Seminararbeit
Statistical Learning

Prognose des wöchentlichen Gehalts von Fußballspielern mit Modellen des Statistical Learnings

vorgelegt der Fakultät für Wirtschaftswissenschaften
der Universität Duisburg-Essen (Campus Essen)
Lehrstuhl für Ökonometrie
Prof. Dr. Christoph Hanck

von:
Paul Drecker und Janine Langerbein
Matrikelnummer: 3072124 und 3061371

Abgabetermin: 27.08.2019

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Für das Fußballsimulationsspiel FIFA erstellt das Entwicklungsunternehmen Electronic Arts umfangreiche Datenbanken mit Spielerstatistiken ("FIFA 17 Player Ratings Blend Stats, Subjectivity" 2016). Neben individuellen Fähigkeiten der Spieler, wie Ballkontrolle oder Schusskraft, befindet sich in dieser Datenbank auch die Variable Gehalt. Dabei handelt es sich um einen Schätzwert, da das Gehalt der meisten Fußballspieler nicht bekannt ist ("Football Leaks: 106 Mio. Euro Für Lionel Messi in Barcelona" 2018). Um die Spielerfahrung so realistisch wie möglich zu gestalten, ist es das Ziel der Entwickler, die Variablen in einem realistischen Verhältnis zueinander festzulegen. Es kann vermutet werden, dass zwischen den individuellen Attributen der Spieler und deren Gehalt eine Beziehung existiert. In der Theorie sollten beispielsweise Spieler mit besseren Fähigkeiten ein höheres Gehalt verhandeln können. Auf Basis dieser Beziehung sollte es möglich sein, das Gehalt der Spieler mit Hilfe eines Modells möglichst gut prognostizieren zu können. Idealerweise sollte dieses Modell in der Lage sein, auch das Gehalt neuer Spieler präzise zu schätzen. Zunächst wird der Datensatz, sowie seine Aufbereitung vorgestellt. Anschließend wird kurz das Kreuzvalidierungsverfahren erläutert, welches in die Modelle integriert sein wird. Für die Wahl eines geeigneten Modells werden verschiedene Modelle aus dem Bereich Statistical Learning miteinander verglichen. Die Prognosegüte wird anhand des jeweiligen Test-*Root Mean Square Error* (RMSE) beurteilt. Bei den vorgestellten Modellen handelt es sich um lineare und nicht-lineare Modelle. Bei den linearen Modellen liegt der Fokus auf den Dimensionsreduktionsmethoden, bei den nicht-linearen Modellen auf Random Forests und Gradient Boosting. Dabei kann ebenfalls grob eingeschätzt werden, ob die Annahme der Linearität die Prognosegüte eines Modells auf Basis dieses Datensatzes beeinträchtigt. Anschließend werden die Ergebnisse zusammengefasst.

2 Daten

Der untersuchte Datensatz beinhaltet die Statistiken aus FIFA 19, einem 2018 erschienenen Teil der FIFA-Reihe. Einzusehen sind sie auf der Website *sofifa.de*. Enthalten sind 15207 Beobachtungen und 89 Variablen. Bei der Variable *Wage* handelt es sich um den wöchentlichen Lohn der Fußballspieler. Um diesen schätzen zu können, müssen die Daten zunächst so aufbereitet werden, dass die anzuwendenden Modelle fehlerfrei arbeiten können.

2.1 Datenaufbereitung

Zunächst werden alle Variablen entfernt, bei welchen eine Beziehung zum Lohn ausgeschlossen werden kann. Dabei handelt es sich um die im Spiel zugeordnete Spieler-ID und die aufsteigende Laufvariable des Spielers im Datensatz. Weitere entfernte Variablen sind Hyperlinks zum Spielerporträt, die Flagge und das Logo des jeweiligen Fußballsclubs. Bei 11 Prozent der Spielern fehlen Angaben zu positionsspezifischen Fähigkeiten. Einige Modelle, wie beispielsweise *Random Forests* würden diese Spieler nicht in ihrer Schätzung berücksichtigen. Dadurch würde die restliche Information durch diese Spieler bei anderen Variablen verloren gehen. Die Variablen der positionsspezifischen Fähigkeiten werden daher aus dem Datensatz entfernt. Des Weiteren enthält der Datensatz einige wenige Spieler, welche keine individuellen Attribute angegeben haben. Diese werden über eine beliebige Variable, welche diese Attribute erfasst, identifiziert und ebenfalls

aus dem Datensatz entfernt. Gelöscht wurden auch 1,2 Prozent der Spieler, welche keinem Club angehören. Bei diesen existiert kein gültiger Arbeitsvertrag und sie beziehen, eventuell unabhängig von ihren Fertigkeiten, kein Gehalt. Diese Spieler könnten somit die Prognosegüte senken. Da *sofifa.de* einige Fehler bei den Angaben der Variable *Body Type* machte, wird die gesamte Variable entfernt. Die Information über den Figurtyp kann außerdem auch ungefähr aus den Variablen *Height* und *Weight* gewonnen werden. Einige Spieler wurden an einen anderen Club verliehen. Ihren Vertrag schlossen sie jedoch mit dem verleihenden Club. Der aktuelle Club hatte wahrscheinlich keinen Einfluss auf die vorherigen Gehaltsverhandlungen. Aus diesem Grund wird für alle ausgeliehenen Spieler der Club aus der Variable *Loaned From* in die *Club*-Variable gespeichert. Da sich die Leihe im Fußball mittlerweile zu einem Spekulationsgeschäft entwickelt hat, ist es dennoch anzunehmen, dass diese sich auf das Gehalt auswirken könnte(Bierschwale 2019). Um diese Auswirkung zu erfassen, wird eine Dummyvariable erstellt. Diese nimmt den Wert 1 an, wenn der Spieler verliehen wurde und 0, wenn nicht. Die Variablen zur Vertragslaufzeit enthalten bei verliehenen Spieler die Dauer des Ausleihgeschäfts. Diese Daten wahrscheinlich in keiner Beziehung zum Lohn, zum anderen werden die Variablen dadurch insgesamt inkonsistent. Wie bereits erwähnt, können diese Werte für betreffende Spieler nicht problemlos auf *NA* gesetzt werden. Daher werden alle Variablen, welche die Vertragslaufzeit beschreiben entfernt. Die Variablen *Wage*, *Value* und *Release Clause* sind als Klasse *character* gespeichert. Dies wurde korrigiert, indem die Werte ausgeschrieben und das Eurozeichen entfernt wurde. Anschließen wurden die Variablen als *numeric* gespeichert. Hat ein Spieler bei *Release Clause* keine Angabe, wird diese Variable für ihn auf 0 gesetzt. Größe und Gewicht der Spieler sind in Pfund und Fuß, sowie als *character* angegeben. Sie werden in Kilogramm und Centimeter umgerechnet und anschließend als *numeric* gespeichert. Für jeden Spieler ist die Nationalität bekannt. Kleinere Nationen haben beispielsweise jedoch eine geringere Anzahl an Spielern vorzuweisen. Dadurch sinkt die Information zu dieser Nation und es kann zu Fehleinschätzungen kommen. Des Weiteren führt die Berücksichtigung jeder Ausprägung der Variable *Nationality* zu einem komplizierteren Modell mit wahrscheinlich geringem zusätzlichen Informationsgehalt. Es wird daher eine Land-Region Zuordnung der Weltbank genutzt, um die einzelnen Ausprägungen in der Variable *Region* in größere Gruppen zusammenzufassen (Bank n.d.). Dadurch kann die Anzahl an Dummyvariablen verringert und die Information pro Ausprägung gesteigert werden. Im letzten Schritt der Datenaufbereitung wird der bereinigte Datensatz in einen Trainings- und einen Testdatensatz geteilt. Der Trainingsdatensatz enthält 70 Prozent und der Testdatensatz 30 Prozent der Daten. Dadurch können die Modelle an den Trainingsdatensatz angepasst und ihre Prognosegüte anhand des Testdatensatzes geprüft werden.

2.2 Datenübersicht

Nach der Bereinigung umfasst der Datensatz noch 14970 Beobachtungen und 53 Variablen. Darunter befindet sich die zu prognostizierende Variable *Wage*. Im folgenden wird die Struktur dieser Variable, sowie ihr Verhältnis zu den anderen Variablen betrachtet.

Bei der Betrachtung des Boxplots (Abbildung 1) fallen einige Ausreißer außerhalb der Whisker auf. Da es sich wahrscheinlich nicht um Messfehler handelt werden diese im Datensatz belassen. Des Weiteren befinden sich die meisten Datenwerte am linken Rand der Grafik. Der Median ist ebenfalls nicht in der Mitte der Box, sondern weiter links. Die Verteilung der Variable *Wage* ist somit rechtsschief. Eine Normalverteilung ist damit auszuschließen.

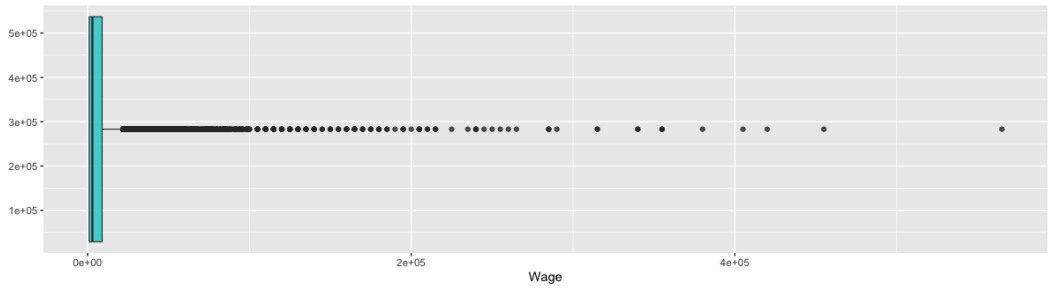

Abbildung 1: Boxplot

Die Korrelationsmatrix (Abbildung 2) zeigt eine geringe Korrelation zwischen *Wage* und den meisten anderen Variablen. Lediglich die Variablen *Value*, *Release.Clause*, *International.Reputation* und *Overall* haben eine mittelstarke, positive Korrelation mit dem Gehalt der Spieler. Die Korrelationen mit den restlichen Variablen ist gering und meist positiv. Wenn die erklärenden Variablen keine oder nur geringe Beziehungen zur abhängigen Variable aufweisen, kann dies die Vorhersage erschweren. Die erklärenden Variablen sind untereinander teilweise stark korreliert. Vor allem Variablen, welche die individuellen Fertigkeiten der Spieler beschreiben scheinen eine hohe Korrelation aufzuweisen. Die Richtung der Korrelation ist dabei abhängig von der Position des jeweiligen Spielers. So sind die Fähigkeiten der Feldspieler positiv untereinander, aber negativ mit den Fähigkeiten der Torhüter korreliert. Die spezifischen Fähigkeiten der Torhüter sind untereinander ebenfalls positiv stark korreliert. In der Korrelationsmatrix befinden sich nur die signifikanten Werte.

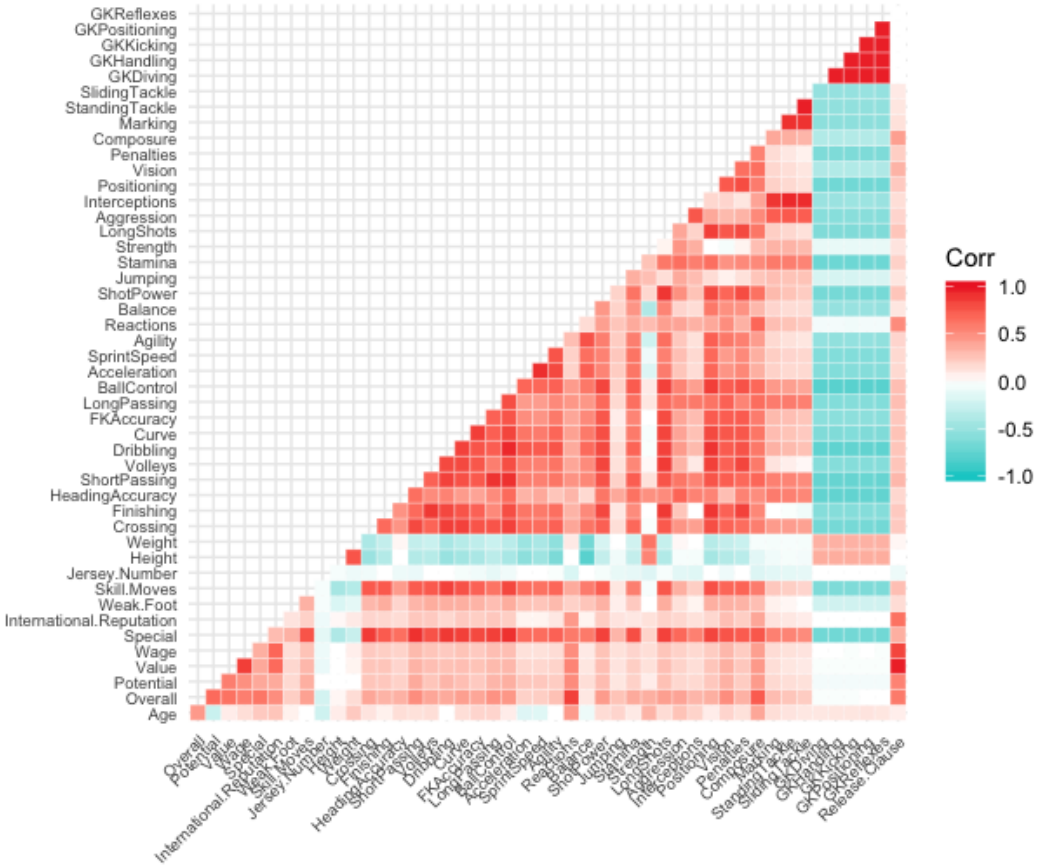

Abbildung 2: Korrelationsmatrix

Insgesamt zeigen sich mehrere Aspekte, welche die Prognose der Variable *Wage* an Hand der vorhandenen Variablen beeinträchtigen können. Es gilt somit, Modelle zu verwenden, welche diese Schwierigkeiten abmildern können.

3 Kreuzvalidierung

Der bereinigte Datensatz wurde in einen Trainings- und einen Testdatensatz unterteilt. Auf Basis des Trainingsdatensatzes werden die Modelle angepasst. Anschließend soll das Gehalt der Spieler des Testdatensatzes geschätzt werden. Die Schätzwerte werden dann mit den wahren Werten des Testdatensatzes verglichen. Durch die Berechnung des RMSE wird die Prognosegüte der einzelnen Modelle vergleichbar.(James et al. 2017, S.181) Im Gegensatz zum MSE ist der RMSE besser interpretierbar, da er der durchschnittlichen Abweichung entspricht. Trainings- und Testdatensatz wurden zufällig gewählt. Es kann passieren, dass die Wahl des Modells davon abhängt, welche Daten sich im Trainings-, und welche im Testdatensatz befinden. Um innerhalb der Modelloptimierung ein Modell zu wählen, welches robust gegenüber Änderungen der Zusammensetzung

des Trainingsdatensatzes ist. Eine Maßnahme dafür ist die wiederholte Kreuzvalidierung (CV). Die CV basiert auf dem gleichen Prinzip wie die Aufteilung in Trainings- und Testdatensatz. Zuerst wird der Datensatz zufällig in eine vorher festgelegte Anzahl an k Teilmengen aufgeteilt. Anschließend wird aus diesen Teilmengen die Menge $k-1$ genutzt, um das Modell zu schätzen. Die Teilmenge, welche nicht verwendet wird, dient als Testdatensatz. Somit kann der CV-RMSE berechnet und für jedes Modell verglichen werden. Dieses Prinzip wird so oft wiederholt, dass jede Teilmenge k einmal als Testdatensatz genutzt wurde. Der RMSE des gesamten Modells ergibt sich dann über das arithmetische Mittel aller k RMSE. (James et al. 2017, S.181) Wie der Test-RMSE ist der CV-RMSE abhängig von der zufälligen Unterteilung der Daten. Durch die iterative Nutzung der k Teilmengen als Testdatensatz wird diese Abhängigkeit reduziert. Um die Modelle mit einer größeren Variation der Daten zu testen, wird die wiederholte CV verwendet. Mit jeder Wiederholung wechselt die zufällige Aufteilung der Daten. Die Varianz der Datensätze steigt. Der endgültige RMSE der Modelle ist dann das arithmetische Mittel über die Anzahl der Wiederholungen. Nachfolgend wird bei den linearen Modelle die zehnfache CV mit drei Wiederholungen durchgeführt.

4 Lineare Modelle

Trotz einiger restriktiver Annahmen, wie insbesondere einer linearen Beziehung zwischen der abhängigen und der erklärenden Variable, bieten lineare Modelle oftmals eine ausreichende Approximation des wahren Verhältnisses der Daten. Ein geeigneter Einstieg für die Einschätzung einer potenziellen linearen Beziehung zwischen Antwortvariable und Regressoren ist die lineare Regression. Ausführlicher wird anschließend auf die Dimensionsreduktionsmethoden eingegangen.

4.1 Multiple lineare Regression

Die lineare Regression ist eine simple Form des überwachten Lernens. Die multiple lineare Regression ist eine Erweiterung der einfachen linearen Regression. Es werden mehrere Regressoren zur Erklärung der abhängigen Variable verwendet. Somit hat das Modell mit p Regressoren die Form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon, \tag{1}$$

bei der X_j den j-ten Regressor und β_j den Regressionskoeffizient darstellen. Letzterer misst den Einfluss der eklärenden Variablen auf Y, der abhängigen Variable. (James et al. 2017, S. 71f) Die Variable *Wage* wird zunächst auf alle 52 erklärenden Variablen des Trainingsdatensatzes regressiert. Das Modell wird mit Hilfe des wiederholten 10-fachen Kreuzvalidierungsverfahrens ausgewählt. Das gewählte Modell hat ein angepasstes R-Quadrat von 0,8867. Die F-Statistik ist signifikant mit einem Wert von 112 auf dem 0.001 Signifikanzniveau. Dies lässt darauf schließen, dass mindestens eine der erklärenden Variablen einen signifikanten Einfluss auf das wöchentliche Gehalt der Spieler hat.

Von den 740 erklärenden Variablen (inklusive der unterschiedlichen Ausprägungen der Dummy-Variablen) haben nur 94 einen signifikanten Einfluss auf die abhängige Variable. In Kombination mit den hohen Werten des angepassten R-Quadrats und der signifikanten F-Statistik kann dies

auf die Kollinearität von zwei oder mehreren Regressoren deuten. Kollinearität erschwert es dem Modell, den individuellen Einfluss korrelierter Regressoren auf die zu erklärende Variable zu bestimmen. Dadurch sinkt die Genauigkeit der Schätzer und deren Standardfehler steigt. Da die t-Statistik aus dem Quotient von $\hat{\beta}_j$ und dessen Standardfehler besteht, fällt die t-Statistik bei Vorliegen von Kollinearität. Somit sinkt die Macht des t-Tests.(James et al. 2017, S. 99-101) Die Auswirkungen von Kollinearität auf die Voraussage von Daten sind umstritten. Während einige Quellen einen Einfluss negieren, gehen andere von einem Einfluss in einigen Szenarien aus (Mundfrom, Smith, and Kay 2018, S. 26).

Bereits die Korrelationsmatrix in Kapitel 2 deutet auf die Kollinearität der Regressoren hin. Eine weitere Möglichkeit zur Aufdeckung von Multikollinearität ist der Varianzinflationsfaktor (VIF). Dieser berechnet sich als Quotient der Varianz von $\hat{\beta}_j$ des gesamten Modells und der Varianz von $\hat{\beta}_j$, wenn der Regressor des Modells nur j ist. Der kleinste Wert, den der VIF annehmen kann, ist 1. Ab Werten von 5 bis 10 wird die Multikollinearität innerhalb des Modells als problematisch eingestuft, da es zu den oben genannten Auswirkungen kommen kann. (James et al. 2017, S. 101f.)

Die Berechnung des VIF für das multiple lineare Modell ergibt, dass 44 erklärende Variablen einen Wert von fünf oder höher haben. Bei 36 davon liegt der Wert sogar über zehn. Es gibt zwei Möglichkeiten, diese Multikollinearität zu beseitigen. Zum einen können problematische Variablen aus dem Modell entfernt werden. Auf Grund der Kollinearität ist der Informationsgehalt dieser Variablen für das Modell gering. Dennoch ist es mitunter schwer, zu entscheiden, welche der kollinearen Variablen entfernt werden soll. Eine Entfernung der *falschen* Variable, könnte die Performance des Modells negativ beeinflussen. Eine weitere Möglichkeit ist die Zusammenfassung dieser Variablen in einem einzelnen neuen Regressor.(James et al. 2017, S. 102)

Mit diesem multiplen linearen Regressionsmodell werden, basierend auf dem Testdatensatz, neue Werte prognostiziert. Anschließend wird der Test-RMSE berechnet. Dieser liegt bei 7878,359. Der Wert entspricht somit fast dem arithmetischen Mittel der Variable Wage.

4.2 Hauptkomponentenregression

Um die potenziell problematische Multikollinearität in den Regressoren zu beseitigen wird eine Hauptkomponentenregression (PCR) durchgeführt. Dabei werden die Regressoren zu Komponenten zusammengefasst, welche eine lineare Kombination dieser darstellen. Anschließend wird die abhängige Variable auf die Komponenten regressiert. Eine Entfernung der korrelierten Variablen aus dem Modell garantiert nicht, dass lineare Kombinationen von Regressoren nicht mit weiteren Regressoren korrelieren. Bei der PCR hingegen sind die einzelnen Komponenten unkorreliert. (Kuhn and Johnson 2013) Bei der PCR handelt es sich um eine Dimensionsreduktionsmethode. Die Dimension der ursprünglichen Regressoren wird von $p+1$ Regressoren auf $M+1$ Komponenten reduziert. Dabei sind $Z_1, Z_2, ..., Z_M$ die $M < p$ linearen Kombinationen der p Regressoren. Daraus ergibt sich

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j \tag{2}$$

für die Konstanten $\phi_{1m}, \phi_{2m}, ..., \phi_{pm}, m = 1, ..., M$. Nach der Transformation der Regressoren wird eine kleinste Quadrate Regression durchgeführt

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{im} + \epsilon_i, \qquad i = 1, ..., n, \tag{3}$$

wobei $\theta_0, \theta_1, ..., \theta_M$ die Regressionskoeffizienten sind. (James et al. 2017, S. 229) Die erste Hauptkomponente ist die lineare Kombination der Regressoren, welche die höchste Varianz hat. Selbiges gilt für die zweite Hauptkomponente, unter der Bedingung, dass diese mit ersterer unkorreliert ist. Durch diese Bedingung stehen die Hauptkomponenten orthogonal zueinander. (James et al. 2017, S. 231-233)

Aus der Konstruktion der Hauptkomponenten ergibt sich ein absteigender Informationsgehalt. Während die erste Komponente den größten Teil der Varianz der Regressoren erklärt, nimmt dieser Erklärungsgehalt für jede folgende Komponente ab. Dies bedeutet, dass oftmals eine geringe Anzahl an Komponenten ausreicht, um einen Großteil der Variabilität der Daten, sowie der abhängigen Variable zu erklären. Bei hoher Kollinearität ist der Informationsgehalt der Daten meist in geringerer Dimension enthalten als der ursprünglichen. In diesem Fall führt die PCR zu einem besseren Ergebnis als die multiple lineare Regression. (James et al. 2017, S. 233)

Vor der Durchführung der PCR werden die numerischen Variablen des Trainingsdatensatzes zunächst skaliert und um ihren Mittelwert zentriert. Eine unterschiedliche Skalierung kann sich auf die PCR auswirken. Bei unterschiedlichen Maßeinheiten könnten bestimmte Variablen mit einer hohen Varianz beispielsweise eine bedeutendere Rolle in der Richtung der Hauptkomponente einnehmen (James et al. 2017, S. 236)). Anschließend wird erneut die Variable Wage mit Hilfe eines PCR-Modells auf die erklärenden Variablen des Trainingsdatensatzes regressiert. Die maximale Anzahl an Komponenten wird auf 100 festgelegt. Mit Hilfe der wiederholten CV wurde das beste Modell an Hand des CV-RMSE gewählt. In diesem Fall ist der CV_RMSE mit einem Wert von 10511,77 bei $M = 100$ Hauptkomponenten, der maximal möglichen Anzahl, am geringsten. Die erste Komponente ist bereits in der Lage, 43,07 Prozent der Variabilität in den Regressoren zu erklären (siehe Abbildung 3). Wie erwartet nimmt der zusätzliche Informationsgehalt mit steigender Komponentenanzahl ab. Die zweite Komponente kann nur noch 10,31 Prozent der Regressorvariabilität erklären. Ab circa $M = 75$ Komponenten verläuft die Kurve fast horizontal. Die erklärte prozentuale Regressorvarianz mit jeder zusätzlichen Komponente ist hier verschwindend gering.

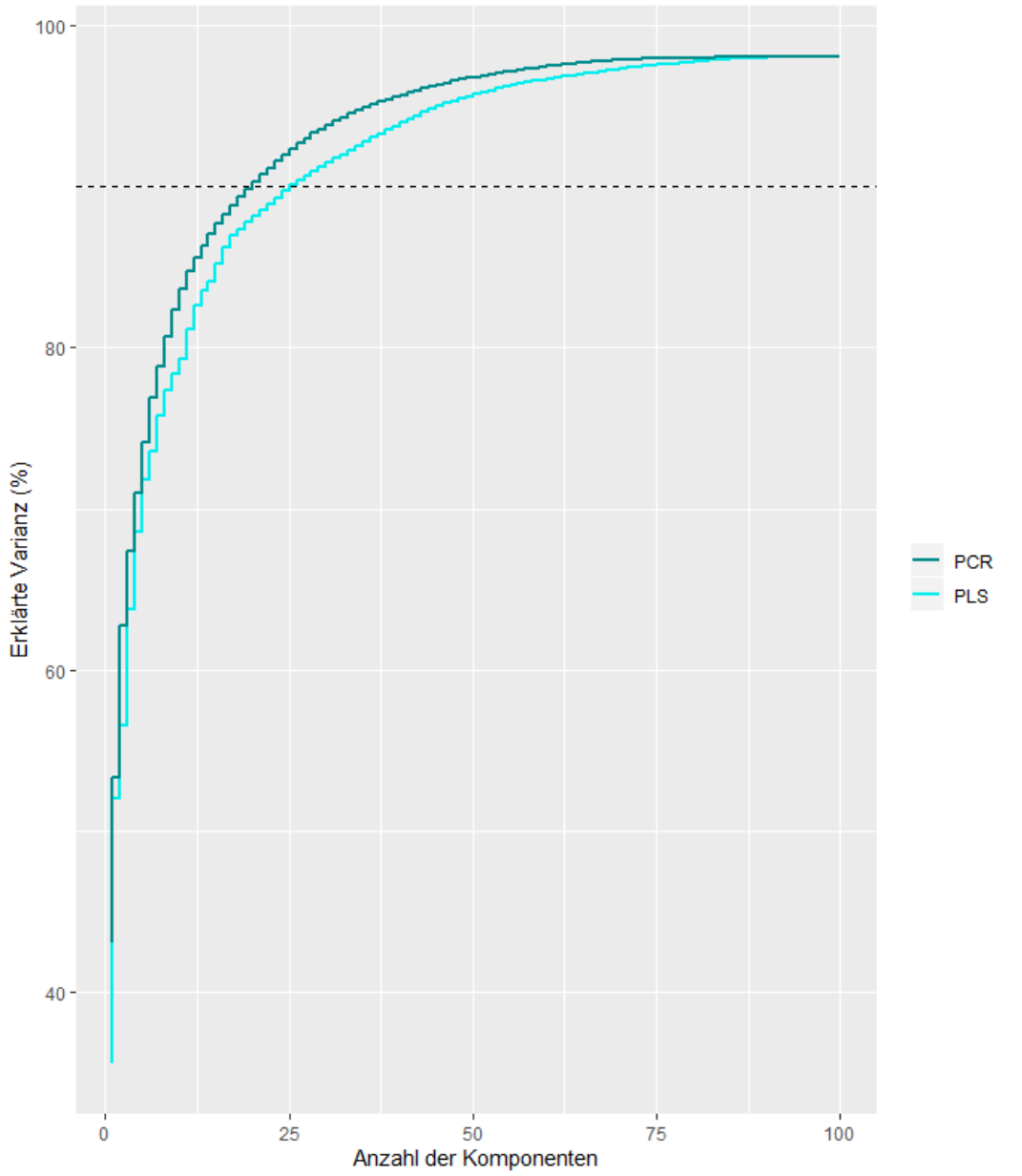

Abbildung 3: Erklärte Regressorvarianz je Komponentenanzahl

Die zentrale Idee der PCR ist es, möglichst viel Regressorvarianz mit möglichst wenigen Komponenten zu erklären. Um das Modell so simpel wie möglich zu halten wird $M = 21$ als finale Anzahl der Komponenten gewählt. Dies ist die geringste Komponentenanzahl, welche mindestens 90 Prozent der Varianz erfasst. Es können 90,31 Prozent der Regressorvarianz erklärt werden. Abbildung 4 zeigt die 15 einflussreichsten Variablen in diesem PCR-Modell. Am höchsten ist die Bedeutung der Variablen *Value*, *Release.Clause* und *International.Reputation*. Bei diesen 15 Variablen handelt es sich um die 15 numerischen Variablen, welche die höchste Korrelation mit *Wage* haben.

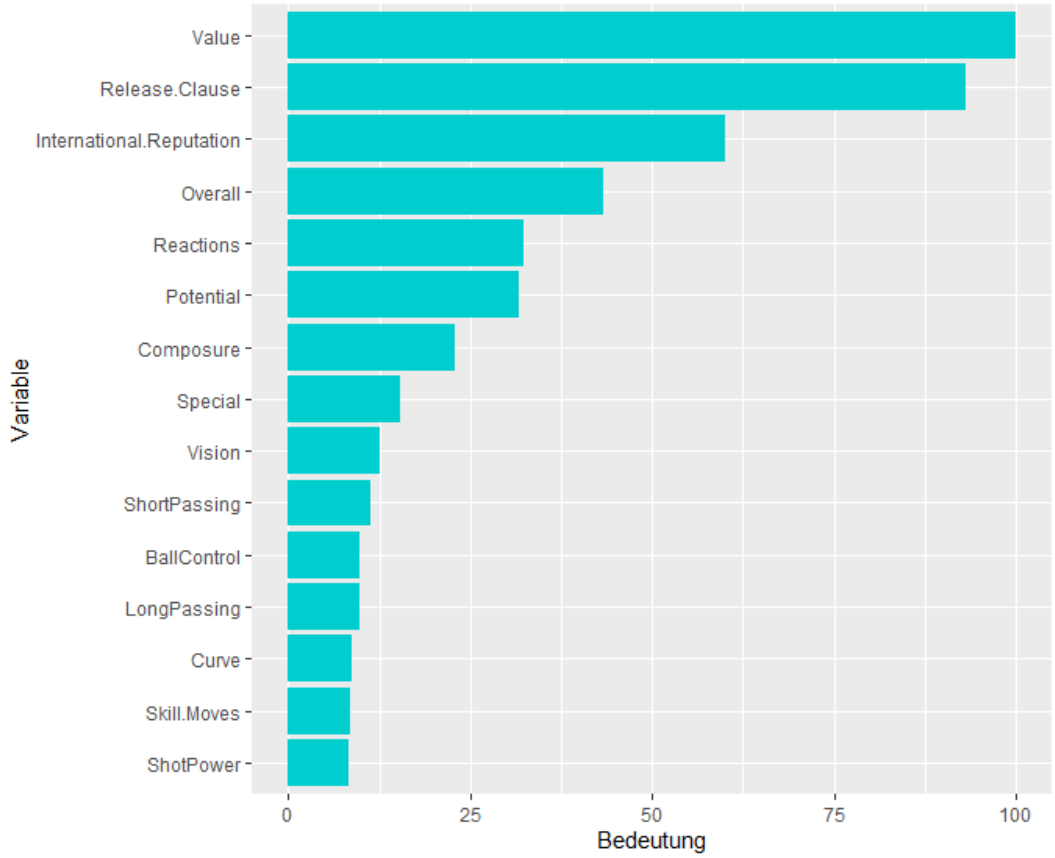

Abbildung 4: Einflussreichste Variablen im PCR-Modell

Auf Basis des 21-Komponentenmodells wird eine Prognose des Lohns für den Testdatensatz erstellt. Der RMSE beträgt 11483,9 und liegt damit weit über dem RMSE des multiplen linearen Regressionsmodells. Ein möglicher Grund für die schlechte Prognosegüte des Modells kann an einem Nachteil der PCR liegen: Bei der Bildung der Komponenten wird nur die Varianz der Regressoren berücksichtigt. Es muss nicht sein, dass diese Komponenten auch die Varianz der Antwortvariable erklären können. Abbildung 5 zeigt, dass das PCR-Modell bei 21 Komponenten 74,28 Prozent der Varianz der Antwortvariable erklären kann. Nichteinmal das gemäß CV-RMSE optimale Modell mit 100 Komponenten schafft es, 80 Prozent dieser Varianz zu begründen. Vor allem bei kleiner Komponentenanzahl ist die Aussagekraft des Modells bezüglich der Varianz der Antwortvariable gering.

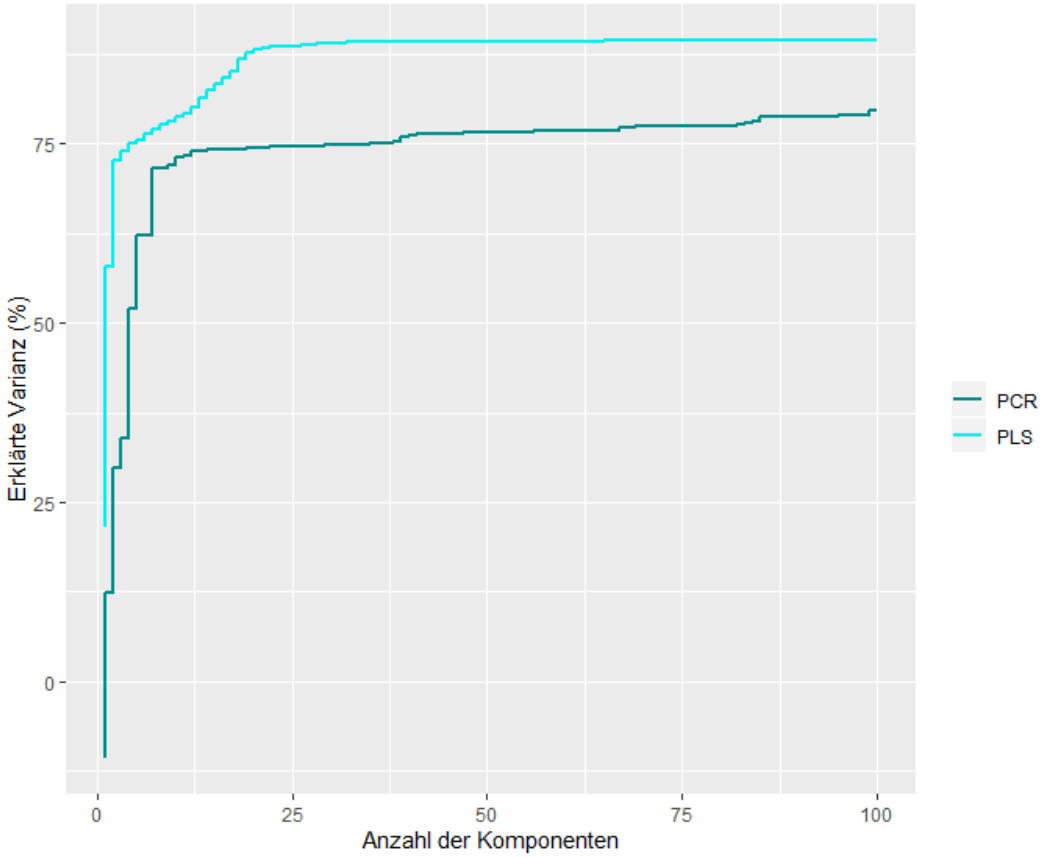

Abbildung 5: Erklärte Varianz der Antwortvariable nach Komponentenanzahl

Das Ziel der Modelle ist in diesem Fall nicht die Erklärung der Regressorvarianz, sondern die bestmögliche Prognose der Antwortvariable. Zumindest auf Basis des vorliegenden Datensatzes scheint PCR dazu weniger geeignet. Eine Alternative zu PCR, welche auch die Antwortvariable bei der Komponentenbildung berücksichtigt, ist die Regression der partiellen kleinsten Quadrate (PLS).

4.3 Regression der partiellen kleinsten Quadrate

Im Gegensatz zur PCR identifiziert die PLS Komponenten, welche zusätzlich die Antwortvariable gut erklären können. Zur Berechnung der ersten Komponente Z_1 wird zunächst jedes ϕ_{j1} gleich dem Koeffizienten der linearen Regression von Y auf X_j gesetzt. Dadurch gewichtet PLS Variablen mit stärkerer Beziehung zu Y stärker. Anschließend wird jede Variable auf Z_1 regressiert. Die Residuen dieser Regression werden als verbliebene Information interpretiert. Auf Basis dieser orthogonalisierten Daten wird analog zu Z_1 die zweite Komponente Z_2 berechnet. Auf die M Komponenten kann dann die abhängige Variable Y regressiert werden. PLS ist somit eine überwachte Alternative zur PCR. (James et al. 2017, S. 337f)

Bei der PLS werden ebenfalls die standardisierten und zentrierten numerischen Regressoren verwendet. Die maximale Anzahl an Komponenten wird analog zum PCR-Modell auf 100 begrenzt. Das am CV-RMSE gemessene beste Modell hat eine Komponentenanzahl von $M = 93$ und einen CV-RMSE von 7826,032. Es benötigt somit sieben Komponenten weniger, als das optimale

PCR-Modell. Zur Erklärung einer Regressorvarianz von mindestens 90 Prozent benötigt das PLS-Modell 26 Komponenten, dies sind fünf mehr als bei der PCR (siehe Abbildung 3). Generell kann PLS bei gleicher Komponentenanzahl minimal weniger Regressorvarianz begründen als PCR. Dies kann daran liegen, dass die Komponenten nicht mehr nur auf Basis der Regressorvarianz gebildet werden, sondern auch die Antwortvariable *Wage* miteinbezogen wird.

Bei der Erklärung der Varianz der Antwortvariable erreicht PLS bei gleicher Komponentenanzahl ein deutlich besseres Ergebnis als PCR. Mit 26 Komponenten wird 88,64 Prozent dieser Varianz aufgeklärt. Das sind 14,11 Prozent mehr, als beim entsprechenden PCR-Modell. Generell kann PLS hier mit kleinerer Komponentenanzahl einen vergleichweise deutlich höheren Anteil der Varianz der Antwortvariable abbilden (Abbildung 5).

Abbildung 6 zeigt, dass die 15 wichtigsten Variablen im PLS-Modell ähnlich der des PCR-Modells sind. *Value*, *Release.Clause* und *International.Reputation* sind weiterhin die einflussreichsten drei Variablen. Bei der PLS-Regression haben jedoch auch einige qualitative Variablen einen stärkeren Einfluss. Diese sind Ausprägungen der Variablen *Preferred.Foot*, *Region*, *Work.Rate* und *Club*. Des Weiteren haben hier alle 15 Variablen eine stärkere Bedeutung für das Modell. Im PCR-Modell nahm die Bedeutung der Variablen nach der viertwichtigsten Variable stark ab.

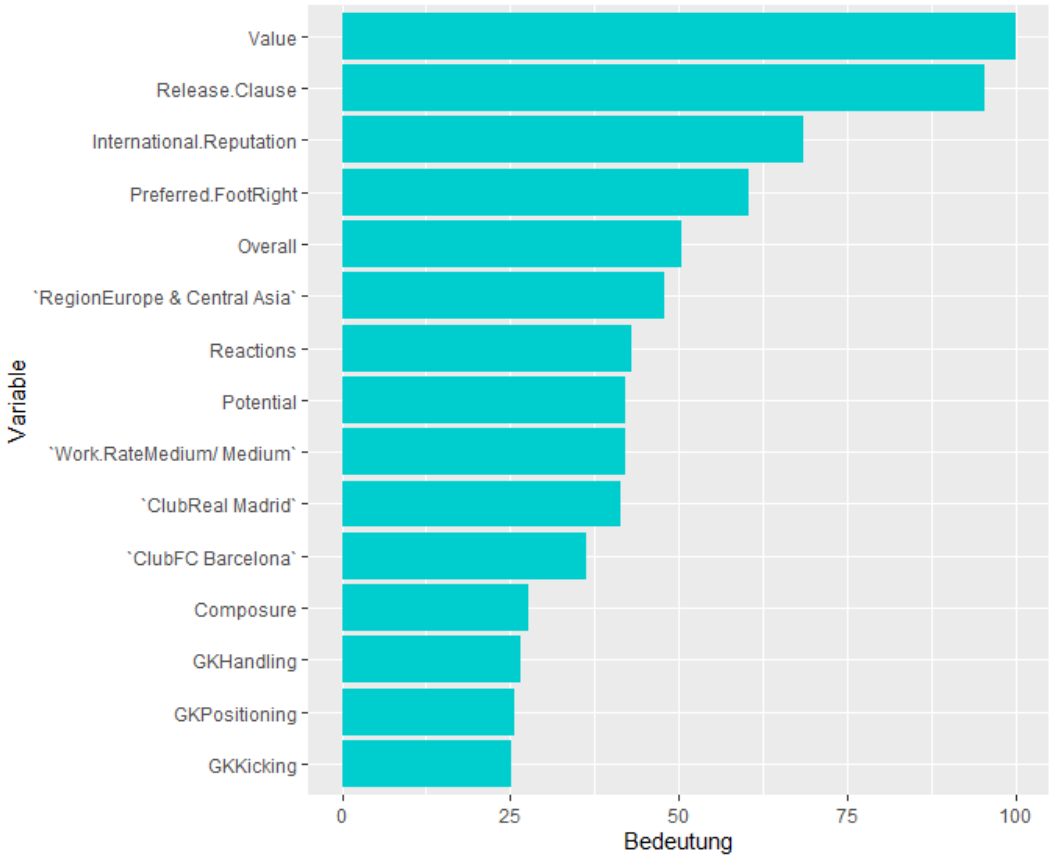

Abbildung 6: Einflussreichste Variablen im PLS-Modell

Das 26 Komponenten-Modell erreicht bei der Prognose der Werte des Testdatensatzes einen RMSE von 8295,128. Das ist deutlich geringer als im PCR-Modell. Dennoch unterliegen beide Modelle dem multiplen linearen Regressionsmodell. Erst bei $M = 100$ schafft es das PLS-Modell

sich dem RMSE der multiplen linearen Regression anzunähern.

4.4 Vergleich linearer Modelle

Gemessen am Test-RMSE schneiden PCR und PLS bei der Prognose vergleichweise schlecht ab 7. Selbst die gemäß CV-RMSE optimalen Modelle mit höherer Komponentenanzahl sind in der Prognose nicht besser, als die multiple lineare Regression.

Modell	Lin. Reg.	PCR	PLS	Ridge	Lasso
RMSE	7878.359	11483.9	8295.128	7962.344	7886.491

Abbildung 7: Test-RMSE der linearen Modelle

Wie zuvor erwähnt lässt sich dies für PCR mit der fehlenden Berücksichtigung der Antwortvariable *Wage* bei der Komponentenbildung erklären. Des Weiteren ist die Performance von PCR abhängig von der Struktur des Trainingsdatensatzes. Die Antwortvariable kann zuverlässig vorhergesagt werden, wenn eine kleine Komponentenanzahl ausreicht, um einen Großteil der Varianz von Regressoren und Antwortvariable zu erklären (James et al. 2017, S. 335). Auf Grund der ähnlichen Vorgehensweise bei der Komponentenbildung lässt sich dies auch bei PLS vermuten. In beiden Modellen wurde auf Basis des Trainigsdatensatzes mit $M = 21$ und $M = 26$ keine geringe Komponentenanzahl gewählt. Das liegt unter anderem daran, dass selbst die ersten Komponenten beider Modelle verhältnismäßig wenig Information enthalten. Die lineare Kombination der Regressoren mit der höchsten Varianz kann somit nur einen geringen Teil der Gesamtvarianz der Regressoren abbilden.

Ein weiterer Grund für die schlechte Performance von PCR und PLS ist möglicherweise die Dimension des Datensatzes. Dimensionsreduktionsmethoden können die Varianz der angepassten Koeffizienten bei einer hohen Anzahl an Regressoren und verhältnismäßig geringer Anzahl an Observationen senken. Des Weiteren besteht die Möglichkeit, dass die geschätzten Koeffizienten bei Dimensionsreduktionsmethoden nicht erwartungstreu sind. (James et al. 2017, S. 330) Der hier verwendete Datensatz hat mit 14970 Beobachtungen und 53 Variablen keine hohe Dimension. Daher ist es denkbar, dass die Varianz nicht ausreichend gesenkt werden kann und die Verzerrung der Schätzer überwiegt.

Des Weiteren schließen Dimensionsreduktionsmethoden keine Variablen aus dem Modell aus. PCR und PLS ähneln damit eher der Ridge Regression, statt dem Lasso. Ridge Regression schrumpft die Koeffizienten stufenlos, im Gegensatz zu den diskreten PCR- und PLS-Modellen (Hastie, Friedman, and Tibshirani 2001, S. 82). Lasso ist die einzige der genannten Methoden, welche die Koeffizienten auf 0 setzen kann. (James et al. 2017, S. 219) In diesem Fall könnte dies die bessere Prognosegüte des Lassos erklären, da redundante Variablen einfach vom Modell ausgeschlossen werden können. So kann ebenfalls die Multikollinearität beseitigt werden.

Da der Test-RMSE der multiplen linearen Regression hier am geringsten ist lässt sich vermuten, dass Multikollinearität in den linearen Modellen die Prognose kaum beeinflusst.

Trotz der Verwendung unterschiedlicher linearer Modelle unterscheiden sich die Test-RMSE, mit Ausnahme des PCR-Modells, nur marginal voneinander. Die Prognosen der vier besten Modelle weichen im Durchschnitt fast um ein arithmetisches Mittel und sogar dem doppelten Median von den wahren Werten der Variable *Wage* ab. Keines der Modelle scheint optimal zu sein, um die Variable *Wage* möglichst genau auf Basis neuer Daten vorherzusagen.

Eine mögliche Erklärung ist die bereits erwähnte geringe Korrelation zwischen der Antwortvariable *Wage* und den Regressoren. Es ist außerdem möglich, dass relevante Regressoren nicht berücksichtigt werden konnten, da sie nicht im Datensatz enthalten sind. Durch diese Fehlspezifikation entstünde eine Verzerrung der Schätzfunktion. Auch die Rechtsschiefe der Verteilung der Antwortvariable, sowie die Ausreißer können zu der schlechten Prognosegüte beigetragen haben. Denkbar wäre hier eine Transformation der Variable *Wage*. Diese würde hier jedoch den Rahmen der Arbeit überschreiten. Ein weiterer Grund könnte ebenfalls das Vorliegen einer nichtlinearen Beziehung zwischen dem Gehalt der Spieler und den Regressoren sein. In diesem Fall scheinen die linearen Modelle keine ausreichende Approximation der wahren Beziehung abbilden zu können. Aus diesem Grund sollten nichtlineare Modelle in Betracht gezogen werden. Da sowohl Polynome, als auch Splines nicht auf die vorhandenen Faktorvariablen angewendet werden können, wird zunächst eine Prognose mit baumbasierten Methoden versucht.

5 Entscheidungsbäume

Auf Grund der potenziell nicht linearen Zusammenhänge und der Multikollinearität der Variablen bieten sich Entscheidungsbaum basierte Regressionsmodelle zur Schätzung der Variable *Wage* an. Diese teilen die Datenpunkte der abhängigen Variable in Teilbereiche $R_1, R_2, ..., R_J$ auf, welche auf Regeln zur Ausprägung der unabhängigen Variablen basieren (James et al. 2017, S. 306). Zur Verringerung der Rechenzeit erfolgt die Aufteilung in Bereiche rechteckig. Ziel dieser Aufteilung ist es, die Residuenquadratsumme (RSS) zu minimieren. In jedem Bereich wird zunächst das arithmetische Mittel gebildet und anschließend der positive Abstand der Datenpunkte zu diesem betrachtet.

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{4}$$

Auf Grund der hohen Anzahl an möglichen Kombinationen der Bereiche, ist eine Berechnung aller Möglichkeiten nicht praktikabel. Deshalb wird der *Top-Down*-Ansatz verwendet. Es wird erst ein Schätzer und eine Regel ausgewählt, die den Gesamtbereich so teilen, dass die größtmögliche Reduzierung des RSS erreicht wird. Die darauf entstandenen Bereiche werden getrennt betrachtet. Die darauf folgende Aufteilung wird nach diesem Prinzip fortgeführt. (James et al. 2017, S. 306). Letztendlich führt dies zu einem Entscheidungsbaum, welcher für jede Ausprägung der Daten einen Knoten besitzt. Der RSS ist somit minimal. Jedoch käme es so zu einem *Overfitting*. Der Trainingsdatensatz wird zwar perfekt beschrieben, die Prognosegüte für andere Datensätze ist aber gering. Um dem vorzubeugen, kann beispielsweise festgelegt werden, dass jeder neue Knoten eine festgelegte Anzahl von Beobachtungen enthalten muss (James et al. 2017, S. 306). Anschließend wird an jedem Endknoten eines Baumes ein Durchschnitt gebildet, welcher die entsprechende Schätzung für neue Daten darstellt. In Abhängigkeit vom Datensatz kann die optimale Anzahl an Knotenpunkten für eine Schätzung neuer Daten bereits überschritten sein.

In diesem Fall muss der Baum entsprechend gekürzt werden, um *Overfitting* zu vermeiden. Jede Teilungsregel stellt einen Knoten im Entscheidungsbaum dar. Diese Regeln sind abhängig von der Datenstruktur. Die Varianz der Schätzung ist über einen einzelnen Baum entsprechend hoch. Die Modelle gelten als nichtlinear, da gleiche oder ähnliche Ausprägungen einer Variable in unterschiedlichen Bereichen des Baumes liegen können. Da in jedem Schritt jeweils eine Variable betrachtet wird, beeinflusst Multikollinearität die Schätzung nicht. Multikollinearität ist dennoch bei der Interpretation der Regeln eines Baumes zu berücksichtigen. So kann es beispielsweise sein, dass eine Variable an einem frühen Knoten des Baumes den RSS deutlich reduziert. Dieser Effekt kann jedoch auch nur wegen einer hohen Korrelation dieser Variable mit einer anderen Variable existieren.

5.1 Random Forest

Um die Abhängigkeit von der Struktur des Trainingsdatensatzes zu reduzieren, wird das Bootstrapverfahren verwendet. Dieses zieht aus dem Datensatz mehrfach, zufällig und mit zurücklegen einen neuen Stichprobendatensatz mit der gleichen Anzahl an Ausprägungen. Dadurch erhöht sich die Variation der Daten. Mit jeder gezogenen Stichprobe wird ein separater Entscheidungsbaum berechnet. (James et al. 2017, S. 190) Die Variablen, welche bei der Aufteilung der Daten im Entscheidungsbaum zur Verfügung stehen, sind über alle Bäume gleich. Somit ist die entstehende Variation in den Bäumen allein durch die Datenvariation bedingt. Dieses Verfahren heißt *Bootstrap aggration (Bagging)*. Bei der Anwendung des *Bagging* auf neue Daten wird jeder Entscheidungsbaum mit den entsprechenden Daten berechnet. Der Schätzwert ist der Mittelwert der verschiedenen Ergebnisse. (Breiman 1996, S. 123) Beim *Bagging* werden in jedem Knoten und für jeden Baum sowohl dieselbe, als auch die maximale Anzahl an möglichen Variablen für das Unterteilen der Daten genutzt ($p = m$, mtry in R). Aus dieser Anzahl an möglichen Variablen wird auf Grundlage der RSS-Minimierung eine ausgewählt. Dadurch entsteht in der Wahl der Variablen eine Korrelation. Außerdem werden insbesondere in den ersten Knoten des Baumes häufiger dieselben Variablen gewählt. Durch Hinzufügen eines Zufallsprozesses in der Variablenauswahl jedes Knotens, reagiert das Modell flexibler auf die Variation der Daten. Diese Erweiterung des *Baggings* wird *Random Forest* genannt. (Breiman 2001, S. 6.) Der größtmögliche Zufallsprozess ist hier, dass in jedem Knoten nur eine zufällige Variable zur Verfügung steht ($m = 1$), auf deren Grundlage die Daten geteilt werden müssen. Die Anzahl der Variablen bewegt sich dann zwischen einer möglichen Variable oder allen verfügbaren. Die Variablenanzahl stellt somit eine Möglichkeit dar, das Modell zu optimieren. Als eine gute Annäherung für die Anzahl der möglichen Variablen m hat sich $\frac{p}{3}$ herausgestellt.(Probst, Wright, and Boulesteix 2019, S. 2)

Da Entscheidungsbäume zu *Overfitting* neigen, weisen sie bei der Anwendung auf andere Datensätze oft eine hohe Verzerrung auf. In einem *Random Forest* wird die Wahrscheinlichkeit des *Overfittings* durch zwei Eigenschaften gesenkt. Zum einen führt das *Bootstrapping* dazu, dass die Bäume nicht auf einen einzigen Datensatz angepasst werden. Zum anderen sinkt die Korrelation zwischen den Entscheidungsbäumen durch die zufällige Auswahl der möglichen Variablen in jedem Knoten. Durch Erhöhung der Varianz kann die Korrelation zwischen den Bäumen weiter gesenkt werden (Probst, Wright, and Boulesteix 2019, S. 3). Wenn nicht der vollständige *Bootstrap*-Trainingsdatensatz verwendet wird, sondern nur eine Teilmenge, steigt die Variation des Datensatzes. Dadurch reagiert der *Random Forest* flexibler auf die vorhandene Varianz der unbekannten Datensätze. Die Genauigkeit und Stabilität eines *Random Forest*-Modells wird ebenfalls

durch die Anzahl der Knoten der Entscheidungsbäume beeinflusst. Eine hohe Anzahl an möglichen Knoten oder eine geringe Anzahl an Beobachtungen in jedem geteilten Bereich führt zu einer eher höheren Anpassung an den Trainingsdatensatz (Probst, Wright, and Boulesteix 2019, S. 3). Eine höhere Anzahl an Beobachtungen in jedem Bereich oder eine geringe Anzahl an maximalen Knoten können die Stabilität des Modells erhöhen. Die Genauigkeit sinkt dadurch jedoch. Die Berechnung verschiedener Parameterkombinationen erfolt hier über das R-Pakte *ranger*. Es basiert auf dem *Random Forest*-Algorithmus (Wright and Ziegler 2017). Die Vorteile von *ranger* gegenüber *caret* sind eine deutlich geringere Rechenzeit. Außerdem kann mit dem *carte*-Paket nur eine geringere Anzahl an Hyperparametern festgelegt werden. Im Gegensatz zum *Random Forest*-Algorithmus lässt *ranger* die Verwendung von Faktorvariablen mit mehr als 32 Ausprägungen zu. Dies ist ein großer Vorteil, da die Auswertung des PLS-Modells die Bedeutung bestimmter Ausprägungen der Variable *Club* angab. Im ersten Schritt werden verschiedene Kombinationen der Hyperparameter ausgewählt. Es können $(200, 400, 600, 800, 1000)$ Bäume gewählt werden. Die Anzahl an wählbaren Variablen besteht aus einer vierschrittigen Sequenz zwischen 20 und 52. Die minimale Knotenanzahl beträgt $(3, 4, 5)$ und die maximale Tiefe ist $(10, 20, 30)$. Die Stichprobengröße ist $(0, 63, 0.75, 1)$. Zum Vergleich der unterschiedlichen Modelle wird der *Out-of-Bag-RMSE* (OOB-RMSE) verwendet. Dieser wird berechnet, indem die im *Bootstrap* nicht gezogenen Beobachtungen als Testdatensatz verwendet werden. Anders als bei den vorherigen Modellen wird auf Kreuzvalidierung verzichtet, da die Rechnenzeit sonst unverhältnismäßig stark ansteigen würde. Der OOB-RMSE ist jedoch mit dem CV-RMSE vergleichbar. Das Modell mit dem geringsten OOB-RMSE von 10085,24 hat die Hyperparameter $mtry = 44$, mindestens drei Ausprägungen an jedem Knoten, eine maximale Knotenanzahl von 30, sowie einen Trainingsdatensatz von 100 Prozent und 1000 Bäume. Es fällt auf, dass mit steigender Variablenanzahl, die an jedem Knoten zur Verfügung steht, der OOB-RMSE sinkt. Dies könnte mit der geringen Korrelation des Lohnes mit den erklärenden Variablen zusammenhängen. Das Modell könnte also eine größere Anzahl an möglichen Variablen benötigen, um sich den Daten anzupassen. (Bernard, Heutte, and Adam 2009, S. 177) Um zu überprüfen, wie das Modell auf einen unbekannten Datensatz reagiert, wird der Test-RMSE berechnet. Für dieses Modell ergibt sich ein Test-RMSE von 9399,45 Dieser liegt deutlich über dem Test-RMSE der linearen Modelle. Dieses Ergebnis könnte mit der angesprochenen geringen Korrelation der zwischen *Wage* und den erklärenden Variablen liegen. Eine Schwäche des *Random Forest* Algorithmus ist es, bei einem niedrigen Signal-zu-Rausch Verhältnis die geringen Informationen aus dem Rauschen zu filtern.(Reis, Baron, and Shahaf 2018)

Ein Vorteil des Entscheidungsbaumes ist die leichte Interpretierbarkeit des Modells. Durch die Anwendung des *Bootstrap* und der Vielzahl an Bäumen ist im *Bagging* oder im *Random Forest* eine einzelne Betrachtung eines Entscheidungsbaumes nicht mehr aussagekräftig für die Variablenbedeutung im Modell (James et al. 2017, S. 319). Um die Bedeutung der Variablen im Modell dennoch einordnen zu können, wird untersucht in welchem Maß diese zur Reduktion des RSS beitragen. Hierfür wird das arithmetische Mittel über die RSS-Reduktion für jede Variable über alle Bereichsteilungen gebildet (Abbildung 8).(James et al. 2017, S. 319) Es ist festzustellen, dass vor allem die Variablen *Overall*, *Value* und *Club* einen hohen Einfluss auf den RSS haben. Das Ergebnis für diese drei Variablen ähnelt der Variablenbedeutung des PLS-Modells. Im Vergleich zu PCR und PLS ist aber auffällig, dass sich der hohe Einfluss auf weniger Variablen verteilt. Für die Variablen, welche in diesem Ranger-Modell den stärksten Effekt hatten, konnte zuvor keine starke Korrelation mit anderen Variablen festgestellt werden. Multikollinearität beeinflusst hier also nicht die Interpretation der Variablen.

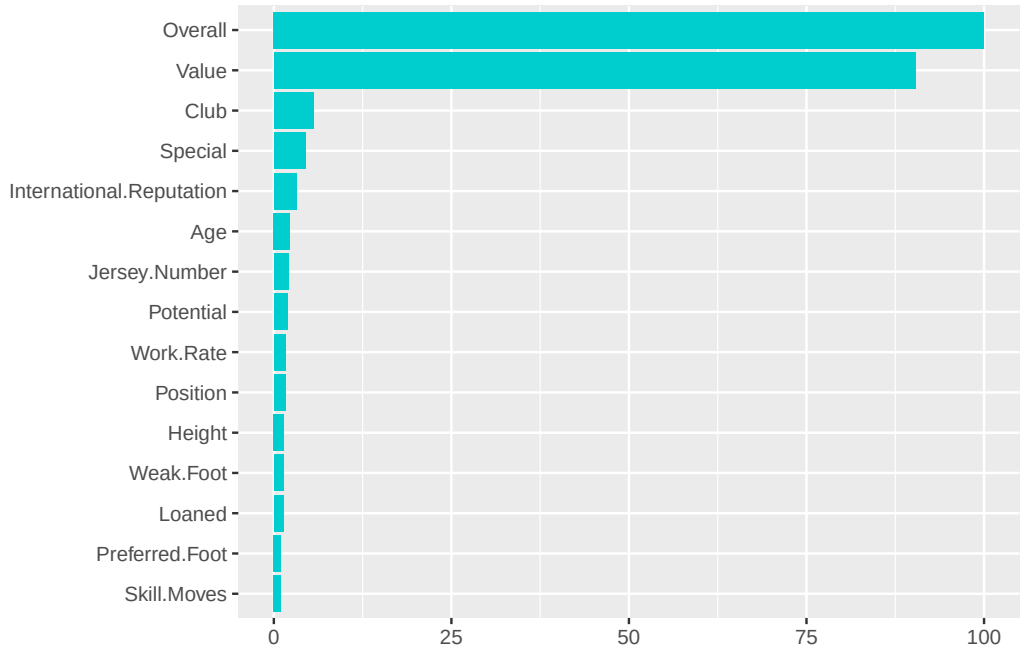

Abbildung 8: Bedeutung Random Forest

5.2 Boosting

Ein weiteres Entscheidungsbaum basiertes Modell zur Schätzung einer Regression ist *Boosting*. Die besondere Stärke des *Boostings* ist es durch ein kleinschrittiges Vorgehen schwache Informationsdichten abbilden zu können. Im Gegensatz zum *Random Forest* kommt es dabei nicht zum unabhängigen Entstehen von Entscheidungsbäumen. Die aufeinanderfolgenden Entscheidungsbäume basieren auf den Vorhersagen vorheriger Entscheidungsbäume. Zur Vorhersage der Variable *Wage* wird *Gradient Boosting* verwendet (Ridgeway, n.d.). Dieses basiert auf einem mehrstufigen, sich wiederholenden Algorithmus (Friedman 1999). Im Laufe dieses Prozesses wird die Verlustfunktion $L(y(F(x))$, in diesem Fall die quadratische Abweichung, minimiert.

$$L(y(F(x)) = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{5}$$

Der erste Schritt wird auf Basis der Verlustfunktion ein Schätzer für die abhängige Variable, das Gehalt der Fußballspieler berechnet. In diesem ersten Schritt entspricht diese $F_0(x)$ dem Mittelwert über alle Spielergehälter.(Friedman 1999, S. 2)

$$F_0(x) = argmin_p \sum_{i=1}^{n} L(y_i, p) \tag{6}$$

Der zweite Schritt findet iterativ statt. Jede Iteration entspricht einem Entscheidungsbaum ($m =$ 1bis M). Der erste Teil des zweiten Schritts ist die Berechnung der Residuen der aktuellen Schät-

zung (m) (Friedman 1999, S. 2).

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad fuer\ i = 1, ...n \tag{7}$$

Die berechneten Residuen werden nun anstelle des Einkommens genutzt, um einen Entscheidungsbaum zu bestimmen. Es ist nicht auszuschließen, dass in den Blättern (j) des Entscheidungsbaumes verschiedene Residuen (R) liegen. Daher wird die Verlustfunktion genutzt, um den Schätzwert (γ) der einzelnen Blätter zu bestimmen. (Friedman 1999, S. 2) Aufgrund der quadratischen Abweichung als Verlustfunktion, ist der Durchschnitt der Residuen in den einzelnen Blättern der Schätzwert.

$$\gamma_{jm} = argmin_\gamma \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma) \tag{8}$$

Anschließend werden alle bisherigen Schätzungen summiert. Dabei wird jede Schätzung, außer der ersten ($F_0(x)$), mit der *Learning Rate* (ν) multipliziert, welche zwischen 0 und eins liegt. (Friedman 1999, S. 2) Eine geringe *Learning Rate* reduziert den Effekt jedes einzelnen Entscheidungsbaumes auf die Schätzung. Damit wird die Wahrscheinlichkeit des *Overfittings* reduziert. Aus dem so geschätzten Wert und dem wahren Wert werden wiederum die Residuen bestimmt und anschließend der Algorithmus iteriert.

$$F_m(x) = F_{m-1}(x) + \nu \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \tag{9}$$

Im Vergleich zum *Random Forest* soll beim *Boosting* die Schätzgenauigkeit durch die Kombination geringer Informationsdichten verbessert werden. Zur Verringerung der Varianz wird dem *Gradient Boosting* eine Kreuzvalidierung vorausgestellt. Bei der Anwendung des *gbm*-Paketes (Ridgeway, n.d.) über das *caret*-Paket optimiert das Modell automatisch die Anzahl der Bäume, sowie die maximale Anzahl der Teilungen. Im Gegensatz zum *Random Forest* kann es durch eine hohe Anzahl an Bäumen zum *Overfitting* kommen, da sich die Schätzung durch häufiges Wiederholen von Schritt zwei langsam, aber stetig an den Trainingsdatensatz anpasst. Die Anzahl der Teilungen pro Baum kontrolliert die Komplexität der Entscheidungsbäume. Eine hohe Anzahl an Teilungen führt zu einer Senkung der Varianz der Schätzung, bei gleichzeitigem höheren Risiko des *Overfittings*. Wie beim *Random Forest* besteht beim *Gradient Boosting* die Möglichkeit weitere Parameter zu optimieren. Wie bereits erwähnt wird die Wahrscheinlichkeit des *Overfittings* durch eine geringere *Learning Rate* reduziert, da sich mit jedem Baum kleinschrittiger an den wahren Wert des Trainingsdatensatzes angenähert wird. So wird es ermöglicht, eine hohe Anzahl an Bäumen zu modellieren. Die Tiefe der Entscheidungsbäume kann, neben der Anzahl der Teilungen, auch über die Mindestanzahl der Beobachtungen in jedem Knoten angepasst werden. Eine geringe Mindestanzahl an Beobachtungen führt zu einer sinkenden Varianz der Schätzung. Gleichzeitig steigt aber auch die Möglichkeit einer Überanpassung an den Trainingsdatensatz. Um ein *Gradient Boosting*-Modell mit geringem CV-RMSE zu finden, werden verschiedene Kombinationen von Hyperparametern berechnet. Aus Effizienzgründen beinhalten das Modell nur eine dreifache Kreuzvalidierung. Auf Grund der Erfahrung mit dem *ranger*-Modell wird eine geringere Anzahl an Bäumen gewählt. Diese beträgt $(300, 500, 700)$, mit einer maximalen Anzahl an Teilungen von $(4, 5, 6)$. Die mögliche *Learning rate* beträgt $(0,01, 0,05, 0,1)$ Prozent und die

Mindestanzahl der Beobachtungen pro Knoten ist $(3, 4, 5)$. Das gewählte Modell besitzt einen CV-RMSE von 7314,727. Die Anzahl der benötigten Bäume ist 500, bei einer Interaktionstiefe von 6. Die minimale Anzahl an Beobachtungen an den Endknoten beträgt 3. Die *Learning Rate* liegt bei 0,1. Damit konnte das *Gradient Boosting* im Vergleich zum *Random Forest* in *ranger* bei gleicher Anzahl von Bäumen einen besseren OOB- bzw. CV-RMSE erreichen. Bei der Prognose kann das *Gradient Boosting*-Modell einen Test-RMSE von 5360,765 erreichen. Damit handelt es sich mit Abstand um das Modell mit der besten Prognosegüte. Im Vergleich zum *Random Forest* lässt sich dieses Ergebnis unter anderem auf die geringe Korrelation zwischen der Variable *Wage* und den erklärenden Variablen zurückführen. Durch den kleinschrittigen Ansatz ist *Boosting* besser in der Lage, geringe Information in die Schätzung mit einzubeziehen.

Beim *Boosting* lässt sich ebenfalls berechnen, welche Variablen den RSS innerhalb des Modells am stärksten reduziert haben. Im Vergleich zum *Random Forest* sinkt die Bedeutung von *International.Reputation* und die der *Release.Clause* steigt. *Value* und *Overall* sind weiterhin wichtige Variablen bei der Erstellung des Modells (Abbildung 9). Wie beim der PLS zeigt sich, dass besonders einzelne Clubs eine große Bedeutung auf die Prognose von *Wage* haben. In beiden Modellen sind dies vor allem Clubs die bekannt sind für sportliche Erfolge und hohe Ausgaben für Spieler. Von den zehn am besten verdienenden Spielern des Datensatzes stehen acht beim FC Barcelona oder bei Real Madrid unter Vertrag. Besonders hoher wöchentlicher Lohn kann somit im *Boosting* präziser abgebildet werden.

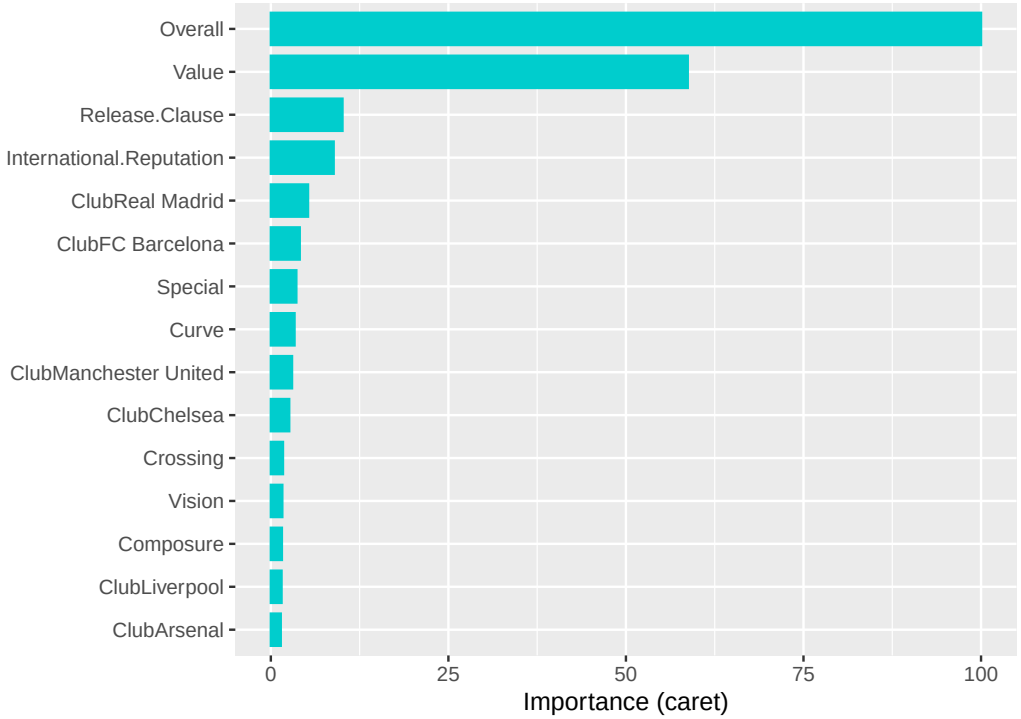

Abbildung 9: Bedeutung GBM

6 Zusammenfassung

Das Ziel dieser Arbeit war es, das wöchentliche Gehalt der Fußballspieler des FIFA19-Datensatzes vorherzusagen. Dazu wurden unterschiedliche Modelle des *Statistical Learnings* angewandt. Bei der Untersuchung des gesamten Datensatzes zeigten sich einige Ausreißer mit besonders hohen Gehältern. Außerdem konnte eine hohe Korrelation einiger erklärenden Variablen festgestellt werden. Die Korrelation der Antwortvariable *Wage* mit den Regressoren war, bis auf wenige Ausnahmen, gering. Die Berechnung des VIF stützte die These einer multikollinearen Beziehung zwischen den erklärenden Variablen. Da diese Kollinearität das multiple lineare Modell beeinflussen könnte, wurde ebenfalls eine Schätzung mit PCR und PLS durchgeführt. Im Vergleich mit anderen linearen Modellen wiesen diese beiden jedoch den höchsten Test-RMSE auf. Dies lag vermutlich daran, dass auf Grund der Datenstruktur die ersten Komponenten verhältnismäßig wenig Information beinhalteten. Daher wurden recht viele Komponenten benötigt. Bei der PCR kam eine unzureichende Erklärung der Varianz der Antwortvariable durch die Komponenten hinzu. Insgesamt war der Test-RMSE des multiplen linearen Modells am geringsten, auch wenn die Unterschiede (außer bei PCR) nur gering waren. Daher kann vermutet werden, dass der Einfluss der Multikollinearität auf die Prognose gering ist.

Insgesamt ist der Test-RMSE als hoch einzuschätzen. Gründe dafür können die geringe Korrelation der Antworvariable *Wage* mit den Regressoren sein. Außerdem kann eine Nichtlinearität vorliegen, sodass die linearen Modelle die wahre Beziehung der Daten nicht ausreichend abbilden können.

Aus diesem Grund wurden die nichtlinearen, entscheidungsbaumbasierten Modelle *Random Forest* und *Boosting* untersucht. Basierend auf der Theorie des *Random Forest*-Algorithmus wurden unterschiedliche Optimierungsparameter abgestimmt. Es wurde die Kombination gewählt, welche den OOB-RMSE minimiert. Der Test-RMSE der Prognose war jedoch schlechter, als der der linearen Modelle.

Beim *Boosting* wurden ebenfalls verschiedene Parameterkombinationen untersucht und das Modell mit dem geringsten CV-RMSE gewählt. Der Test-RMSE dieses Modells ist der niedrigste aller hier getesteten Modelle. Dies liegt vermutlich daran, dass *Boosting* auf Grund seines Algorithmus' geeignet ist, besonders schwache Zusammenhänge zu erkennen.

Die Modelle unterscheiden sich in der Bedeutung der einzelnen Variablen. In den linearen Modellen haben deutlich mehr Variablen eine höhere Importanz. Beim *Random Forest* und *Boosting* haben die Variablen *Overall* und *Value* mit hohem Abstand den höchsten Einfluss. Generell sind diese beiden Variablen, zuzüglich der Variable *International.Reputation*, in allen Modellen von hoher Bedeutung. Dies sind auch die Variablen, welche die höchste Korrelation mit der Antwortvariable *Wage* aufweisen können. An Hand der Variablenbedeutung zeigt sich, dass Variablen mit einer geringen Korrelation zu *Wage* keinen starken Einfluss auf das Modell haben. Da dies die meisten Variablen des Datensatzes betrifft, kann dies tatsächlich der Grund sein, warum die meisten Modelle keine hohe Prognosegüte haben. Die wenigen Variablen mit hoher Korrelation scheinen für eine präzise Vorhersage nicht auszureichen. Es ist ebenfalls denkbar, dass relevante Variablen nicht im Datensatz enthalten sind, und somit die Schätzfunktion verzerrt wird.

Wie bereits erwähnt, kann *Boosting* selbst aus schwachen Zusammenhängen lernen. Eine weitere Anpassung der Parameter, beispielsweise durch eine Erhöhung der Anzahl der Entscheidungsbäume, kann dieses Modell sicherlich weiter optimieren. Auf Grund mangelnder Rechenleistung musste an dieser Stelle darauf verzichtet werden.

Eine weitere Möglichkeit zur Verbesserung der Prognose sind Modelle, welche ein niedriges Signal-Rausch Verhältnis gut abbilden können. Hier würde sich ein auf Wahrscheinlichkeitsverteilungen beruhender *Probabilistic Random Forest* anbieten (Reis, Baron, and Shahaf 2018).

Der vorliegende Datensatz könnte außerdem mit zusätzlichen Variablen angereichert werden. So könnte das Hinzufügen von bisherigen ausgelassenen Variablen potenzielle Verzerrungen reduzieren.

Literaturverzeichnis & Software

Bank, World. n.d. "World Bank Data Help Desk." Accessed August 20, 2019. https://datahelpdesk.worldbank.org/ knowledgebase/articles/906519.

Bernard, Simon, Laurent Heutte, and Sébastien Adam. 2009. "Influence of Hyperparameters on Random Forest Accuracy." In *Multiple Classifier Systems*, edited by Jón Atli Benediktsson, Josef Kittler, and Fabio Roli, 5519:171–80. Berlin, Heidelberg: Springer Berlin Heidelberg.

Bierschwale, Jens. 2019. "Fußball-Leihspieler: Wenn Minderjährige Zum Spekulationsobjekt Verkommen." *DIE WELT*, July. https://www.welt.de /sport/fussball/article197174447/Fussball-Leihspieler-Wenn-Minderjaehrige-zum-Spekulationsobjekt-verkommen.html.

Breiman, Leo. 1996. "Bagging Predictors." *Machine Learning* 24 (2): 123–40.

———. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

"FIFA 17 Player Ratings Blend Stats, Subjectivity." 2016. *ESPN.com*. https://www.espn.com /soccer/blog/espn-fc-united/68/post/2959703/fifa-17-player-ratings-system-blends-advanced-stats-and-subjective-scouting.

"Football Leaks: 106 Mio. Euro Für Lionel Messi in Barcelona." 2018. https://www.faz.net /aktuell/sport/fussball/football-leaks-106-mio-euro-fuer-lionel-messi-in-barcelona-15397130.html.

Friedman, Jerome H. 1999. "Stochastic Gradient Boosting." https://statweb.stanford.edu/~jhf/ftp/stobst.pdf.

Hastie, Trevor, Jerome Friedman, and Robert Tibshirani. 2001. "Linear Methods for Regression." In *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, edited by Trevor Hastie, Jerome Friedman, and Robert Tibshirani, 41–78. Springer Series in Statistics. New York, NY: Springer New York.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2017. *An Introduction to Statistical Learning: With Applications in R*. 1st ed. 2013, Corr. 7th printing 2017. New York: Springer.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. New York, NY: Springer New York.

Mundfrom, Daniel, Michelle Smith, and Lisa Kay. 2018. "The Effect of Multicollinearity on Prediction in Regression Models." *General Linear Model Journal* 44 (January): 24–28.

Probst, Philipp, Marvin N. Wright, and Anne-Laure Boulesteix. 2019. "Hyperparameters and Tuning Strategies for Random Forest." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9 (3): e1301.

Reis, Itamar, Dalya Baron, and Sahar Shahaf. 2018. "Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Datasets." *The Astronomical Journal* 157 (1): 16. http://arxiv.org/abs/1811.05994.

Ridgeway, Greg. n.d. "Generalized Boosted Models: A Guide to the Gbm Package," 15.

Wright, Marvin N., and Andreas Ziegler. 2017. "Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R." *Journal of Statistical Software* 77 (1).