Vergleich der Güte von unterschiedlichen Klassifikationsfamilien

Katharina Schober

Seminar KI: gestern, heute, morgen
Angewandte Informatik, Otto-Friedrich-Universität Bamberg

Matrikelnummer: 1740817
Studiengang: Computing the Humanities
Semester: 2. Fachsemester

4. Februar 2018

Zusammenfassung. Für das Klassifikationslernen findet man in der Literatur sehr viele verschiedene Algorithmen. Mit Hilfe der Güte der Klassifikatoren soll in diesem Artikel die Existenz einer so großen Auswahl an Klassifikatoren hinterfragt werden. Für die Analyse dieser Problematik werden die Untersuchungen aus dem Artikel "Do we need hundreds of classifieres to solve real world classification problems" von Fernández-Delgado et al. betrachtet. Hierfür werden 179 Klassifikatoren aus 17 Klassifikationsfamilien (Diskriminanzanalyse, Bayes, Neuronale Netze, Support Vector Machines, Entscheidungsbäume, regelbasierte Methoden, Boosting, Bagging, Stacking, Random Forest, andere Sets, generalisierte lineare Modelle, Methode des nächsten Nachbarn, partiell kleinste Quadrate und Regression der Hauptkomponenten, logistische und multinominale Regression, multivariate adaptive Regressionssplines, andere Methoden) mit Hilfe von 121 Datensätzen verglichen. Diese werden hauptsächlich aus dem "UCI Machine Learning Repository" gewonnen, es werden außerdem vier reale Datensätze herangezogen. Für die Untersuchung finden die Programme C, C++, Matlab, R (mit und ohne caret) und Weka Verwendung. Die Verfahren Friedman Ranking, PAMA, P95 und PMA vergleichen die Güte der Klassifikatoren und bringen diese in eine Reihenfolge. Das Friedman Ranking wird auch zur Betrachtung der Klassifikationsfamilien herangezogen. Hier lässt sich feststellen, dass der Random Forest die beste Klassifikationsfamilie ist. Sein Ranking Intervall ist sehr schmal und niedrig. Sehr gute Klassifikatoren dieser Familie sind hierbei der parRF_t, rf_t und rforest_R. An zweiter Stelle folgt der Support Vector Machine, mit den sehr guten Klassifikatoren svm_C, LibSVM_w und svmRadicalCost_t. Die drittbeste Klassifikationsfamilie sind die Neuronalen Netze, wobei hierbei die Klassifikatoren elm_kerne_m und C5.0_t nennenswert sind.

Schlüsselwörter: Klassifikator, Güte, Random Forest, Neuronale Netze, Support Vector Machine

1 Einleitung

Bei der Datenanalyse kommt es immer wieder zum selben Problem. Man hat einen Datensatz und möchte diesen klassifizieren, aber weiß nicht welchen Klassifikatoren man dafür am besten verwendet. Es gibt eine große Auswahl an sehr unterschiedlichen Klassifikatoren aus verschiedenen Bereichen wie beispielsweise die Mathematik, Informatik oder Statistik und es ist oftmals schwer den passenden auszuwählen. Außerdem spielt das verwendete Analyseprogramm eine wichtige Rolle, da dort die Klassifikatoren unterschiedlich vorhanden sind. Doch warum gibt es überhaupt so viele Klassifikatoren? Zudem lassen sich die Klassifikatoren in Klassifikationsfamilien zusammenfassen, was das Ganze noch weiter verkompliziert. Aufgrund der beschriebenen Problematiken stellt sich folgende Forschungsfrage:

> "Bestätigt die Güte der Klassifikatoren die Existenz einer so großen Anzahl an möglichen Klassifikatoren?"

Basierend auf den Artikel "Do we need hundreds of classifieres to solve real world classification problems" von Fernández-Delgado et al. ist eine Prüfung von 179 Klassifikatoren aus 17 Klassifikationsfamilie mit Hilfe von 121 Datensätze zu vorhanden (Fernández-Delgado u. a., 2014). Dieser wird herangezogen um die Güte der Klassifikatoren zu vergleichen und somit eine Antwort auf die Forschungsfrage zu geben. Im Folgenden werden auf die untersuchten Klassifikatoren eingegangen. Außerdem wird eine Auswahl an Klassifikationsfamilien genauer betrachtet. Darauf folgt die Beschreibung der verwendeten Datensätze und die Analyse. Hierfür wird in Kapitel 4 die Güte mit Hilfe von vier verschiedenen Verfahren verglichen um im Anschluss auch auf die Güte bezüglich der Klassifikationsfamilien einzugehen. Abschließend wird in einem Fazit die Forschungsfrage beantwortet.

2 Klassifikationsfamilien

Um die Güte der Klassifikatoren vergleichen zu können, müssen zuerst die Klassifikationsfamilien, denen die einzelnen Klassifikatoren angehören, erläutert werden. Insgesamt werden 179 Klassifikatoren verglichen, welche in C, C++, Matlab, R (mit und ohne caret) und Weka implementiert werden (Fernández-Delgado u. a., 2014). Diese sind auf 17 Klassifikationsfamilien verteilt (Fernández-Delgado u. a., 2014). Im Folgenden werden die einzelnen Klassifikationsfamilien mit der jeweiligen Anzahl an Klassifikatoren tabellarisch aufgeführt.

Tabelle 1. Klassifikationsfamilien

Klassifikationsfamilie	Anzahl
Diskriminanzanalyse	20
Bayes	6
Neuronale Netze	21
Support Vector Machines	10
Entscheidungsbäume	14
regelbasierte Methoden	12
Boosting	20
Bagging	24
Stacking	2
Random Forest	8
andere Sets	11
generalisierte lineare Modelle	5
Methode des nächsten Nachbarn	5
partiell kleinste Quadrate und Regression der Hauptkomponenten	6
logistische und multinominale Regression	3
multivariate adaptive Regressionssplines	2
andere Methoden	10

Es ist zu erkennen, dass versucht wird eine große Bandbreite an Klassifikationsfamilien abzudecken. Auffällig ist dabei, dass die Klassifikatoren sehr unterschiedlich auf die einzelnen Klassifikationsfamilien verteilt sind. Die Anzahlen reichen von zwei (z.B. multivariate adaptive Regressionssplines) bis hin zu 24 (z.B. Bagging) Klassifikatoren pro Klassifikationsfamilie. Im weiteren Verlauf sollen nun einzelne Klassifikationsfamilien beispielhaft vorgestellt werden.

2.1 Diskriminanzanalyse

Die Diskriminanzanalyse kommt aus der Statistik und ist ein multivariates Verfahren zur Analyse von Gruppenentscheidungen (Backhaus u. a., 2016a). Hierbei handelt es sich um ein struktur-prüfendes Verfahren, wobei die Abhängigkeit einer nominal skalierten Variable von metrischen Variablen untersucht wird (Backhaus u. a., 2016a). Bei der Diskriminanzanalyse führen 6 Teilschritte zum Ergebnis. Diese lauten wie folgt:

1. Definition der Gruppen,
2. Formulierung der Diskriminanzfunktion,
3. Schätzung der Diskriminanzfunktion,
4. Prüfung der Diskriminanzfunktion,
5. Prüfung der Merkmalsvariablen,
6. Klassifikation neuer Elemente.

(Backhaus u. a., 2016a, vgl. S. 219)

Die Diskriminanzfunktion ist definiert als

$$Y = b_0 + b_1 X_1 + ... + b_J X_J,$$

wobei Y die Diskriminanzvariable, X_j die Merkmalsvariabe, b_j den Diskriminanzkoeffizient für die Merkmalsvariable j und b_0 das konstante Glied darstellt (Backhaus u. a., 2016a, S. 221). Eine entscheidende Rolle spielt die Güte der Diskriminanzanalyse, und wie diese gemessen wird. Zur Veranschaulichung dient die folgende Graphik (Backhaus u. a., 2016a, S. 224).

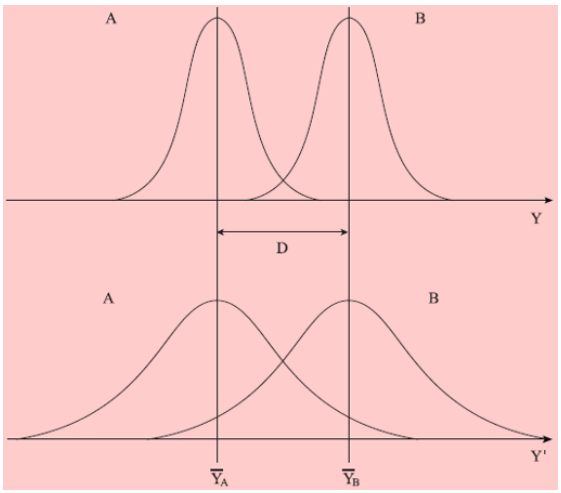

Abb. 1. Diskriminanzanalyse

In Abb. 1 ist zu erkennen, dass die obere Hälfte der Grafik eine sehr gute Güte hat, wohingegen die untere Hälfte das Gegenteil zeigt. Begründen lässt sich dies mit dem Ziel der Diskriminanzanalyse Gruppenunterschiede darzustellen (Backhaus u. a., 2016a). Dieser ist umso besser, wenn die Varianz innerhalb der Gruppen so gering wie möglich, und die Varianz zwischen den Gruppen so groß wie möglich ist (Backhaus u. a., 2016a). Das bedeutet, dass sich die Gruppen am wenigsten vermischen und der gewollte Gruppenunterschied sichtbarer wird. Wie beschrieben hat die obere Hälfte der Graphik eine sehr geringe Varianz innerhalb der Gruppen und eine hohe Varianz zwischen den Gruppen, weswegen die Graphen sich kaum überschneiden.

2.2 Neuronale Netze

Eine zweite Klassifikationsfamilie sind die Neuronalen Netze. Sie beruhen auf dem Vorbild des biologischen Nervensystems (Mitchell, 1997). Hierbei gibt es viele verschiedene Varianten, wobei die bekannteste das Dreischichten-Modell ist. Dieses beinhaltet eine Eingabeschicht (Input-Layer), eine verdeckte Schicht (Hidden-Layer) und eine Ausgabeschicht (Output-Layer) (Backhaus u. a., 2016a).

In Abb. 2 ist das Dreischichten-Model graphisch dargestellt (Backhaus u. a., 2016a, S. 605).

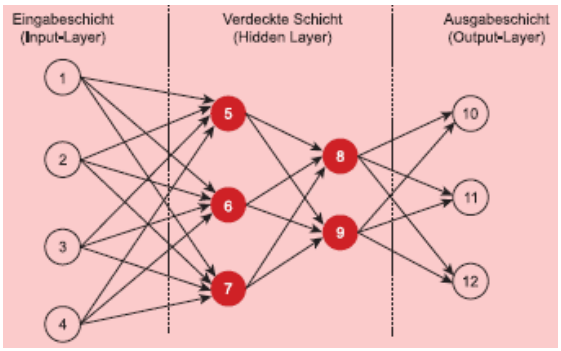

Abb. 2. Neuronale Netze

Im Rahmen der Neuronalen Netze sollte der Backpropagation Algorithmus unbedingt Erwähnung finden. Hierbei wird die Eingabe durch das Netz propagiert und daraufhin die Ausgabe mit der Wunschausgabe verglichen (Backhaus u. a., 2016b). Die Differenz davon stellt den Fehler des Netzes dar (Backhaus u. a., 2016b). Zur Minimierung dieses Fehlers werden Gewichte eingesetzt (Backhaus u. a., 2016b). Durch mehrmaliges Durchlaufen des Netzes soll der Fehler, mit Hilfe der Anpassung der Gewichte, minimiert werden (Backhaus u. a., 2016b). Das Ganze wird abgebrochen, wenn der quadratische Fehler sehr gering ist, also beispielsweise unter 0,1% liegt (Backhaus u. a., 2016b). Abschließend ist zu erwähnen, dass eine hohe Fallzahl Voraussetzung ist, um das Neuronal Netz trainieren zu können (Backhaus u. a., 2016a).

2.3 Support Vector Machine

Eine weitere Klassifikationsfamilie ist die Support Vector Machine. Diese ist besonders geeignet, wenn es sich um Big Data Klassifikationsprobleme handelt (Suthaharan, 2016). Die Support Vector Machine zählt zu den mathematisch und rechnerisch aufwändigen Verfahren (Suthaharan, 2016). Deshalb wird im Folgenden nur ein kurzer Überblick gegeben.

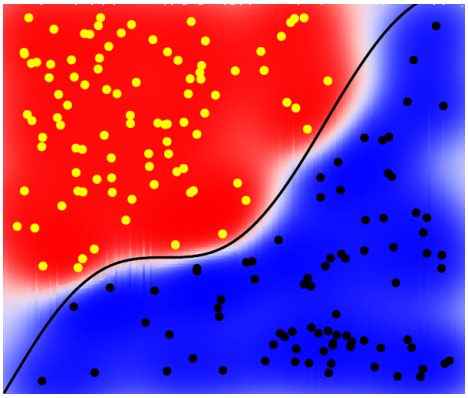

Abb. 3. SVM

Wie in Abb. 3 zu sehen, geht es bei der Support Vector Machine im Allgemeinen darum zwei Klassen bestmöglich voneinander zu trennen (Steinwart u. Christmann, 2008, S. 297). Generell lassen sich zwei wichtige Arten ausmachen, nämlich lineare Support Vector Machine und nichtlineare Support Vector Machine (James u. a., 2013). Ersteres ist der Fall, wenn zur Klassentrennung die Datendomäne linear geteilt werden kann (Suthaharan, 2016). Hierfür müssen zwei Schritte durchlaufen werden. Zuerst muss die Datendomäne in einem Antwort-Set abgebildet werden um dann die Datendomäne trennen zu können (Suthaharan, 2016). Da es sich hier um ein lineares Verfahren handelt, wird die lineare Gleichung $y = wx' + \gamma$ verwendet (Suthaharan, 2016). Die nichtlineare Support Vector Machine ist der Fall, wenn die Datendomäne in einem Merkmalsraum abgebildet werden muss um die Klassen trennen zu können (Suthaharan, 2016). Dieses Vorgehen beinhaltet drei Schritte. Zuerst wird die Datendomäne, mit Hilfe der Kernelfunktion, in einem Merkmalsraum abgebildet (Suthaharan, 2016). Dann muss die gebildete Mermalsraumdomäne in einem Antwort-Set abgebildet werden, um im Folgenden eine Trennung durchzuführen (Suthaharan, 2016). Die zugehörige Gleichung lautet $y = w\phi(x') + \gamma$ (Suthaharan, 2016). (Suthaharan, 2016).

2.4 Random Forest

Das letzte nennenswerte Verfahren des Machine Learnings ist Random Forest, was übersetzt so viel bedeutet wie "zufällig generierte Entscheidungsbäume".Der Name betont bereits die Wichtigkeit der Entscheidungsbäume in diesem Verfahren(Suthaharan, 2016). Hierbei handelt es sich um eine Kombination von mehreren Entscheidungsbäumen, welche von einem Zufallsvektor abhängen (Breiman, 2001). Allgemein lässt sich das Verfahren wie folgt definieren:

> Random Forest ist ein Klassifikator, der aus einer Sammlung von baumstrukturierten Klassifikatoren besteht $\{h(x, \theta_k, k = 1, ...\}$,

> wobei $\{\theta_k\}$ unabhängige, identisch verteilte Zufallsvektoren sind und jeder Baum eine Einheitsstimme für die häufigste Klasse am Input x ausgibt.
> (Breiman, 2001, vgl. S. 6)

Generell wird eine Lernstichprobe (Bootstrap) ausgewählt (Suthaharan, 2016). Aus dieser wird ein Entscheidungsbaum erstellt, der mit einer zufälligen Anzahl an Einflussvariablen an den Knoten als Splitting-Variablen ausgestattet ist (James u. a., 2013). Hierbei ist zu beachten, dass die Anzahl der Prädiktoren, die bei jedem Split berücksichtigt werden, ungefähr der Quadratwurzel der Gesamtzahl der Prädiktoren entspricht (James u. a., 2013). Diese Schritte werden nun mehrmals parallel und unabhängig voneinander ausgeführt, so dass aus einem Baum ein ganzer "Wald" entsteht. Die Zufälligkeit spielt bei diesem Verfahren eine wichtige Rolle, da so versucht wird Korrelationen zu minimieren, was zur Folge hat, dass die Genauigkeit verbessert wird (Breiman, 2001). Der Verallgemeinerungsfehler, welcher beim Random Forest entstehen kann, hängt von der Stärke der einzelnen Entscheidungsbäume und von der Korrelation zwischen diesen ab (Breiman, 2001). Außerdem handelt es sich um ein Verfahren, das robuster gegen "noise" ist (Breiman, 2001). Ein nennenswerter Vorteil des Random Forest ist das Verhindern von Overfitting durch das Gesetz der großen Zahlen (Breiman, 2001). Im Vergleich zu Bagging hat Random Forest den Vorteil, dass er durch eine kleine Optimierung den Baum dekorreliert und somit verbessert (James u. a., 2013). Abschließend ist festzuhalten, dass sich das Verfahren vor allem bei einer großen Anzahl an Prädiktoren anbietet (James u. a., 2013).
Nachdem ein paar der untersuchten Klassifikationsfamilien vorgestellt wurden, stellt sich nun die Frage nach den passenden Daten, um eine sinnvolle Aussage über die Güte der Klassifikatoren treffen zu können. Diese Frage soll im folgenden Kapitel beantwortet werden.

3 Datensätze

Eine der wichtigsten Informationen zur Messung der Güte sind die dafür verwendeten Datensätze. Im Artikel von Fernández-Delgado et al. werden größtenteils Datensätze von der "UCI Machine Learning Repository" verwendet (Fernández-Delgado u. a., 2014). Hierbei handelt es sich um eine sehr hilfreiche Datensatzquelle. Sie bietet insgesamt 289 Datensätze zur Auswahl wobei nach sämtlichen Kriterien, wie beispielsweise Variablenart oder Variablenauswahl, gefiltert werden kann (UCI, 2007). 2014, als der Artikel von Fernández-Delgado et al. veröffentlicht wurde, waren bei der "UCI Machine Learning Repository" lediglich 167 Datensätze zur Verfügung (Fernández-Delgado u. a., 2014). Folglich hat sich die Anzahl der Datensätze in drei Jahren fast verdoppelt, was wiederum für den "UCI Machine Learning Repository" als Datensatzquelle spricht. Von den damaligen 165 Datensätzen wurden 57 verworfen (Fernández-Delgado u. a., 2014). Dies hat verschiedene Gründe, beispielsweise sind die Datensätze zu groß, da sie eine hohe Anzahl an *inputs* oder *patterns* besitzen (Fernández-Delgado u. a., 2014). Andere

Gründe sind z.B. Klassen ohne, oder nur mit einem *pattern*, oder nur einen *input* (Fernández-Delgado u. a., 2014). Neben den 108 Datensätzen aus dem "UCI Machine Learning Repository" werden in dem Text von Fernández-Delgado et al. auch vier reale Datensätze (oocMerl4D, oocMerl2F, oocTris2F, oocTris5B) verwendet (Fernández-Delgado u. a., 2014). Diese beinhalten die Fruchtbarkeitsschätzung im Gebiet der Fischerei (Fernández-Delgado u. a., 2014). Die inhaltliche Thematik ist allerdings irrelevant für die Untersuchung der Güte der Klassifikatoren. Insgesamt kommt man somit auf $165 - 57 + 4 = 112$ Datensätze. Auf Grund von Klassenspaltungen können bei manchen Datensätzen mehrere Klassenprobleme betrachtet werden (Fernández-Delgado u. a., 2014). Daraus ergeben sich 121 Datensätze zur Analyse der Güte (Fernández-Delgado u. a., 2014). Ein Vorteil dieser Auswahl an Datensätzen ist die Vielfältigkeit. So sind von sehr kleinen, bis sehr großen Datensätzen alles vertreten. Außerdem liegen große Intervalle in den *patterns* (10 - 130.064), *inputs* (3 - 262) und Klassen (2 - 100) vor (Fernández-Delgado u. a., 2014). Bevor die Untersuchung stattfinden kann, müssen noch einige wenige Vorarbeiten geleistet werden. In diesem Fall beinhaltet dies, den Mittelwert auf null und die Standardabweichung auf eins zu setzen (Fernández-Delgado u. a., 2014). Außerdem werden fehlende Werte als ein null Wert behandelt (Fernández-Delgado u. a., 2014). Die hier beschriebenen 121 Datensätze mit den getroffenen Vorarbeiten bilden die Basis der folgenden Untersuchung der Güte der Klassifikatoren.

4 Vergleich der Güte

Der Hauptteil dieser Arbeit widmet sich der Messung und dem Vergleich der Güte der einzelnen Klassifikatoren und deren Klassifikationsfamilien. Den oben beschriebenen Kapiteln lassen sich die wichtigsten Eckdaten entnehmen. Der Text von Fernández-Delgado et al. untersucht 179 Klassifikatoren aus 17 Klassifikationsfamilien, welche mit Hilfe von 121 Datensätzen überprüft werden (Fernández-Delgado u. a., 2014). Folglich ergeben sich 21.659 Kombinationen die durchlaufen werden müssen. Allerdings sind davon 449 Kombinationen, aus verschiedenen Gründen, fehlerhaft und werden deshalb bei der Auswertung nicht berücksichtigt (Fernández-Delgado u. a., 2014). Es stellt sich nun die Frage, ob es eine Berechtigung für so viele verschiedene Klassifikatoren gibt, oder ob ein paar wenige Verfahren ausreichen würden. Dies soll mit Hilfe der Messung der Güte beantwortet werden.

4.1 Verfahren und Ergebnisse

Im Folgenden werden die vier, im Artikel von Fernández-Delgado et al. verwendeten, Verfahren zum Vergleich der Güte erläutert und die jeweiligen Ergebnisse vorgestellt.

Friedman Ranking Das Friedman Ranking ist ein nichtparametrischer Test zum Vergleich von Stichproben (Bühl, 2016). Folglich ist eine Normalverteilung

keine notwendige Voraussetzung. Außerdem müssen immer mehr wie zwei Stichproben vorliegen (Bühl, 2016). Die Nullhypothese bei diesem Verfahren lautet "Es besteht kein Unterschied in der Lage der Stichproben." (Bühl, 2016). Das Ergebnis wird mit Hilfe von Rangreihen dargestellt, welche fallweise für die Werte der beteiligten Variablen ermittelt werden. Die mittleren Rangplätze bedeuteten folglich, dass große Werte hohe Rangplätze und kleine Werte niedrige Rangplätze erhalten (Bühl, 2016). Wichtig hierbei ist, dass niedrige Ränge große Unterschiede in der Lage der Stichproben bedeuten.

Tabelle 2. Friedman Ranking

	Ranking	Klassifikator	Klassifikationsfamilie
1	32,9	parRF_t	Random Forest
2	33,1	rf_t	Random Forest
3	36,8	svm_C	Support Vector Machine
4	38,0	svmPoly_t	Support Vector Machine
5	39,4	rforest_R	Random Forest
6	39,6	elm_kernel_m	Neuronale Netze
7	40,3	svmRadialCost_t	Support Vector Machine
8	42,5	svmRadial_t	Support Vector Machine
9	42,9	C5.0_t	Boosting
10	44,1	avNNe_t	Neuronale Netze

Wendet man nun das beschriebene Friedman Ranking auf den Vergleich der Güte der Klassifikatoren an, erhält man die Tabelle 2. Diese enthält die besten zehn Klassifikatoren, sortiert nach den Rängen. Die Bedeutungen und Unterschiede der einzelnen Abkürzungen der Klassifikatoren sind dem Artikel "Do we need hundreds of classifiers to solve real world classification problems" nach Fernández-Delgado et al. zu entnehmen (Fernández-Delgado u. a., 2014). Zum Verständnis sind in der obenstehenden Tabelle 2 die dazugehörigen Klassifikationsfamilien angegeben. Die zwei besten Verfahren gehören der Klassifikationsfamilie Random Forest an, wobei ersteres eine parallele Implementierung und zweiteres das Caret benutzt (Fernández-Delgado u. a., 2014). An fünfter Stelle lässt sich ebenfalls ein Klassifikator der Random Forest Familie finden, welcher im Programm R angewandt wurde (Fernández-Delgado u. a., 2014). Unter den Top Ten des Friedman Rankings, bezüglich der Güte der Klassifikatoren, ist die Klassifikationsfamilie Support Vector Machine sehr stark vertreten. Sie belegt den dritten, vierten, siebten und achten Platz und beinhaltet unter anderem den Gauß-Kernel (Fernández-Delgado u. a., 2014). Außerdem sind noch die Neuronalen Netze und Boosting vertreten (Fernández-Delgado u. a., 2014). Um allerdings eine Aussage darüber treffen zu können, ob einige wenige Klassifikatoren sehr gut sind, oder ob auch andere durchaus ihre Berechtigung haben genügt es nicht nur das Friedman Ranking zu betrachten.

PAMA Aus diesem Grund wird zum Vergleich der PAMA ermittelt. PAMA steht für "probability of achieving the maximum accuracy" (Fernández-Delgado u. a., 2014). Aus dem Namen lässt sich bereits die Bedeutung ableiten. Er schätzt für jeden Klassifikator die Wahrscheinlichkeit des Erreichens der maximalen Genauigkeit (Fernández-Delgado u. a., 2014). Er wird berechnet, indem die Anzahl der Datensätze, für die die höchste Genauigkeit erreicht wird, durch die Gesamtzahl der Datensätze geteilt wird (Fernández-Delgado u. a., 2014).

Tabelle 3. PAMA

	Wahrscheinlichkeit	Klassifikator	Klassifikationsfamilie
1	13, 2%	elm_kernel_m	Neuronale Netze
2	10, 7%	svm_C	Support Vector Machine
3	9, 9%	parRF_t	Random Forest
4	9, 1%	C5.0_t	Boosting
5	9, 1%	adaboost_R	Boosting
6	8, 3%	rforest_R	Random Forest
7	6, 6%	nnet_t	Neuronale Netze
8	6, 6%	svmRadialCost_t	Support Vector Machine
9	5, 8%	rf_t	Random Forest
10	5, 8%	RRF_t	Random Forest

Die obenstehende Tabelle 3 beinhaltet die zehn besten Klassifikatoren des PAMA in Bezug auf deren Güte (Fernández-Delgado u. a., 2014). Die Angaben sind ähnlich wie bei dem bereits genannten Friedman Ranking, nur werden hier, statt Ränge, geschätzte prozentuale Wahrscheinlichkeiten des Erreichens der maximalen Genauigkeit angegeben. Auffällig ist hierbei, dass der maximale Prozentsatz lediglich bei 13, 2% liegt. Dieser Wert gehört einem von zwei Verfahren der Neuronalen Netze und beinhaltet das Gauß-Kernel (Fernández-Delgado u. a., 2014). Das dazugehörige zweite Verfahren liegt auf dem siebten Platz und beinhaltet Caret (Fernández-Delgado u. a., 2014). An zweiter und achter Stelle liegen Klassifikatoren der Support Vector Machine Familie (Fernández-Delgado u. a., 2014). Beide sind bereits aus dem Friedman Ranking bekannt. Auch der Random Forest ist wieder durch den dritten, sechsten, neunten und zehnten Platz stark vertreten und beinhalten auch hier Caret und parallele Implementierung (Fernández-Delgado u. a., 2014). Ein Unterschied zum Friedman Ranking ist, dass in den Top Ten des PAMA zwei Klassifikatoren des Boostings enthalten sind (Fernández-Delgado u. a., 2014).

P95 Es fällt bereits eine gewisse Ähnlichkeit beim Vergleich von Friedman Ranking und PAMA auf. Um dies bestätigen, oder widerlegen zu können wird das dritte Verfahren P95 hinzugezogen. Dieses lautet "probability of achieving more than 95% of the maximum accuracy" und beinhaltet folglich die Wahrscheinlichkeit mehr wie 95% der maximalen Genauigkeit zu erreichen (Fernández-Delgado u. a., 2014). Zur Berechnung wird die Anzahl der Datensätze, in denen 95%

oder mehr der maximalen Genauigkeit erreicht wird, durch die Gesamtzahl der Datensätze geteilt (Fernández-Delgado u. a., 2014).

Tabelle 4. P95

	Wahrscheinlichkeit	Klassifikator	Klassifikationsfamilie
1	71, 1%	parRF_t	Random Forest
2	70, 2%	svm_C	Support Vector Machine
3	68, 6%	rf_t	Random Forest
4	65, 3%	rforest_R	Random Forest
5	63, 6%	Bagging-LibSVM_w	Bagging
6	63, 6%	svmRadialCost_t	Support Vector Machine
7	62, 8%	svmRadial_t	Support Vector Machine
8	62, 8%	svmPoly_t	Support Vector Machine
9	62, 0%	LibSVM_w	Support Vector Machine
10	61, 2%	C5.0_t	Boosting

Die Tabelle 4 zeigt die jeweiligen prozentualen Wahrscheinlichkeiten des P95 bezüglich der Güte der einzelnen Klassifikatoren (Fernández-Delgado u. a., 2014). Den höchsten Wert, mit 71, 1%, hat das bereits bekannte Verfahren parRF_t, welches einen von drei enthaltenen Klassifikatoren der Random Forest Familie darstellt (Fernández-Delgado u. a., 2014). Auch die beiden anderen - auf Platz drei und vier - sind bereits erläutert. Den zweiten, sechsten, siebten, achten und neunten Platz belegen Klassifikatoren der Support Vector Machine (Fernández-Delgado u. a., 2014). Somit gehören knapp die Hälfte der Top Ten dieser Klassifikationsfamilie an. Nur zwei davon sind noch unbekannt, wobei diese ebenfalls das Kernel enthalten (Fernández-Delgado u. a., 2014). Bei dem P95 Verfahren ist unter den zehn besten Klassifikatoren ein Mitglied der Klassifikationsfamilie Bagging auf dem fünften Platz (Fernández-Delgado u. a., 2014).

PMA Es fällt bereits bei den letzten drei Verfahren zum Vergleich der Güte eine gewisse Wiederholung sowohl in den Klassifikatoren, als auch in den Klassifikationsfamilien auf. Abschließend soll das vierte Verfahren der PMA behandelt werden. PMA steht für "percentage of the maximum accuracy" und bedeutet Prozentsatz der maximalen Genauigkeit (Fernández-Delgado u. a., 2014). Folglich beinhaltet er den Prozentsatz der maximalen Genauigkeit, der von jeder Klasse erreicht wird, gemittelt über die gesamte Datensammlung (Fernández-Delgado u. a., 2014).

Tabelle 5. PMA

	Wahrscheinlichkeit	Klassifikator	Klassifikationsfamilie
1	94,1%	parRF_t	Random Forest
2	93,6%	rf_t	Random Forest
3	93,3%	rforest_R	Random Forest
4	92,5%	C5.0_t	Boosting
5	92,5%	RotationForest_w	Random Forest
6	92,3%	svm_C	Support Vector Machine
7	92,1%	mlp_t	Neuronale Netze
8	91,7%	LibSVM_w	Support Vector Machine
9	91,4%	RRF_t	Random Forest
10	91,4%	dkp_C	Neuronale Netze

Die obenstehende Tabelle 5 beinhaltet die zehn besten Klassifikatoren laut dem PMA (Fernández-Delgado u. a., 2014). Der Prozentsatz dieser erstreckt sich von 91,4% bis 94,1%. Auffallend ist hier, dass die Hälfte der aufgelisteten Klassifikatoren der Random Forest Familie entstammen, unteranderem auch die besten drei (Fernández-Delgado u. a., 2014). Der einzige unbekannte Klassifikator aus den Fünfen ist ein Rotation Forest, was bedeutet, dass sich die Parameterachsen drehen (Fernández-Delgado u. a., 2014). Wie beim PAMA ist auch hier der Boosting Klassifikator C5.0_t an vierter Stelle. Außerdem sind jeweils zwei Klassifikatoren aus der Support Vector Machine Familie und den Neuronalen Netzen vertreten, wobei letztgenannte noch unbekannt sind. Der mlp_t beinhaltet ein mehrschichtiges Perzeptron und der dkp_C Klassifikator meint ein direktes Kernelperzeptron (Fernández-Delgado u. a., 2014).

Auffällig ist, dass bei vier unterschiedlichen Berechnungen zum Vergleich der Güte der Klassifikatoren größtenteils die gleichen Klassifikatoren unter den Top Ten erscheinen. In der Klassifikationsfamilie Random Forest sind das die Klassifikatoren parRF_t, rf_t, rforest_R und RRF_t. Bei Support Vector Machine werden svm_C, LibSVM_w, svmRadialCost_t, svmRadial_t und svmPoly_t mindestens zweimal genannt. Sowohl bei den Neuronalen Netzen, als auch beim Boosting taucht lediglich ein Klassifikator mehrfach auf, nämlich der elm_kernel_m und der C5.0_t. Die einzige weitere Klassifikationsfamilie, die in den Top Ten genannt wird ist das Bagging. Zusammenfassend ist deutlich zu erkennen, dass nur wenige Klassifikatoren, aus wenigen Klassifikationsfamilien zu den Besten zählen. Dies erklärt nicht die vorliegende Vielfalt an Klassifikatoren.

4.2 Güte der Klassifikationsfamilien

Nun sollen explizit die Klassifikationsfamilien betrachtet werden. Dabei wird jeweils auch auf die Klassifikatoren eingegangen. Generell wird zur Untersuchung lediglich das Friedman Ranking herangezogen. Zur Visualisierung bietet sich ein BoxPlot an.

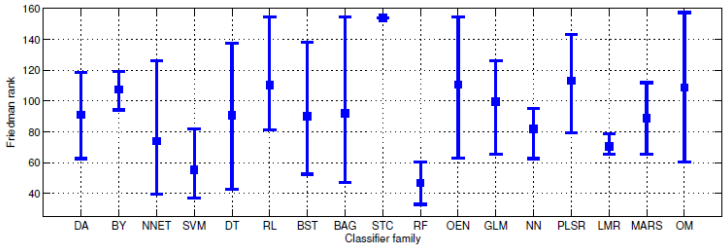

Abb. 4. Friedman Ranking der Klassifikationsfamilien visualisiert durch einen BoxPlot

Der BoxPlot aus Abb. 4 beinhaltet die Ergebnisse des Friedman Ranking Intervalls für alle Klassifikatoren in Bezug auf die einzelnen Klassifikationsfamilien (Fernández-Delgado u. a., 2014). Das Minimum zeigt dabei den niedrigsten und das Maximum den höchsten Rang. Ist das Intervall sehr klein und liegt eher niedrig bedeutet dies, dass die Klassifikationsfamilie sehr gut ist. Dementsprechend ist deutlich zu erkennen, dass der Random Forest am besten ist. Er besitzt ein Minimum von 32,9 und ein Maximum von 60,5 mit dem Mittelwert 46,7 (Fernández-Delgado u. a., 2014). An zweiter Stelle steht die Klassifikationsfamilie Support Vector Machine deren Minimum bei 36,8, das Maximum bei 81,6 und der Mittelwert bei 55,4 liegt (Fernández-Delgado u. a., 2014). Die letzte hier betrachtete Klassifikationsfamilie steht an dritter Stelle und heißt Neuronale Netze. Diese besitzt ein Minimum von 39,6, ein Maximum von 126,2 und einen Mittelwert von 73,8 (Fernández-Delgado u. a., 2014). Alle drei Klassifikationsfamilien sind bereits aus dem Kapitel "Verfahren und Ergebnisse" bekannt. Auch hier liegen Klassifikatoren aus den genannten Familien unter den Top Ten der einzelnen Messungen. Die oben erläuterte statistische Klassifikationsfamilie, die Diskriminanzanalyse, ist beim Friedman Ranking im Mittelfeld anzuordnen. Der kleinste Rang liegt hier bei 62,5 und der größte bei 118,3 (Fernández-Delgado u. a., 2014). Zusammenfassend lässt sich feststellen, dass, wie erwartet, die Klassifikationsfamilien vorne liegen, die auch schon bei den oben beschriebenen Verfahren des Friedman Rankings, des PAMA, des P95 und dem PMA sehr gut abgeschnitten haben. In diesem Kapitel konnten die am häufigsten genannten Klassifikationsfamilien in eine Reihenfolge gebracht werden.

5 Fazit

Die Klassifikatoren, dessen Güte untersucht wurden, können sehr deutlich in eine Reihenfolge gebracht werden. In dem Kapitel "Vergleich der Güte" ist deutlich geworden, dass eine Unterteilung in gute Klassifikatoren und nicht so aussagekräftige Klassifikatoren getroffen werden kann. Dies bedeutet, dass man tatsächlich einen Klassifikator, bzw. eine Klassifikationsfamilie als empfehlenswert beurteilen kann und nicht unbedingt jeder Klassifikator gleich gut, oder gleich angemessen ist. Somit lässt sich die Forschungsfrage, ob die Güte der

Klassifikatoren die Existenz einer so großen Anzahl an möglichen Klassifikatoren bestätigt, mit Nein beantworten. Ganz im Gegenteil es lässt sich kein Beweis für die Notwendigkeit der schwachen Klassifikatoren und Klassifikationsfamilien finden. Warum es trotzdem so eine große Auswahl gibt und wo welche Klassifikatoren ihre Begründung finden ist leider dem Artikel von Fernández-Delgado et al. nicht zu entnehmen (Fernández-Delgado u. a., 2014).
Es kann nur eine Vermutung im Hinblick auf die Vorteile des Random Forest aufgestellt werden, warum diese Klassifikationsfamilie sehr gut abgeschnitten hat. Ein Vorteil ist, dass es ein weites Spektrum an Variablen-, bzw. Modellierungsarten gibt, die durch den Random Forest abgedeckt werden können (Deeb, 2015). Außerdem handelt es sich, wegen der großen Anzahl an unabhängigen Entscheidungsbäumen, um ein sehr unempfindliches Verfahren (Deeb, 2015). Des Weiteren entsteht auf Grund des Gesetzes der großen Zahlen kein Overfitting (Breiman, 2001). Außerdem ist Random Forest robust gegen Noise"(Breiman, 2001). Zu guter Letzt stellen die parallelisierbaren Entscheidungsbäume einen großen Vorteil dar (Deeb, 2015). Durch diese Tatsache können große Datenmengen gleichzeitig verarbeitet werden, was eine gewisse Schnelligkeit mit sich bringt (Deeb, 2015). Dies sind einige Vorteile, die zu dem guten Ergebnis des Random Forest geführt haben könnten. Allerdings handelt es sich hierbei nur um Vermutungen.
Abschließend sollte noch bedacht werden, dass es oftmals nicht den einen passenden Klassifikator gibt, sondern durchaus Kombinationen in Betracht gezogen werden müssen.

Literaturverzeichnis

Backhaus, Klaus ; Erichson, Bernd ; Plinke, Wulff ; Weiber, Rolf: *Multivariate Analysemethoden. Eine anwendungsorientierte Einführung.* Berlin Heidelberg : Springer-Verlag, 2016

Backhaus, Klaus ; Erichson, Bernd ; Weiber, Rolf: *Fortgeschrittene Multivariate Analysemethoden. Eine anwendungsorientierte Einführung.* Berlin Heidelberg : Springer-Verlag, 2016

Breiman, Leo: Random Forest. In: *Machine Learning* 45 (2001), S. S. 5 – 32

Bühl, Achim: *SPSS23. Einführung in die moderne Datenanalyse.* Hallbergmoos : Pearson Deutschland GmbH, 2016

Deeb, Ahmed E.: *Rants on Machine Learning.* `https://medium.com/rants-on-machine-learning/the-unreasonable-effectiveness-of-random-forests-f33c3ce28883`, 2015. – Eingesehen am 23.11.2017

Fernández-Delgado, Manuel ; Cernadas, Eva ; Barro, Senén ; Amorim, Dinani: Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? In: *Journal of Machine Learning Research* 15 (2014), S. S. 3133 – 3181

James, Gareth ; Witten, Daniela ; Hastie, Trevor ; Tibshirani, Robert: *An Introduction to Statistical Learning. with Applications in R.* New York : Springer Science+Business Media, 2013

Mitchell, Tom M.: *Machine Learning.* New York : McGraw-Hill, 1997

Steinwart, Ingo ; Christmann, Andreas: *Support Vector Machine.* New York : Springer Science+Business Media, LLC, 2008

Suthaharan, Shan: *Machine Learning Models and Algorithms for Big Data Classification. Thinking with Examples for Effective Learning.* New York : Springer Science+Business Media, 2016

UCI: *UCI Machine Learning Repository. Center of Machine Learning and Intelligent Systems.* `http://archive.ics.uci.edu/ml/datasets.html?task=cla`, 2007. – Eingesehen am 04.11.2017