bei über 150.000 Bäumen entweder initialer Knoten oder erster Kindknoten. ORtg und DRtg sind die einzigen anderen Variablen, die in mehr als 50% der Bäume vorkommen. Laut dieser Graphik sind die beiden Ratings und TEAM_VORP die bedeutendsten Variablen, gefolgt von FG%_OPPONENT, TS% und eFG%. Insgesamt existieren nur vier Variablen, die im Schnitt vor dem zehnten Kindknoten gewählt werden.

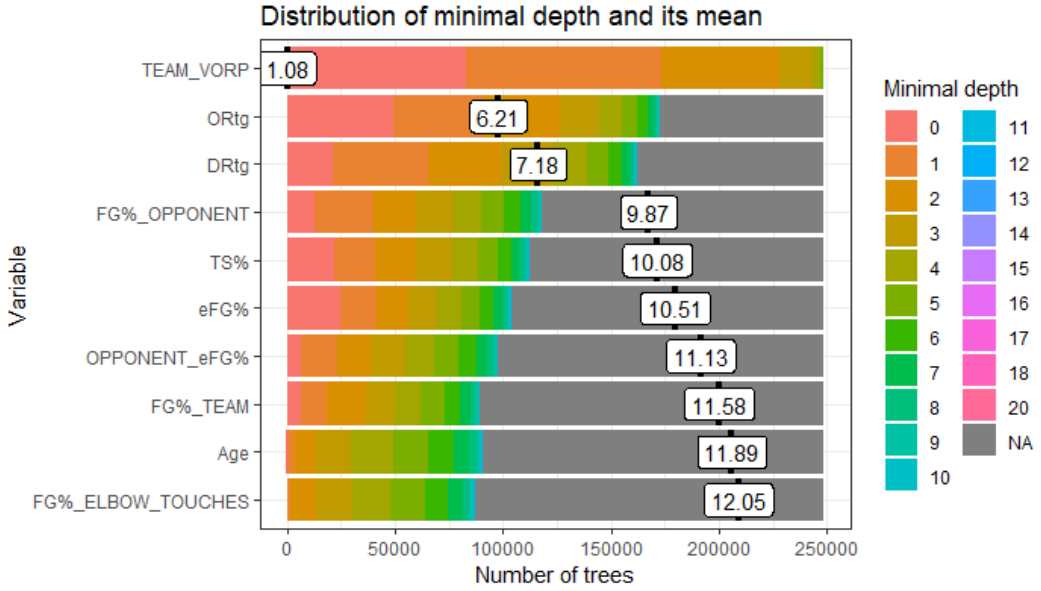


Abbildung 28: Verteilung der minimalen Tiefe

Das forest Objekt, das von der Funktion randomForest zurückgegeben wird, enthält zwei Variablen anhand deren gemessen werden kann, wie beachtenswert eine Einflussvariable ist:

- *IncNodePurity*: Gibt an, um wie viel sich die Unreinheit aller Knoten verringert durch Durchführung einer Teilung mit einer bestimmten Variable. Verwendet die Residuenquadratsumme für die Berechnung.
- *%IncMSE*: Ist robuster und informativer als IncNodePurity. Gibt an wie sich der MSE der Vorhersagen durch die Permutation einer Variable erhöht. In anderen Worten: Es gibt an, wie sich die Genauigkeit eines Modells durch Auslassen einer Variable erhöht.

Bei den Modellen wird nur die *IncNodePurity* berechnet. Die drei bedeutendsten Variablen sind wieder TEAM_VORP und die beiden Ratings. Insgesamt finden sich acht Variablen aus der letzten Graphik wieder, lediglich Age und FG%_ELBOW_TOUCHES werden durch 2P%_TEAM beziehungsweise
2P%_OPPONENT ersetzt.

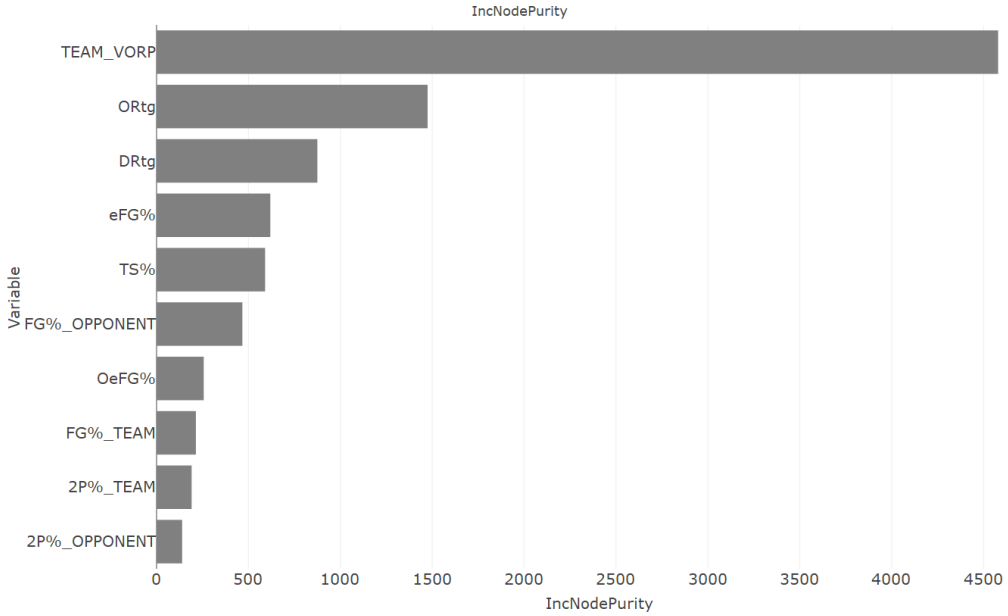
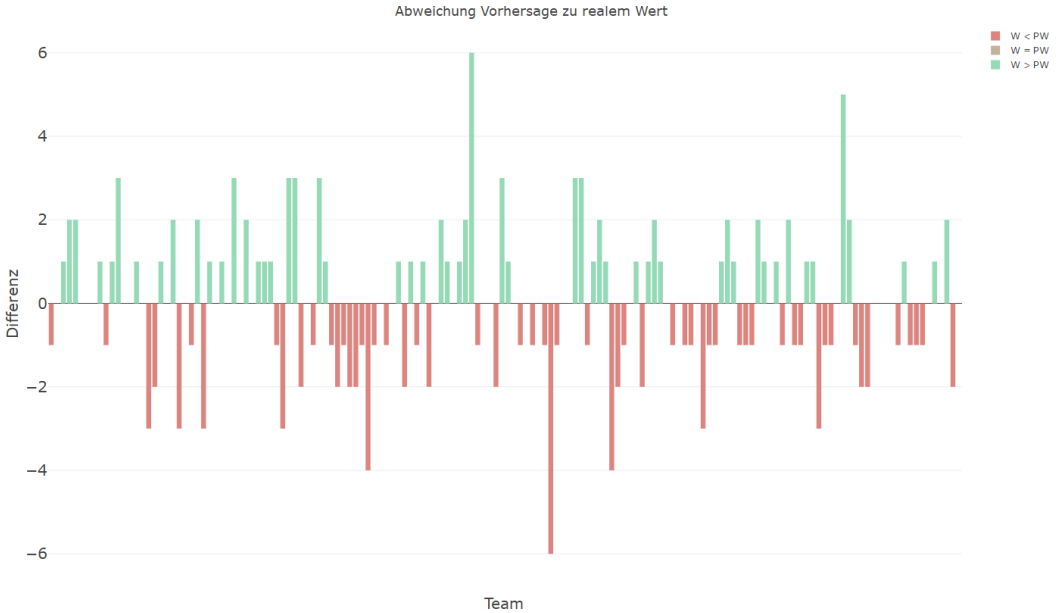


Abbildung 29: Die zehn wichtigsten Variablen nach beiden Kriterien

Da keine Aufteilung in Training und Test stattfindet, wird die Anzahl der Siege von allen 30 Mannschaften in den letzten fünf Saisons geschätzt. Die Abweichungen zwischen Prognose und realem Wert sind in folgender Graphik gegeben:



Absolute Abweichung	0	1	2	3	4	5	6	abs(AVG)	AVG	RMSE
Häufigkeit	38	66	27	14	2	1	2	1.25	-0.00667	1.691

Abbildung 30: Abweichungen von Vorhersage zu echtem Wert

Bei über 25% der Mannschaften wird die Anzahl der Siege korrekt vorhergesagt

und bei mehr als zwei Dritteln weicht die Vorhersage um ±1 ab. Bei lediglich fünf Mannschaften ist die Abweichung größer als drei. Wie bereits bei den letzten beiden Modellen wird mit einem Datensatz experimentiert, der die Variable TEAM_VORP nicht enthält. Dabei fallen die Ergebnisse jedoch distinkt schlechter aus.

5.1.6 Überblick über die bisherigen Modelle

In diesem kurzen Abschnitt werden die bisherigen Modelle noch einmal kurz zusammengefasst und verglichen. Dabei werden nicht nur Modelle mit einer zufälligen Einteilung sondern auch Modelle mit einer bewusste Einteilung vorgestellt. Bei der bewussten Einteilung findet jeweils ein Training basierend auf den letzten vier Saisons statt. Für den Test wird die aktuelle Saison verwendet. Da diese Modelle jedoch jedes Mal schlechter abschneiden als die zufällig aufgeteilten, werden diese nicht extra vorgestellt. Im Anhang befindet sich die Tabelle der aktuellen Saison und die Prognosetabellen der vier Modelle. Für die Lasso Regression wird das Modell mit dem Datensatz ohne TEAM_VORP mit einbezogen, da dieses besser abschneidet als das mit der Variable.

Zufällige Aufteilung

Methode	RMSE	R^2
Linear	3.629	0.9356
Lasso	3.055	0.9495
Elastic net	3.050	0.9489

Bewusste Aufteilung

Methode	RMSE	R^2
Linear	3.873	0.9349
Lasso	3.469	0.9538
Elastic net	3.454	0.9539
Random forest	1.699	0.9818

Bei allen Modellen schneidet die zufällige Aufteilung besser ab als die bewusste Aufteilung. Die linearen Modelle haben distinkt höhere RMSE-Werte als die restlichen Modelle. Bei den restlichen Modellen wird durch die gewollte Aufteilung zum einen mehr Varianz erklärt, zum anderen ergeben sich aber auch höher RMSE-Werte. Das mit Abstand beste Modell ist der random forest. Der RMSE ist fast halb so gering wie bei der elastic net Regression und für R^2 wird der höchste Wert erreicht.

5.2 Modelle für die Vorhersage einzelner Spiele

Ein zentraler Aspekt bei Wettanbietern ist das Wetten auf den Ausgang einzelner Spiele. Dabei kann zum Beispiel auf die gewinnende Mannschaft oder die Anzahl der erzielten Punkte einer Mannschaft getippt werden. Je nachdem welche Teams spielen gibt es verschiedene Quoten, die die Auszahlung bei einer richtigen Wette vorhersagen. Diese sind nicht zufällige Werte sondern richten sich meistens nach der erwarteten Leistung einer Mannschaft. Hinter all diesen Quoten befinden sich oftmals Regressionsmodelle. Das mögliche Aussehen solcher Modelle soll in diesem Teilabschnitt thematisiert werden.
Die verwendeten Datensätze sind die Spielpläne der einzelnen Saisons. In jeder Saison finden 1230 Spiele statt. (Ausnahme 2011 mit 990 Spielen) Für jede Saison werden ca. die ersten 125 Spiele entfernt. Ein Grund hierfür ist, dass für die ersten Spiele einer Saison die Einflussvariablen noch nicht existieren, wodurch eine Entscheidung einem Münzwurf gleicht. Für jede der letzten fünf Saisons wird ein Modell, basierend auf den drei Saisons davor, berechnet und dann die vorhergesagten Ergebnisse der einzelnen Spiele verglichen. Bei den Einflussgrößen handelt es sich immer um die realen Werte, das heißt die Prognose eines Spiels von Mitte Januar hat keinen Einfluss auf die Prognose eines Spiels im März.
Die Einflussgrößen sind in allen Modellen identisch, sofern keine Selektion von dem Modell durchgeführt wird. Da der computationale Aufwand für die Vorhersage einzelner Statistiken wie Assists, Rebounds, usw. zu groß ist, wird lediglich die Anzahl der Punkte, die von den jeweiligen Mannschaften erzielt wird, vorhergesagt und dann adäquat entschieden.

Einflussgrößen
Die Einflussgrößen beziehen sich allesamt auf die Leistung einer Mannschaft in einer Saison. Ein offensichtlicher Leistungsindikator ist die bisherige Anzahl an Siegen einer Mannschaft. Bei einem Basketball Spiel kommt es darauf an, mehr Punkte zu erzielen als der Gegner. Sinnvolle Kovariablen sind somit die Anzahl der Punkte, die eine Mannschaft pro Spiel erzielt beziehungsweise erlaubt. Ein anderer Effekt, der im Basketball häufig beobachtet wird, ist *home court advantage*. Laut diesem spielen Mannschaften in ihrer eigenen Arena besser. Zurückzuführen ist der Effekt auf Faktoren wie die Vertrautheit der heimischen Einrichtungen oder die Unterstützung der Fans. (Moore & Brylinsky, 1995) Aufgrund des Effekts sind zwei weitere Einflussgrößen der Gewinnanteil der heimischen Mannschaft daheim und der Gewinnanteil der gastierenden Mannschaft als Gast. Zum Schluss soll noch ein Trendeffekt eingebaut werden. Dieser gibt an, ob eine Mannschaft in letzter Zeit mehr Siege oder Niederlagen geholt hat. Trifft eine Mannschaft, die ihre letzten zehn Spiele gewonnen hat, auf eine Mannschaft, die ihre letzten zehn Spiele verloren hat, ist eher von einem Anhalten der Siegesserie der ersten Mannschaft auszugehen.
Eine Übersicht über die Kovariablen:

1. Der bisherige Anteil an Siegen der Mannschaften.
2. Der Anteil an Siegen einer Mannschaft in den letzten n Spielen. (Flexibler Parameter)
3. Die Anteil an Siegen der Heimmannschaft als Heim beziehungsweise der Gastmannschaft als Gast.

4. Die Anzahl der durchschnittlich pro Spiel erzielten Punkte der Mannschaften.

5. Die Anzahl der durchschnittlich pro Spiel erlaubten Punkte der Mannschaften.

Evaluierung der Modelle
Um die Modelle zu evaluieren, braucht es eine Vergleichslinie. Ein Spiel hat zwei Ausgänge:

1. Mannschaft A gewinnt

2. Mannschaft B gewinnt

Dies ähnelt einem Münzwurf und somit sollte nur mit Raten eine Genauigkeit von 50% erreicht werden. Eine weitere Möglichkeit ist die Entscheidung basierend auf den bisherigen Ergebnissen der Saison. Zwei mögliche Methoden sind:

1. Das Team wählen, das bis dato häufiger gewonnen hat. (Relative Häufigkeit)

2. Das Team wählen, bei dem die Differenz zwischen erzielten und zugelassenen Punkten höher ist.

Bei Verwendung dieser Methoden ergeben sich folgende Werte für die letzten fünf Saisons:

Saison	Höhere Sieg%	Höhere Differenz
2013-14	0.696	0.641
2014-15	0.701	0.649
2015-16	0.696	0.639
2016-17	0.641	0.594
2017-18	0.678	0.584
Durchschnitt	0.6824	0.6214

Tabelle 8: Vorhersagegenauigkeit mit simplen Methoden

In ca. 68% der Fälle ist der Tipp auf die erfolgreichere Mannschaft der richtige Tipp. Somit ist das Ziel, ein Modell zu finden, das diese Zahl übersteigt.

5.2.1 Bivariate lineare Regression

Als Grundlinie für den Vergleich der Modelle ist wieder ein lineares Modell von Interesse. Die multivariate lineare Regression ist einer Erweiterung des univariaten Modells, das in Kapitel 5.1.1 verwendet wurde. Im Vergleich zu diesem Modell existieren mehrere abhängige Variablen und die gleichen Prädiktoren. Es wird angenommen, dass jede Zielgröße einem eigenen Regressionsmodell folgt:

$$\begin{aligned} y_1 &= \beta_{01} + \beta_{11}x_1 + ... + \beta_{r1}x_r + \varepsilon_1 \\ y_2 &= \beta_{02} + \beta_{12}x_1 + ... + \beta_{r2}x_r + \varepsilon_2 \\ y_p &= \beta_{0p} + \beta_{1p}x_1 + ... + \beta_{rp}x_r + \varepsilon_p \end{aligned} \tag{8}$$

Der Störterm ε hat den Erwartungswert 0 und die Varianzmatrix $\Sigma_{p\times p}$. Die Matrix der abhängigen Variablen ergibt sich durch:

$$y_{n\times m} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1p} \\ y_{21} & y_{22} & \cdots & y_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{np} \end{pmatrix} = \begin{pmatrix} y_{(1)} & y_{(2)} & \cdots & y_{(n)} \end{pmatrix} \tag{9}$$

$y_{(i)}$ ist ein Vektor der n Messungen der i-ten Variable beinhaltet. Des weiteren ist:

$$\beta_{(r+1)\times m} = \begin{pmatrix} \beta_{01} & \beta_{02} & \cdots & \beta_{0m} \\ \beta_{11} & \beta_{12} & \cdots & \beta_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{r1} & \beta_{r2} & \cdots & \beta_{rm} \end{pmatrix} = \begin{pmatrix} \beta_{(1)} & \beta_{(2)} & \cdots & \beta_{(m)} \end{pmatrix} \tag{10}$$

$\beta_{(i)}$ sind die $(r+1)$ Koeffizienten des Modells für die i-te Variable. Die Matrix der Fehler ist analog zu der aus (8):

$$\varepsilon = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \cdots & \varepsilon_{1p} \\ \varepsilon_{21} & \varepsilon_{22} & \cdots & \varepsilon_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n1} & \varepsilon_{n2} & \cdots & \varepsilon_{np} \end{pmatrix} = \begin{pmatrix} \varepsilon_{(1)} & \varepsilon_{(2)} & \cdots & \varepsilon_{(n)} \end{pmatrix} = \begin{pmatrix} \varepsilon_1^T \\ \varepsilon_2^T \\ \vdots \\ \varepsilon_n^T \end{pmatrix} \tag{11}$$

Der p-dimensionale Vektor ε_j^T beinhaltet die Residuen für alle p Zielvariablen für die j-te Beobachtung. Die Regressionsgleichung, mit $x_{n\times(r+1}$ als Designmatrix, lautet:

$$y_{n\times p} = x_{n\times(r+1)}\beta_{(r+1)\times p} + \varepsilon_{n\times p} \tag{12}$$

(Maitra, 2012)

Das Modell

Die beiden Zielvariablen folgen einer Normalverteilung, was aus dem folgenden Q-Q-Plot hervorgeht:

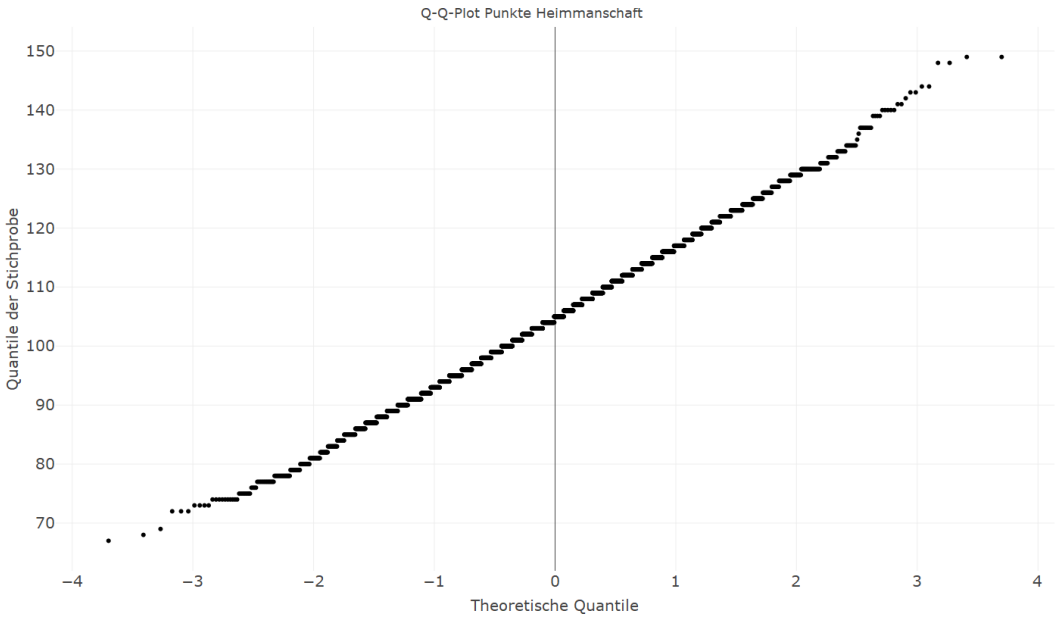


Abbildung 31: Die Anzahl der Punkte folgt einer Normalverteilung

Analog dazu folgt auch die Anzahl der Punkte der Gäste einer Normalverteilung. (siehe Anhang) Die Regression kann in R ohne spezielle Pakete durchgeführt werden, durch folgenden Befehl:

```
model = lm(cbind(PTS_HOME_SCORED, PTS_GUEST_SCORED) ~
            PTSHOME+PTSROAD+`W%HOME`+
            `W%ROAD`+`W%10HOME`+`W%10ROAD`+
            `W%HOMEHOME`+`W%ROADROAD`,
                 data = testSet)
```

Nachdem für jedes einzelne Spiel die erzielten Punkte der Mannschaften vorhergesagt werden, wird entsprechend entschieden. Es werden Modelle für die letzten 1-10 Spiele der Mannschaften betrachtet und mit den obigen Werten verglichen. Die Vorhersagegenauigkeit ist am geringsten bei Verwendung der Ergebnisse der letzten fünf Spiele. Die höchste Genauigkeit ergibt sich, wenn der Ausgang der letzten zehn Spiele relevant ist.

Saison	Höhere Sieg%	Bivariate lineare Regression
2013-14	0.696	0.683
2014-15	0.701	0.705
2015-16	0.696	0.723
2016-17	0.641	0.666
2017-18	0.678	0.683
Durchschnitt	0.6824	0.692

Tabelle 9: Vorhersagegenauigkeit der bivariaten linearen Regression

Es ist eine Verbesserung um ungefähr ein Prozent zu erkennen, was 52 Spielen innerhalb der fünf Saisons entspricht. Eine weitere Möglichkeit um die Leistung des Algorithmus zu evaluieren, ist es die Vorhersagegenaugikeiten in bestimmten Situationen betrachten. Vor einem Spiel existieren drei mögliche Ausgangssituationen:

1. Die Heimmannschaft hat einen höheren Siegesanteil.
2. Die Gastmannschaft hat einen höheren Siegesanteil.
3. Die Mannschaften haben einen identischen Siegesanteil.

Die Entscheidungshäufigkeiten in Abhängigkeit dieser Situationen sind in den folgenden Tabellen dargestellt. Dabei sind auf der x-Achse die Vorhersagen dargestellt und auf der y-Achse die wahren Häufigkeiten.

H>G	Heim	Gast	
Heim	2075	6	2081
Gast	631	3	634
	2706	9	2715

H<G	Heim	Gast	
Heim	486	603	1089
Gast	436	1220	1656
	922	1823	2745

H=G	Heim	Gast	
Heim	45	0	45
Gast	30	0	30
	75	0	75

Tabelle 10: Vorhersagegenauigkeiten unter bestimmten Voraussetzungen

Auffallend ist, dass in Situation 1 und 3 fast ausschließlich Siege für die Heimmannschaft vorhergesagt werden. Hat die Gastmannschaft einen höheren Siegesanteil, sind die Vorhersagen ausgeglichener. Von den 1089 Siegen der Heimmannschaft werden circa 44.6% korrekt prognostiziert und von den 1656 Siegen der Gastmannschaft werden ungefähr 73.7% der Siege richtig vorhergesagt. Insgesamt werden für den ersten Fall 76.5% richtige Vorhersagen getätigt, 62.1% für den zweiten Fall und 60% für den letzten Fall.

5.2.2 Generalisierte lineare Regression

Bei einem linearen Modell wird der bedingte Erwartungswert $\mathrm{E}(Y|X)$ mit Hilfe der Linearkombination $x^T\beta$ modelliert. Dabei findet keine Verknüpfung zwischen Erwartungswert und Prädiktor statt. Bei einem generalisierten linearen Modell, existiert diese Verknüpfung. Dabei gilt:

1. Für den Erwartungswert: $\mu = h(\eta)$
2. Für den Prädiktor: $\eta = g(\mu)$

Dabei wird $h(\cdot)$ als Reponse-Funktion und $g(\cdot)$ als Link-Funktion bezeichnet. Des weiteren lässt sich die Verteilung der Zielvariable als einparametrische Exponentialfamilie darstellen:

$$f(y_i|x_i) = \exp\left(\frac{y_i\vartheta_i - b(\vartheta_i)}{\phi_i}w_i + c(y_i, \phi, w_i)\right) \tag{13}$$

Damit muss ein GLM zwei Annahmen erfüllen:

1. Strukturannahme: Verknüpfung von $x^T\beta$ und dem Parameter durch Response-/Linkfunktion.
2. Verteilungsannahme: Dichte von $y|x$ folgt der Form einer einparametrischen Exponentialfamilie.

(Mayr, 2017)
Nachdem bei der linearen Regression die erzielten Punkte der beiden Mannschaften modelliert wurden, soll jetzt direkt der Sieger modelliert werden. Die Responsevariable *Winner* ist binär, intuitiv wäre eine logistische Regression ein sinnvolles Modell. Bei einer logistischen Regression ist die Zielvariable Bernoulli Verteilt, das heißt

$y_i|x_i \sim \mathrm{B}(\pi(x_i))$. Response- und Linkfunktion sind gegeben durch:

$$h(\eta) = \frac{\exp x^T\beta}{1+\exp x^T\beta}$$
$$g(\pi) = \log\frac{\pi}{1-\pi}$$

Bei der Bernoulli Verteilung handelt es sich auch um eine Exponentialfamilie:

$$\begin{aligned}
\gamma(y|x_i) &= \pi(x_i)^{y_i} + (1-\pi(x_i))^{1-y_i} \\
&= \exp\left(y_i - \log(\pi(x_i)) + (1-y_i)\log(1-\pi(x_i))\right) \\
&= \exp\left(y_i \cdot \log\frac{\pi(x_i)}{1-\pi(x_i)} + \log(1-\pi(x_i))\right) \\
\vartheta_i &= \log\frac{\pi(x_i)}{1-\pi(x_i)} \\
b(\vartheta_i) &= -\log(1-\pi(x_i)) \\
\phi_i &= 1 \\
w_i &= 1 \\
c(y_i,\phi,w_i) &= 0
\end{aligned} \tag{14}$$

Damit sind beide Annahmen erfüllt. Für die Implementierung in R muss die Zielvariable erst in die Form eines Faktors umgewandelt werden, ehe die Regression durchgeführt wird. Das Modell wird wieder mit Hilfe des *caret* Pakets getuned.

```
f=as.formula("Winner~PTSHOME+PTSROAD+W.Home+W.Road+
              Last.Home+Last.Road+W.HomeHome+W.RoadRoad+
              W.Home:W.HomeHome+W.Road:W.RoadRoad")
fit = train(form = f, data = dataset, method = "glm",
            trControl = trainControl(method = "cv",
            number = 10),family = binomial(link = "logit"))
```

Für die Regression wird unter Einbezug der letzten zwei Spiele die größte Genauigkeit erreicht. Für die einzelnen Saisons ergibt sich:

Saison	Höhere Sieg%	Logistische Regression
2013-14	0.696	0.696
2014-15	0.701	0.714
2015-16	0.696	0.709
2016-17	0.641	0.672
2017-18	0.678	0.703
Durchschnitt	0.6824	0.6988

Tabelle 11: Vorhersagegenauigkeit der logistischen Regression

Aufnahme einer Talentvariable

Ein Problematik der bisherigen Modelle ist es, dass nicht bekannt ist, wie viele Stars eine Mannschaft hat beziehungsweise wie viel Talent. Die Aufnahme von TEAM_VORP wäre zu aufwendig, da der Wert vor jedem Spiel neu berechnet werden müsste, was ein zu hoher computationaler Aufwand ist. Am besten wäre also

eine konstante Zahl.
Jedes Jahr ist im Februar das sogenannte „All Star Game“. Dabei handelt es sich um ein Spiel, bei dem die besten Spieler der Liga gegeneinander Spielen. Die Teams setzen sich jeweils aus zwölf Spielern aus dem Osten und Westen zusammen. Um Teil der Mannschaft zu werden, gibt es zwei Möglichkeiten:

1. Die fünf Startspieler werden von den Fans gewählt. Dabei wird in zwei grundlegende Positionen unterschieden, Guards und Forwards. Die beiden beziehungsweise drei Spieler mit den meisten Stimmen werden Startspieler. Das Problem ist, dass oft die beliebtesten Spieler Startspieler sind. Deswegen werden seit der letzten Saison die Stimmen der Fans gleich gewichtet mit den Stimmen der Trainer.

2. Die Bankspieler werden von den Trainern gewählt.

Eine dritte Möglichkeit existiert, falls sich ein Spieler verletzt. In diesem Fall wird ein Ersatzspieler von den Trainern gewählt. Für die Datensätze wird überprüft, welche Mannschaft wann wie viele All-Stars beherbergt. Dabei werden die Zahl der Spieler immer Mitte Februar, also nach dem Spiel, aktualisiert. Die Variable ist jedoch nicht perfekt, da auch All-Stars sich einmal verletzen oder aus anderen Gründen ein Spiel verpassen. Für die logistische Regression kann immerhin eine sehr geringe Verbesserung, die zwölf korrekt prognostizierten Spielen innerhalb der fünf Saisons entspricht, erreicht werden.

Saison	Höhere Sieg%	Logistische Regression
2013-14	0.696	0.693
2014-15	0.701	0.713
2015-16	0.696	0.714
2016-17	0.641	0.682
2017-18	0.678	0.703
Durchschnitt	0.6824	0.701

Tabelle 12: Vorhersagegenauigkeit der logistischen Regression

Die Vorhersagegenauigkeit der logistischen Regression ist nochmals um fast 1% größer, als bei der linearen Regression. Wie bereits bei dem linearen Modell, werden in Spielen, in denen die Heimmannschaft einen höheren Siegesanteil hat, fast ausschließlich Siege für diese vorhergesagt. Insgesamt werden in dieser Situation 76.6% der Sieger korrekt prognostiziert. Die logistische Regression schnitt, im Vergleich mit der linearen Regression, besser ab, wenn die Gastmannschaft einen höheren Anteil an Siegen hat, als die Heimmannschaft. Bei dem linearen Modell werden 62.1% korrekt vorhergesagt, bei dem logistischen 64%. Treffen zwei gleich gute Teams aufeinander, werden 56% korrekte Vorhersagen getätigt.

H>G	Heim	Gast	
Heim	2064	17	2081
Gast	618	16	634
	2682	33	2715

H<G	Heim	Gast	
Heim	439	650	1089
Gast	338	1318	1656
	777	1968	2745

H=G	Heim	Gast	
Heim	40	5	45
Gast	28	2	30
	68	7	75

Tabelle 13: Vorhersagegenauigkeiten unter bestimmten Voraussetzungen

5.2.3 Vergleich der Modelle

Von den vorgestellten Modellen schneidet die logistische Regression am besten ab. Neben den beiden Modellen wird noch eine Klassifizierung mittels einem random forest versucht. Es kann jedoch keine weitere Verbesserung mehr erreicht werden. Das größte Problem ist, dass es zu viele Variablen gibt, die schwer in Modelle aufgenommen werden können. Darunter fallen zum Beispiel Verletzungen oder Spiele, in denen Teams ihre Stars nicht spielen lassen, damit sie für die Playoffs fit sind. Eine Berechnung direkt vor Spielbeginn, in die alle möglichen Variablen aufgenommen werden, wäre vermutlich genauer, ist aber mit einem zu hohen Aufwand verbunden. Die Erstellung dieser Regressionsmodelle ist inspiriert von dem Paper NBA Oracle von Matthew Beckler, aus diesem Grund wird noch ein kurzer Vergleich der jeweiligen Ergebnisse durchgeführt. Beckler verwendete lineare Regression, logistische Regression, support vector machines und artificial neural networks und berechnete ebenfalls Ergebnisse für fünf Saisons(1992/93-1996/97). Die besten Ergebnisse erreicht er mit der linearen Regression, mit einer Vorhersagegenauigkeit in Höhe von 0.7009. Er verglich seine Resultate mit denen von Websites, Experten, einfachen Methoden und Expertenmeinungen. Sein finaler Vergleich sah wie folgt aus:

Methode	Zufall	Sieg%	Websites	Andere Modelle	Experten	Oracle
Genaugikeit	50%	62%	65%	70%	71%*	Bis zu 73%

(Beckler, Wang & Papamichael, 2013)
Auffällig ist zunächst einmal, dass bei der Wahl des Teams mit dem höheren Sieganteil nur 62% korrekt vorhergesagt werden. Das könnte darauf hindeuten, dass die Liga zu diesem Zeitpunkt ausgeglichener war, während es heute wenige Topteams und viele schlechte Teams gibt. Experten wird eine Vorhersagegenauigkeit von 71% zugewiesen, diese attributiert er jedoch dem Fakt, dass Experten nicht jedes Spiel vorhersagen, sondern Begegnungen zwischen gleichstarken Teams auslassen und so ihre Quoten inflationiert sind. Seiner eigenen Methode gibt er eine Genauigkeit von bis zu 73%, jedoch konnte er diese nur für eine Saison erreichen. Bei seinen linearen Modellen erreichte er in je zwei der fünf Saisons eine Vorhersagegenauigkeit von 70% oder mehr. Mit dieser Logik könnte der logistischen Regression auch eine Vorhersagegenauigkeit von bis zu 72% zugeschrieben werden. Des weiteren haben die Modelle von Beckler eine sehr geringe Varianz im Bezug auf die Vorhersagegenauigkeit in für die einzelnen Saisons. Bei den hier berechneten Modellen liegt eine größere Varianz vor, da die Saison 2016-17 immer die geringste Vorhersagegenauigkeit vorweist,

teilweise bis zu 4% geringer als die beste Vorhersage. Würde diese Saison entfernt werden, wäre eine Vorhersagegenauigkeit von 70.7% bei ca. 4400 Spielen über vier Jahre hinweg erreicht.

Saison	Höhere Sieg%	Linear	Logistisch	Random forest
2013-14	0.696	0.680	0.693	0.674
2014-15	0.701	0.704	0.713	0.688
2015-16	0.696	0.720	0.714	0.697
2016-17	0.641	0.668	0.682	0.668
2017-18	0.678	0.687	0.702	0.678
Durchschnitt	0.6824	0.6918	0.701	0.681

Tabelle 14: Vorhersagegenauigkeit der jeweiligen Modelle

5.3 Simulation einer Saison

Bei vielen Wettanbietern ist es möglich, auf Spiele zu tippen, die in der entfernten Zukunft liegen. Die Vorhersage eines Spiels in drei Wochen ist komplizierter als die Vorhersage eines Spiels am selben Tag, dementsprechend kann auch mehr Geld gewonnen werden. Das Problem dabei ist, dass für die Vorhersage eines Spiels die vorherigen Leistungen der beiden Mannschaften und die vorherigen Leistungen derer Gegner einbezogen werden müssen. Um zu prognostizieren, wie diese Spiele ausgehen, müssen die Spiele davor prognostiziert werden, eine Simulation wird also notwendig.
Die Simulation:
Für die Simulation einer Saison werden je nach Verfahren entweder ein oder zwei Modelle aufgestellt, die darauf abzielen die erzielten Punkte einer Mannschaft vorherzusagen. Für die letzten drei Saisons werden Simulationen durchgeführt und mit den realen Saisons verglichen.

```
Chronologische Durchlaufung der Saison mittels Schleife:
     1. Ermitteln der bisherigen Statistiken der
        beiden Mannschaften
     2. Vorhersage der Sieger, entweder direkte Vorhersage
        oder durch Vorhersage der Punkte
     3. Statistiken aktualisieren und zurück zu Schritt 1
```

Die Vergleichsbasis für die Simulationen wird wieder durch die Entscheidung zugunsten der höheren Siegeshäufigkeit gebildet. Ist diese bei zwei Mannschaften gleich hoch, entscheidet der Zufall.

Saison	Höhere Sieg%
2015-16	0.646
2016-17	0.593
2017-18	0.561
Durchschnitt	0.6

Tabelle 15: Vorhersagegenauigkeit der Simulation

Ein Problem bei dieser Methode ist jedoch, dass vor allem Mannschaften die fast alle Spiele gewonnen/verloren haben, immer gewinnen beziehungsweise verlieren. Ein gutes Beispiel dafür sind die Golden State Warriors und Philadelphia 76ers. Diese hatten zum Beginn der Simulation der Saison 2015-16 nur Siege beziehungsweise Niederlagen. Am Ende der Saison hatten die Golden State Warriors mit 73 Siegen den Rekord für die meisten Siege in einer Saison aufgestellt, während Philadelphia in der Saison lediglich zehn Siege holen konnte. Somit gibt es für die beiden Teams zusammen nur 19 falsche Prognosen, was zu einer höheren Vorhersagegenauigkeit beiträgt.

5.3.1 Bivariate lineare Regression

Im letzten Kapitel wurde die multivariate multiple lineare Regression eingeführt, um die Anzahl der erzielten Punkte der beiden Mannschaften vorherzusagen. Dieser Ansatz wird erneut durchgeführt, mit folgendem Modell:

```
model = lm(cbind(PTS_HOME_SCORED, PTS_GUEST_SCORED) ~
           PTSHOME:PTSROAD+PTSAHOME:PTSAROAD+`W%HOME`+
           `W%ROAD`+`W%1HOME`+`W%1ROAD`+
           `W%HOMEHOME`+`W%ROADROAD`,
                data = testSet)
```

Die Vorhersagegenauigkeit der einzelnen Saisons kann der nachfolgenden Graphik entnommen werden:

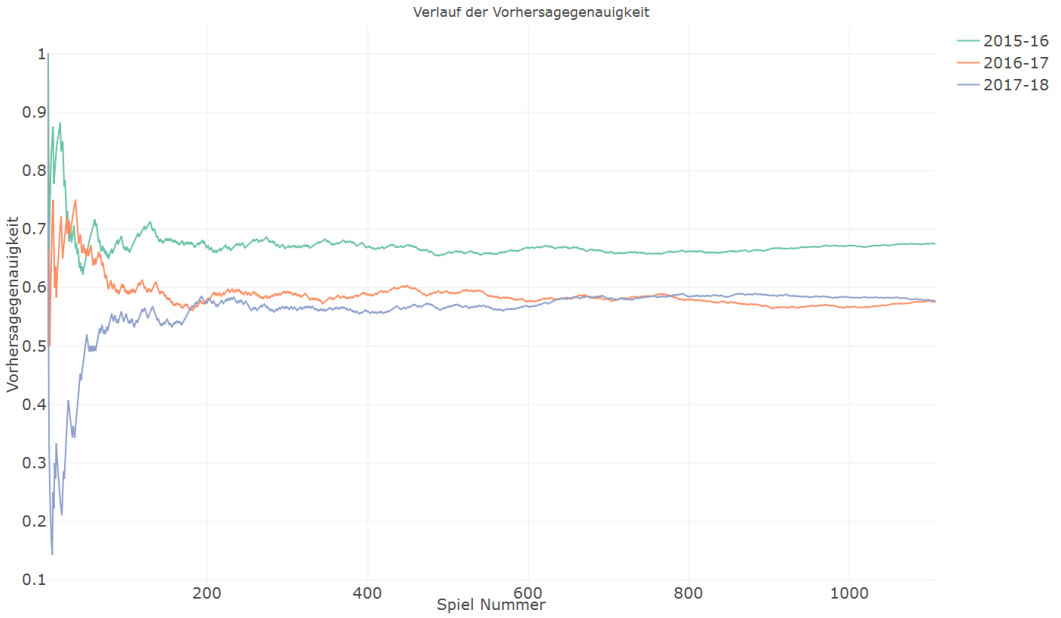


Abbildung 32: Vorhersagegenauigkeiten der drei Saisons mit bivariater linearer Regression

Die Saison 2015-16 wird am genauesten simuliert, 67.6% der Ergebnisse sind auch in der Realität eingetreten. Für die anderen beiden Saisons werden ungefähr 57.6% beziehungsweise 60.3% korrekt prognostiziert. Damit liegen diese Werte über denen aus Tabelle 13. Insgesamt werden 60.9% der 3321 Spiele korrekt vorhergesagt. Als Alternative zu der bivariaten linearen Regression werden wieder zwei einzelne lineare Regressionen durchgeführt, eine für jede Mannschaft. Die Gleichungen sehen wie folgt aus:

$$\begin{aligned} PTS\_HOME\_SCORED &= \beta_0 + \beta_1 x_{PTSHOME} + \beta_2 x_{PTSAROAD} + \beta_3 x_{W\%HOME} \\ &\quad + \beta_4 x_{W\%8HOME} + \beta_5 x_{W\%HOMEHOME} \\ PTS\_GUEST\_SCORED &= \beta_0 + \beta_1 x_{PTSROAD} + \beta_2 x_{PTSAHOME} + \beta_3 x_{W\%ROAD} \\ &\quad + \beta_4 x_{W\%8ROAD} + \beta_5 x_{W\%ROADROAD} \end{aligned}$$

Durch diese Methode kann eine Verbesserung in der Saison 2016-17 erreicht werden, die Ergebnisse der anderen beiden Saisons sind ähnlich der des ersten Modells.

Saison	Ein Modell	Zwei Modelle
2015-16	0.675	0.674
2016-17	0.575	0.598
2017-18	0.577	0.580
Durchschnitt	0.609	0.617

Tabelle 16: Vorhersagegenauigkeiten der beiden Methoden

5.3.2 Hauptkomponentenregression

Manche der Variablen weisen eindeutige Korrelationen zueinander auf, beispielsweise der Anteil der Siege von Mannschaft A und der Anteil der Heimsiege von Mannschaft A. Die Hauptkomponentenregression basiert auf der Hauptkomponentenanalyse, deren Ziel es ist, die Anzahl der Kovariablen zu verringern und möglichst wenig Information dabei zu verlieren. Die erste Hauptkomponente ist die Lösung der Gleichung

$$y = a^T x \tag{15}$$

mit maximaler Varianz $V(y) = a^T \Sigma a$. Die Varianz soll maximiert werden, unter der Nebenbedingung $a^T a = 1$:

$$\begin{aligned} L(a) &= a^T \Sigma a - \lambda \left(a^T a - 1\right) \\ \frac{\partial L}{\partial a} &= 2\Sigma a - 2\lambda a \\ \Rightarrow (\Sigma - \lambda I)\, a &= 0 \end{aligned}$$

λ ist Eigenwert von Σ und a der zugehörige Eigenvektor. Daraus folgt:

$$a^T \Sigma a = a^T \lambda a = \lambda$$

Die Lösung des Maximierungsproblems ist somit der größte Eigenwert λ_1. Damit lautet die erste Hauptkomponente $y_1 = a_1^T x$, mit a_1 als Eigenvektor des größten Eigenwerts von λ. Falls die Dateninformationen nicht von einem Vektor ausreichend erfasst werden, müssen weitere Linearkombinationen berechnet werden. Für die Berechnung der zweiten Hauptkomponente muss folgende Gleichung gelöst werden:

$$y_2 = a_2^T x$$

Die Kovarianz zwischen y_1 und y_2 hat dabei eine Kovarianz von 0:

$$\begin{aligned} \operatorname{Cov}(y_1, y_2) &= 0 \\ \Leftrightarrow \operatorname{Cov}\left(a_1^T x, a_2^T x\right) &= 0 \\ \Leftrightarrow a_1^T a_2 &= 1 \end{aligned}$$

Außerdem gilt $a_2^T a_2 = 1$. Als nächstes wird wieder $a_2^T \Sigma a_2$ unter diesen Nebenbedingungen maximiert:

$$\begin{aligned} L(a_2) &= a_2^T \Sigma a_2 - \lambda \left(a_2^T a_2 - 1\right) - \delta \left(a_2^T a_1\right) \\ \frac{\partial L}{\partial a_2} &= 2\left(\Sigma - \lambda_2 I\right) a_2 - \gamma a_1 = 0 \\ \Rightarrow (\Sigma - \lambda_2 I)\, a_2 &= 0 \end{aligned}$$

λ_2 ist der zweitgrößte Eigenwert und a_2 der entsprechende Eigenvektor. Dieses Verfahren kann theoretisch bis zur p-ten Hauptkomponente fortgeführt werden. (Küchenhoff, 2018)
Die Hauptkomponentenregression funktioniert wie folgt:

1. Durchführung einer Hauptkomponentenanalyse mit der Datenmatrix.
2. Berechnung eines Regressionsmodells, beispielsweise mit der Methode der kleinsten Quadrate.
3. Retransformation der Hauptkomponenten in die ursprünglichen Variablen.

(Wehrens & Mevik, 2007)
Eine Hauptkomponentenregression kann in R mit Hilfe des Pakets *pls* durchgeführt werden. Wie bereits bei der linearen Regression, wird wieder mit zwei separaten Regressionsmodellen gearbeitet. Die Einflussvariablen setzen sich aus den Variablen der letzten beiden Modelle und der Anzahl der All Star Spieler der jeweiligen Mannschaft zusammen. Die Modelle werden wieder mit Hilfe des Pakets *mlr* optimiert. Bei der Berechnung der Modelle werden die Ergebnisse der letzten zehn Begegnungen einer Mannschaft berücksichtigt. Für die drei Saisons wird wieder für 2015-16, mit 67.6%,die höchste Anzahl an korrekten Spiele vorhergesagt. Die Komponenten für die erzielten Punkte der jeweiligen Mannschaft sehen wie folgt aus:

Variable	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6
PTSHOME	-0.943	0.319				
PTSAROAD	-0.325	-0.944				
W%HOME				-0.458	-0.279	-0.841
W%10HOME				-0.699	0.695	0.156
W%HOMEHOME				-0.537	-0.662	0.518
AllStarHome			0.990	0.106		
Erklärte Varianz	0.6696	0.3107	0.0179	0.0013	0.0003	0.0001

Tabelle 17: Hauptkomponenten für die erzielten Punkte des Heimteams

Die ersten beiden Komponenten erklären bereits ungefähr 98% der Varianz zusammen, was normalerweise bereits mehr als genug ist. Die genauesten Prognosen ergeben sich jedoch bei der Verwendung der ersten vier Komponenten. Die Hauptkomponenten für die erzielten Punkte der Gastmannschaft ähneln denen für die Punkte der Heimmannschaft:

Variable	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5	Comp 6
PTSROAD	0.788	-0.611				
PTSAHOME	0.613	0.790				
W%ROAD				-0.472	-0.290	-0.827
W%10ROAD				-0.683	0.708	0.153
W%ROADROAD				-0.532	-0.644	0.541
AllStarRoad			0.984	0.163		
Erklärte Varianz	0.5940	0.3822	0.0214	0.0019	0.0004	0.0001

Tabelle 18: Hauptkomponenten für die erzielten Punkte des Gastteams

Mit der Hauptkomponentenregression kann für die anderen beiden Saisons auch über 60% der Spielausgänge korrekt simuliert werden, wie der nachfolgende Plot illustriert:

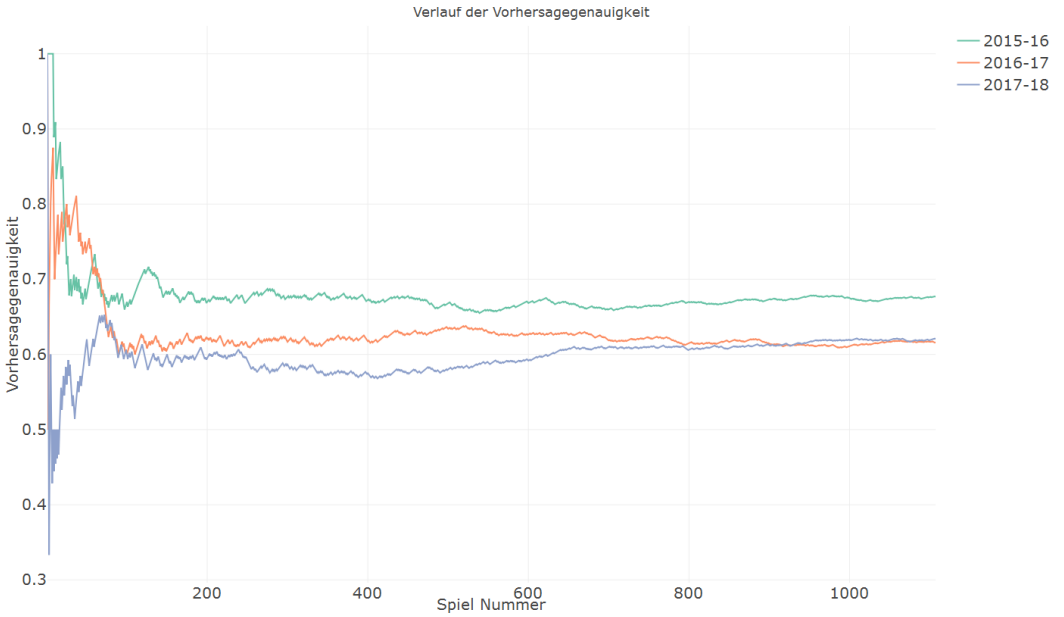


Abbildung 33: Vorhersagegenauigkeiten der drei Saison mit Hauptkomponentenregression

Saison	Hauptkomponentenregression%
2015-16	0.677
2016-17	0.616
2017-18	0.621
Durchschnitt	0.638

Tabelle 19: Vorhersagegenauigkeit der Simulation

In der Saison 2015-16 ist die Vorhersagegenauigkeit einmal mehr am höchsten, tatsächlich unterschied sich bei neun Mannschaften die Anzahl der simulierten Siege um maximal ±5 von der realen Zahl in diesem Zeitraum. Die Golden State Warriors haben zum Beispiel zum Start der Simulation neun Siege und keine Niederlagen. In den 73 simulierten Spielen wird 68 mal ein Sieg der Warriors vorhergesagt, in der Realität holte das Team 64 Siege in der Zeit. Das bedeutet jedoch nicht, dass 64 mal ein Siege der Warriors korrekt prognostiziert wird. Tatsächlich werden ungefähr 83.5% der Spiele der Mannschaft korrekt prognostiziert. Wie groß die Streuung der Differenz der prognostizierten Siege und realen Siege in den drei Jahren ist, offenbart folgende Graphik:

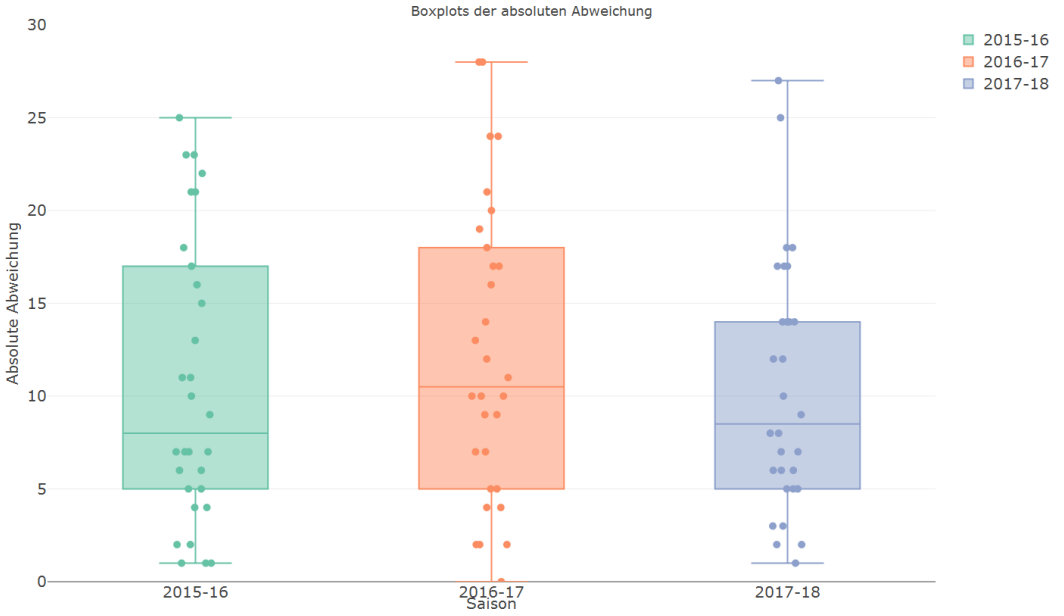


Abbildung 34: Boxplot der absoluten Differenzen der Siege

5.3.3 Vergleich der Modelle

Die berechneten Modelle können vor allem in der Saison 2015-16 bei vielen Spielen das korrekte Ergebnis vorhersagen, was bereits bei den Modellen mit den realen Daten der Fall ist. Mit einer Hauptkomponentenregression wird das beste Ergebnisse erreicht, andere Verfahren weisen schlechtere Resultate auf:

Saison	Höhere Sieg%	Linear	PCR	Random Forest
2015-16	0.646	0.674	0.677	0.667
2016-17	0.593	0.598	0.616	0.585
2017-18	0.561	0.580	0.621	0.556
Durchschnitt	0.600	0.617	0.638	0.603

Tabelle 20: Vorhersagegenauigkeiten der Simulationen

Wie kompliziert es jedoch ist, nach nur zehn Spielen den Rest einer Saison zu prognostizieren, illustriert die letzte Graphik. Innerhalb der letzten drei Saison wird nur von 25 Mannschaften eine Anzahl an Siegen prognostiziert, die um maximal fünf neben der realen liegt. Wie bereits in Kapitel 5.2 ist der Grund, dass zu viel in dem Verlauf einer Saison passieren kann. Spieler verletzen sich und wechseln Teams, was Einflussgrößen sind, die nicht berechnet werden können. Die besten Chancen auf eine akkurate Vorhersage der restlichen Saison nach so wenigen Spielen, ergeben sich bei Mannschaften die entweder schlecht oder sehr gut sind. Aber auch hierbei soll darauf aufgepasst werden, keine Mannschaft zu wählen, die nach zehn Spielen einen Sieg oder weniger hat. In den meisten Fällen werden für diese Mannschaft am Ende der Saison weniger als zehn Siege prognostiziert, was im realen Leben fast nie der Fall ist.

6 Shiny App

Zusätzlich zu den bisherigen Analysen und Prognosen wird eine Web-App in R programmiert, die es ermöglicht, individuelle Graphiken und Prognosen zu erstellen. Für die Erstellung der Applikation werden die R-Pakete *shiny* und *shinydashboard* verwendet. Die App kann lokal ausgeführt werden oder ist unter

`https://nicohahn.shinyapps.io/appNBA/`

erreichbar. Da der Webserver lediglich mit einem Arbeitsspeicher in Höhe von einem Gigabyte arbeitet, hat die lokale App eine bessere Laufzeit. Die App beinhaltet fünf verschiedene Funktionen. Diese und deren Funktionsweise werden im Verlauf dieses Kapitels vorgestellt.
Zu Beginn importiert die App eine lokale Umgebung, auf der die Namen der Spieler, Regressionsmodelle und bestimmte Statistiken für die Regressionsmodelle hinterlegt sind. Nach dem Laden werden alle benötigten Pakete gegebenenfalls installiert und bereitgestellt. Zudem wird die devtools-Version von *plotly* heruntergeladen, falls diese nicht installiert ist. Des weiteren befinden sich im Ordner der App weitere csv- und html-Dateien, die für die Erstellung der Graphiken benötigt werden.

6.1 Streudiagramme

Die ersten beiden Tabs„Team stats“ und „Season stats“, ermöglichen es, Streudiagramme zu erstellen. In dem ersten Tab kann der User eine Saison auswählen und bis zu vier verschiedene Statistiken graphisch darstellen. Dadurch kann dargestellt werden, welche Mannschaften bei welchen Statistiken am besten abschnitten. Zusätzlich kann noch zwischen den Statistiken der Mannschaft selbst und den Statistiken aus Sicht der gegnerischen Mannschaften entschieden werden. Wird der Mauszeiger über die Punkte bewegt, werden die jeweiligen Statistiken der Punkte angezeigt. Außerdem können Kästchen innerhalb der Plots gezogen werden, um nur den Inhalt innerhalb dieser anzuzeigen. Mit einem Doppelklick wird wieder heraus gezoomt.
Die Streudiagramme in dem zweiten Tab ermöglichen es, die Statistiken aller Mannschaften, die je in der NBA spielten, anzuzeigen. Dabei können die Mannschaften entweder gruppiert werden - das heißt, es werden nur die durchschnittlichen Statistiken der einzelnen Saisons angezeigt - oder es kann jede Mannschaft in die Graphik aufgenommen werden. Dieses Feature eignet sich, um die zeitliche Veränderung der Liga in verschiedenen Aspekten darzustellen. Werden die Mannschaften gruppiert, hat der User erneut vier Graphikparameter. Werden die Teams nicht gruppiert, ist wiederum die Unterscheidung zwischen Statistiken der Mannschaft selbst und der Gegner möglich. Die Punkte werden automatisch nach den Jahrzehnten eingefärbt. Durch einen Klick auf die Jahreszahl in der Legende wird das jeweilige Jahrzehnt aus dem Streudiagramm entfernt, beziehungsweise in das Diagramm aufgenommen. Durch einen Doppelklick auf eine Jahreszahl wird nur das jeweilige Jahrzehnt dargestellt.

6.2 Animationen

Neben Scatterplots können im zweiten Tab auch Animationen erstellt werden. Nach der Auswahl einer Statistik kann auf den Play-Button gedrückt werden. Für die

Erstellung einer Animation wird ein Datensatz geladen, der die durchschnittlichen Werte der jeweiligen Saisons enthält. Zu diesem Datensatz wird eine Variable *frame* hinzugefügt, die die Werte der jeweiligen Saison annehmen kann. Als nächstes werden die Beobachtungen der jeweiligen Saisons vervielfacht, um die Werte ihrer *frame*-Variable festzulegen. Gibt es beispielsweise zehn Saisons, befindet sich die älteste Saison zehnmal in dem neuen Datensatz. Dabei unterscheiden sich die zehn Beobachtungen anhand der Variable *frame*, die immer eine andere Saison als Wert erhält. Die zweitälteste Saison befindet sich neunmal in dem neuen Datensatz, mit den neun aktuellsten Saisons. Folglich nimmt die Variable *frame* zehnmal den Wert für die aktuelle Saison an und einmal den Wert für die älteste Saison. Für die Erstellung des Datensatzes wird die Funktion **accumulate_by()** von der Website `https://plot.ly/r/cumulative-animations/` verwendet.

6.3 Shotcharts

Shotcharts sind hervorragend, um zu zeigen, auf welche Art von Würfen sich Teams in der Offensive spezialisieren. Diese Art von Plots können in dem gleichnamigen Tab produziert werden. Dabei wird auf die API der NBA zurückgegriffen. In dieser existieren die Wurfdaten für alle Spieler ab der Saison 2000-01. Für jeden Wurf wird neben X-Y-Koordinaten unter anderem dokumentiert, ob der Wurf ein Treffer ist und aus welcher Zone er kommt. Für die Darstellung des Shotcharts eines Spielers wird eine Spieler ID benötigt, die ebenfalls in der API zu finden ist. Um den Shotchart einer Mannschaft zu bekommen, müssen alle Würfe aller Spieler, die in dieser Saison für die Mannschaft auf dem Feld standen, dargestellt werden. Somit muss zunächst der Kader der Mannschaft gefunden werden. Dafür wird als erstes auf einen Teil der API zurückgegriffen, der den aktuellen Kader einer Mannschaft enthält. Als nächstes muss herausgefunden werden, welche Spieler im Verlauf der Saison bei dieser Mannschaft unter Vertrag standen und ob diese auch Spiele für das Team absolviert haben. Auf der Website *basketball-reference.com* existiert eine Seite, die alle Transaktionen der Saison enthält. Für die Saison 2017-18 ist dies die Folgende:

`https://www.basketball-reference.com/leagues/NBA_2018_transactions.html`

Diese Website wird dann mit Hilfe des Pakets *rvest* in R eingelesen. Die Transaktionen werden als nächstes in einen Vektor umgewandelt, bei dem nur Elemente enthalten sind, die den Namen der Mannschaft enthalten. Alle Transaktionen beginnen mit dem Wort „The“ oder „In“, weswegen alle jene Elemente aus dem Vektor entfernt werden, die nicht mit einem dieser Wörter beginnen. Beginnt ein Element mit „The“, kann es sich um ein Tauschgeschäft mit Spielern handeln, um die Verpflichtung oder Entlassung eines Spielers oder um eine personelle Veränderung des Teams. Für alle diese Fälle gibt es wieder genaue Wortlaute, weswegen erneut gefiltert wird. Aus den verschiedenen Tauschgeschäften (Draft Picks, Geld, Trainer, Spieler, etc.) werden die Spielertransaktionen extrahiert. Um auf die Namen der beteiligten Spieler zu kommen, wird zunächst auf die API zurückgegriffen. Mit Hilfe dieser wird ein Vektor erstellt, der die Namen aller Spieler enthält, die in der jeweiligen Saison bei einer Mannschaft unter Vertrag standen. Diese Namen werden dann mit den Transaktionen abgeglichen, um nur die Spielernamen zu erhalten. Zum Schluss müssen die zu den Spielern korrespondierenden IDs ausfindig gemacht werden, um den Teildatensatz zu bekommen. Dieser Datensatz wird mit dem der API kombiniert, um einen

Datensatz zu erhalten, der alle Spieler enthält, die im Verlauf einer Saison unter Vertrag standen. Jetzt werden die Wurfdaten der einzelnen Spieler gedownloadet und zu einem großen Datensatz zusammengefügt. Bei ehemaligen Spielern werden nur die Würfe aufgenommen, die diese für die Mannschaft erzielt hatten.
Nach der Erstellung des Datensatzes können Wurfcharts produziert werden. Dabei kann entweder ein Diagramm mit allen Würfen, aufgeteilt nach Treffer und Nicht-Treffer, eine Heatmap oder ein Diagramm angezeigt werden, das die Treffergenauigkeit mit dem Durchschnitt der Liga in Relation stellt. Für die Erstellung der Graphiken werden die Pakete *ggplot2*, *grid* und *jpeg* verwendet. Der Basketballplatz wird ebenfalls mit ggplot erstellt. Dafür wird ein Skript von Todd Schneider verwendet, das auf github verfügbar ist. (Schneider, 2016)
Die Heatmap stellt alle genommenen Würfe einer Mannschaft dar. Dafür wird die zweidimensionale Dichte der Würfe auf den Basketballplatz projiziert.
Um die durchschnittliche Treffergenauigkeit einer Mannschaft zu berechnen, wird der Basketballplatz zuerst in verschiedene Bereiche eingeteilt. In diesen Bereichen wird dann die durchschnittliche Wurfquote aller Mannschaften in der jeweiligen Saison berechnet. Diese wird dann mit der Quote der ausgewählten Mannschaft in diesem Bereich verglichen. Wie der Platz aufgeteilt wird, geht aus der folgenden Graphik hervor:

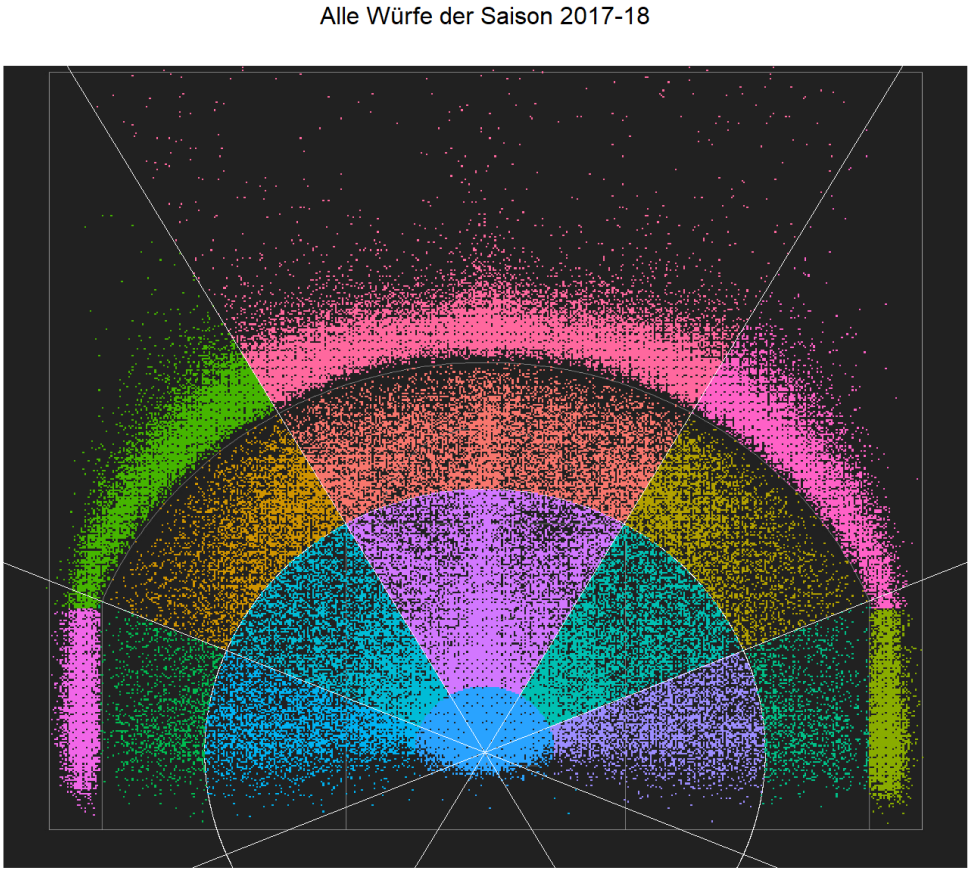


Abbildung 35: Aufteilung des Platzes in verschiedene Bereiche

6.4 Radarchats

In dem vierten Tab kann der User zwei Teams auswählen und diese vergleichen. Die Teams werden mit dem VORP Wert verglichen, um festzustellen, welche Mannschaft auf welcher Position besser besetzt ist. Dabei können alle Mannschaften ab der Saison 1973-74 ausgewählt werden. Nach der Auswahl werden zwei Radarcharts erstellt, die die jeweiligen summierten VORP Werte der Positionen enthält. So kann einfach erkannt werden, auf welcher Position welche Mannschaft bessere Spieler beherbergt. Zusätzlich wird der Kader der Mannschaft mit den jeweiligen Werten der Spieler tabelliert dargestellt.

6.5 Prognose

In dem letzten Tab kann eine individuelle Mannschaft zusammengestellt werden und eine Prognose erstellt werden, wie viele Siege diese hypothetische Mannschaft erreichen könnte. Dafür wählt der User zwölf Spieler aus, von jeder Position mindestens zwei, und teilt den Spielern eine Anzahl an Minuten zu. Bevor der zweite Point Guard ausgewählt werden kann, muss der erste Point Guard ausgewählt werden. Selbiges muss auch bei den anderen Positionen berücksichtigt werden. Spieler 11 und 12 können erst gewählt werden, nachdem die anderen zehn Spieler gewählt wurden. Mit den Slidern unter den jeweiligen Spielern können Minuten zugeteilt werden. Hierbei ist der User sozusagen der Coach, der sagt, wie viele Minuten pro Spiel ein Spieler eingesetzt wird. Insgesamt müssen 240 Minuten zugewiesen werden. Nachdem genau diese Anzahl an Minuten erreicht ist, erscheint ein Button, mit dem die Anzahl der Siege prognostiziert wird. Nachdem dieser gedrückt wird, erscheint ein Fenster das die erwartete Anzahl an Siegen nach einem linearen Modell, einem Lasso Modell und einem random forest enthält. Zusätzlich dazu wird eine Tabelle ausgegeben, die die hochgerechneten Statistiken der Spieler enthält, die Statistiken des Teams und die durchschnittlichen Statistiken der Liga. Für die Berechnung der Werte wird der Datensatz relVars2 verwendet, der in dem zu Beginn geladenen Environment ist. Dieser enthält alle für die Regressionen relevanten Variablen von über 500 Spielern. Dabei werden bei jedem Spieler die Werte pro 36 Minuten verwendet.

7 Fazit

Ziel dieser Arbeit war es, zu untersuchen, wie sich die National Basketball Association im Verlauf der Zeit geändert hat, was eine Mannschaft gut macht und wie Spielausgänge prognostiziert werden können.
Zu Beginn wurde mit einer deskriptiven Analyse untersucht, wie sich die Liga verändert hat. Dabei konnte festgestellt werden, dass sich die Liga in den 50er und 60er Jahren durch ein sehr hohes Tempo ausgezeichnet hat, das danach rapide fiel. Während das Tempo fiel, konnte ein Anstieg der Effizienz beobachtet werden. Durch die Einführung des Dreipunktewurfs änderten sich die Angriffsstrategien einzelner Mannschaften in Bezug darauf, dass Teams immer mehr Dreipunktewürfe versuchten. Ein Trend, der bis heute anhält.
Als nächstes wurde untersucht, welche Positionen in den verschiedenen Äras am dominantesten waren. Begonnen wurde damit in den 1970er Jahren, wo vor allem sehr große Spieler dominierten. Über die nächsten vier Jahrzehnte stellte sich mehr Ausgeglichenheit ein. Erst in dem aktuellen Jahrzehnt konnte wieder die Dominanz einer einzelnen Position festgestellt werden. Nachdem vor 50 Jahren die großen Spieler dominiert haben, sind es heute jedoch mit den Point Guards die kleinen, die dominieren.
In Kapitel 5.1 wird mittels einer Regressionsanalyse untersucht, welche Variablen den meisten Einfluss auf den Erfolg eines Teams hatten. Dafür werden die Ergebnisse der letzten fünf Saisons verwendet. Als erstes wird das lineare Regressionsmodell vorgestellt. Bei dieser Methodik gibt es nur eine Einflussvariable, die das Talentlevel einer Mannschaft wiedergeben soll, was zu einer ungenauen Prognose führt. Daraufhin werden über 130 Kovariablen in das Modell aufgenommen, die die Performance einer Mannschaft in den verschiedensten Bereichen beschreiben.
Aufgrund der hohen Variablenanzahl wird eine Variablenselektion durchgeführt. Dafür werden das Lasso und elastic net Modell vorgestellt. Mit den penalisierten Modellen konnte eine Verbesserung des linearen Modells erreicht werden.
Zum Abschluss der Regressionsanalyse wird ein random forest berechnet. Dieser schnitt mit Abstand am besten ab. Aus dem resultierenden Modell konnten die Variablen herausgelesen werden, die den größten Einfluss auf den Erfolg einer Mannschaft haben. Dabei war mit Abstand das Talentlevel einer Mannschaft am bedeutendsten, gefolgt von den offensiven Fähigkeiten der Mannschaft und als drittes den defensiven. Der deutliche Abstand zwischen den Offensiv- und Defensivfähigkeiten einer Mannschaft legt nahe, dass die Offensive heutzutage wichtiger ist als die Defensive, wodurch der Titel dieser Arbeit vielleicht doch nicht so korrekt ist.
In Kapitel 5.2 wird die Prognose einzelner Spielausgänge behandelt. Dafür wurden über 5000 Spielausgänge der letzten fünf Jahre prognostiziert. Dabei wurde als erstes das multivariate multiple lineare Regressionsmodell vorgestellt, mit dem ca. 69% der Spiele korrekt vorhergesagt wurden.
Als nächstes wurde ein logistisches Regressionsmodell verwendet, das 70% der Spiele richtig vorhersagen konnte. In beiden Modellen konnte der bekannte Heimspieleffekt beobachtet werden, der besagt, dass Teams zuhause besser spielen als auswärts. Des weiteren wurde deutlich, wie viele Faktoren in die Prognose eines Spiels einfließen, die nicht vorhersehbar oder latent sind. Dazu zählen unter anderem Verletzungen und Tagesform der Spieler.
Das konnte vor allem in Kapitel 5.3 nochmals bestätigt werden, als der restliche Ver-

lauf einer Saison nach 10% der Saison simuliert wurde. Mit einem linearen Ansatz konnten hier nur 61.7% der Spiele in den letzten drei Saisons zutreffend vorhergesagt werden.
Die Hauptkomponentenregression wurde als nächstes vorgestellt, mit der es gelingt 63.8% der Spielausgänge richtig zu simulieren. Dies war somit das beste Modell für die Simulation einer kompletten Saison.
Neben den Analysen wurde ebenfalls eine shiny-App in R entwickelt, mit der zusätzlichen Graphiken oder auch Prognosen erstellt werden konnten. Die Funktionsweise dieser App wird in Kapitel 6 thematisiert.
Die statistische Analyse ist schon lange ein fester Bestandteil von Basketballmannschaften. Immer mehr Teams stellen Statistiker an, die dabei helfen sollen, die Offensive und Defensive eines Teams zu optimieren, oder die optimale Taktik gegen bestimmte Gegner zu finden. Modelle zur Prognose von Spielergebnissen finden vor allem in der Wettbranche Verwendung, wo mit geeigneten Modellen viel Geld gemacht werden kann.

8 Verwendete Statistiken und Abkürzungen

Punkte: Anzahl der Punkte die eine Mannschaft pro Spiel erreicht.
Rebound: Fangen eines erfolglosen Korbwurfs. Unter eigenem Korb defensiver Rebound, unter gegnerischem Korb offensiver.
Assist: Vorangegangener Pass, der direkt zu einem Korberfolg führt
Turnover: Verlust des Ballbesitzes an die gegnerische Mannschaft.
Block: Abwehren eins Korbwurfs, bevor der Ball den höchsten Punkt seiner Flugkurve überschreitet.
Steal: Erzwingen eines Turnover durch Abfangen eines gegnerischen Passes oder Wegnahme des Balles beim Dribbling.
Field Goal: Ein erfolgreicher Korb durch einen Zwei- oder Dreipunktewurf.
Dist Miles: Gibt die Anzahl an Meilen an, die ein Team in der Offensive oder Defensive zurücklegt.
Avg Speed: Durchschnittliches Lauftempo eines Teams in der Offensive bzw. Defensive. **Drive**: Zug eines Spielers zum Korb aus dem Halbfeld. Der Drive muss von dem Spieler selbst initialisiert werden und beginnt nicht unterhalb des Korbes.
DFG%: Die Wurfquote des Gegners unter dem Korb. (nba.com, 2017)
Pace: Der Pace Faktor gibt an, wie oft ein Team innerhalb von 48 Minuten in Ballbesitz ist.
Possession: Jedes Mal wenn die 24-Sekunden Wurfuhr auf 24 gesetzt wird beginnt ein Possession.
TS%: $\frac{PTS}{FGA+0.44 \cdot FTA}$
eFG%: $\frac{FGM+0.5 \cdot 3PM}{FGA}$
SOS: Gibt an wie schwer die Spielplan einer Mannschaft ist.
(basketball reference.com, 2014b)
Freiwurfrate: Freiwürfe pro Feldwurf: $\frac{FT}{FGA}$
(basketball reference.com, 2010)
Dreipunktewurfrate: Relativer Anteil an Dreipunktewürfen: $\frac{3PA}{FGA}$
PER: Ein Pro-Minute Rating, das alle positiven Eigenschaften eines Spielers summiert, alle negativen abzieht und ein Rating für die Performance eines Spielers ausgibt. Für die Berechnung wird zuerst das „unadjusted PER“ berechnet:

$$\begin{aligned} uPER = \frac{1}{MP} \cdot \Bigg(& 3PM + \frac{2}{3} AST + \left(2 - fac \cdot \frac{AST\_tm}{FGM\_tm}\right) \cdot FGM \\ & + FTM \cdot 0.5 \cdot \left(1 + \left(1 - \frac{AST\_tm}{FGM\_tm}\right) + \frac{2}{3} \cdot \frac{AST\_tm}{FGM\_tm}\right) \\ & - VOP \cdot TOV - VOP \cdot DRBP \cdot (FGA - FGM) \\ & - VOP \cdot 0.44\,(0.44 + (0.56 \cdot DRBP)) \cdot (FTA - FTM) \\ & + VOP \cdot (1 - DRBP) \cdot (TRB - ORB) + VOP \cdot DRBP \cdot ORB \\ & + VOP \cdot STL + VOP \cdot DRBP \cdot BLK \\ & - PF \cdot \left(\frac{FTM\_lg}{PF\_lg} - 0.44 \cdot \frac{FTA\_lg}{PF\_lg} \cdot VOP\right)\Bigg) \end{aligned} \tag{16}$$

tm steht für Teamvariablen, lg für ligaweite Variablen. Des weiteren gilt:

$$factor = \frac{2}{3} - \frac{0.5 \cdot \frac{AST\_lg}{FGM\_lg}}{2 \cdot \frac{FGM\_lg}{FTM\_lg}} \tag{17}$$

$$VOP = \frac{PTS\_lg}{FGA\_lg - ORB\_lg + TOV\_lg + 0.44 \cdot FTA\_lg} \tag{18}$$

$$DRBP = \frac{TRB\_lg - ORB\_lg}{TRB\_lg} \tag{19}$$

Das nicht adjustierte PER wird jetzt noch durch die Pace einer Mannschaft bereinigt:

$$PER = \left(uPER \cdot \left(\frac{Pace\_lg}{Pace\_tm}\right)\right) \cdot \left(\frac{15}{uPER\_lg}\right) \tag{20}$$

(basketball reference.com, 2005)
BPM: Box Plus/Minus misst die Qualität und den Beitrag eines Spielers zu einer Mannschaft. Die Statistik wird auf 100 Possessions aggregiert und basiert auf einer linearen Regression. Eine genaue Beschreibung kann **hier** nachgelesen werden.
VORP: Value over replacement player gibt an wie viel ein Spieler zu einem Team beigetragen hat im Vergleich zu einem durchschnittlichen Spieler. Für die Berechnung wird BPM verwendet:

$$VORP = (BPM - (-2.0)) \cdot \frac{MP}{\frac{MP\_tm}{5}} \frac{Games\_tm}{82} \tag{21}$$

(basketball reference.com, 2014a)
Elbow: Der Elbow ist an den Ecken der Freiwurflinie.

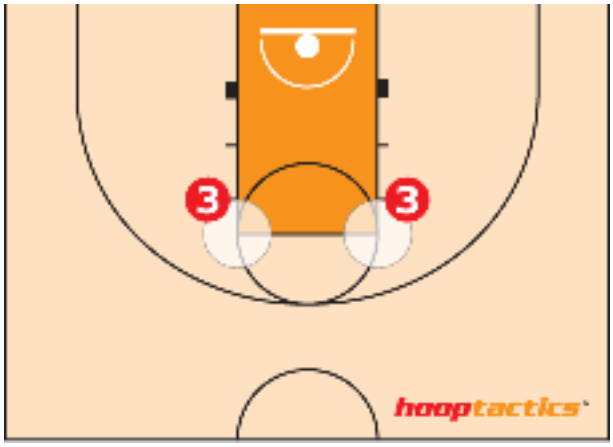


Abbildung 36: Elbow

Paint: Die Paint ist die gefärbte Fläche unterhalb der Freiwurflinie.

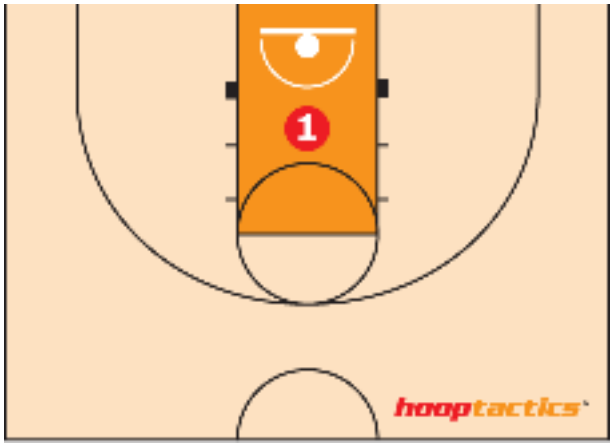


Abbildung 37: Painted area

Restricted area: Der Bereich zwischen dem Halbkreis unterhalb des Korbes und dem Korb.

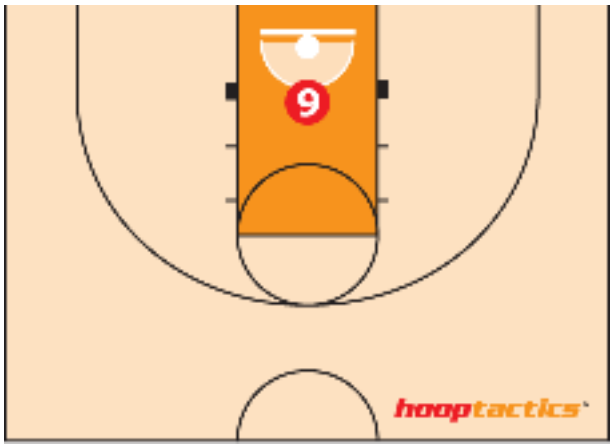


Abbildung 38: Restricted are

(hooptactics.com, 2009)
ORtg und DRtg: Offensives und defensives Rating gibt an, wie viele Punkte ein Spieler über 100 Possessions generiert bzw. zulässt. Die Berechnung kann **hier** nachgelesen werden. (basketball reference.com, 2013)

9 Anhang

9.1 Zusätzliche Tabellen

9.1.1 Komplette Vorhersagen der Saison

Tabelle der aktuellen Saison
Anmerkungen: Haben zwei Mannschaften die gleiche Anzahl an Siegen, entscheidet sich die Position nach zwei Kriterien:

1. Das Team, das in der regulären Saison öfters gegen das andere gewonnen hat, erhält den höheren Setzplatz.

2. Haben beide Mannschaften die gleiche Anzahl an Siegen gegeneinander, entscheidet das Los.

	Western Conference	
	Team	W
	Houston Rockets	65
	Golden State Warriors	58
	Portland Trail Blazers	49
	Oklahoma City Thunder	48
	Utah Jazz	48
	New Orleans Pelicans	48
	San Antonio Spurs	47
	Minnesota Timberwolves	47
	Denver Nuggets	46
	Los Angeles Clippers	42
	Los Angeles Lakers	35
	Sacramento Kings	27
	Dallas Mavericks	24
	Memphis Grizzlies	22
	Phoenix Suns	21

	Eastern Conference	
	Team	W
	Toronto Raptors	59
	Boston Celtics	55
	Philadelphia 76ers	52
	Cleveland Cavaliers	50
	Indiana Pacers	48
	Miamit Heat	44
	Milwaukee Bucks	44
	Washington Wizards	43
	Detroit Pistons	39
	Charlotte Hornets	36
	New York Knicks	29
	Brooklyn Nets	28
	Chicago Bulls	27
	Orlando Magic	25
	Atlanta Hawks	24

Tabelle 21: Tabellen der aktuellen Saison

Lineares Modell

Western Conference		Eastern Conference	
Team	W	Team	W
Houston Rockets (-)	63	Toronto Raptors (-)	59
Golden State Warriors (-)	56	Philadelphia 76ers (+1)	52
Utah Jazz (+2)	54	Boston Celtics (-1)	47
Oklahoma City Thunder (-)	49	Cleveland Cavaliers (-)	46
San Antonio Spurs (+2)	47	Detroit Pistons (+4)	43
Portland Trail Blazers (-3)	47	Indiana Pacers (-1)	43
Minnesota Timberwolves (+1)	46	Miami Heat (-1)	42
New Orleans Pelicans (-2)	46	Milwaukee Bucks (-1)	41
Denver Nuggets (-)	45	Washington Wizards (-1)	41
Los Angeles Clippers (-)	41	Charlotte Hornets (-)	40
Los Angeles Lakers (-)	38	Brooklyn Nets (+1)	33
Dallas Mavericks (+1)	34	New York Knicks (-1)	31
Memphis Grizzlies (+1)	25	Atlanta Hawks (+2)	29
Sacramento Kings (-2)	22	Orlando Magic (-)	28
Phoenix Suns (-)	16	Chicago Bulls (-2)	24

Tabelle 22: Tabellen der aktuellen Saison nach dem linearen Modell

Lasso Regression

Western Conference		Eastern Conference	
Team	W	Team	W
Houston Rockets (-)	62	Toronto Raptors (-)	59
Golden State Warriors (-)	56	Philadelphia 76ers (+1)	52
Utah Jazz (+2)	51	Boston Celtics (-1)	49
Oklahoma City Thunder (-)	49	Cleveland Cavaliers (-)	45
San Antonio Spurs (+2)	49	Indiana Pacers (-)	44
Portland Trail Blazers (-3)	47	Miami Heat (-)	42
Minnesota Timberwolves (+1)	45	Washington Wizards (-)	42
New Orleans Pelicans (-2)	45	Charlotte Hornets (+2)	41
Denver Nuggets (-)	44	Milwaukee Bucks (-1)	40
Los Angeles Clippers (-)	41	Detroit Pistons (-1)	39
Los Angeles Lakers (-)	35	New York Knicks (-)	32
Dallas Mavericks (+1)	33	Brooklyn Nets (-)	32
Memphis Grizzlies (+1)	26	Orlando Magic (+1)	29
Sacramento Kings (-2)	23	Atlanta Hawks (+1)	27
Phoenix Suns (-)	18	Chicago Bulls (-2)	23

Tabelle 23: Tabellen der aktuellen Saison nach dem Lasso Modell

Elastic net Regression

Western Conference			Eastern Conference		
	Team	W		Team	W
	Houston Rockets (-)	62		Toronto Raptors (-)	59
	Golden State Warriors (-)	56		Philadelphia 76ers (+1)	52
	Utah Jazz (+2)	51		Boston Celtics (-1)	49
	Oklahoma City Thunder (-)	49		Cleveland Cavaliers (-)	45
	San Antonio Spurs (+2)	48		Indiana Pacers (-)	44
	Portland Trail Blazers (-3)	47		Miami Heat (-)	42
	Minnesota Timberwolves (+1)	45		Washington Wizards (-)	42
	New Orleans Pelicans (-2)	45		Charlotte Hornets (+2)	41
	Denver Nuggets (-)	44		Milwaukee Bucks (-1)	40
	Los Angeles Clippers (-)	41		Detroit Pistons (-1)	39
	Los Angeles Lakers (-)	36		New York Knicks (-)	32
	Dallas Mavericks (+1)	33		Brooklyn Nets (-)	32
	Memphis Grizzlies (+1)	26		Orlando Magic (+1)	29
	Sacramento Kings (-2)	23		Atlanta Hawks (+1)	27
	Phoenix Suns (-)	18		Chicago Bulls (-2)	23

Tabelle 24: Tabellen der aktuellen Saison nach dem elastic net Modell

Random forest Regression

Western Conference			Eastern Conference		
	Team	W		Team	W
	Houston Rockets (-)	60		Toronto Raptors (-)	57
	Golden State Warriors (-)	58		Boston Celtics (-)	53
	Utah Jazz (+2)	50		Philadelphia 76ers (-)	53
	Portland Trail Blazers (-1)	49		Cleveland Cavaliers (-)	49
	New Orleans Pelicans (+1)	48		Indiana Pacers (-)	46
	Oklahoma City Thunder (-2)	47		Miami Heat (-)	44
	San Antonio Spurs (-)	47		Milwaukee Bucks (-)	44
	Denver Nuggets (+1)	47		Washington Wizards (-)	43
	Minnesota Timberwolves (-1)	47		Detroit Pistons (-)	40
	Los Angeles Clippers (-)	43		Charlotte Hornets (-)	37
	Los Angeles Lakers (-)	37		New York Knicks (-)	30
	Dallas Mavericks (+1)	27		Brooklyn Nets (-)	29
	Sacramento Kings (-1)	26		Chicago Bulls (-)	26
	Memphis Grizzlies (-)	24		Orlando Magic (-)	26
	Phoenix Suns (-)	22		Atlanta Hawks (-)	25

Tabelle 25: Tabellen der aktuellen Saison nach dem random forest Modell

9.2 Zusätzliche Graphiken

Abbildung 39: VORP Radarchart der 1970er

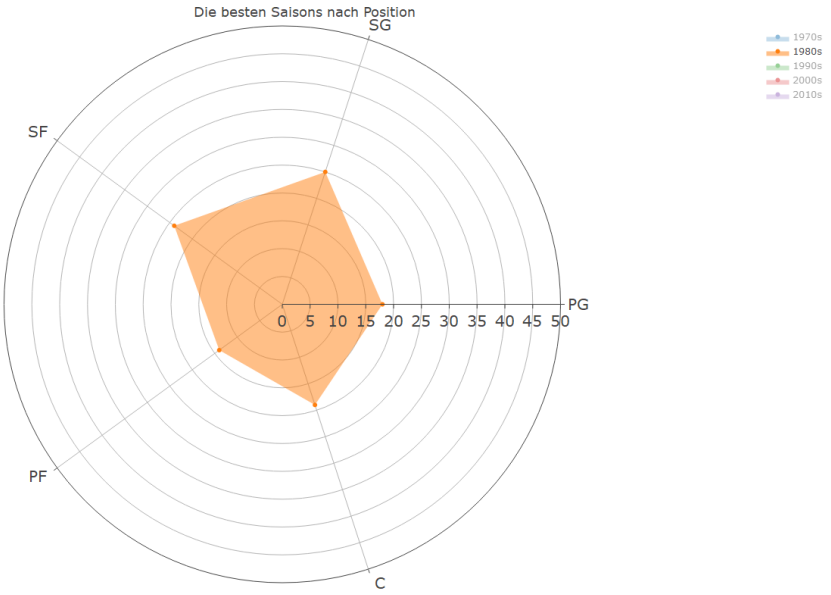


Abbildung 40: VORP Radarchart der 1980er

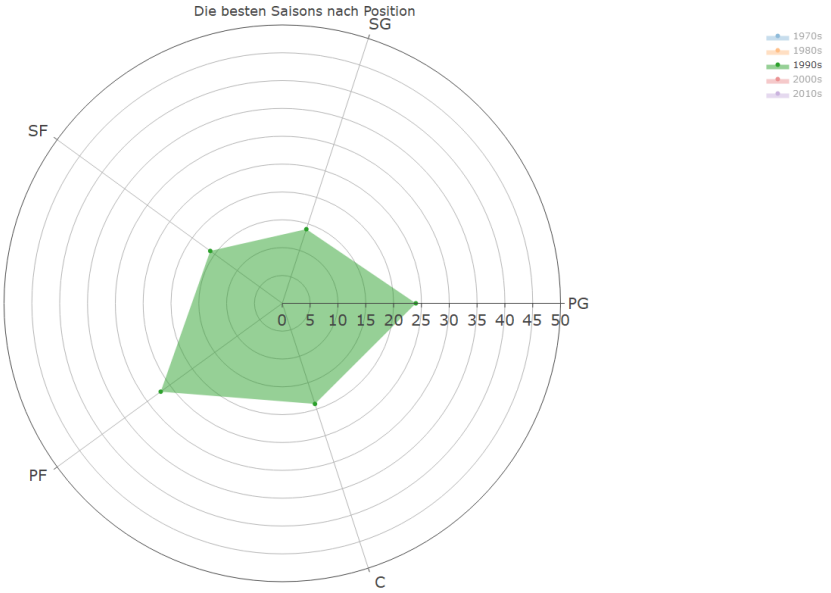


Abbildung 41: VORP Radarchart der 1990er

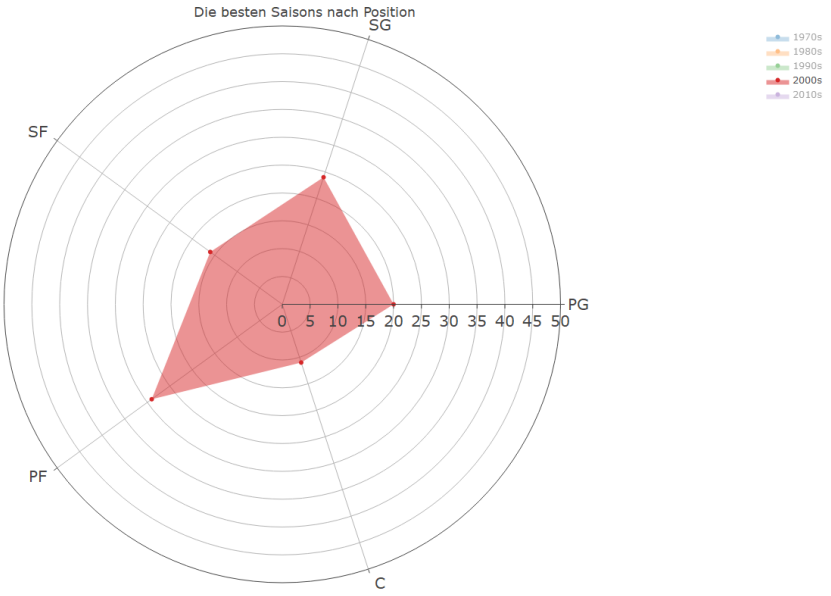


Abbildung 42: VORP Radarchart der 2000er

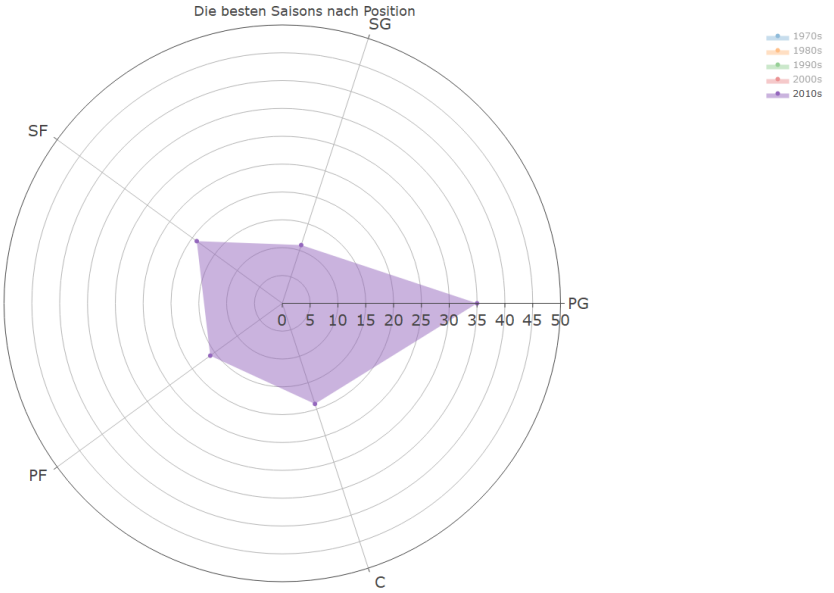


Abbildung 43: VORP Radarchart der 2010er

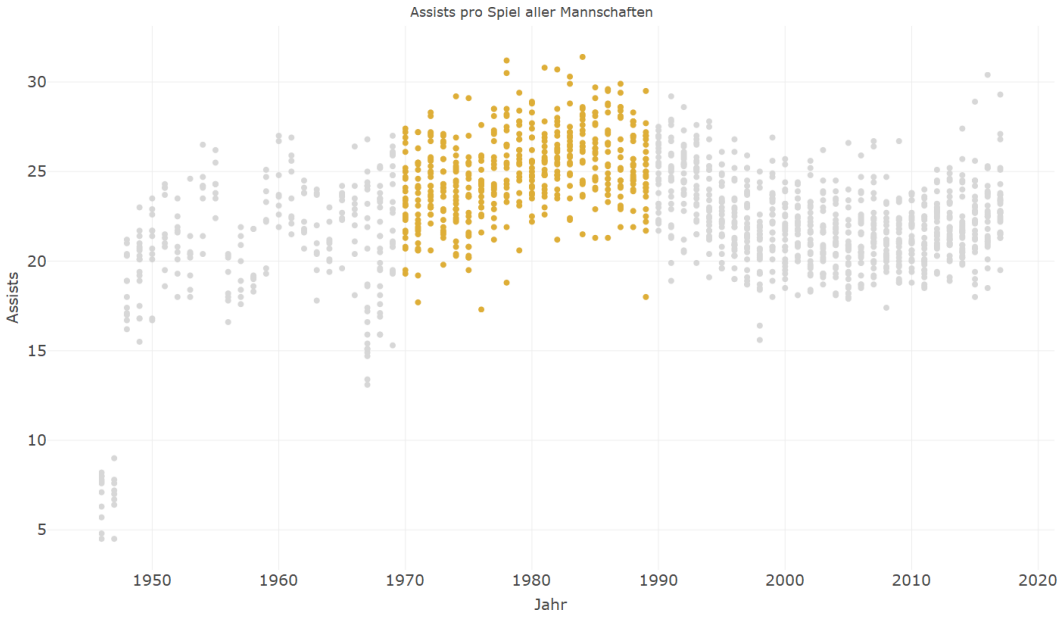


Abbildung 44: Assists pro Spiel

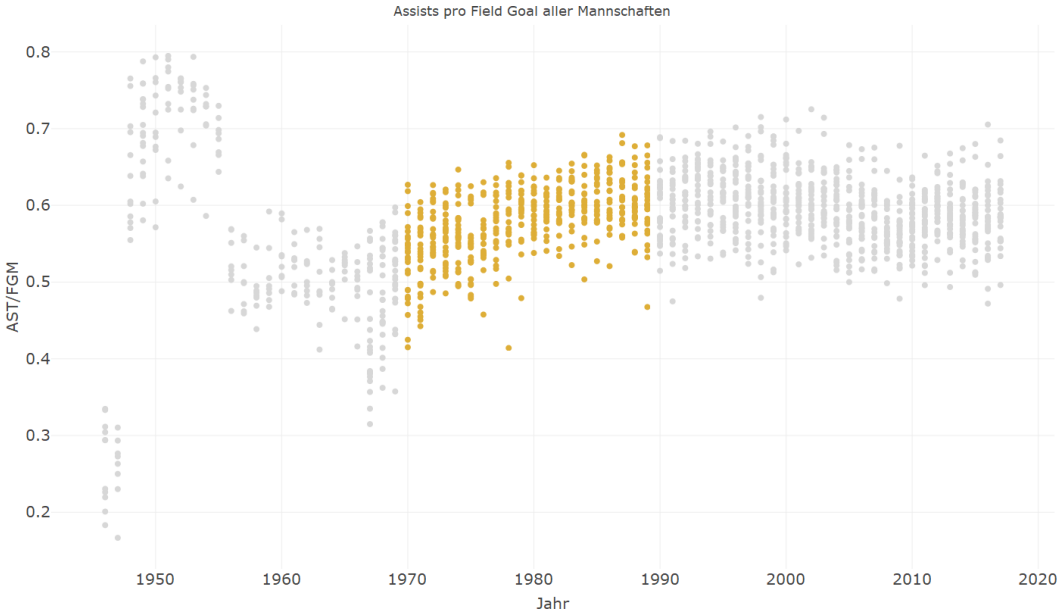


Abbildung 45: Assists pro Field Goal

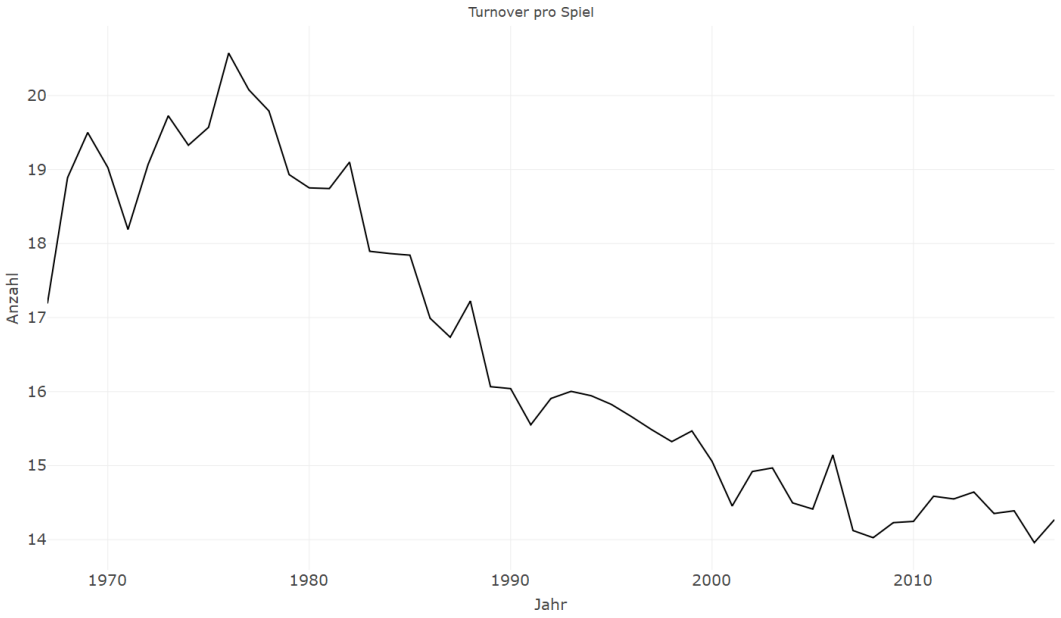


Abbildung 46: Sinkende Anzahl der Turnover

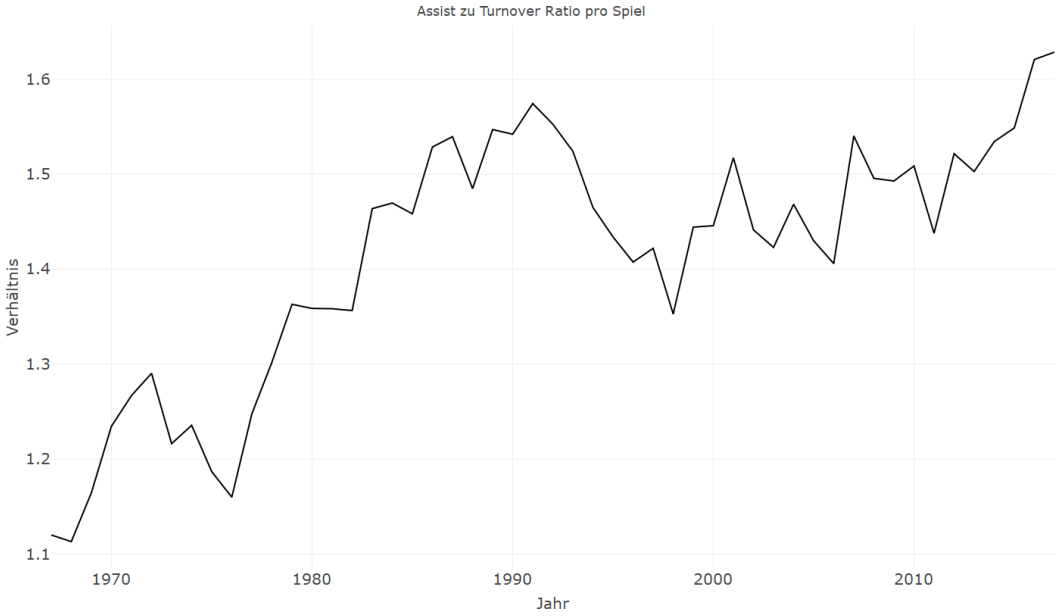


Abbildung 47: Verbessertes AST/TOV Ratio

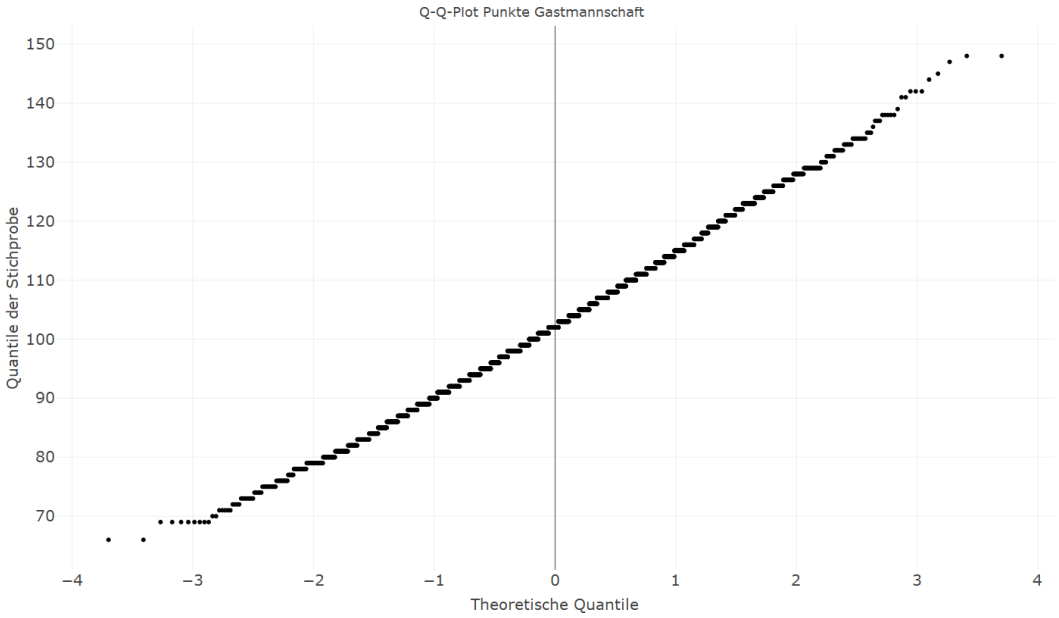


Abbildung 48: Q-Q-Plot der Anzahl der Punkte Gastmannschaft

9.3 Verwendete R-Pakete

- caret (from Jed Wing et al., 2018)
- devtools (Wickham, Hester & Chang, 2018)
- DT (Xie, Cheng, Tan & Girlich, 2018)

- e1071 (Meyer, Dimitriadou, Hornik, Weingessel & Leisch, 2017)
- ggplot2 (Wickham, 2009)
- glmnet (Friedman et al., 2010)
- grid (R Core Team, 2018)
- jpeg (Urbanek, 2014)
- mlr (Bischl et al., 2016)
- packrat (Ushey, McPherson, Cheng, Atkins & Allaire, 2018)
- parallel (R Development Core Team, 2008)
- parallelMap (Bischl & Lang, 2015)
- PKI (Urbanek, 2017)
- plotly (Sievert, 2018)
- plotmo (Milborrow, 2018)
- pls (Wehrens & Mevik, 2007)
- plyr (Wickham, 2011)
- randomForest (Liaw & Wiener, 2002)
- randomForestExplainer (Paluszynska & Biecek, 2017)
- Rcpp (Eddelbuettel & Balamuta, 2017)
- RJSONIO (Lang, 2014)
- rlang (Henry & Wickham, 2018)
- rvest (Wickham, 2016)
- shiny (Chang et al., 2017)
- shinydashboard (Chang & Borges Ribeiro, 2018)
- tidyverse (Wickham, 2017)

Inhalt des USB-Sticks

Folgende Dateien befinden sich auf dem beigefügten USB-Stick:

- Ordner Bachelorarbeit

  Enthält die Bachelorarbeit pdf-Datei

- Ordner R

  Enthält alle Analysen, Datensätze und die shiny App. Unterteilung in:

  - App
  - Datensätze
  - Deskriptive Analyse
  - Erstellung Datensätze
  - Funktionen
  - Modelle
  - Pakete

- Des weiteren befindet sich eine readme.txt Datei auf dem USB-Stick, welche die Beschreibungen der jeweiligen Codes enthält.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die Bachelor Thesis selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die aus fremden Quellen direkt oder indirekt übernommenen Gedanken als solche kenntlich gemacht habe.
Die Arbeit habe ich bisher keinem anderen Prüfungsamt in gleicher oder vergleichbarer Form vorgelegt. Sie wurde bisher nicht veröffentlicht.

München, den 16.07.2016

Literaturverzeichnis

basketball reference.com. (2005). *Calculating per.* https://www.basketball-reference.com/about/per.html. (Accessed: 2018-26-05)

basketball reference.com. (2010). *Four factors.* https://www.basketball-reference.com/about/factors.html. (Accessed: 2018-26-05)

basketball reference.com. (2013). *Calculating individual offensive and defensive ratings.* https://www.basketball-reference.com/about/ratings.html. (Accessed: 2018-26-05)

basketball reference.com. (2014a). *About box plus/minus (bpm).* https://www.basketball-reference.com/about/bpm.html. (Accessed: 2018-26-05)

basketball reference.com. (2014b). *Glossary.* https://www.basketball-reference.com/about/glossary.html. (Accessed: 2018-26-05)

basketball reference.com. (2018). *Nba & aba league index.* https://www.basketball-reference.com/leagues/. (Accessed: 2018-07-05)

Beckler, M., Wang, H. & Papamichael, M. (2013). Nba oracle. *Zuletzt besucht am*, *17* (20082009.9).

Bischl, B. & Lang, M. (2015). parallelmap: Unified interface to parallelization back-ends [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=parallelMap (R package version 1.3)

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, J., Studerus, E., ... Jones, Z. M. (2016). mlr: Machine learning in r. *Journal of Machine Learning Research*, *17* (170), 1-5. Zugriff auf http://jmlr.org/papers/v17/15-066.html

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24* (2), 123–140.

Breiman, L. & Cutler, A. (2004). *Random forests.* https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. (Accessed: 2018-26-05)

Chang, W. & Borges Ribeiro, B. (2018). shinydashboard: Create dashboards with 'shiny' [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=shinydashboard (R package version 0.7.0)

Chang, W., Cheng, J., Allaire, J., Xie, Y. & McPherson, J. (2017). shiny: Web application framework for r [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=shiny (R package version 1.0.5)

Eddelbuettel, D. & Balamuta, J. J. (2017, aug). Extending extitR with extitC++: A Brief Introduction to extitRcpp. *PeerJ Preprints*, *5*, e3188v1. Zugriff auf https://doi.org/10.7287/peerj.preprints.3188v1 doi: 10.7287/peerj.preprints.3188v1

Fahrmeir, L., Kneib, T. & Lang, S. (2009). *Regression - modelle, methoden und anwendungen.* Berlin Heidelberg New York: Springer-Verlag.

Friedman, J., Hastie, T. & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, Articles*, *33* (1), 1–22. Zugriff auf https://www.jstatsoft.org/v033/i01 doi: 10.18637/jss.v033.i01

from Jed Wing, M. K. C., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., ... Hunt., T. (2018). caret: Classification and regression training [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=caret (R package version 6.0-79)

Henry, L. & Wickham, H. (2018). rlang: Functions for base types and core r and 'tidyverse' features [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=rlang (R package version 0.2.0)

history.com. (2009a). *American basketball association debuts.* https://www.history.com/this-day-in-history/american-basketball-\association-debuts. A+E Networks. (Accessed: 2018-07-05)

history.com. (2009b). *Nba is born.* https://www.history.com/this-day-in-history/nba-is-born. A+E Networks. (Accessed: 2018-07-05)

hoopedia.com. (o. J.). *Nba roots.* http://bit.do/nba_roots. (Accessed via Wayback Machine: 2018-07-05)

hooptactics.com. (2009). *Basketball court areas.* http://hooptactics.com/Basketball_Basics_Court_Areas. (Accessed: 2018-26-05)

Küchenhoff, H. (2018). *Multivariate verfahren.* Vorlesung.

Lang, D. T. (2014). Rjsonio: Serialize r objects to json, javascript object notation [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=RJSONIO (R package version 1.3-0)

Liaw, A. & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2* (3), 18-22. Zugriff auf https://CRAN.R-project.org/doc/Rnews/

Maitra, R. (2012). *Multivariate linear regression models.* Vorlesung.

Mayr, A. (2017). *Generalisierte regressionsmodelle.* Vorlesung.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A. & Leisch, F. (2017). e1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=e1071 (R package version 1.6-8)

Milborrow, S. (2018). plotmo: Plot a model's response and residuals [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=plotmo (R package version 3.3.7)

Moore, J. C. & Brylinsky, J. (1995, Dec). Facility familiarity and the home advantage. *Journal of Amateur Sport*, *18* (4). Zugriff auf http://www.biomedsearch.com/article/Facility-familiarity-home-advantage/17782544.html

nba.com. (2017). *Glossary.* https://stats.nba.com/help/glossary/. (Accessed: 2018-26-05)

nba.com. (2018). *Mark cuban fined.* http://www.nba.com/article/2018/02/21/dallas-mavericks-owner-mark-cuban-fined-comments-tanking. (Accessed: 2018-26-05)

Paluszynska, A. & Biecek, P. (2017). randomforestexplainer: Explaining and visualizing random forests in terms of variable importance [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=randomForestExplainer (R package version 0.9)

R Core Team. (2018). R: A language and environment for statistical computing [Software-Handbuch]. Vienna, Austria. Zugriff auf https://www.R-project.org/

R Development Core Team. (2008). R: A language and environment for statistical computing [Software-Handbuch]. Vienna, Austria. Zugriff auf http://www.R-project.org (ISBN 3-900051-07-0)

Schneider, T. (2016). *Ballr: Interactive nba shot charts with r and shiny.* https://github.com/toddwschneider/ballr. GitHub.

Sievert, C. (2018). plotly for r [Software-Handbuch]. Zugriff auf https://plotly-book.cpsievert.me

statista.com. (2018). *Total nba league revenue* from 2001/02 to 2016/17 (in billion u.s. dollars).* https://www.statista.com/statistics/193467/total-league-\revenue-of-the-nba-since-2005/. (Accessed: 2018-07-05)

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73* (3), 273–282.

Urbanek, S. (2014). jpeg: Read and write jpeg images [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=jpeg (R package version 0.1-8)

Urbanek, S. (2017). Pki: Public key infrastucture for r based on the x.509 standard [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=PKI (R package version 5.1)

Ushey, K., McPherson, J., Cheng, J., Atkins, A. & Allaire, J. (2018). packrat: A dependency management system for projects and their r package dependencies [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=packrat (R package version 0.4.9-3)

Wehrens, R. & Mevik, B.-H. (2007). The pls package: principal component and partial least squares regression in r.

Wickham, H. (2009). *ggplot2: Elegant graphics for data analysis.* Springer-Verlag New York. Zugriff auf http://ggplot2.org

Wickham, H. (2011). The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, *40* (1), 1–29. Zugriff auf http://www.jstatsoft.org/v40/i01/

Wickham, H. (2016). rvest: Easily harvest (scrape) web pages [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=rvest (R package version 0.3.2)

Wickham, H. (2017). tidyverse: Easily install and load the 'tidyverse' [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=tidyverse (R package version 1.2.1)

Wickham, H., Hester, J. & Chang, W. (2018). devtools: Tools to make developing r packages easier [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=devtools (R package version 1.13.5)

Xie, Y., Cheng, J., Tan, X. & Girlich, M. (2018). Dt: A wrapper of the javascript library 'datatables' [Software-Handbuch]. Zugriff auf https://CRAN.R-project.org/package=dt (R package version 0.4)