Aus der Klinik für Radiologie

der Medizinischen Fakultät Charité – Universitätsmedizin Berlin

DISSERTATION

Prädiktive Radiomics-Modelle zur Dignitätsklassifizierung mediastinaler Lymphknoten im CT bei Adeno- und Plattenepithelkarzinomen der Lunge.

Predictive radiomics models for dignity classification of mediastinal lymph nodes in CT in adeno and squamous cell carcinomas of the lungs.

zur Erlangung des akademischen Grades

Doctor medicinae (Dr. med.)

vorgelegt der Medizinischen Fakultät

Charité – Universitätsmedizin Berlin

von

Hendrik Philipp Becker, geb. Makoschey

aus Bonn

Datum der Promotion: 18.12.2020

Inhaltsverzeichnis

Abstract

Bei der mediastinalen Ausbreitungsdiagnostik von Plattenepithel- und Adenokarzinomen der Lunge ist das PET-CT zur Identifizierung von suspekten Lymphknoten vor deren histologischer Sicherung Goldstandard.

Radiomics, also die automatisierte Bewertung relevanter Bildstrukturen mittels Klassifikationsalgorithmen, könnte eine diagnostische Alternative zum komplexen und nicht flächendeckend verfügbaren PET-CT sein.

Ziel der vorliegenden Forschungsarbeit war es, Klassifikationsalgorithmen mit unterschiedlichen *machine learning*-Techniken zu entwickeln, die anhand von Bildmerkmalen (*radiomic features*) aus CT-Scans die Dignität mediastinaler Lymphknoten klassifizieren können, und dabei eine ähnlich genaue Vorhersageleistung wie die Klassifikation durch PET-CT erreichen.

Dafür wurde die Dignität von insgesamt 1799 Lymphknoten aus Kontrastmittel-PET-CT-Scans von 381 Patienten mit der Diagnose eines Adeno- oder Plattenepithelkatzinoms der Lunge, mittels PET-CT klassifiziert. In einem zweiten Schritt wurden die klassifizierten Lymphknoten aus dem CT-Scan extrahiert und für die Entwicklung von 24 CT-basierten Vorhersagealgorithmen genutzt. Dabei wurden 4 *radiomic feature*-Selektionsmechanismen mit 6 unterschiedlichen *machine learning*-Algorithmen kombiniert, auf einem Trainingsdatensatz entwickelt, und an einem Testdatensatz validiert.

Zusätzlich wurden die 24 Klassifikationsalgorithmen auf das Vorhandensein von *radiomic features* mit einem hohen prädiktiven Wert überprüft, mit den Vorhersageleistungen von zwei Radiologen der Charité auf dem identischen Testdatensatz verglichen und in die Dignitäts-Vorhersage der Radiologen integriert, wenn diese angaben, den Lymphknoten nicht sicher kategorisieren zu können.

Es gelang, 24 stabile Klassifikationsalgorithmen zu entwickeln, die auf dem Testdatensatz eine durchschnittliche Treffergenauigkeit von 77%, bei einer Sensitivität von 71% und einer Spezifität von 80%, erreichen konnten.

Dabei gelang außerdem die Identifizierung einiger *radiomic features* mit einem hohen prognostischen Wert.

Im Vergleich mit den Klassifikationsleistungen der Radiologen erreichten die Klassifikationsalgorithmen höhere Sensitivitäten, bei geringeren Spezifitäten und eine vergleichbare Treffergenauigkeit. Die Integration der Algorithmen in die Vorhersage der Radiologen in unsicheren Kategorien resultierten in signifikant höheren Sensitivitäten und signifikant niedrigeren Spezifitäten bei unveränderten Treffergenauigkeiten.

Die Erstellung von 24 stabilen *machine learning*-Klassifikationsalgorithmen war erfolgreich und ist eine Möglichkeit, bisher ungenutzte Informationen aus CT-Scans automatisiert zu bewerten. Es gelang jedoch nicht, die Vorhersageleistung der PET-CT zu erreichen.

Die Forschungsergebnisse zeigen aber, dass *Radiomics*-Klassifikationsalgorithmen, je nach diagnostischen Anforderungen, mit signifikant höheren Sensitivitäten in unsicheren Klassifikationsfällen, schon heute eine zusätzliche Unterstützung für Radiologen sein könnten.

English abstract

In mediastinal lymph node staging of adeno and squamous-cell carcinoma of the lung, PET-CT analysis for the identification of suspicious lymph nodes prior to their histological confirmation is currently gold-standard.

Radiomics, the automated evaluation of image structures through classification algorithms could be a diagnostic alternative to the complex and not nationwide available PET-CT analysis.

The aim of the study was to develop classification algorithms with different *machine learning* techniques that can classify the dignity of mediastinal lymph nodes using only imaging features (*radiomic features*) from CT-scans and reaching an equally precise prediction performance like PET-CT.

Therefore, we classified the dignity of 1799 lymph nodes from contrast enhanced PET-CT scans of 381 patients that were diagnosed with an adeno or squamous-cell carcinoma of the lung, using PET-CT. Afterwards we extracted the classified lymph nodes from the CT-scan and used them as a basis to develop 24 CT-based classification algorithms. Here we combined 4 *radiomic feature* selection methods and 6 *machine learning* algorithms, that were developed on a training data set and validated on a test data set.

Additionally, the 24 classification algorithms were screened for single *radiomic features* that hold a high predictive value, the performances of the 24 algorithms were compared with the performances of 2 radiologists and integrated in the radiologist's prediction in cases where they were uncertain about the dignity of a lymph node.

We successfully developed 24 stable classification algorithms that achieved an average accuracy of 77%, a sensitivity of 71% and a specificity of 80% on the test data set.

It was also possible to identify *radiomic features* with a high predictive value.

In comparison with the radiologist's performance, the classification algorithms achieved higher sensitivities with lower specificities and a comparable accuracy. The integration of the algorithms in the radiologist's prediction in uncertain categories resulted in significantly higher sensitivities and significantly lower specificities with unchanged accuracies.

The development of 24 stable *machine learning* classification algorithms was successful and is a way to automatically assess previously unused information from CT scans. However, the predictive performance of PET-CT was not reached.

Yet, the research results demonstrate that *radiomics* classification algorithms, depending on the diagnostic demands, could already be a supplementary tool for radiologists today by reaching significantly higher sensitivities in cases of uncertain classification.

1 Einleitung

1.1 Formen und Relevanz des Lungenkarzinoms in Deutschland

Bösartige Neubildungen der Lunge und des Bronchialsystems zählen in Deutschland und weltweit zu den häufigsten Todesursachen bei beiden Geschlechtern (1).

1.1.1 Formen

Allgemein wird, ausgehend von der Zellteilungsrate, zwischen zwei Tumortypen unterschieden: kleinzelligen und nicht-kleinzelligen Karzinomen. Dabei stellt sich das kleinzellige Lungenkarzinom (englisch: Small cell lung cancer, SCLC) mit einem Gesamtanteil von ca. 15% (1) aller Lungenkarzinome histologisch als neuroendokriner Tumor dar (2).

Die nicht-kleinzelligen Karzinome (englisch: Non small cell lung cancer, NSCLC) lassen sich histologisch weiter in die Subgruppe der Plattenepithelkarzinome und der Adenokarzinome, sowie als seltenere Variante in die großzelligen Lungenkarzinome unterteilen, welche zusammen rund 85% (1) der Lungenkarzinome in Deutschland ausmachen (2).

Plattenepithelkarzinome finden ihren Ursprung in den epithelialen, wandauskleidenden Zellen des Respirationstrakts, Adenokarzinome hingegen in den Drüsenzellen des Bronchialsystems (2).

1.1.2 Epidemiologie

(Alle Epidemiologische Daten sind aus dem *Bericht zum Krebsgeschehen in Deutschland 2016* des Robert Koch Instituts (1) entnommen.)

1.1.2.1 Inzidenz

Die absolute Zahl der Neuerkrankungen des Lungenkarzinoms in Deutschland im Jahr 2013 betrug bei Männern 34.690 (58,6 pro 100.000 Einwohner) wobei die Prognose für 2018 mit -1,2% abfallend ist. Bei Frauen lag die Zahl der Neuerkrankungen im selben Zeitraum bei 18.810 (28,6 pro 100.000 Einwohner). Hier ist die Prognose für 2018 jedoch mit +3,1% steigend, was laut *Bericht zum Krebsgeschehen in Deutschland 2016* des Robert Koch Instituts, durch das unterschiedliche Rauchverhalten der beiden Geschlechter bis zur Jahrtausendwende bedingt ist (1).

Bei beiden Geschlechtern hat der Anteil der Adenokarzinome gegenüber den Plattenepithelkarzinomen zugenommen und stellt nun sowohl bei Männern (seit 2008) als auch bei Frauen (seit 2000) den häufigsten Subtyp dar (1), somit sind die Plattenepithelkarzinome nur noch der zweithäufigste Subtyp.

1.1.2.2 Mortalität

Die Mortalität des Lungenkarzinoms in Deutschland betrug im Jahr 2013 bei Männern 29.708 (48,8 pro 100.000 Einwohnern) und war damit die häufigste Krebstodesursache (1). Bei Frauen beträgt die Mortalität im selben Jahr 15.140 (21,7 pro 100.000 Einwohnern) und war somit die zweithäufigste Krebstodesursache (1). Außerdem lag das relative 5-Jahres-Überleben der erkrankten Männer bei 16%, das der erkrankten Frauen bei 21% (1).

Es lässt sich also festhalten, dass das Lungenkarzinom in Deutschland bei beiden Geschlechtern eine große epidemiologische Relevanz hat und insgesamt mit einer schlechten Prognose verbunden ist.

1.1.3 Ätiologie

Bei der Risikoanalyse des Lungenkarzinoms unterscheidet man zwischen exogenen und endogenen Risikofaktoren.

1.1.3.1 Exogene Risikofaktoren

Das Robert Koch Institut stellt in seinem *Bericht zum Krebsgeschehen in Deutschland 2016* die herausragende Bedeutung des exogenen Risikofaktors Tabakrauchen fest (1). Aktivrauchen ist der mit Abstand häufigste Risikofaktor und für 85% (3) der Lungenkarzinome in Europa verantwortlich, wobei das Risiko steigt, je länger und je häufiger geraucht wird und sinkt, je früher mit dem Rauchen aufgehört wird, vgl. (2-4).

Dabei sind, nach de Groot et al., bis heute mindestens 60 Karzinogene aus dem Tabakrauch identifiziert worden, die wichtigste Gruppe stellten die polyzyklischen aromatischen Kohlenwasserstoffe dar (5).

Auch das Passivrauchen ist, nach Secretan B. et al., kanzerogen (6). Bei Frauen, die durch einen Zigaretten-rauchenden Partner exponiert waren, erhöht sich das Risiko laut einer Metaanalyse der *IARC Working Group on the Evaluation of Carcinogenic Risk to Humans* proportional zur Dauer der Exposition im Vergleich zur nicht-exponierten Normalbevölkerung (7). So weisen Ehefrauen für je 10 gerauchten Zigaretten des

Partners pro Tag, ein um 10-23% erhöhtes relatives Risiko auf, an Lungenkrebs zu erkranken, verglichen mit Ehefrauen, dessen Partner Nichtraucher waren (7).

Als weitere exogene Risikofaktoren werden Radon, ionisierende Strahlen, Feinstaub, Asbest, polyzyklische aromatische Kohlenwasserstoffe und Arsen-, Chrom-, und Nickelverbindungen betrachtet, vgl. (3, 5).

1.1.3.2 Endogene Risikofaktoren

Auch Nichtraucher erkranken an Lungenkrebs, wobei die genauen Anteile je nach Studien stark variieren. Als Nichtraucher werden, nach Pallis et al., Personen bezeichnet, die in ihrem Leben insgesamt weniger als 100 Zigaretten geraucht haben, was auf 25% aller Lungenkrebspatienten zutrifft (4).

Patienten die mit einem Lungenkrebspatienten erstgradig verwandt sind, haben, laut Pallis et al., selbst ein 2.5-fach erhöhtes Erkrankungsrisiko (4). Neue Studien deuten auf mögliche genetische Prädispositionen hin, sind aber noch nicht in ausreichendem Umfang vorhanden. Bisher konnten jedoch einige Single-Nucleotid Polymorphismen (SNPs) identifiziert werden, welche mit Lungenkrebs assoziiert sind: 2010 identifizierten Li et al. den SNP (rs2352028) auf Chromosom 13q31.3, der für bis zu 10% aller Lungenkarzinome in Nichtrauchern verantwortlich sein könnte (4).

Zusätzlich mögliche, endogene Risikofaktoren sind Mutationen von Akute-Phase Proteinen, DNA-Reparaturgenen und Zellstoffwechsel-Enzymen.

Außerdem rücken im Zuge der molekularpathologischen Krebsforschung erworbene, somatische Mutationen weiter in den Vordergrund, vgl. (8). Beispielsweise könnten Veränderungen von HER-2 Rezeptoren (aus der Gruppe der Epidermal-Growth-Factor Rezeptoren) und ROS1 Rezeptoren (aus der Gruppe der Insulin Rezeptoren) in einer erhöhten Proliferationsrate der mutierten Zellen resultieren, sodass sich mit Tyrosin-Kinase Inhibitoren wie Erlotinib als sogenannte „targeted agents“ (deutsch: gezielte (Arznei)Mittel) (4), neue therapeutische Optionen eröffnen, vgl. (2, 4, 8).

1.2 Die Diagnostik und Behandlung des Lungenkarzinoms

1.2.1 Diagnostik

1.2.1.1 Anamnese, Körperliche Untersuchung und Klinische Laborparameter

Der diagnostische Prozess beginnt in der Regel mit nicht-invasiven Maßnahmen, wie der Patientenanamnese, der klinischen Untersuchung sowie der Bestimmung klinischer Laborparameter.

1.2.1.2 Bildgebung

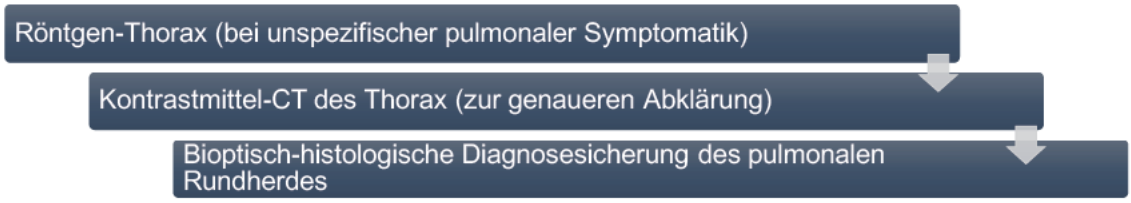


Abbildung 1: Regelhafter Verlauf der Bildgebung bei pulmonal verdächtigen Strukturen bis hin zur finalen Sicherung der Diagnose durch Biopsie und die nachfolgende histologische Untersuchung.

1.2.1.2.1 Röntgen-Thorax

Ob als Zufallsdiagnostik bei ursprünglich anderer klinischer Fragestellung oder als initiale Diagnostik zur Abklärung bei Verdacht auf Lungenkarzinom: Häufig steht das Röntgen-Thorax am Anfang der Bildgebung und wird als primäre Bildgebung in der aktuellen S-3 Leitlinie von 2018 empfohlen (3).

1.2.1.2.2 Kontrastmittel-CT

Die weiterführende Bildgebung der ersten Wahl ist das Kontrastmittel-CT von Thorax und Oberbauch. Durch charakteristische, maligne Tumor-Morphologien wie „nekrotische Areale, umschriebene Pleuraverdickung[en], pleurale Retraktion[en], Inhomogenität, Bronchuszeichen“, sowie „Spikulae“ und „Gefäßzeichen“ (3), kann laut S3-Leitline der Verdacht auf einen malignen Prozess durch eine hohe Sensitivität erhärtet werden, ohne jedoch die Diagnose mangels ausreichender Spezifität zu sichern (3).

1.2.1.2.3 MRT

Da die Magnetresonanztomografie eine höhere Weichteilauflösung bei fehlender Strahlenbelastung bietet, stellt sie eine diagnostische Alternative zur Computertomographie dar. Beispielsweise liefert, laut S3-Leitlinie, „eine dynamische MRT während fortgesetzter Atmung deutlich bessere Erkenntnisse zum Vorliegen einer Infiltration von Thoraxwand oder Mediastinum als die CT in Atemstillstand.“ (3).

1.2.1.3 Bipotisch-histologische Diagnosesicherung

Bei erhärtetem Verdacht auf einen malignen Prozess erfolgt die Diagnosesicherung, nach S3-Leitlinie, bioptisch-histologisch, wobei die Bronchioskopie mit der transbronchialen Biopsie die erste Wahl ist (3). Mit Hilfe der bioptischen Zellgewinnung kann die histologische Unterscheidung zwischen kleinzelligem und nicht-kleinzelligem Tumor inklusive weiterer Unterteilung in Adeno- oder Plattenepithelkarzinom gelingen, vgl. (2, 3).

In den letzten Jahren hat sich ein weiterer Aspekt hervorgetan: Die molekulare Diagnostik neuer Therapie-Targets, wie zum Beispiel die Epidermal-Growth-Factor-Rezeptor Mutationen, welche Pathologen zusätzlich als Erweiterung der bisherigen Diagnostik nutzen können und aus denen sich neue therapeutische Möglichkeiten ergeben, vgl. (5, 8, 9).

1.2.1.4 Staging

Die Ausbreitungsdiagnostik des Lungenkarzinoms lässt sich in drei Hauptaspekte unterteilen: den T-Status, also die lokoregionäre Tumorausdehnung, den N-Status, also das Vorhandensein von Lymphknotenmetastasen, sowie den M-Status, also das Vorhandensein von Fernmetastasen, vgl. (10, 11).

Ein exaktes Staging ist von großer Bedeutung, da es eine Einteilung in ein entsprechendes UICC8-Stadium erlaubt, vgl. (2, 11). Das Tumorstadium ist essenziell für die Wahl der weiteren Behandlung und dient maßgeblich als Entscheidungsgrundlage zwischen einem operativen oder palliativen Behandlungskonzept, vgl. (10, 11). Außerdem gibt es Auskunft über die statistische Überlebenszeit in Form der 5-Jahre-Überlebensrate (1).

1.2.1.4.1 T-Staging

Der T-Status kann radiologisch mit Hilfe des CTs oder gegebenenfalls MRTs bestimmt werden (3). Dabei sind der Durchmesser der Tumormasse und die lokale Infiltration in umliegende Gewebe entscheidend (11).

1.2.1.4.2 N-Staging

*Abbildung 2: Diese Grafik zeigt den **bisher typischen Verlauf des N-Stagings** eines Lungenkarzinoms. Nach der ersten CT-Untersuchung wird eine **PET** oder eine integrierte PET-CT Untersuchung durchgeführt. Nach Detektion der lokalen Tracer-Aufnahmen und der Korrelierung zu anatomischen Strukturen aus dem CT kann eine gezielte Biopsie erfolgen. Der zusätzliche diagnostische Schritt der PET soll die Wahrscheinlichkeit einer erfolgreichen Metstasen-Biopsie erhöhen um so zu einem möglichst genauen Staging-Befund zu gelangen.*

*Abbildung 3: **Alternativ** zu dem in Abbildung 2 gezeigten Verlauf könnte die komplexe PET durch eine automatisierte Analyse von bisher ungenutzten Bildparametern (features) ersetzt werden. So würde dem Patienten, falls die PET überhaupt verfügbar ist, ein komplexer und fehleranfälliger Diagnostikschritt erspart bleiben. Ist die **Radiomics-Analyse** vergleichbar genau, könnte das Ziel der effizienten Lymphknotenbiopsie ressourcen- und patientenschonend erreicht werden. Aktuelle Anwendungsbeispiele aus der Forschung finden sich im Kapitel 1.4.1.1.*

Der N-Status kann auf vielfältige Weise bestimmt werden, wobei die alleinige, in der Computertomographie ermittelte, Lymphknotengröße wenig aussagekräftig ist und erst in Kombination mit dem FDG-PET (siehe Kapitel *Einleitung*, Absatz *Exkurs*) als funktionelle Diagnostik aussagekräftiger wird (12).

Mit Hilfe des PET-CTs lässt sich abschätzen, welche Lymphknoten biopsiert und histologisch untersucht werden sollten, wobei bei der Befundung die Faktoren Lymphknotengröße und die Tracer-Aufnahme des Lymphknotens im Vordergrund stehen.

Die Lymphknotenbiopsie wird unter anderem mittels Mediastinoskopie, Thorakoskopie, Brochoskopie und Ultraschall-gestützten Verfahren durchgeführt (3).

1.2.1.4.3 M-Staging

Mit Hilfe der einzeitigen, integrierten Ganzkörperdiagnostik durch das FDG-PET-CT (siehe Kapitel *Einleitung*, Absatz *Exkurs*) können Fernmetastasen, laut S-3 Leitlinie mit „eine[r] mittlere[n] Sensitivität von 93 % und Spezifität von 96%“ (3) nachgewiesen

werden. Für die Detektion von Hirnmetastasen ist die Magnetresonanztomographie wegen des besseren Weichteilkontrasts sowie einer geringeren Artefaktbildung, laut S3-Leitlinie, die erste Wahl (3).

1.2.2 FDG-PET-CT in der Diagnostik des Lungenkarzinoms

Die Positronen-Emissions-Tomographie (PET), mit dem Radioisotop und Glukoseanalogon F-18-Fluoro-Desoxy-Glukose (F-18-FDG) als Marker, ist ein Verfahren zur Detektion solider Tumoren, Lymphknotenmetastasen sowie von fernmetastasiertem Tumorgewebe. In Kombination mit der Computertomographie - als integriertes PET-CT - stellt es ein nicht-invasives, einzeitiges Schnittbildverfahren dar, welches sowohl zur Diagnostik als auch zum Staging des NSCLC geeignet ist, vgl. (3, 13-15). Eine zusätzliche Verwendung findet das PET-CT in der Rezidivdiagnostik nach behandeltem Lungenkarzinom mit der Frage nach neu aufgetretenen Metastasen.

1.2.2.1 Physikalische Grundlagen der PET

Die physikalische Grundlage der PET ist die sogenannte Vernichtungsstrahlung. Vernichtungsstrahlung ist Gamma-Strahlung und entsteht durch das Aufeinandertreffen eines Positrons (emittiert durch beta+ Zerfall des injizierten Radionukleoids) mit einem Elektron (Valenzelektron eines Atoms im Körper des Patienten), es resultiert die Entstehung zweier Photonen. Dieser Vorgang wird als Annihilation bezeichnet. Da dieser Vorgang erst durch die Interaktion von Positron und Elektron abläuft, wird nicht der physikalische Vorgang der Positronenemission, sondern die aus der Annihilation entstandene Gamma-Strahlung detektiert.

Die Strecke, die ein Positron zurücklegt, bis es auf ein Elektron trifft wird als „range“ (16) (deutsch: Reichweite) bezeichnet und kann in vivo zwischen „0.2 and 2.6 mm“ (16) betragen. Folglich schränkt dies die örtliche Auflösung der PET ein (16).

Entstanden sind zwei Photonen mit einer Energie von jeweils 511keV (511keV entsprechen der Energie jeweils eines Elektrons oder Positrons) die sich in einem Winkel von meist 180 Grad vom Ort ihrer Entstehung wegbewegen. Treffen diese Photonen fast gleichzeitig im sogennanten „[…] timing window of the coincidence circuit” (typically 3–15 ns)“ (16) (deutsch: Zeitfenster der Koinzidenzschaltung) auf jeweils gegenüberliegende Seiten eines Ringdetektors, bezeichnet man dies als Koinzidenz.

Die entstandene Gerade, auf der sich die zwei Photonen auf den Detektor zubewegen, nennt man „line of response (LOR) or the coincidence line“ (16) (deutsch: Line of response oder Koinzidenzlinie) und dient zur Berechnung des Positronen-Emissions-Orts.

Anhand der Verteilung dieser Koinzidenzen und deren Laufzeitunterschiede bei Auftreffen auf den Detektor kann auf die räumliche Verteilung des Positronen-emittierenden Radionukleotids im Körper des Patienten geschlossen werden.

1.2.2.2 Physikalische Grundlagen der Computertomographie

Das physikalische Grundprinzip der Computertomographie besteht in der überlagerungsfreien Darstellung menschlichen Gewebes durch Röntgenstrahlung. Dies wird erreicht, indem ein Ringdetektor mit einer gegenüberliegenden Röntgenquelle (auch „tube“ (17) genannt) um 180° beziehungsweise 360° transversal um den Körper des Patienten rotiert, um aus verschiedenen Winkeln transversale Absorptionsprofile der Körpermasse zu erstellen.

Durch Vorschub des Patienten durch den Ringdetektor oder durch eine helikale Drehung des Ringdetektors, samt Röntgenquelle, können multiple Profile aller Körperregionen erstellt werden. Nachfolgend integriert ein Computer diese Absorptionsprofile zu einem überlagerungsfreien, dreidimensionalen Bild. Darin sind nun auch die Sagittal- und Axialebene enthalten, die entstandenen dreidimensionalen Bildpunkte bezeichnet man als Voxel.

Für jedes Voxel errechnet der Computer den dazugehörigen Schwächungsgrad der Röntgenstrahlung und visualisiert diesen Wert durch einen Grauton (17). Folglich ergeben sich eine Vielzahl von Graustufen, die mittels Houndsfield-Einheiten (HU) beschrieben werden. Ein Voxel aus Wasser hat, nach Goldman et al., demnach eine HU von -1000, ein Voxel aus dichtem, kortikalem Knochen eine HU von +1000 (17). Da das menschliche Auge nicht 2000 verschiedene Graustufen gleichzeitig unterscheiden kann, erlaubt die Bildwiedergabe-Software eine Fensterung in relevante HU-Bereiche. Wann benachbarte Strukturen eine unterschiedliche HU erhalten beschreiben Goldman et al. wie folgt: „Tatsächlich muss sich ein Material mindestens um 1% von seiner Umgebung unterscheiden, um einen unterschiedlichen Grauwert zu erhalten.“ (Übersetzung durch den Autor) (17).

Das PET stellt laut Basu et al. vor allem funktionale Prozesse des menschlichen Körpers dar mit jedoch unzureichender räumlicher Auflösung, durch das CT hingegen werden keine dynamischen, körperlichen Funktionen dargestellt, sondern eine anatomische Karte des Patienten mit hoher Ortsauflösung erstellt: „Glücklicherweise kann dieses Hindernis durch die Integration der funktionalen Information aus dem PET in die anatomische Information, die aus strukturabbildenden Methoden wie dem CT gewonnen werden kann, überwunden werden.“ (Übersetzung durch den Autor) (16), beschreiben Basu et al. die Vorteile einer Diagnostik durch integriertes PET-CT.

1.2.2.3 Ablauf der PET-CT-Untersuchung

Der Ablauf einer PET-CT Untersuchung besteht aus nachfolgenden Schritten: Zuerst muss der Patient für mindestens 4-6 Stunden fasten, um eine ausreichende Verstoffwechselung der zu injizierenden Glukose sicherzustellen, vgl. (18). Durch das Fasten sinkt die Blutglukosekonzentration, folglich muss die injizierte radioaktive Glukose (FDG), laut Basu et al., nicht mit großen Mengen bereits vorhandener Blutglukose um die Aufnahme in die Zelle konkurrieren (16). Zusätzlich minimiert nach Auffassung der Autoren ein niedriger Insulinspiegel den FDG-Uptake in andere Bereiche des Körpers, wie Herz, Leber und Muskeln, was in einer verringerten Hintergrundaktivität resultiere (16). Muskelkontraktionen führen zu einer gesteigerten zellulären Glukoseaufnahme durch Energieverbrauch des kontraktilen Apparates, welche wiederum in einer gesteigerten Hintergrundaktivität resultiert.

Nach der venösen FDG-Injektion wird vor dem ersten Scan bis zu eine Stunde gewartet, damit sich die FDG in Körper anreichern kann, vgl. (16, 18). Danach erfolgt ein Transmissions- und Emissionsscan.

Der zuvor durchgeführte „Blank scan (reference scan)“ (deutsch: Referenzscan) (16) sowie der Transmissionsscan helfe dabei, den „attenuation correction factor (ACF)“ (16) (deutsch: Abschwächungskorrekturfaktor) zu bestimmen, um so zu verhindern, dass fälschlicherweise externe Strahlung als patienteneigene Strahlung detektiert werde (16). Laut Harders et al. ergibt sich daher für einen vollständigen PET-Scan von der Schädelkalotte bis zum Oberschenkel eine Dauer von ungefähr 45 Minuten (18).

Als grundlegender pathophysiologischer Mechanismus hinter dem FDG-Uptake ist der erhöhte Tumorstoffwechsel zu nennen, der sich meist in Form einer gesteigerten zellulären Glukoseaufnahme darstellt.

Dieses Phänomen ist, nach Basu et al., hauptsächlich durch die gesteigerte Glykolyse von proliferierenden Tumorzellen begründet, welche zu einer Überexpression von Glukose transportierenden Membranproteinen (hauptsächlich GLUT1 und GLUT3) führe (16). Weitere Faktoren, die einen FDG-Uptake begünstigen, seien zudem der regional erhöhte Blutfluss sowie der verstärke Arbeitsumsatz glykolytischer Enzyme (16).

1.2.2.4 Messung und Interpretation der zellulären Glukoseaufnahme im PET/CT

Durch die regional erhöhte Glukoseaufnahme, welche durch die Strahlung der Radioisotope (wie dem F-18-FDG) von einem Ringdetektor aufgezeichnet wird, ist es möglich, lokal erhöhte Stoffwechselraten in einem definierten Bereich zu messen. Diese definierten Bereiche werden als region-of-interest (ROI) (deutsch: Regionen von Interesse) bezeichnet.

Die radioaktives FDG aufnehmenden Körperstrukturen, lassen sich visuell und semi-quantitaiv (als standardized uptake value (SUV), deutsch: Standardisierter Aufnahmewert), darstellen, vgl. (13, 16, 18). Weiterhin ergeben sich die Möglichkeiten, entweder durchschnittliche SUVs zu berechnen (SUVmean) oder in einer definierten ROI den maximalen Uptake des Markers in einem Voxel (SUVmax) zu identifizieren (14).

Die quantitative Bestimmung der SUV, hat sich in Studien als nützlicher Parameter in Diagnostik, Staging und Prognose gezeigt (15), kann die visuelle Analyse jedoch nicht gänzlich ersetzen (18, 19).

1.2.2.5 PET-CT in der Diagnostik des Lungenkarzinoms

1.2.2.5.1 M-Staging

Für die Detektion von Lungenherden durch das integrierte PET-CT zeigen Metaanalyse eine Sensitivität von ca. 90% und eine Spezifität von ca. 78%, bei malignen Herden ab einem Durchmesser von 10 mm sogar eine Sensitivität von 98% (3). Nach Madsen et al. ist das PET-CT mit seiner hohen Sensitivität dazu geeignet, Malignität in Lungenrundherden auszuschließen (20).

1.2.2.5.2 N-Staging

Anders verhält es sich bei der Frage nach einer mediastinalen Lymphknotenmetastasierung. Madsen et al. bewerten hier das PET-CT als

unzureichend, um mediastinale Metastasierung auszuschließen (20): die in ihrer Metaanalyse von 2016 herausgearbeiteten PET-CT-Sensitivitäten für das N-Staging liegen zwischen 50-91% (Tabelle 1, S.6, (20)).

Fünf weitere Metanalysen (Fischer et al. 2001; Gould et a. 2001; Hellwig et al. 2001; Wahidi et al. 2007; Ung et al. 2007, angeführt auf S.96 in der S3- Leitlinie Lungenkarzinom von 2018 (3)) , die das Lymphknotenstaging mit PET-CT untersuchten, stellen „eine Sensitivität und Spezifität von 74-85 % bzw. 85-92 % für die Unterscheidung eines N0/1- gegenüber einem N2/3-Status“ (3) fest.

Bei der Frage nach mediastinalen Lymphknotenmetastasen fungiert das PET-CT als Orientierungshilfe, um die Dignität mediastinaler Lymphknoten zu bestimmen. Dann erfolgt eine genauere Untersuchung der Lymphknoten mittels histologischer Sicherung durch ein bioptisches Verfahren.

1.2.2.5.3 Klinische Folgen des PET-CT-Stagings

Im Vergleich zum alleinigem Staging durch die Computertomographie zeigten Takeuchi et al. 2014 in einer Studie mit 592 Patienten, die an nicht-kleinzelligem Lungenkrebs erkrankt waren, dass durch eine Erweiterung der Diagnostik durch eine PET in 28,7% der Fälle der Tumor einem anderen Stadium zugeordnet werden könnte (21). In 16,4% der Fälle erfolgte eine Einstufung in höhere Stadien, in 12,3% in niedrigere Stadien.

In der *PLUS Studie* von Tinteren et al. führte die Erweiterung der Diagnostik, anders als bei Takeuchi et al. 2014, vorranging zur Höherstufung des Tumorstadiums (22).

In Folge des genaueren Stagings ist eine verbesserte Anpassung der Therapie möglich, welche sich hauptsächlich am Tumorstadium orientiert, vgl. (2, 10, 11).

Tinteren et al. belegten in der 2002 durchgeführten *PLUS Studie* erstmals einen konkreten Vorteil des PET-CT für Patienten mit NSCLC. Durch ein erweitertes Staging mit zusätzlichem PET, welches in einer Versuchsgruppe das konventionelle CT als Bildgebung ergänzte, konnte die Häufigkeit „unnötiger“ (22) (Übersetzung durch den Autor) Thorakotomien von 41% in der konventionellen CT Vergleichsgruppe auf 21% in der PET-CT Vergleichsgruppe gesenkt werden, was einer relativen Risikoreduktion von 51% entspricht (22).

Aus der Metanalyse von Madsen et al. geht in Bezug auf die Malignitätsbestimmung von Lungenrundherden folgendes Fazit hervor: „Das PET-CT […] reduziert den Bedarf einer Biopsie, wenn negativ, wegen einer höheren Spezifität als das CT“ (20).

Herderer et al. konnten 2006 zeigen, dass ein integriertes PET-CT direkt zu Beginn der Diagnostik zwar die Anzahl invasiver Operationen sowie der Mediastinoskopien reduzieren kann, die absolute Anzahl diagnostischer Tests sowie die damit verbundenen Kosten sich aber nicht signifikant vom konventionellen Diagnostikschema (ohne PET) unterschieden (23).

1.2.3 Behandlung

Die Behandlung des NSCLC erfolgt abhängig von der Stadieneinteilung des Lungenkarzinoms nach *UICC8* (*Union internationale contre le cancer*) (2, 24).

Bis zum Stadium $IIIA_{3,}$ welches durch eine lokale Tumorausbreitung bis T3 gekennzeichnet ist, erfolgt eine operative Therapie. Das Tumorstadium T3 ist durch einen maximalen Tumordurchmesser von 5-7cm oder die Infiltration von Thoraxwand, Perikard, Nervus phrenicus oder einen zusätzlichen Tumor im identischen Lungenlappen, definiert, vgl. (11).

Ab Stadium $IIIA_4$, welches durch eine Tumorausdehnung über den Hemithorax hinaus oder das Vorhandensein von unförmigen oder auf verschiedenen Etagen befallene Lymphknoten, in gemeinsamem Auftreten mit einem Tumor, der über die Lunge hinaus ausgebreitet ist, vorliegt, kommt ein palliatives Behandlungskonzept zu Anwendung, vgl. (2, 3, 10, 11, 25).

Die kurative Therapie erfolgt in Form der operativen Resektion oder durch Bestrahlung. Möglich sind Lobektomien (Lappenresektionen) bis hin zu Pneumektomien (Lungenflügelresektionen) inklusive einer systematischen, mediastinalen Lymphknotendissektion und adjuvanter (nachfolgender) Kombinationschemotherapie ab einem Tumordurchmesser von 4 cm sowie einem positiven Nodalstatus bis N2 (2, 3). Eine Alternative zur chirurgischen Resektion ist die stereotaktische Bestrahlung, welche außerdem in palliativen Situationen bei Hirnmetastasen zur lokalen Tumorkontrolle eingesetzt werden kann. Das allgemeine palliative Therapieschema ist die definitive Radiochemotherapie, welche durch funktionserhaltende Operationen begleitet werden kann (2, 3).

Ein alternativer Therapieansatz macht sich neuere Forschungsergebnisse der Molekularbiologie zu Nutze. Neue, spezifische Therapien haben Treibermutationen, wie zum Beispiel EGF-Rezeptormutationen (EGF: Epidermal Growth Factor), als Ziel und inhibieren dessen Signalkaskaden, in diesem Fall mit Tyrosinkinase-Inhibitoren, vgl. (3, 8, 26, 27).

1.3 Künstliche Intelligenz in der Medizin

1.3.1 Definitionen

1.3.1.1 Künstliche Intelligenz

Grundsätzlich beschreibt das Gebiet der Künstlichen Intelligenz (KI, englisch: artificial intelligence, kurz: AI) den Versuch, mit mathematischen Modellen das menschliche Gehirn mitsamt seinen Fähigkeiten nachzubilden. Insbesondere die Simulation menschlicher Entscheidungsprozesse durch Algorithmen ist hierbei von großem Interesse. Folglich findet KI in nahezu allen Gebieten mögliche Anwendung, in denen Menschen versuchen, systematische Entscheidungen, im Kontext von Problembearbeitungen, zu treffen.

Saurabh Jha von der University of Pennsylvania sieht Künstliche Intelligenz, ähnlich zum Turing-Test (28), vor allem durch die Nachahmung menschlicher Kognition definiert: „Künstliche Intelligenz – die Imitation menschlicher Erkenntnis durch Computer." (Übersetzung durch den Autor) (29).

Z. Ghahramani von der University of Cambridge sieht intelligentes Handeln vordergründig im richtigen Umgang mit Wahrscheinlichkeit und dem Abwägen von Unsicherheit: „Letzten Endes beruht Intelligenz auf dem Verstehen und Handeln in einer nur unvollständig wahrgenommen und unsicheren Welt." (Übersetzung durch den Autor) (30). Nach Ghahramani können Maschinen, auf Grundlage umfangreicher Datenmengen, durch probabilistische Modelle Wahrscheinlichkeiten für das Eintreten bestimmter Ereignisse errechnen, die unter diesen Voraussetzungen rational begründbar sind (30).

In diesem Kontext können Algorithmen und Maschinen also durchaus in der Lage sein, intelligente Abwägungen über die Wahrscheinlichkeit unsicherer Ereignisse zu vollziehen. Diese Abwägungen basieren dann auf Daten, die aktuelle Modelle begründen, bis neue Daten alte Modelle erweitern oder ablösen. In Zeiten von Big Data und

evidenzbasierter Medizin, in der Ärzte und Ärztinnen mit Risikoanalysen und Prognose-Scores arbeiten, scheint eine Implementierung Künstlicher Intelligenz in die Medizin daher möglich.

1.3.1.2 Medizinische, empirische Definition von Künstlicher Intelligenz

In der Medizin hat eine exakte Definition menschlicher Intelligenz für die Entwicklung medizinische nutzbarer Algorithmen aktuell nur begrenzte Relevanz. Hier kommt es primär auf die Leistungsfähigkeit, Sicherheit und Verlässlichkeit eines Systems an.

Mögliche sind, in Analogie zu Ghahramanis Ansatz, empirische Definitionen, die die Treffsicherheit von Aussagen eines Systems, bezogen auf eine spezifische Fragestellung, bewerten, und diese mit den Ergebnissen einer menschlichen Expertengruppe abgleichen. So entstehen Vergleichswerte, wobei der Goldstandard die menschliche Expertise in einem bereits etablierten Prognose- oder Diagnosesystem sein kann.

Die erfolgreiche Bearbeitung einer solchen Fragestellung durch ein künstliches System kann zwar mit dem Begriff Intelligenz beschrieben werden, dies ist aber nicht zwingend notwendig, da hier nicht primär nach der Intelligenz eines Systems gefragt wird, sondern dessen quantifizierbarer Erfolg als Endpunkt dient.

1.3.2 Formen der Künstlichen Intelligenz in der Radiologie

1.3.2.1 Radiomics

Radiomics ist, nach Lambin (2012), ein Teilbereich der KI in der Medizin und beschreit die Extraktion quantifizierbarer Information von regions-of-interest (ROIs) aus radiologischen Bildern durch einen mathematischen Algorithmus (31). *Radiomics* muss aber nicht zwingend auf KI basieren, sondern kann auch mit einfachen, statistischen Modellen zur Anwendung kommen.

Das Gebiet der *Radiomics* wurde schon im Jahre 1999 in den *artificial intelligence in medicine (AIME) conferences* als „knowledge discovery from databases“ (deutsch: Wissensentdeckung durch Datenbanken), „data mining“ oder „intelligent data analysis“ (32) bezeichnet und beschreibt das ungerichtete Sammeln großer Datenmengen, um sie nachfolgend auf statistisch relevante Merkmale hin zu untersuchen, vgl. (32, 33).

Laut Peek et al. stellt diese unselektive Herangehensweise eine Neuerung gegenüber traditioneller Forschung dar, da hier nicht, im Kontext einer spezifischen Hypothese, selektiv Daten zu dessen Überprüfung gesammelt würden (32).

Ferreira et al. beschreiben *Radiomics* als Korrelation von Daten, basierend auf quantitativen und qualitativen Merkmalen, von denen, in Verbindung mit Patienteninformationen, diagnostische und prognostischen Informationen abgeleitet werden könnten (34).

Durch die Nutzung künstlicher Intelligenz in der Radiologie verschiebt sich, laut Saurabh Jha, das Tätigkeitsfeld der zukünftigen Radiologen weg von der einfachen, visuellen Bildanalyse hin zu „information specialists" (29) (deutsch: Informationsspezialisten), die durch Automatisierung generierte Bildbefunde überprüften und mit umfangreichen Patienteninformationen in den klinischen Kontext integrierten.

Es wird allgemein zwischen zwei Analyseformen unterschieden: dem *machine learning* und dem *deep learning*, vgl. (33, 35). In unserer Arbeit verwenden wir *machine learning* mit *supervised learning*.

1.3.2.2 machine learning

machine learning ist ein Verfahren, bei dem ein Algorithmus Datensätze auf vorab definierte Merkmale (*features*) prüft und nachfolgend Informationen extrahiert.

Die *features* können in der Radiologie als „imaging biomarkers“ (36) verstanden werden, die, nach Zwanenburg et al., die Eigenschaften relevanter Bildbereiche quantifizieren können. Die *features* sind von Experten exakt vorgegeben und der Algorithmus auf deren Erkennung spezifisch programmiert.

Damit ist diese Version der KI deterministisch, also vorhersagbar, da bei identischer Dateneingabe identische Ergebnisse ermittelt werden. In einem nachfolgenden Schritt werden dann die statistisch relevantesten *features* ausgewählt. Ein Vorteil dieses Verfahrens ist, dass die vom Algorithmus zu suchenden *features* von medizinischen Experten bestimmt und kontrolliert werden. Diesen extern-kontrollierten Prozess nennt man auch *supervised learning* (33) (deutsch: Beaufsichtigtes Lernen). Weil zu Beginn der Entwicklung eines Vorhersagemodells das Analyseziel (auch *feature* oder *target* genannt, vgl. (33, 35)) bereits bekannt ist, sind weniger Daten notwendig, um ein stabiles Detektionssystem zu erstellen, vgl. (35).

Nachteilig kann sich diese vorgegebene Begrenztheit der festgelegten *features* jedoch auswirken, wenn gewisse Abnormitäten im Bild durch diese nur unzureichend erfasst werden, andere hingegen redundant sind, vgl. (34, 37, 38).

Zusammenfassend lässt sich sagen, dass beim *machine learning* der Blickwinkel der Analyse in Form von *features* und der Modellierung des Algorithmus vorgegeben ist und daher die Ergebnisse stark vom gewählten Blickwinkel abhängen.

Deep learning ähnelt in vielen Punkten dem machine learning, jedoch werden hier komplexere Algorithmen teilweise ohne explizite Merkmalsdefinitionen vorab verwendet, wodurch es sich von *machine learning* maßgeblich unterscheidet. Ziel des *deep learning* ist es, die Ausprägung von Bildmerkmalen aus Datensätzen zu identifizieren, die vorher nicht definiert wurden und kann deshalb als Ergänzung des *machine learnings* betrachtet werden.

1.3.3 Radiomics: Von qualitativ-subjektiver zu quantitativ-objektiver Bild-Analyse

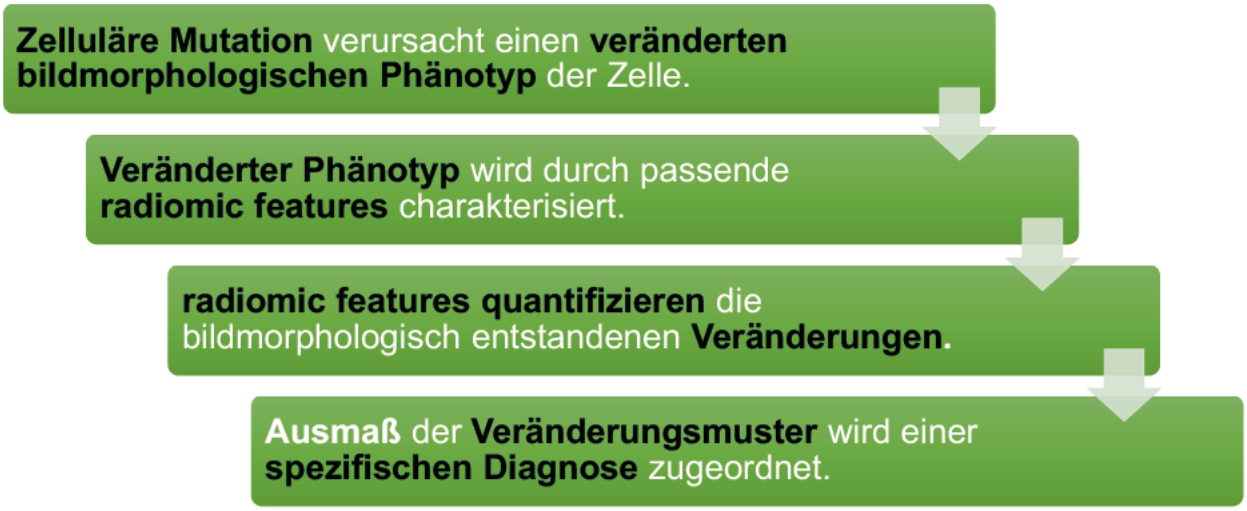


Abbildung 4: Hier wird das Grundprinzip von radiomic features dargestellt. Im ersten Schritt kommt es in Folge einer zellulären Mutation zu einer tumorösen Entartung, die sich in einer veränderten bildmorphologischen Struktur im CT darstellen kann. Diese im veränderten zelluläre Phänotyp entstandenen Bildmerkmale können durch radiomic features dargestellt werden und das Ausmaß der bildmorphologischen Veränderungen quantifizieren. Anhand des Ausmaßes der Veränderungsmuster kann dem neuen Bildbefund mit einer gewissen Wahrscheinlichkeit eine Diagnose zugeordnet werden.

1.3.3.1 Von der Mutation zum radiologischen Phänotyp

Die Zellen des menschlichen Körpers mit ihrer genetischen Informationen im Zellkern unterliegen einem ständigen Wechselspiel zwischen genetischen und epigenetischen Mutationen mit nachfolgenden Reparaturvorgängen, um den physiologischen Metabolismus der Zelle aufrecht zu erhalten, vgl. (27). Gelingt den zellulären

Reparaturmechanismen in Form verschiedener Enzymfamilien, wie zum Beispiel das Zellzyklusprotein p53, keine Korrektur der genetischen Information, wird der physiologische Zelltod, die Apoptose eingeleitet, vgl. (26).

Bleibt hingegen eine Mutation unerkannt und betrifft die Proliferationseigenschaften der Zelle, kann diese entarten und sich unkontrolliert vermehren. Diesen Zustand ungebremsten und teilweise invasiven Wachstums bezeichnet man als Krebs, welcher sich beispielsweise in Form eines soliden Tumors darstellen kann. Durch Mutationen, also genetische Veränderungen, kann so eine Zelle mit veränderten Merkmalsausprägungen entstehen, welche sich von ihrer ursprünglichen Zellreihe unterscheidet. Die daraus entstehenden, oft visuell erkennbaren, Unterschiede bezeichnet man in der Bildgebung als Heterogenität. *Features*, die diese Heterogenität darstellen, korrelieren mit Zellzykluswegen die eine erhöhte Proliferationsrate bedingen, vgl. (39) (40).

Die genetische Tumorheterogenität verändert also den Phänotyp der Zelle sowie zelluläre Stoffwechselvorgänge und hat diagnostische, prognostische und therapeutische Relevanz, vgl. (26, 27, 39-42). Aerts et al. entdeckten 2014 eine signifikante Assoziation zwischen *radiomic features* und Genexpressionsmustern (Tabelle 4b, S.5 in (40)) sowieo *radiomic features* und Tumorgenen, die darauf hindeuteten, dass diese *features* tumorbiologische Prozesse darstellten (40). Nach Aerts et al. sind *Radiomics* deshalb in der Lage, einen allgemeinen, prognostischen bedeutenden Tumor-Phänotyp zu quantifizieren, der wahrscheinlich auch auf andere Krebsarten übertragbar sei (40).

In der radiologischen Bildgebung stellt sich diese Heterogenität in der Computertomographie durch unterschiedliche Dichtestufen in Form von Grauwerten dar, in der nuklearmedizinischen Bildgebung im FDG-PET durch eine unterschiedliche Uptake-Varianz des Tracers.

Beide Verfahren verwendet man, teilweise kombiniert, in der Diagnostik (siehe Kapitel *Einleitung,* Absatz *Diagnostik des Lungenkarzinoms*), Therapieplanung und für die Prognose des Lungenkarzinoms, vgl. (3, 14, 43).

1.3.3.2 Vom radiologischen Phänotyp zum quantifizierbaren radiomic feature

Leitidee der *Radiomics* ist es, bisher ungenutzte quantitative Information aus radiologischer Bildgebung zu identifizieren und nutzbar zu machen.

Um mögliche neue, relevante Parameter zu extrahieren, welche für das *machine learning* Grundvoraussetzung sind, bedarf es einer systematischen Analyse. Das *National Cancer Institute* (NCI) des *U.S. Department of Health and Human Services* versucht, laut Chen et al., mit den *Quantitative Imaging Network guidelines* einen systematischen Zugang zu diesen *features* zu etablieren: Zu Beginn soll die standardisierte Bildgebung erfolgen, in der dann relevante Bildbereiche segmentiert werden („lesion segmentation"), welche nun auf vielfältige *features* hin quantitativ untersucht werden („feature extraction, and quantitative data analysis") (37).

Trotz der teilweise noch uneinheitlich definierten Merkmalgruppen sowie die uneinheitliche Bearbeitung ermittelter Merkmale, gibt es weit verbreitete und häufig genutzte Feature Arten, die im Feature-Set von PyRadiomics erfolgreich zusammengeführt werden (https://pyradiomics.readthedocs.io/en/latest/features.html, letzter Aufruf: 05.06.2019). Dabei bilden einige *features* geometrische Strukturen ab, andere zeigen bestimmte Pixel- oder Voxel-Konfigurationen an und sind durch mathematische Funktionen darstellbar.

Hosny et al. beschreiben diese „predefined engineered features" (deutsch: vordefinierte, hergestellte Merkmale) als „set of context-based human-crafted features designed to represent knowledge regarding a specific data space."(35) (Übersetzung durch den Autor: Set aus Kontext-basierenden, durch Menschen hergestellte Merkmale, die dazu dienen, Wissen über einen spezifischen Datenbereich anzuzeigen.).

Eine detaillierte Erläuterung zu *radiomic features* ist im Kapitel *Material und Methoden des Klassifizierungsalgorithmus* im Absatz *Radiomic-Feature-Set* zu finden.

1.3.3.3 Feature-Analyse

Die ermittelten *features* werden schließlich einer quantitativen Datenanalyse unterzogen um statistisch signifikante Ergebnisse zu identifizieren. Schlussendlich soll es so möglich sein, eine aussagekräftige „virtual biopsy" (34) (deutsch: virtuelle Biopsie) der Läsion zu erhalten, vgl. (44). Ein Vorteil dieser Methode gegenüber einer konventionellen Tumorbiopsie ist, dass der gesamte Bereich der Läsion untersucht wird, anstatt per konventioneller Biopsie nur einen kleinen Teil der Tumormasse zu erhalten, von der man hoffen müsse, sie sei identisch mit dem histopathologischen Typ des übrigen Tumorgewebes, was laut Aerts et al. häufig nicht der Fall ist (40).

Zusätzlich nutzt man mit der CT-Aufnahme bereits vorhandene Daten und erspart dem Patienten somit die Risiken eines invasiven Eingriffs, vgl. (34, 40). Auch mehrzeitige Aufnahmen, beispielsweise zur Therapiekontrolle, sind dadurch möglich, vgl. (34, 35, 37, 38, 40, 45).

1.4 CT-*Radiomics* im Kontext des Nicht-Kleinzelligen Lungenkarzinoms

1.4.1 Aktueller Stand der Forschung

1.4.1.1 N-Staging

Toney et al. stellten 2014 ein neuronales Netzwerk vor, welches auf der Grundlage von vier FDG-PET und CT Parametern („(Node sizes) + (nodal uptake/Bkg) + (primary SUVmax) + (primary size)“ (46)) mit einer Genauigkeit von 99.2% den chirurgisch-pathologischen Nodalstatus bei Nicht-Kleinzelligen Lungenkarzinomen voraussagen konnte. Bei gleicher Aufgabe erreichte ein „expert PET reader“ (46) (deutsch: erfahrener PET-Befunder) nur eine Genauigkeit von 72.4%. Diese Studie legt nahe, dass KI auch in der Bestimmung des Nodalstatus bei Lungenkarzinom im CT eine Rolle spielen kann, wobei hier, neben CT-Features, noch auf zwei Uptake-Parameter des FDG-PETs zurückgegriffen wurde.

Flechsing et al. konnten 2014 in ihrer Arbeit *Quantitative Volumetric CT-Histogram Analysis in N-Staging of 18F-FDG–Equivocal Patients with Lung Cancer* den zusätzlichen Nutzen einer volumetrischen CT-Histogramm-Analyse zur Identifizierung des Nodalstatus bei Lungenkarzinomen zeigen (47). Die mediane Dichte ist demnach bei histologisch-malignen Lymphknoten mit „33.2 Houns-field units [HU]“ im Vergleich zu histologisch-benignen Lymphknoten mit „10.1 HU“ signifikant erhöht (47). War der FDG-Uptake eines Lymphknoten uneindeutig, lag bei „cutoff value of 20 HU“ die Inzidenz von malignem Lymphknotenbefall bei „88%“ (47).

Im Jahr 2017 konnten Flechsing et al. ihre Ergebnisse von 2014, unabhängig vom Subtyp des Lungenkarzinoms, bestätigen: „Maligne Lymphknoten hatten eine signifikant höhere mediane CT-Dichte […] im Vergleich zu benignen Lymphknoten […], unabhängig vom histologischen Subtyp.“ (Übersetzung durch den Autor) (48).

Giesel et al. bestätigten 2017, dass PET-positive Lymphknoten (gemessen durch die SUVmax) bei Krebspatienten (unter anderem bei: Lungenkrebs, Malignem Melanom,

Prostatakarzinom) mit einer erhöhten Dichte im CT korrelien, was als zusätzlicher Parameter bei einem uneindeutigen PET-Befund nutzbar sein könne (48).

Diese erhöhte CT-Dichte positiver Lymphknoten lässt sich auch durch *radiomic features* quantifizieren (Beispielsweise im Feature-Set von PyRadiomics, siehe Kapitel *Material und Methoden des* Klassifizierungsalgorithmus, Absatz *Radiomic-Feature-Set*) und ist Gegenstand aktueller Forschungen.

1.4.1.2 Tumorcharakterisierung

Swensen et al. stellten im Jahr 2000 fest, dass pulmonale Rundherde mit einer Kontrastmittelaufnahme über dem Schwellenwert von 15 Houndsfield-Einheiten mit einer Sensitivität von 98% maligne Neoplasien anzeigen, wogegen eine fehlende Kontrastmittelanreicherung mit einer Spezifität von 58% nach Meinung der Autoren zwar ein Hinweis für Gutartigkeit sein kann, jedoch nicht ausreicht, um einen Befund endgültig als benigne klassifizieren zu können (49).

1.4.1.3 Entartungsrisiko

Hawkins et al. entwickelten 2016 einen Klassifizierungsalgorithmus, der mit Hilfe von 23 *radiomic features* die Wahrscheinlichkeit bestimmen sollte, mit der ein pulmonaler Rundherd in der Zukunft maligne entartet (50). Für einen Vorhersagezeitraum von einem und zwei Jahren ergaben sich Genauigkeiten von „80% (area under the curve 0.83) and 79% (area under the curve 0.75)“ (50). Somit ist, nach Aussage der Autoren, der genutzte Algorithmus genauer als das *Lung Imaging Reporting and Data System* (Lung-RADS), entwickelt vom *American College of Radiology*, zur Standardisierung des Lungenkrebs-CT-Screenings, sowie eine Entartungs-Prognose, die ausschließlich auf Volumenparametern basiert (50).

1.4.1.4 Histologischer Subtyp

Aerts et al. entdeckten 2014, dass *radiomic features* in Lungen- und Kopf/Hals Tumoren signifikant mit verschiedenen tumorbiologischen Genmustern assoziiert sind und folglich diese Features verschiedene biologische Mechanismen darstellten (40). Insbesondere die Heterogenität abbildenden Features korrelieren, laut den Autoren, stark mit Zellzykluswegen, was eine verstärkte Proliferation heterogener Tumoren anzeige (40).

Diese Ergebnisse von Aerts et al. konnten im Jahr 2017 durch Grossmann et al. bestätigt werden: *radiomic features*, die intratumoröse Heterogenität anzeigten, seinen demnach

in der Lage, die Aktivität der RNA-Polymerase vorherzusagen (AUC 0.62) (39), welche als ein Maß für die Zellproliferation angesehen wird und typischerweise in Tumorzellen erhöht ist.

Velazquez et al. entdeckten 2017 eine „radiomic siganture“ (42) (deutsch: Radiomics Unterschrift), die zwischen EGFR-positiven und EGFR-negativen (AUC 0.69) sowie zwischen EGFR-positiven und KRAS-positiven (AUC 0.80) Lungentumorgen unterscheiden kann (42).

Forschungsergebnisse wie diese könnten in Zukunft eine noch effektivere molekularpathologische Diagnostik und Therapie ermöglichen, vgl. (2, 3, 8).

Flechsing et al. gelang es im Jahr 2017 nicht, den Tumorsubtyp eines Lungenkarzinoms anhand von „volumetric density analysis“ (51) (deutsch: volumetrischer Dichteanalyse) zu ermitteln.

1.4.1.5 Prognose

Aerts et al. konnten in ihrer Studie *Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach* (40) von 2014 feststellen, dass *radiomic features* aus CT-Scans, die die Heterogenität im Primärtumor (Lungen- und Kopf/Hals Tumoren) beschreiben, mit einer schlechteren Überlebensrate assoziiert sind. Kompakte oder kugelförmige Primärtumoren seien hingegen mit einem besseren Gesamtüberleben assoziiert (40).

Im weiteren Verlauf der Studie isolierten Aerts et al. aus den 100 stabilsten features für jede der vier Feature Kategorien („(I) tumour image intensity, (II) shape, (III) texture and (IV) multiscale wavelet“ (40)) dasjenige mit der jeweils besten Testleistung und erhielten so eine „radiomic signature“ (40), bestehend aus vier *radiomic features*.

Verglichen mit dem Volumen des Primärtumors war diese „radiomic signature“ (40) als alleiniger Parameter sowie in Kombination mit dem Volumen ein signifikant besserer Prognoseparameter für das Gesamtüberleben als das isoliert betrachtete Tumorvolumen. Auch in Kombination mit dem TNM-Status erwies es sich in allen Kohorten als überlegener Prognoseparameter im Vergleich zum alleinigen TNM-Staging, in zwei Kohorten (Lung2, H&N2) war sogar die isoliert betrachtete „radiomic signature“ dem TNM-Staging prognostisch überlegen. Dabei blieb, laut Aerts et al., die Aussagekraft

auch zwischen den unterschiedlichen Behandlungsgruppen gleich, was den zusätzlichen Wert der *features* zeige (40).

Balagurunathan et al. zeigten 2014, dass viele *radiomic features* reproduzierbar sind und sich damit als „potential quantitative imaging biomarkers“ (52) (deutsch: potentielle quantitative Bildgebungs-Biomarker) eigneten, wobei das Merkmal *run-length gray-level nonuniformity* Patienten mit Nicht-kleinzelligem Lungenkarzinom in verschiedene Überlebensgruppen aufteilen konnte (52).

Nach Coroller et al. sind *radiomic features*, die die Homogenität von Lymphknotenmetastasen messen, ein geeigneter Parameter für die Vorhersage des Therapieansprechens auf eine neoadjuvante Chemotherapie, und besser dazu geeignet, als der prädiktive Wert von *radiomic features* des Primärtumors (53).

2 Fragestellung und Zielsetzung

Adeno- und Plattenepithelkarzinome der Lunge sind eine der weltweit häufigsten Tumoren mit einer schlechten Überlebensprognose (1). Bei der Ausbreitungs- und Rezidivdiagnostik kommen vielfältige Methoden zu Einsatz mit dem Ziel, anhand des Tumorstadiums kurative von palliativen Situationen unterscheiden zu können (3, 24).

Auf der Suche nach potentiellen metastasierten, mediastinalen Lymphknoten in der Erst- und Rezidivdiagnostik, wird eine orientierende Untersuchung des Mediastinums mittels PET-CT durchgeführt um den Nodalstatus der Lymphknoten zu klassifizieren, bevor mögliche Lymphknotenmetastasen histologisch in einer Biopsie gesichert werden (3).

Im Rahmen zunehmender Digitalisierungs- und Automatisierungsprozesse der Medizin bieten *Radiomics*-Diagnostiksysteme, die auf *machine*- und *deep learning*-Ansätzen der Künstlichen Intelligenz beruhen, neue diagnostische Möglichkeiten (31, 54).

2.1.1 Zielsetzung

Ziel der Forschungsarbeit war es, *Radiomics*-Klassifikationsalgorithmen mit *machine learning*-Techniken zu entwickeln, die anhand von CT-Scans den Nodalstatus mediastinaler Lymphknoten klassifizieren können, und dabei eine ähnlich genaue Vorhersageleistung wie die Klassifikation im PET-CT erreichen.

2.1.2 Fragestellung

In der Studie wurden folgende Fragestellungen bearbeitet:

- Gelingt es, reproduzierbare *Radiomics*-Klassifikationsalgorithmen ausschließlich anhand von CT-Daten zu erstellen?
- Was sind die Leistungsparameter der Nodalstatus-Vorhersage durch die Algorithmen?
- Gibt es einzelne Bildmerkmale im CT (*radiomic features*) mit besonderen, prädiktiven Eigenschaften?
- Wie gut klassifizieren die Algorithmen den Lymphknoten-Datensatz im Vergleich zur Klassifikationsleistung von zwei Radiologen?
- Entsteht durch die Lymphknotenklassifikation durch Algorithmen in Fällen, in denen die Radiologen unsicher waren, ein diagnostischer Zusatznutzen?

- Gibt es signifikante Unterschiede zwischen der Lymphknotenklassifikation der Algorithmen und der Lymphknotenklassifikation der Radiologen in Fällen, in denen die Radiologen unsicher waren?

3 Material und Methoden der Klassifizierungsalgorithmen

3.1 Kohorte

Die Stichprobe umfasst alle 381 Patienten, die zwischen Dezember 2011 und Mai 2018 im Lungentumorboard des Virchow-Klinikums der Charité Berlin mit einem Adeno- oder Plattenepithelkarzinom der Lunge vorgestellt wurden und innerhalb von 100 Tagen nach Diagnosestellung ein Ganzkörper- oder Thorax-PET-CT mit Kontrastmittel erhielten. Das durchschnittliche Alter der eingeschlossenen Patienten lag bei 65,88 Jahren bei einer Spannweite von 32 – 88 Jahren. Die Geschlechterverteilung lag bei 143 weiblichen sowie 238 männlichen Patienten.

Einschlusskriterien	
Diagnose	Adeno- oder Plattenepithelkarzinom der Lunge
Zeitraum	Dezember 2011 – Mai 2018
Ort	Lungentumorboard Virchow-Klinikum, Charité, Berlin
Bildgebungszeitraum	< 100 Tage nach Diagnosestellung
Bildgebungsmodalität	Ganzkörper- oder Thorax PET-CT-Scan
CT mit Kontrastmittel	vorhanden

Abbildung 5: Auflistung der Einschlusskriterien der Studie.

3.2 Organisation der Stichprobe

Die durchgeführten PET-CT-Aufnahmen der Lungenkrebspatienten-Stichprobe wurden anonymisiert in digitale Patienten-Ordner exportiert. Alle Patienten-Ordner erhielten eine Kennnummer, mit der eine Zuordnung zum jeweiligen histologischen Subtyp, trotz Anonymisierung, möglich ist.

Die Aufteilung der Subtypen bleibt auch nach Integration der anonymisierten Patientenkennnummern in die Übersichtstabelle erhalten. Diese Übersichtstabelle wurde in einem zweiten Schritt durch eine Einzelüberprüfung aller Stichproben um den Modus des CT-Scans der Patienten ergänzt.

Die Unterteilung in die Subtypen Adeno- und Plattenepithelkarzinom erfolgte, damit die Möglichkeit einer Subtyp-spezifischen Analyse in Zukunft bestehen bleibt. Durch die

Trennung beider Entitäten wäre es perspektivisch möglich, dass neue Klassifikationsalgorithmen ausschließlich segmentierte Lymphknoten eines einzelnen Subtyps für die Erstellung von Klassifikationssystemen nutzen und so prüfen könnten, ob jeder Subtyp eigene *radiomic features* exprimiert.

Da in die Stichprobe alle CT-Scans mit Kontrastmittel eingeschlossen werden, wurden Stichproben mit Kontrastmittel-CT dementsprechend mit ‚ok' kodiert. Dabei spielte es keine Rolle, ob ein Ganzkörper-CT oder nur ein Thorax-CT vorhanden ist. Alle Thorax-CTs erhielten die Kennung ‚nur Lunge ok'.

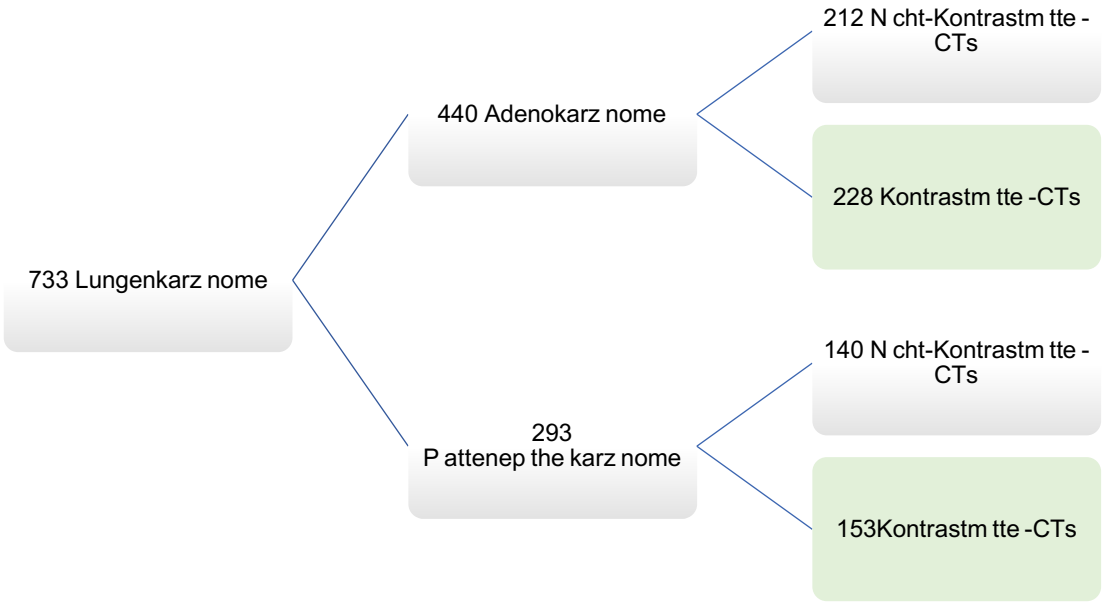


Abbildung 6: Schematisch dargestellter Ablauf der Auswahl aller geeigneten Kontrastmittel-CTs aus ursprünglich 733 möglichen Stichproben. Anfangs wurden insgesamt 733 Lungenkarzinom-Fälle des Tumorboards des retrospektiv gewählten Zeitraums von 2011-2018 in Adeno- und Plattenepithelkarzinome unterteilt. Dann erfolgte die Auswahl aller 381 Fälle, in denen ein geeignetes Kontrastmittel-CT vorhanden war (grün markierte Felder). In diesen Fällen wurden im Anschluss die Lymphknotensegmentierungen vorgenommen.

3.3 PET-CT Scan

Für alle PET/CT Scans wurde ein *Gemini Astonish TF 16 PET/CT* Scanner der Marke Phillips Medical Systems (Best, Niederlande) im 3D Aufnahmemodus verwendet.

3.4 Segmentierung

Insgesamt wurden in 381 Kontrastmittel-CT-Aufnahmen 1799 Lymphknoten segmentiert. 1349 Lymphknoten wurden mit Hilfe des PET-Scans als negativ klassifiziert, 450 Lymphknoten als positiv.

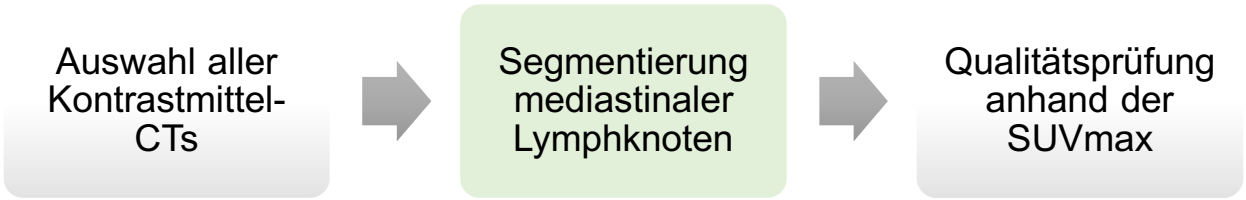


Abbildung 7: Ablauf der Lymphknotensegmentierung. Nach der Auswahl aller relevanten Kontrastmittel-CTs erfolgte in diesen die Segmentierung der mediastinalen Lymphknoten. Die segmentierten Dateien wurden danach mittels der SUVmax-Bestimmung einem Qualitätscheck unterzogen. Dort wurde durch eine Nuklearmedizinerin überprüft, ob der in der Segmentierung zugeordnete Nodalstatus mit dem FDG-Uptake Verhalten des Lymphknotens übereinstimmt.

3.4.1 Software

Die Segmentierung mediastinaler Lymphknoten erfolgte mit dem Programm *The Medical Imaging Interaction Toolkit* (MITK). MITK ist eine open source Software, die von der Abteilung der medizinischen und biologischen Informatik vom German Cancer Research Center lizensiert wird und auf der Website:

http://mitk.org/wiki/The_Medical_Imaging_Interaction_Toolkit_(MITK) (letzter Stand: 03.12.2018) zum kostenlosen Download angeboten wird.

Gearbeitet wurde mit der Version *MITK 2016.11 Release Workbench and Toolkit 2016.11* für Windows-Systeme auf einem *Windows 2012 R2 Standard-Betriebssystem*.

3.4.2 Ablauf der Segmentierungen

Im ersten Schritt wurde die zum jeweiligen Fall gehörende PET-CT-Aufnahme in der MITK-Software geöffnet. Nach dem Öffnen in MITK werden die CT-Aufnahme sowie die PET-Aufnahme als getrennte Bildebenen angezeigt und bleiben einzeln ansteuerbar. Für die visuelle Befundung und Segmentierung wurden die CT-Aufnahmen in einem Weichteilfenster von ca. 60 Houndsfield-Units (HU) im Zentrum und ca. 360 HU im Rahmen betrachtet. Die Fenster der PET-Scans wurden so angepasst, dass eine visuelle Unterscheidung zwischen physiologischen Uptake-Mustern und pathologisch erhöhten Bereichen möglich war.

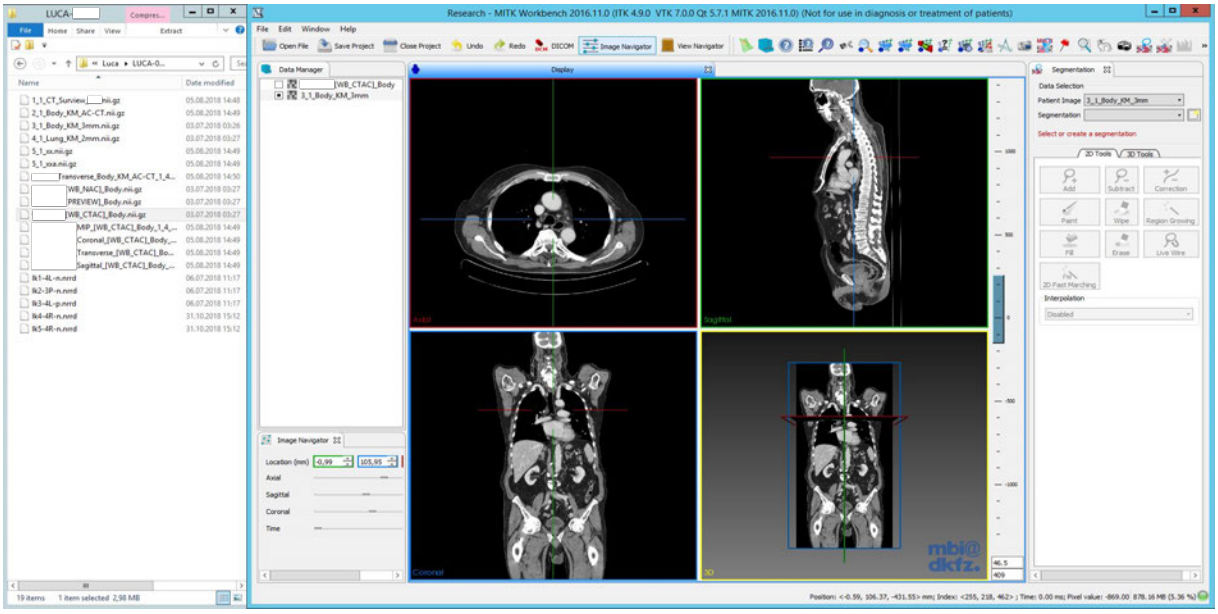


Abbildung 8: Screenshot des zweiten Arbeitsschrittes (Patienten-IDs entfernt). Links im Bild ist der geöffnete Patientenordner zu sehen, rechts im Bild wurden die Scans aus diesem Ordner in MITK geöffnet. Es wurde der CT-Scan mit Kontrastmittel und 3mm Schichtdicke ausgewählt. Nach dem Laden der Aufnahmen in MITK befindet man sich im hier dargestellten Übersichtsmodus. Dort kann durch alle drei Bildebenen navigiert werden. Eine Vergrößerung der Aufnahme sowie die alleinige Darstellung einer Ebene werden in der nächsten Abbildung gezeigt. Die Fensterung des CTs ist rechts neben den Schnittbildern erkennbar und per Mauscursor regulierbar. In der rechten Spalte des MITK-Fensters sehen wir den Segmentationscontroller, hier noch inaktiv.

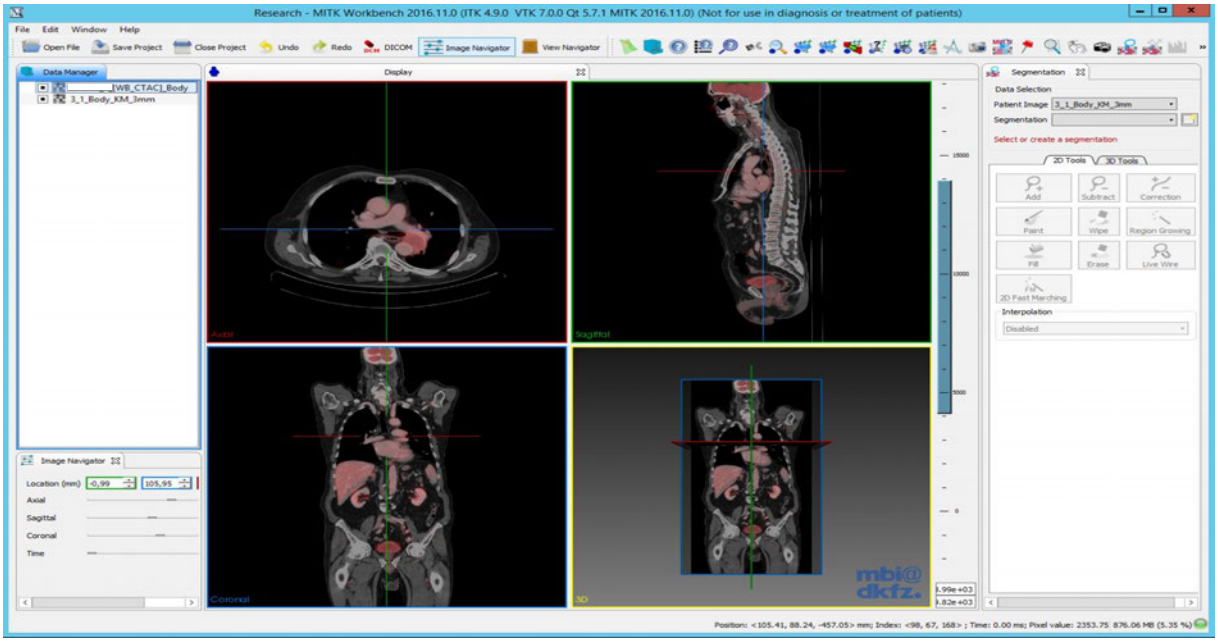


Abbildung 9: Screenshot des importierten PET-CTs in MITK mit Aktivierung beider Bildebenen (Patienten-IDs entfernt). Nun ist eine Navigation auf der Suche nach mediastinalen Lymphknoten durch beide Bildebenen gleichzeitig möglich.

Wurden beide Bildebenen korrekt gefenstert und aktiviert, erfolgte nun die Suche nach mediastinalen Lymphknoten. Dafür wurde die axiale Bildebene geöffnet und dort durch das Mediastinum navigiert.

Fand sich dort ein Lymphknoten wurde dafür im nächsten Schritt eine eigene, in der CT-Bildebene untergeordnete, Datei angelegt. Dieser Schritt ist eine Voraussetzung für die nun durchzuführende Lymphknotensegmentierung, da in dieser erstellten Datei die neu segmentierten Bildbereiche abgespeichert werden müssen. Nach abgeschlossener Segmentierung erhält man so eine zweite Datei, die nur den segmentierten Lymphknoten, ausgeschnitten aus dem ursprünglichen CT-Scan, enhält.

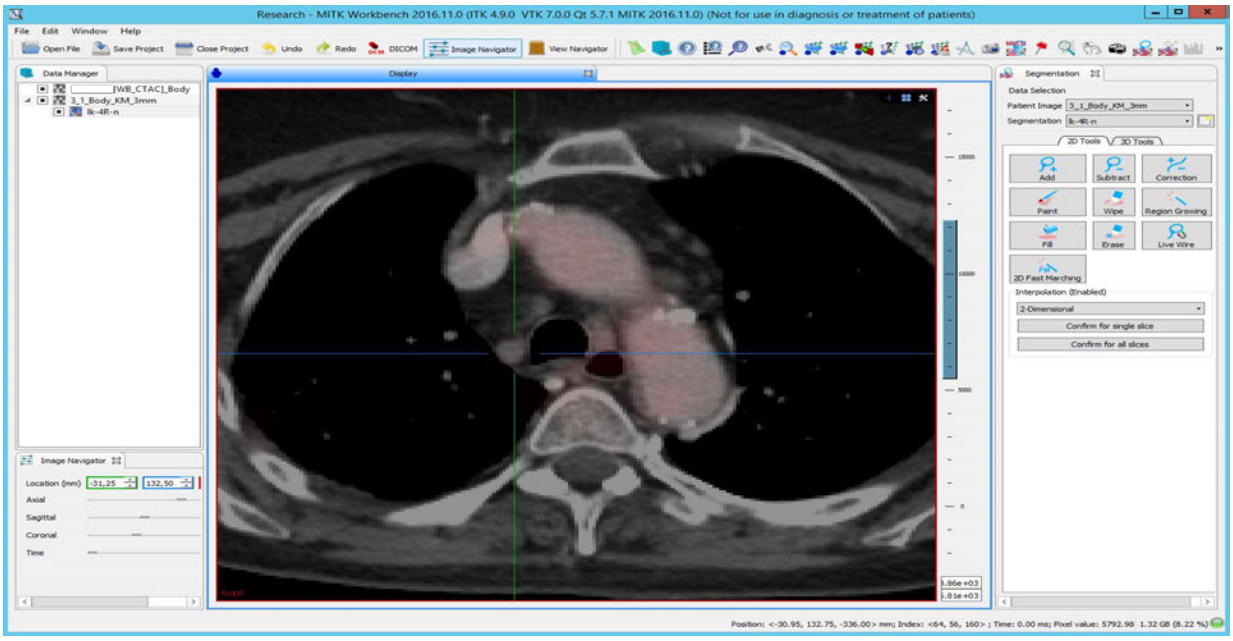


Abbildung 10: Screenshot, der zu segmentierende Lymphknoten liegt in der Mitte des Fadenkreuzes (Patienten-IDs entfernt). In der CT-Ebene wurde eine Unterdatei für die Segmentierung angelegt. Der Name der Datei setzt sich aus der, im nächsten Abschnitt erläuterten, Lymphknoten-Nomenklatur zusammen. Nun sind im Segmentations-Bereich in der rechten Spalte des MITK-Fensters alle Tools aktiviert, es kann mit der Segmentierung begonnen werden.

Die Segmentierung der mediastinalen Lymphknoten erfolgte in der axialen Bildebene manuell per Mauscursor sowie durch semi-automatische 2D-Interpolation von zuvor manuell festgelegten Endpunkten. Vor der finalen Bestätigung einer Interpolation wurde der gesamte, zu interpolierende Abschnitt, auf Vollständigkeit und Korrektheit visuell überprüft.

Die für die Segmentierung verwendeten Werkzeuge in MITK waren *Add* und *Substract* sowie die 2-D-Interpolationsfunktion.

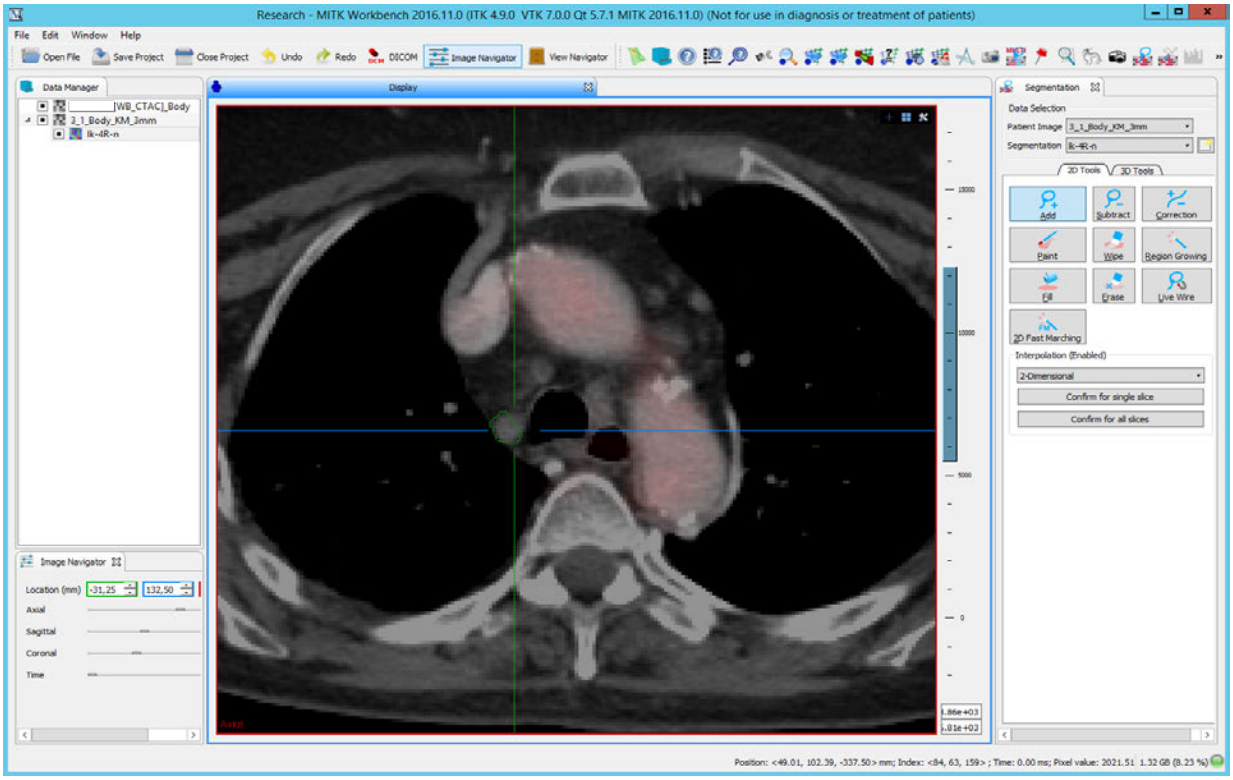


Abbildung 11: Screenshot, Start der manuellen Segmentierung (Patienten-IDs entfernt). Der grüne Rand des segmentierten Bereichs zeigt an, dass die Segmentierung gerade durchgeführt wird. Der Lymphknoten ist schon zu ca. ¾ erfasst.

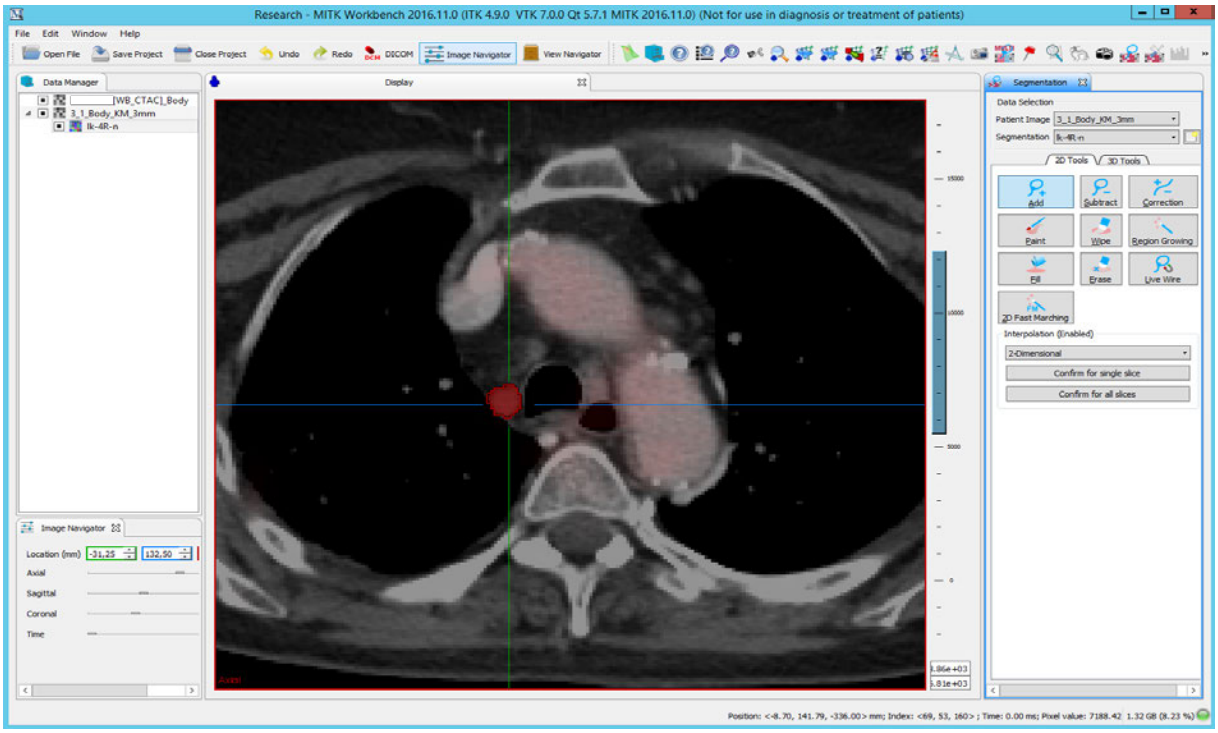


Abbildung 12: Screenshot, die Segmentierung in der gezeigten Ebene ist nun abgeschlossen (Patienten-IDs entfernt). Dies wird durch eine rote Fläche gekennzeichnet, die bei Beendigung der Segmentierung aus der grünen Umrandung hervorgeht.

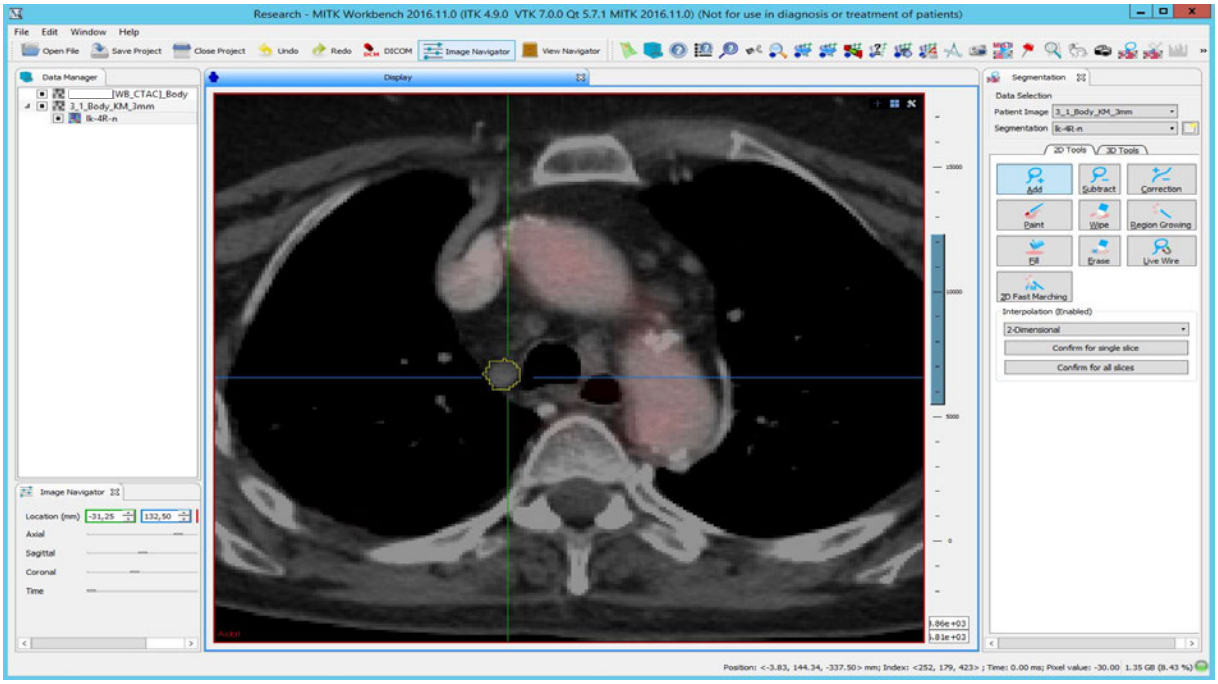


Abbildung 13: Screenshot, die gelbe Umrandung des Lymphknotens visualisiert die zu interpolierende Fläche (Patienten-IDs entfernt). Jetzt findet eine visuelle Prüfung statt, bevor die Interpolation zwischen zwei, zuvor manuell segmentierten, Ebenen bestätigt wird. Eine Bestätigung der Interpolation erfolgt nur dann, wenn die gelbe Linie den Lymphknoten exakt umrandet und vollständig einschließt, ohne jedoch periphere Strukturen zu erfassen.

3.4.2.1 Nomenklatur der Lymphknotendateien

Die systematische Benennung der Datei erfolgte anhand der anatomischen Lokalisation des Lymphknotens im Mediastinum und seines visuell-befundeten Nodalstatus mit Hilfe des Uptake-Verhaltens im PET.

Die Benennung der mediastinalen Lokalisation erfolgte analog zu *The International Association for the Study of Lung Cancer (IASLC) lymph node map* (55). In der Benennung des Nodalstatus steht der Buchstabe *p* für einen positiven, der Buchstabe *n* für einen negativen Nodalstatus. Damit wird sichergestellt, dass, trotz der isolierten Lymphknotendatei, die anatomische Lage des Lymphknotens und sein Nodalstatus nachvollziehbar bleiben.

Zusammenfassend lässt sich sagen, dass die für die Analyse relevanten Bildbereiche im CT ausgeschnitten wurden, um aus diesem Ausschnitt ein neues Bild zu erstellen. Dieser relevante Bildbereich wird auch region-of-interest (ROI) genannt. Die durch Segmentierungen entstandenen Dateien stellen folglich ROIs aus einem größeren Gesamtbild dar und können für die weitere Analyse genutzt werden.

Die Lymphknoten-Dateien wurden abschließend in dem zugehörigen Patientenordner abgespeichert, aus dem zuvor die PET-CT-Aufnahmen entnommen wurden, und können nun von dort für weiter Analysen und Qualitätsprüfungen exportiert werden.

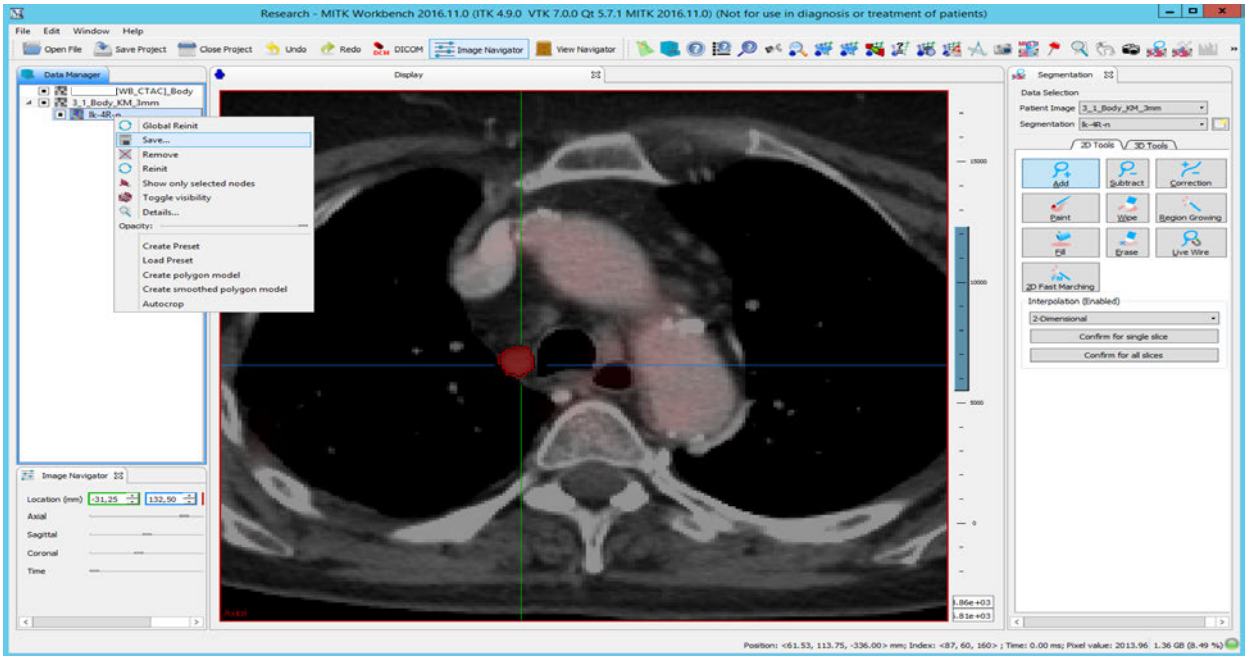


Abbildung 14: Screenshot, die nun vollständige Lymphknotendatei soll abgespeichert werden (Patienten-IDs entfernt). Hierfür wird die Unterdatei in der Spalte 'Data-Manager' von MITK angewählt.

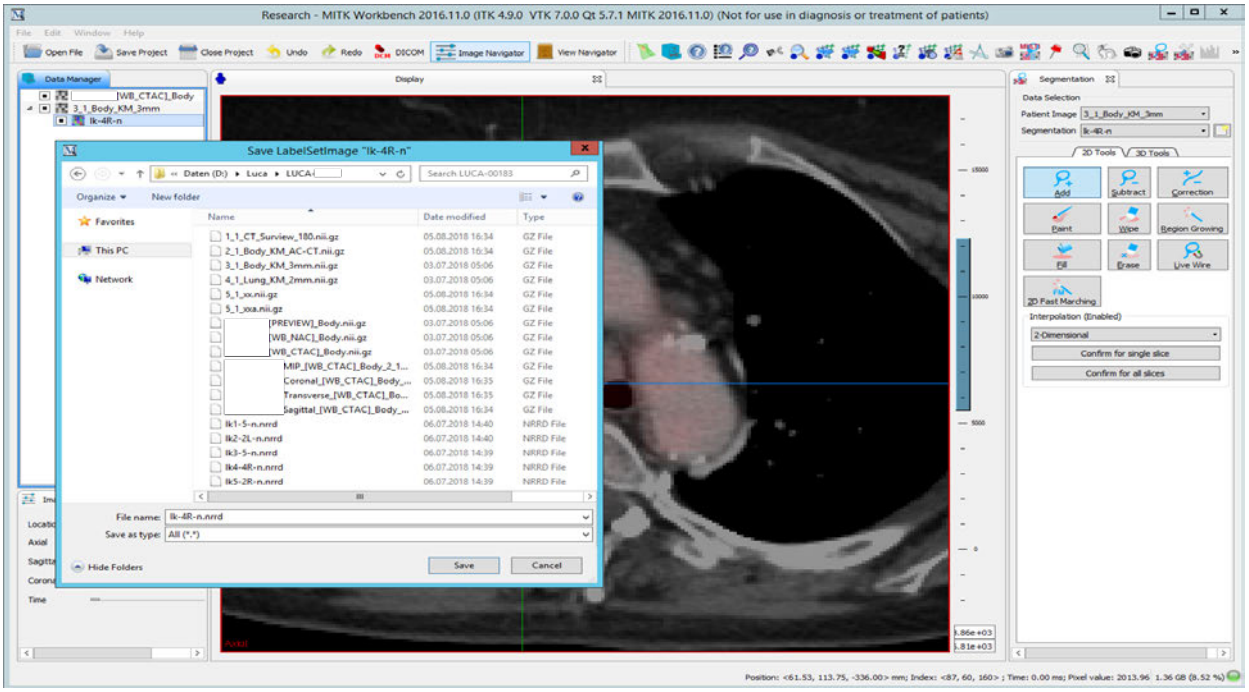


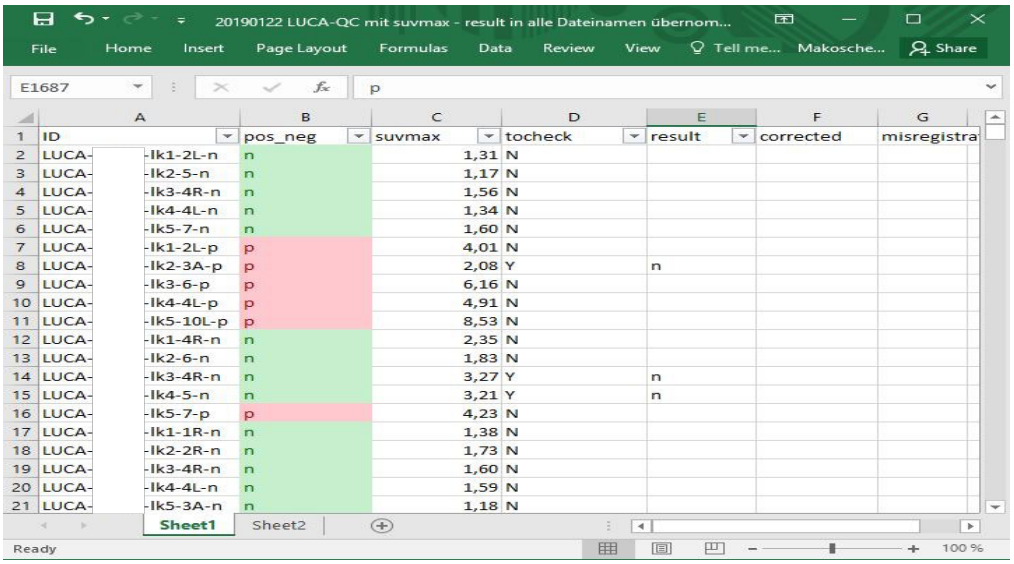
Abbildung 15: Screenshot, nur erfolgt das Speicher der Datei des segmentierten Lymphknotens im zugehörigen Patientenordner und ist dort für weitere Analysen abrufbar (Patienten-IDs entfernt).

3.5 Qualitätsprüfung des Nodalstatus

Nach erfolgter Segmentierung der einzelnen Lymphknoten wurden die SUVmax Werte aus den PET-Scan Daten nach Berechnung der Umrechnungsfaktoren mittels *3D-Slicer* (www.slicer.org, letzter Stand: 24.03.2019) erzeugt. *3D-Slicer* ist eine open-source Bildbearbeitungs- und Visualisierungssoftware für medizinische Bildgebung.

Der Schwellenwert für eine erneute Überprüfung des Nodalstatus lag bei negativ klassifizierten Lymphknoten bei einer SUVmax von > 2 und bei positiv klassifizierten Lymphknoten bei einer SUVmax von < 3. Zusammen mit einer Nuklearmedizinerin wurden diese Fälle nochmals auf Plausibilität geprüft.

Nach der Prüfung liegen 1349 Lymphknoten mit positiven und 450 Lymphknoten mit negativem Nodalstatus vor.



	A	B	C	D	E	F	G
1	ID	pos_neg	suvmax	tocheck	result	corrected	misregistra
2	LUCA- -lk1-2L-n	n	1,31	N			
3	LUCA- -lk2-5-n	n	1,17	N			
4	LUCA- -lk3-4R-n	n	1,56	N			
5	LUCA- -lk4-4L-n	n	1,34	N			
6	LUCA- -lk5-7-n	n	1,60	N			
7	LUCA- -lk1-2L-p	p	4,01	N			
8	LUCA- -lk2-3A-p	p	2,08	Y	n		
9	LUCA- -lk3-6-p	p	6,16	N			
10	LUCA- -lk4-4L-p	p	4,91	N			
11	LUCA- -lk5-10L-p	p	8,53	N			
12	LUCA- -lk1-4R-n	n	2,35	N			
13	LUCA- -lk2-6-n	n	1,83	N			
14	LUCA- -lk3-4R-n	n	3,27	Y	n		
15	LUCA- -lk4-5-n	n	3,21	Y	n		
16	LUCA- -lk5-7-p	p	4,23	N			
17	LUCA- -lk1-1R-n	n	1,38	N			
18	LUCA- -lk2-2R-n	n	1,73	N			
19	LUCA- -lk3-4R-n	n	1,60	N			
20	LUCA- -lk4-4L-n	n	1,59	N			
21	LUCA- -lk5-3A-n	n	1,18	N			

Abbildung 16: Screenshot der Excel-Tabelle mit allen segmentierten Lymphknoten inklusive der SUVmax-Werte. Stimmte der SUVmax-Wert und der visuell ermittelte Nodalstatus nicht überein, erfolgte eine erneute visuelle Prüfung gemeinsam mit einer Nuklearmedizinerin. Die Ergebnisse dieser zweiten Prüfung wurden in die ‚result' Spalte eingepflegt und der Dateiname entsprechend angepasst. Nach der Prüfung liegen 1349 Lymphknoten mit positiven und 450 Lymphknoten mit negativem Nodalstatus vor.

3.6 Grundlagen der statistischen Modellierung der Algorithmen und Vorbereitung des Lymphknoten-Datensatzes

Die statistische Modellierung wurde von Dr. Paul Schmidt in der Programmiersprache R (56) durchgeführt. Seine umfassenden statistischen Modellierungen inklusive des Quellcodes für R finden sich im Anhang im Kapitel 8. In den folgenden Abschnitten erläutere ich das Vorgehen und ordne es in den Kontext der Forschungsarbeit ein.

Das Ziel der statistischen Modellierung war die Entwicklung eines Klassifizierungsmodells, das eine Unterscheidung zwischen einem positiven und negativen Lymphknotenstatus allein anhand von Bildmerkmalen im CT ermöglicht.

3.6.1 Statistische Umsetzung in R

R ist eine kostenlose statistische Programmiersprache mit einem Kommandointerpreter, der statistische Funktionen und Befehle lesen und ausführen kann. Das Programm steht zum freien Download unter https://www.r-project.org (letzter Stand: 16.05.2019) zur Verfügung. Für alle statistischen Analysen wurde die R-Version *3.4.4* auf einem *x86_64-apple-darwin15.6.0* System unter *macOS 10.14.3* genutzt. Außerdem wurden die Add-on Pakete *tidyverse* (57) und *knitr* (58) für die Datenverarbeitung und Darstellung in R genutzt.

3.6.2 Verwendeter Datensatz und Kennzeichnung

Grundlage der Analyse ist der erstellte Datensatz, bestehend aus 1799 Lymphknoten-CT-Dateien und den jeweils zugehörigen Kennzeichnungen des Nodalstatus. Auf dieser Datenbasis erfolgt jede weitere statistische Analyse und Modellierung.

3.6.3 Unterteilung und Balancierung des Datensatzes

3.6.3.1 Unterteilung

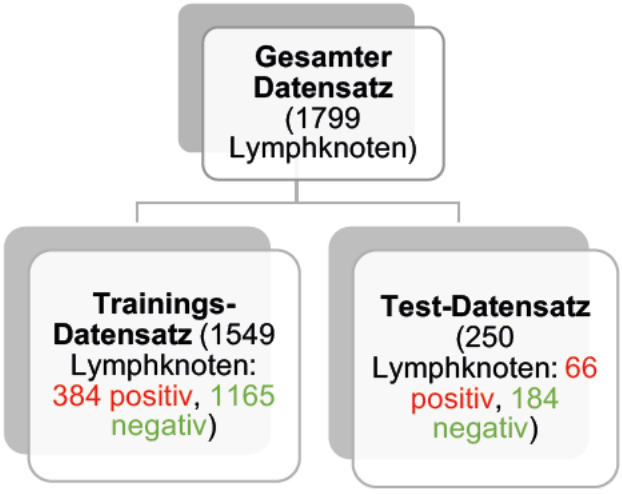


Abbildung 17: Unterteilung des ursprünglichen Datensatzes von 1799 Lymphknoten in einen Trainings- und einen Test-Datensatz. Im Trainingsdatensatz befinden sich nun 1549, im Test-Datensatz 250 segmentierte Lymphknoten.

Bevor mit der eigentlichen Analyse der Daten begonnen werden kann, muss eine Unterteilung des Datensatzes erfolgen. Dies wurde durch die Trennung des Datensatzes in folgende zwei Partitionen erreicht: einen Trainingsdatensatz und einen Testdatensatz.

3.6.3.1.1.1.1 Exkurs: Begriffsdefinition testing

Der Begriff des *testing* ist im machine learning üblich, entspricht aber im medizinischen Kontext eher einer Validierung. Identisches gilt für den Test-Datensatz, der einem Validierungsdatensatz in der klinischen Forschung gleicht. Da der Begriff der Validierung im machine learning jedoch eine andere Bedeutung innehat, werden im Folgenden aus Gründen der Übersichtlichkeit weiterhin die Begriffe testing und Test-Datensatz verwendet.

Der Schritt der Unterteilung ist von großer Bedeutung, da die spätere Überprüfung eines Klassifizierungsmodells nicht auf einem zuvor genutzten Trainingsdatensatz erfolgen kann, vgl. (59). Täte man dies dennoch, erhielte man künstlich gute Vorhersageergebnisse, da das Modell den Datensatz, auf dessen Grundlage es im vorangegangenen Training entwickelt wurde, passgenau abbildet und daher gut vorhersagen kann. Man spräche in diesem Fall von einem „overfitting“ (60) (deutsch: Überanpassung) des Modells.

3.6.3.2 Balancierung

Zusätzlich muss eine Balancierung des Trainingsdatensatzes in zwei identisch große Nodalstatus-Untergruppen (im Folgenden auch Response-Kategorien genannt) erfolgen und somit zur Hälfte aus positiven Lymphknoten und zur Hälfte aus negativen Lymphknoten bestehen. Dadurch kann das Klassifizierungsmodell durch reines Raten keine, sich signifikant vom Zufall unterscheidende, Vorhersagerate treffen, weil beide Antwortmöglichkeiten/Response-Kategorien gleich häufig vorkommen. Da eine Überzahl an positiven Lymphknoten bestand, wurden für das Training der Klassifikationsalgorithmen alle positiven Lymphknoten im Trainingsdatensatz behalten und eine identische Anzahl an negativen Lymphknoten zufällig ausgewählt und ebenfalls behalten. Alle übrigen, negativen Lymphknoten wurden nicht in das Training des jeweiligen Algorithmus eingeschlossen, konnten aber im Training eines anderen Algorithmus wieder zufällig ausgewählt werden.

Abbildung 18: Grundlegender Ablauf der Erstellung eines balancierten Trainingsdatensatzes für das Training eines Klassifizierungsalgorithmus. Das Ziel ist es, das zahlenmäßige Übergewicht der negativen Lymphknoten vor dem Training der Algorithmen auszugleichen. So wird ein neutrales Training garantiert, damit die Orientierung der Vorhersagemodelle an den Häufigkeitsverteilungen der Response-Variablen (positiver und negativer Nodalstatus) und deren Reproduktion nicht zu einem Klassifikationserfolg über 50% führt (entspricht der Zufallswahl in binären Klassifikationssystemen). Dafür wird allen positiven Lymphknoten eine identischen, zufällig ausgewählten, Anzahl negativer Lymphknoten gegenübergestellt. Der so entstandene balancierte Lymphknoten-Datensatz kann nun für das Training eines Algorithmus verwendet werden.

3.7 Feature Selektion

Die hier grundlegende Fragestellung lautet: Welche Features ermöglichen die genauste Vorhersage des Lymphknotenstatus? Ziel ist es, die für diese Fragestellung relevantesten Features zu identifizieren, bevor sie in einen Vorhersagealgorithmus integriert werden.

Abbildung 19: Schematische Darstellung des Prozesses der Feature-Selektion. Der erste Schritt der Feature-Selektion ist die Auswahl eines geeigneten Feature-Sets, in diesem Fall das Set von PyRadiomics. Nun wird der Datensatz auf das Vorhandensein und die Ausprägung der features aus dem gewählten Feature-Set untersucht. Die für unseren Datensatz aussagekräftigsten features können nun als Feature-Subset bezeichnet werden. In einem letzen Schritt werden alle features des Subsets standardisiert und auf ihre Stabilität hin geprüft..

3.7.1 Radiomic-Feature-Set

Für die Feature-Extraktion wird das open source *Radiomic-Feature-Set* von PyRadiomics verwendet (61). Es ist online unter https://pyradiomics.readthedocs.io/en/latest/installation.html (letzter Stand 15.05.2019) zum Download frei verfügbar und als Feature-Set in die statistische Programmiersprache R integrierbar. Erstellt wurde es im Jahr 2017 von van Griethuysen et al. mit folgendem

Ziel: „um einen Referenzstandard für Radiomics-Analysen zu etablieren, eine geprüfte und gewartete Ressource anzubieten“ (Übersetzung durch den Autor) (61). Die meisten Features seien in Einklang mit der *Imaging Biomarker Standardization Initiative* (IBSI) von Zwanenburg et al. (36), welche die zweite große *radiomic feature* Standardisierungs-Initiative darstellt (61). Unterschiede sind online auf https://pyradiomics.readthedocs.io/en/latest/features.html (letzter Stand 16.05.2019) gekennzeichnet und erläutert.

Somit wird in der Arbeit folgendes sichergestellt:

1) Es wird nur mit, in der *Radiomics*-Forschungsgemeinschaft, weit verbreiteten Features gearbeitet.
2) Genaue Definitionen und Herleitungen der *Radiomic*-Features sind online frei einsehbar.
3) Die Plausibilität des Feature-Sets wird von *Radiomics*-Forschungsgemeinschaft fortlaufend extern geprüft.

Das Feature-Set von PyRadiomics (https://pyradiomics.readthedocs.io/en/latest/features.html letzter Stand 17.05.2019) enthält 120 vorprogrammierte Einzel-Features, die in 8 Hauptgruppen unterteilt werden:

1) *First-order-statistics*: In diesen Features erflogt eine statistische Analyse der Intensitäten/Grauwerte von Voxeln der region-of-interest (ROI) im CT. Zwei Beispiel-Parameter aus dieser Gruppe sind: Der maximale Voxel-Grauwert in der ROI und die Energie (gemessen als Summe aller Grauwerte einer ROI).
2) *Shape-Features (3D)*: Definierte Dreiecke formen ein Netz (englisch: mesh) um die ROI und schließen darin alle Voxel ein. Somit wird die Dreidimensionalität der ROI näherungsweise mathematisch abgebildet. Verschiedene Parameter können auch hier bestimmt werden: Anzahl der Voxel im Netz (Grauwerte werden nicht beachtet), Fläche und Volumen des Netzes, Kompaktheit, Sphärische Disproportion, Durchmesser und Flachheit.
3) *Shape-Features (2D)*: Diese Gruppe ist vergleichbar mit Gruppe 2), kommt jedoch ohne eine dritte Dimension im Raum aus.
4) *Gray Level Co-occurrence Matrix (GLCM)-Features*: Sie beschreiben das Auftreten von Pixelkombinationen mit ihren zugehörigen Grauwerten in einem Bildausschnitt. Die Häufigkeit des Auftretens definierter Pixelkombinationen wird

bestimmt. Dabei ist der Abstand der Pixel zueinander, die Grauwerte der Pixel sowie der Winkel der einzelnen Pixel zueinander von Bedeutung. Beispiele für Parameter sind: Kontrast, Auftreten von Clustern, Varianz und Homogenität.

5) *Gray Level Size Zone Matrix (GLSZM)-Features*: Hier werden die einzelnen Zonen im Bild gemessen, die einen identischen Grauwert haben. Parameter quantifizieren beispielsweise kleine Zonen, große Zonen sowie die Variabilität der Zonengrößen.
6) *Gray Level Run Length Matrix (GLRLM)-Features*: Diese Feature-Gruppe misst das Auftreten und die Länge von Voxelkombinationen mit identischen Grauwerten, die in direkter Abfolge eine zusammenhängende Reihe bilden. Gemessen wird unter anderem: Das Auftreten von langen Voxelreihen, die Ähnlichkeit und Varianz zwischen Voxel-Grauwerten in Reihen und die Varianz der Längen konsekutiver Voxelabfolgen.
7) *Neighbouring Gray Tone Difference Matrix (NGTDM)-Features*: Beschreibt die Differenz eines Voxel-Grauwerts und der durchschnittlichen Voxel-Grauwerte in der Umgebung in einem definierten Radius.
8) *Gray Level Dependence Matrix (GLDM) Features*: Sie messen die Abhängigkeit der umgebenden Pixel von einem zentralen Pixel, sowie die Häufigkeit des Auftretens dieser abhängigen Kombinationen. Gemessen wird beispielsweise der Kontrast, berechnet als Grauwertdifferenz zweier benachbarter Voxel. Die Busyness (deutsch: Geschäftigkeit) beschreibt die Häufigkeit der Grauwertwechsel benachbarter Voxel.

Alle 120 Features sind metrisch. Das bedeutet, dass sie numerisch und intervallskaliert sind. Die Features können also durch einen Zahlenwert beschrieben werden, wobei die Intervalle (Abstände) zwischen den einzelnen Skalenwerten immer gleich sind. Die Dimensionen der jeweiligen Feature-Wertebereiche sind jedoch nicht identisch. Das macht, vor deren Implementierung in den Lernalgorithmus, eine Standardisierung notwendig. Diese wird auf den nachfolgenden Seiten genau erläutert.

Viele verschiedene Features quantifizieren identische Bildphänomene und unterscheiden sich daher häufig nur geringfügig voneinander. Deshalb wurden in einem späteren Schritt der statistischen Modellierung redundante Features ausgeschlossen.

3.7.2 Feature-Selektionsmethoden

In diesem Schritt ging es darum, die aussagekräftigsten Features aus dem PyRadiomics-Set für unser Klassifikationsmodell auszuwählen. Um eine Redundanz der Features untereinander und eine spätere Überanpassung des Klassifikationsmodells zu vermeiden, legten wir uns auf eine Anzahl von 20 Features fest. Die entstandene Teilmenge der 120 Ausgangsfeatures wird Feature-Subset genannt. Den Prozess der Feature-Auswahl nennt man folglich Feature-Subset-Selection, vgl. (60, 62).

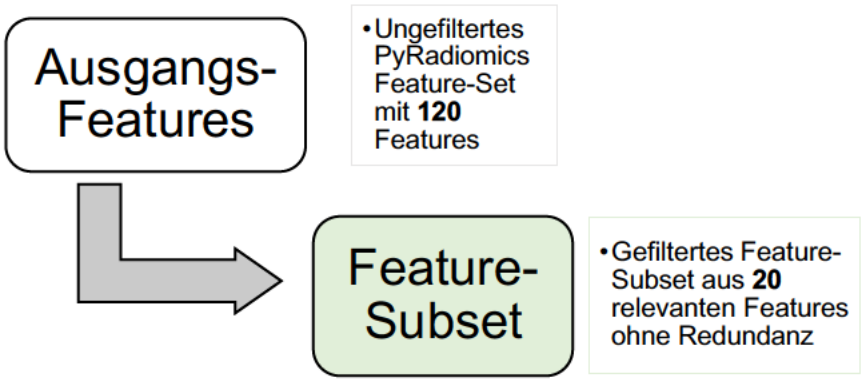


Abbildung 20: Erstellung des Feature-Subsets. Aus vielen Ausgangsfeatures werden die relevantesten Features ausgewählt und zu einem Feature-Subset zusammengefasst, welches nun eine Teilmenge der Ausgangsfeatures darstellt. Dieses Feature-Subset enthält die aussagekräftigsten Merkmale für den vorliegenden Datensatz und kann nun in einem nächsten Schritt in einen Vorhersagealgorithmus integriert werden.

Für die Feature-Subset-Selection wurden vier Filter-Ansätze ausgewählt: *Wilcoxon*, *AUC*, *Mutual Information* und *Maximum Relevance Minimum Redundancy*.

Die Feature-Selektion mit Filter-Methoden zeichnet sich dadurch aus, dass die Feature-Auswahl vor der Implementierung in den Lernalgorithmus erfolgt, vgl. (60). Deshalb ist die Feature-Auswahl unabhängig vom später verwendeten Algorithmus, die Features enthalten also nicht die inhaltlichen Voraussetzungen des Algorithmus, sondern sind allgemeingültig und auf verschiedene Modelle übertragbar.

Exkurs zu alternativen Feature-Selektionsmethoden

> Eine weitere Feature-Selektionsmethode ist der Wrapper-Ansatz. Hier erfolgt die Feature-Auswahl als integrierter Teil des Lernalgorithmus. Folglich sind die ausgewählten Features auch nur im Kontext dieses Modells gültig, ein anderes Modell hätte möglicherweise ein anderes Feature-Subset gewählt, vgl. (60).

3.7.2.1.1 Wilcoxon (wlcx)

Die Wilcoxon-Methode prüft, ob ein Feature die Lymphknotengruppen ‚positiver Nodalstatus' und ‚negativer Nodalstatus' signifikant trennen kann. Dies ist der Fall, wenn die metrischen Features zwischen beiden Gruppen große Differenzen aufweisen.

Es wird also getestet, ob zwei Gruppen, unter Betrachtung durch ein Feature, der gleichen Grundgesamtheit entsprechen oder nicht.

3.7.2.1.2 Area Under The Curve (auc)

Die AUC beschreibt die Wahrscheinlichkeit, eine positive Zufallsstichprobe mit Hilfe des untersuchten Features als positiv einzuordnen.

Bei der Feature-Selektion fungiert das Feature als Vorhersagemodell mit dem Ziel, die Verteilung der Antwort-Gruppen bestmöglich zu trennen. Alle möglichen Werte des Features mit der jeweils dazugehörigen Richtig-Positiv-Rate und Falsch-Positiv-Rate formen den Grafen, welcher auch ROC-Kurve genannt wird. Der Bereich unterhalb der ROC-Kurve ist die AUC und quantifiziert die Vorhersagegenauigkeit dieses spezifischen Features.

3.7.2.1.3 Mutual Information (mi)

Die Mutual Information gibt den statistischen Zusammenhang zweier Variablen an, indem es die Unsicherheit der unbekannten Variable bei Kenntnis der bekannten Variable reduziert.

Bei maximaler Mutual Information lässt sich Variable 2 komplett aus Variable 1 errechnen, also vorhersagen - die Unsicherheit für Variable 2 ist also minimal. Liegt keine Mutual Information vor, handelt es sich um statistisch völlig unabhängige Variablen. Für die Feature-Selektion ist Variable 1 das zu vorhersagende Feature, Variable 2 stellt ‚als vorhergesagtes Element, den Nodalstatus der Lymphknoten dar (entspricht der Antwort-Kategorie des Klassifizierungsmodells).

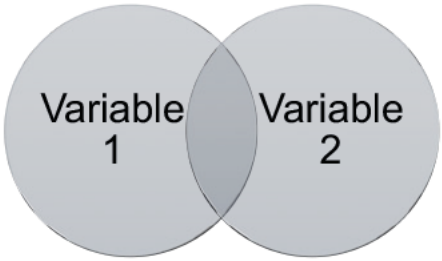


Abbildung 21: Schematische Darstellung der Mutual Information zweier Variablen. Die Wertebereiche von Variable 1 und Variable 2 werden jeweils durch einen grauen Kreis dargestellt. Ihre Mutual Information ergibt sich aus ihrer Schnittmenge, dem dunkelgrauen Bereich. In diesem Bereich haben beide Variablen identische Werte und erlauben somit eine verbesserte Bestimmung der Gesamtwerte der einen Variable aus der jeweils anderen Variable bei Kenntnis der gemeinsamen Schnittmenge.

3.7.2.1.4 Maximum Relevance Minimum Redundancy (mrmi)

Diese Methode inkludiert Features mit hoher Mutual Information mit den Antwort-Kategorien, selektiert also für die Vorhersage relevante Merkmale. Nachfolgend werden Features mit hoher Mutual Information untereinander exkludiert, um Redundanz zu vermeiden.

3.7.3 Feature-Standardisierung

Für die Standardisierung wird der Mittelwert des Features vom Feature subtrahiert. Danach wird durch die Standardabweichung dividiert. So erhalten wir Feature-Werte, die um den Nullpunkt verteilt liegen und eine Streuung von -1 bis 1 aufweisen.

Die absoluten Feature Werte sind nun direkt miteinander vergleichbar. Ohne diese Standardisierung würde ein Lernalgorithmus hohe Feature-Werte künstlich stark und niedrige Werte weniger stark für die Vorhersage gewichten, da die meisten Klassifikationsalgorithmen nur mit absoluten Zahlen operieren können.

3.7.4 Stabilitätsprüfung der Feature-Selektionsmethoden

Für die Stabilitätsprüfung werden die vier Feature-Selektionsmethoden auf zwei zufällig getrennten Hälften des Datensatzes angewandt. Das Teilen des Datensatzes und die Selektion geschehen so häufig, wie die Anzahl der gesuchten Features groß ist (20). 20 Mal wird der Datensatz also zufällig geteilt und die Selektionsmethode ermittelt jedes Mal die besten Features in beiden Hälfen. Anschließend wird bestimmt, wie stark sich die Selektionsergebnisse beider Hälften ähneln. Je größer die Ähnlichkeit der Selektionsergebnisse, desto stabiler die Selektionsmethode.

3.8 Training

Für die Klassifikation des Nodalstatus der Lymphknoten wurden 6 Algorithmen genutzt. Jeder der 6 Algorithmen wurde mit allen 4 Feature-Selektionsmethoden trainiert, sodass wir insgesamt 24 verschiedene Klassifikationsalgorithmen erhalten.

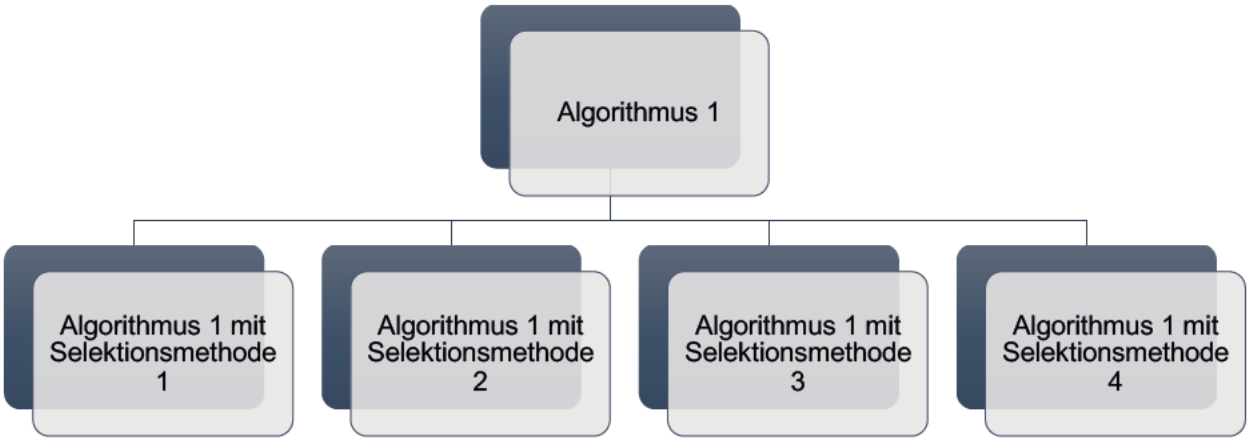


Abbildung 22: Exemplarische Darstellung der Kombination eines Algorithmus mit allen Feature-Selektionsmethoden. Für jeden Algorithmus ergeben sich 4 Varianten, da 4 Feature-Selektionsmethoden vorhanden sind. Das Schema der Grafik ist für jeden Algorithmus gleich, wiederholt sich also 6 mal, sodass wir schlussendlich 24 verschiedene Klassifikationsalgorithmen erhalten.

Alle Algorithmen stammen aus dem Bereich des *machine learning* und werden im Folgenden kurz erläutert.

3.8.1 Algorithmen

3.8.1.1 Support Vector Machines (SVM)

Die Response-Variablen der Stichprobe (hier: die positiven und negativen Lymphknoten) verteilen sich im statistischen Raum anhand ihrer jeweiligen Feature-Ausprägung. Jedes Feature stellt eine Dimension dieses Raumes dar, sodass ab einer Featureanzahl von 4 keine grafische Darstellung mehr erfolgen kann und eine rein mathematische Beschreibung erfolgt.

Das Ziel von *Support Vector Machines* ist die Teilung des Datensatzes in mindestens zwei Klassen, also eine Kategorisierung der Response-Variablen in zwei Antwort-Kategorien (hier: in positiven und negativen Nodalstatus), vgl. (60). Dies wird durch die Trennung der Response-Variablen durch eine Hyperebene realisiert. Die Hyperebene wird mit größtmöglichem Abstand (englisch: margin) zwischen den zwei Vektoren

(englisch: support vectors) der angrenzenden, am nächsten an der anderen Kategorie liegenden, Datenpunkte platziert und teilt nun die Datenpunkte in zwei Gruppen, vgl. (63).

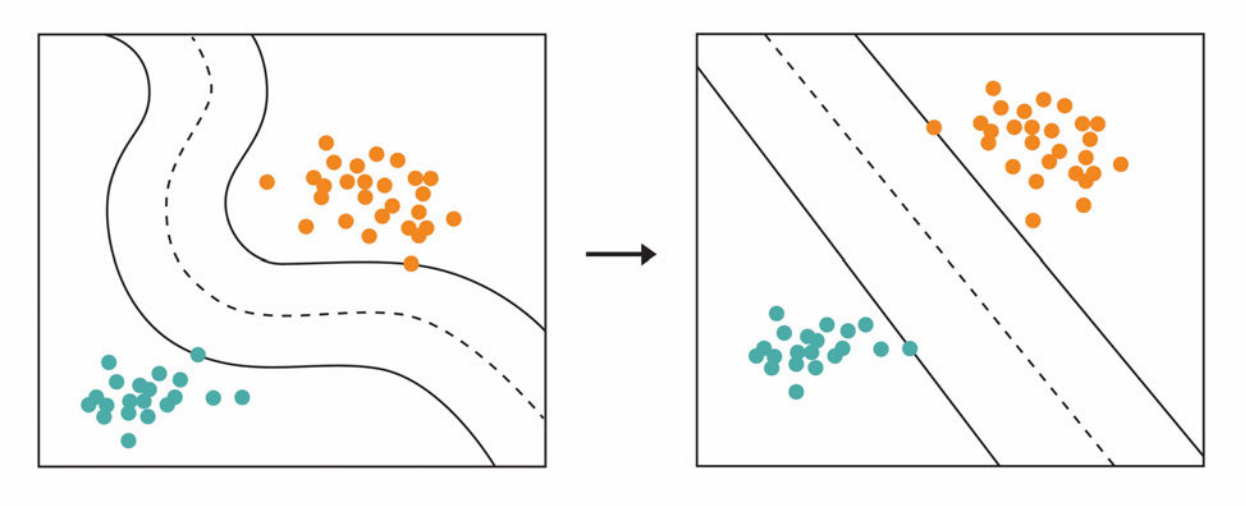


Abbildung 23: In dieser Abbildung ist das Prinzip eines Support Vector Machine-Klassifikationsalgorithmus dargestellt. Die durchgezogenen Linien entsprechen den support vectors (deutsch: Stützvektoren) und haben den maximal möglichen Abstand zu der Hyperebene (gestrichelte Linie). Die einzelnen Variablen sind durch Punkte dargestellt und werden anhand ihrer Position zur Hyperebene kategorisiert. Dabei entsteht in diesem Fall zwei Untergruppen: die Variablen in Kategorie 1 sind grün, die Variablen in Kategorie 2 sind orange gefärbt.

Neue Response-Variablen können nun anhand ihrer Lage zur Hyperebene kategorisiert werden, vgl. (60, 62, 63).

3.8.1.2 Lineare Diskriminanzanalyse (LDA)

Die LDA kreiert eine neue Achse, die mehrere Dimensionen (= *radiomic features*) zusammenführt und die Stichprobe dabei in Klassen trennt (positiv und negativ Lymphknoten), vgl. (64).

Die Formel dieser Achse (lineare Diskriminanzfunktion) ermöglicht bei Input der zugehörigen Feature-Werte eine Klassifikation der jeweiligen Daten in meist 2 Kategorien.

Für eine gute Klassifikation der Stichprobe sollte der Abstand der jeweiligen Mittelwerte der Features zueinander möglichst groß sein und die Streuung der Werte um den Mittelwert so klein wie möglich bleiben, vgl. (64).

3.8.1.3 Neural Network (NN)

Ein Neural Network besteht aus den folgenden drei Schichten: Input – Layers – Output.

Es erlaubt die Klassifikation einer Stichprobe, indem es der Feature-Ausprägung einer Stichprobe (Input) in einem Netz aus Knotenpunkten und Verbindungslinien verarbeitet (layers), vgl. (65, 66).

Die unterschiedliche Gewichtung der Verbindungslinien und der Bias (deutsch: Verzerrung) der Knotenpunkte verarbeiten die Input-Informationen und generieren ein wahrscheinliches Ergebnis (auch Output oder Outcome genannt), vgl. (60). Diese ermittelte Vorhersage kann durch eine Veränderung der Gewichtungen der Verbindungslinien und veränderten Bias-Werten, optimiert werden. Das gelingt durch ein Training des Neural Networks mit bekannten Datensets, vgl. (60, 65).

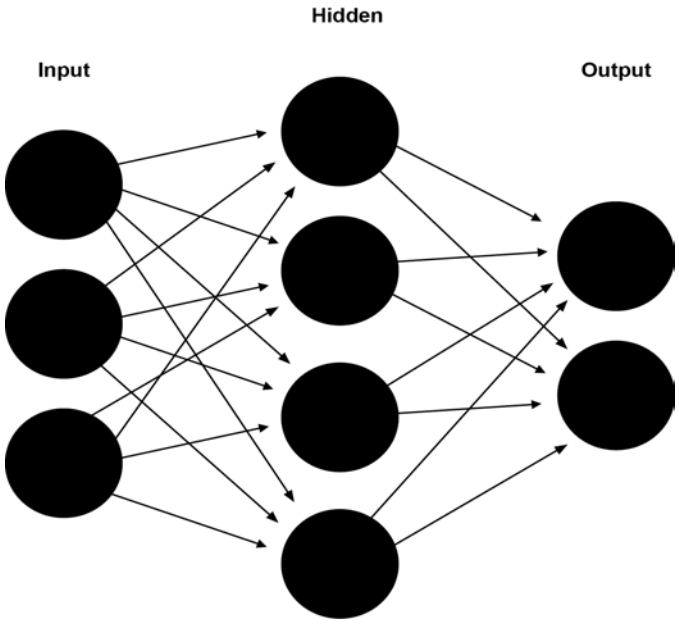


Abbildung 24 Grafische Darstellung des Prinzips eines Neural Networks mit der Dreischichtung aus Eingabe (Input) – Verborgene Schichten (Hidden) – Ausgabe (Output).

3.8.1.4 Klassifikationsbaum (KB)

Ein Klassifikationsbaum unterteilt eine Stichprobe anhand von Variablen solange in Untergruppen (dargestellt durch Knotenpunkte), bis die Klassifikation der Stichprobe optimiert ist, vgl. (64).

Variablen (=*radiomic features*), die die Stichprobe am besten klassifizieren (=in positive und negative Lymphknoten), erscheinen im Entscheidungsbaum oben und werden so zu den am stärksten gewichteten Variablen, vgl. (67).

Sobald ein nachgeschalteter Knotenpunkt keine Verbesserung der Klassifizierung aufweist, wird nicht weiter unterteilt. Die Stichprobe im letzten Knotenpunkt wird dann als Endklassifikation als sogenanntes Blatt dargestellt, vgl. (67).

3.8.1.5 Logistische Regression (Log. Reg.)

Die logistische Regression ermöglicht eine binäre Klassifizierung durch eine logistische Funktion. Dafür wird eine Funktion ausgewählt, die am ehesten alle Datenpunkte der Stichprobe mit einer Funktion abbilden kann, vgl. (68).

Diese logistische Funktion (in S-Form) ermöglicht die Berechnung jeder Wahrscheinlichkeit für das Auftreten von Wert Y (=Stichprobe) bei Wert X (=*radiomic feature*) und erreicht Werte von 0 bis 1. Alle *radiomic feature* -Werte nahe 0 veranlassen eine Klassifizierung der Stichproben in die erste Klasse, alle *radiomic feature* -Werte nahe 1 haben eine Sortierung der Stichprobe in die zweite Klasse zur Folge, wobei für Grenzfälle ein Schwellenwert von 0,5 üblich ist, vgl. (60, 68).

3.8.1.6 Partial Least Squares (PLS)

Auch der Ansatz der Partial Least Squares ermöglicht eine Klassifizierung der Stichprobe.

Die Matrizen der Wertebereiche von X (=*radiomic features*) und Y (=Stichproben) werden in ihre latenten Strukturen zerlegt (zum Beispiel als Datenpunkte in einem Koordinatensystem). Nun wird die Funktion, die die meiste Varianz in Y erklärt, extrahiert. Anschließend wird die Funktion aus X generiert, die Funktion Y am besten erklärt und zusätzlich die meiste Varianz der Datenpunkte in X abbildet, vgl. (69).

So erhält eine Stichprobe mit charakteristischen *radiomic features* einen Wert in Funktion X, welcher nachfolgend einen Wert in Funktion Y abbildet und mit diesem die Stichprobe klassifizieren kann, vgl. (69).

3.8.2 Optimierung der Klassifikation durch Bootstrapping

Hyperparameter sind die Stellschrauben eines Algorithmus. Verändert man diese, verändert sich auch das Ergebnis eines Algorithmus, wodurch eine Verbesserung der Vorhersagegenauigkeit möglich wird, vgl. (60).

Die idealen Hyperparameter-Werte unserer Algorithmen wurden mittels Bootstrapping bestimmt. Bootstrapping ist eine Form des Resamplings (deutsch:

Stichprobenwiederholung), bei der eine Zufallswahl einzelner Elemente der Stichprobe erfolgt. Innerhalb dieser Zufallsauswahl werden weitere statistische Parameter errechnet. Die daraus entstehende statistische Verteilung dieser Parameter sagt nun etwas über die zu erwartende Verteilung der Grundgesamtheit aus, welcher die Stichprobe entstammt, vgl. (60).

Bootstrapping ermöglicht also Aussagen über die Verteilung der Grundgesamtheit. Dies ist besonders hilfreich, wenn die Verteilung der Grundgesamtheit unbekannt ist und man ein möglichst allgemeingültiges Vorhersagemodell, passend für die Grundgesamtheit, entwickeln möchte.

3.9 Testing

Im letzten Schritt wurden alle 26 Klassifikationsalgorithmen auf dem unabhängigen Testdatensatz (Erläuterung siehe Kapitel *Material und Methoden des Klassifizierungsalgorithmus*, Absatz *Unterteilung und Balancierung des Datensatzes*) getestet. Ermittelt wurden folgende statistischen Leistungsparameter:

1) Treffergenauigkeit

2) Sensitivität

3) Spezifität

4) Positiver Prädiktiver Wert

5) Negativer Prädiktiver Wert

4 Material und Methoden der Klassifikation der Lymphknoten durch Radiologen und Vergleich der Klassifikationsleistungen

4.1 Kategorien

Anschließend klassifizierten zwei Radiologen der radiologischen Abteilung des Virchow Klinikums der Charité in Berlin die Lymphknoten des Testdatensatzes nach der Wahrscheinlichkeit des Vorliegens eines positiven Nodalstatus.

Dafür ordneten sie die Wahrscheinlichkeit für das Vorliegen eines positiven Nodalstatus in jeweils eine der vier möglichen Kategorien ein:

Kategorie 0	sicher nicht
Kategorie 1	wahrscheinlich nicht
Kategorie 2	wahrscheinlich
Kategorie 3	sicher

Abbildung 25: Dargestellt sind die vier Kategorien, in die die Radiologen die Lymphknoten je nach Wahrscheinlichkeit für das Vorliegen eines positiven Nodalstatus einordneten. Dabei zeigt Kategorie 0 eine als sicher eingeschätzte Benignität an, Kategorie 3 hingegen eine als sicher eingeschätzte Malignität des befundeten Lymphknotens.

Da wir in den Klassifikationsergebnissen der Radiologen auch die Fälle abbilden wollen, in denen die Radiologen unsicher waren, erweitern wir die binäre Klassifikation um zwei mittlere Kategorien. So ist es möglich, neben einem direkten Vergleich mit den Klassifikationsalgorithmen, die Ergebnisse auch auf einen möglichen Zusatznutzen der Algorithmen in unsicheren Klassifikationskategorien der Radiologen zu untersuchen.

4.2 Vergleich der Klassifikationsleistungen der Algorithmen und der Radiologen

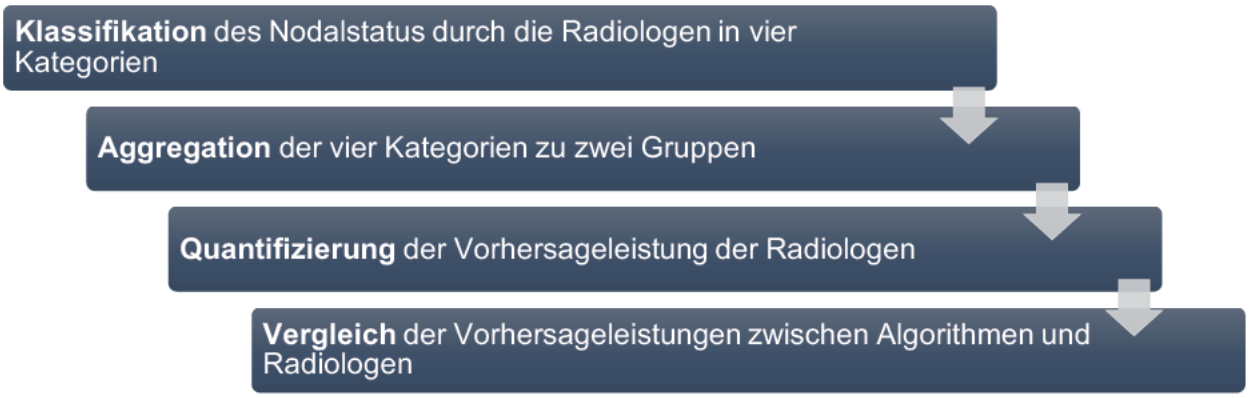


Abbildung 26: Ablauf von der Lymphknotenklassifikation durch die Radiologen bis zum Vergleich der Vorhersageleistungen mit denen der Klassifikationsalgorithmen. Nach der Vorhersage des Nodalstatus der Radiologen werden die vier Gruppen zu zweien aggregiert. Entstanden ist eine binäre Vorhersage, welche wir mit den Vorhersageleistungen der Algorithmen verlgeichen können.

Da wir die Ergebnisse der entwickelten Vorhersagemodelle mit den Ergebnissen der Radiologen vergleichen wollen, werden die vier Outcome-Kategorien zu zwei Gruppen zusammengeführt. Die Kategorien 0-1 werden zur Gruppe 0, die Kategorien 2-3 zur Gruppe 1 aggregiert.

Somit wird die Lymphknoten-Klassifizierung durch die Radiologen ebenfalls binär und kann anhand der vorliegenden Nodalstatus bewertet werden.

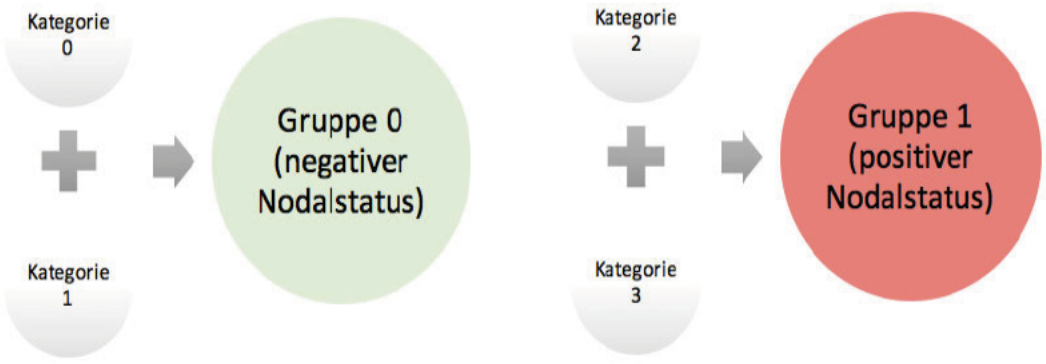


Abbildung 27: Aggregation der vier Klassifikationskategorien der Radiologen. Die Kategorien 0 und 1 (sicher nicht, wahrscheinlich nicht) werden zur Gruppe 0 zusammengeführt (negativer Nodalstatus). Die Kategorien 2 und 3 (wahrscheinlich, sicher) werden als Gruppe 1 zusammengefasst (positiver Nodalstatus). Nun sind die Vorhersagen der Radiologen in zwei Klassifikationskategorien zusammengefasst, die somit binär wurde.

Die Genauigkeit der Vorhersage wird nun durch die gleichen statistischen Parameter wie beim Testing der Vorhersagemodelle quantifiziert und mit der Leistung der Modelle verglichen:

1) Treffergenauigkeit

2) Sensitivität

3) Spezifität

4) Positiver Prädiktiver Wert

5) Negativer Prädiktiver Wert

4.3 Zusatznutzen der Vorhersagemodelle in unsicheren Kategorien der Radiologen

4.3.1 Integration der Algorithmen in die Vorhersage der Radiologen

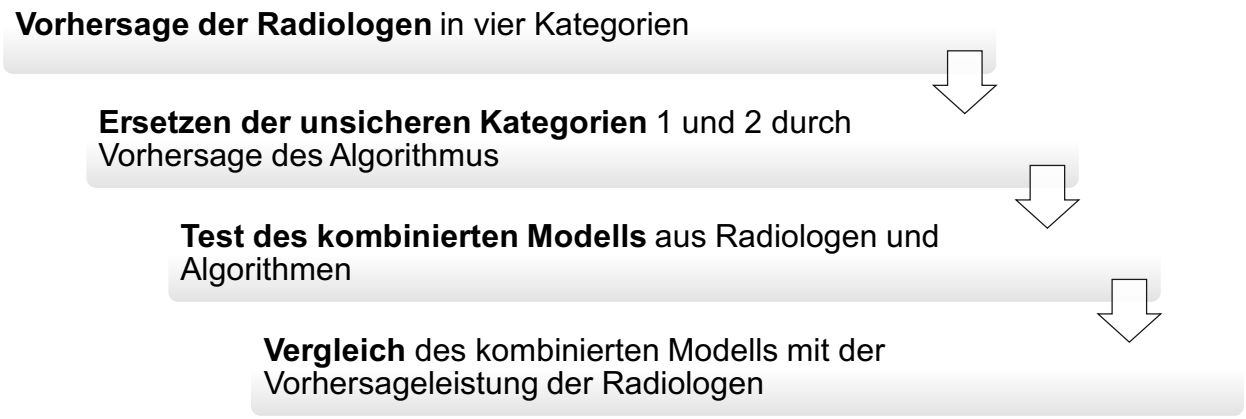


Abbildung 28: Integration der Algorithmen in die Vorhersage der Radiologen. Um einen Zusatznutzen der Vorhersagealgorithmen in unsicheren Kategorien der Radiologen zu überprüfen, werden die unsicheren Klassifikations-Kategorien 1 und 2 der Radiologen durch die binäre Vorhersageleistung der Algorithmen ersetzt. Anschließen werden die Vorhersageleistungen der kombinierten Modelle für unsichere Kategorien getestet und die ermittelten Klassifikationsleistungen mit der alleinigen Vorhersageleistung der Radiologen verglichen.

Als unsichere Kategorien bezeichnen wir Kategorie 1 (Lymphknoten wahrscheinlich nicht positiv) und Kategorie 2 (Lymphknoten wahrscheinlich positiv), als sichere Kategorien 0 (Lymphknoten sicher nicht positiv) und 3 (Lymphknoten sicher positiv).

Wir testen nun die Klassifizierungsleistung der Radiologen in ihren sicheren Kategorien, ersetzen aber die unsicheren Kategorien mit der Klassifizierung durch die Algorithmen für diese Fälle. So entsteht ein drittes, integriertes Vorhersagemodell.

4.3.2 Vergleich der integrierten mit der ursprünglichen Klassifizierungsleistung der Radiologen

Anschließend vergleichen wir die Klassifizierungsleistung des integrierten Vorhersagemodells im Hinblick auf einen möglichen Mehrwert gegenüber der alleinigen Vorhersage der Radiologen bei unsicheren Fällen.

4.4 Testung auf signifikante Unterschiede zwischen den Klassifizierungsleistungen der Algorithmen und der Radiologen in unsicheren Kategorien

Abschließend prüfen wir, ob es signifikante Unterschieden zwischen der Vorhersageleistung der Algorithmen und der Vorhersageleistung der Radiologen in unsicheren Kategorien gibt. Es werden dafür alle fünf ermittelten Leistungsparameter verglichen. Die Testung auf signifikante Unterschiede erfolgt mittels logistischer Regressionsmodelle.

5 Ergebnisse

5.1 Leistungsparameter der Klassifikationsalgorithmen

Nach Anwendung aller 24 Klassifikatoren auf den Testdatensatz erhalten wir folgende Leistungsparameter:

Nummer	Klassifikator	Feature Selektions Methode	Genauigkeit	Sensitivität	Spezifität	Positiver Prädiktiver Wert	Negativer Prädiktiver Wert
1	LDA	WLCX	0,78	0,74	0,79	0,56	0,9
2	LDA	AUC	0,77	0,73	0,79	0,55	0,89
3	LDA	MI	0,79	0,7	0,82	0,58	0,88
4	LDA	MRMI	0,77	0,73	0,79	0,55	0,89
5	Log. Reg.	WLCX	0,78	0,7	0,82	0,57	0,88
6	Log. Reg.	AUC	0,78	0,7	0,82	0,57	0,88
7	Log. Reg.	MI	0,78	0,68	0,81	0,56	0,88
8	Log. Reg.	MRMI	0,75	0,71	0,76	0,52	0,88
9	PLS	WLCX	0,78	0,74	0,79	0,56	0,9
10	PLS	AUC	0,77	0,71	0,79	0,55	0,88
11	PLS	MI	0,79	0,73	0,81	0,58	0,89
12	PLS	MRMI	0,78	0,73	0,8	0,56	0,89
13	SVM	WLCX	0,78	0,7	0,82	0,57	0,88
14	SVM	AUC	0,76	0,7	0,78	0,53	0,88
15	SVM	MI	0,76	0,68	0,79	0,54	0,87
16	SVM	MRMI	0,76	0,68	0,79	0,54	0,87
17	NN	WLCX	0,78	0,7	0,8	0,56	0,88
18	NN	AUC	0,77	0,68	0,8	0,55	0,88
19	NN	MI	0,76	0,74	0,76	0,53	0,89
20	NN	MRMI	0,8	0,67	0,85	0,62	0,88
21	KB	WLCX	0,76	0,79	0,74	0,53	0,91
22	KB	AUC	0,78	0,7	0,8	0,56	0,88
23	KB	MI	0,78	0,73	0,79	0,56	0,89
24	KB	MRMI	0,77	0,71	0,79	0,55	0,88

Abbildung 29: Leistungsparameter der Klassifikatoren, die durch Anwendung auf den Testdatensatz entstanden sind. Die Spalte Klassifikator kennzeichnet den verwendeten Algorithmus. Die jeweils kombinierte Feature-Selektionsmethode findet sich in der Spalte daneben. Abkürzungen: LDA (Lineare Diskriminanzanalyse), Log. Reg. (Logistische Regression), PLS (Partial Least Squares), SVM (Support Vecotor Machines), NN (Neural Network), KB (Klassifikationsbaum). WLCX (Wilxocon), AUC (Area Under The Curve), MI (Mutual Information), MRMI (Maximum Relevance Minimum Redundancy).

In der Tabelle sind für alle 24 Variationen der 6 Algorithmen und 4 Feature-Selektionsmethoden alle Leistungsparameter gelistet.

Alle 24 Klassifikatoren weisen eine vergleichbare Leistung auf.

Leitungsparameter	**Arithmetisches Mittel**	**Spannweite**
Treffergenauigkeit	0,77	0,75-0,80
Sensitivität	0,71	0,67-0,79
Spezifität	0,80	0,74-0,85
Positiver Prädiktiver Wert	0,56	0,52-0,62
Negativer Prädiktiver Wert	0,88	0,87-0,91

Abbildung 30: Übersicht der gemittelten Leistungsparameter aller Klassifikationsalgorithmen und der dazugehörigen Spannweiten.

Die Treffergenauigkeit (englisch: accuracy) beschreibt den Anteil korrekt klassifizierter Variablen von allen klassifizierten Variablen und liegt zwischen 0,75-0,8 (Arithmetisches Mittel: 0,77). Damit liegen die Klassifikatoren in 75-80% der Fälle mit ihrer Vorhersage des Nodalstatus richtig.

Die Sensitivität ist die Anzahl der Richtig-Positiv klassifizierten Variablen, geteilt durch die Anzahl aller positiven Variablen. Hier erreichten die Klassifikatoren Werte von 0,67-0,79 (Arithmetisches Mittel: 0,71). Die höchste Sensitivität mit 79% erreichte der Algorithmus *Klassifikationsbaum* in Kombination mit der Feature-Selektionsmethode nach *Wilcoxon*.

Die Spezifität beschreibt die Anzahl aller Richtig-Negativ klassifizierten Variablen, geteilt durch die Anzahl aller negativen Variablen. Die Klassifikatoren erzielten Spezifitäten von 0,74-0,85 (Arithmetisches Mittel: 0,8), wobei die Kombination aus einem *Neural Network* und der Feature-Selektionsmethode *Maximum Relevance Minimum Redundancy* mit 85% die höchste Spezifität aufwies.

Als Positiver Prädiktiver Wert bezeichnet man die Wahrscheinlichkeit, dass eine Variable bei positivem Testergebnis tatsächlich positiv ist. Hier erreichen die Klassifikatoren Werte von 0,52-0,62 (Arithmetisches Mittel: 0,57).

Der Negative Prädiktive Wert beschreibt die Wahrscheinlichkeit, dass eine Variable bei negativem Testergebnis tatsächlich negativ ist. Hier werden Werte von 0,87-0,91 erzielt (Arithmetisches Mittel: 0,88). Somit sind in bis zu 91% der Fälle negativ klassifizierte Lymphknoten tatsächlich negativ. Dieser Wert wird von dem Algorithmus *Klassifikationsbaum* in Kombination mit der Feature-Selektionsmethode nach *Wilcoxon* erreicht.

Um eine bessere Vergleichbarkeit aller Klassifikatoren zu ermöglichen, stellen wir die Leitungsparameter grafisch dar:

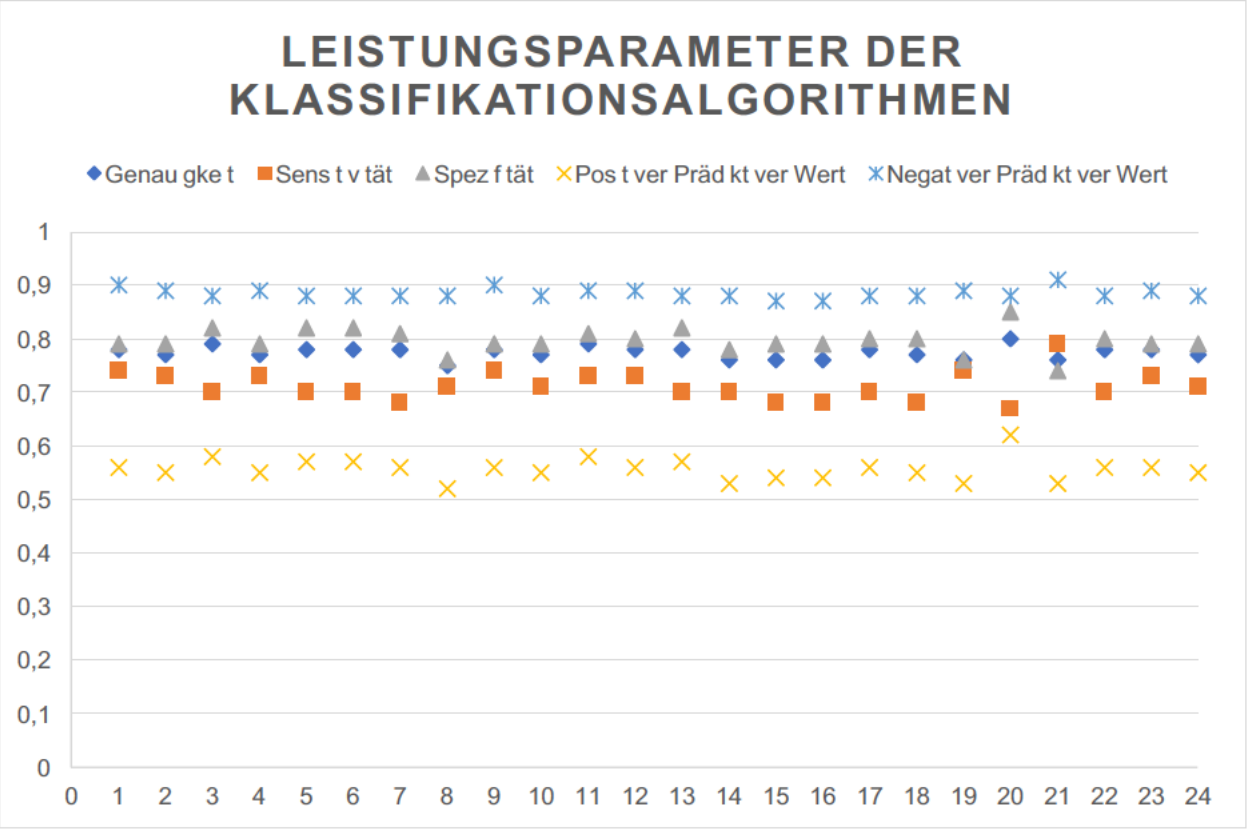


Abbildung 31: Visualisierte Leistungswerte der Klassifikationsalgorithmen. Auf der Ordinate sind die Kennnummern der jeweiligen Klassifikationsalgorithmen mit entsprechender Feature-Selektionsmethode aufgelistet, auf der Abszisse finden sich die dazugehörigen Leistungswerte je nach Test-Kriterium. Da alle Leistungsparameter unterschiedlicher Klassifikatoren für jedes Test-Kriterium nahezu auf einer Horizontalen liegen, bestätigt sich auch visuell der Eindruck, dass alle Klassifikationsalgorithmen eine vergleichbare Leistung zeigen.

5.2 Einzelne *radiomic features* von hohem prognostischen Wert

Die aussagekräftigsten *radiomic features* werden durch die Betrachtung der Klassifikationsbäume nach Beendigung des Trainings ermittelt. Die aussagekräftigsten

Features stellen die ersten Ebenen der Klassifikationsbäume dar, da sie die Stichprobe am besten kategorisieren können.

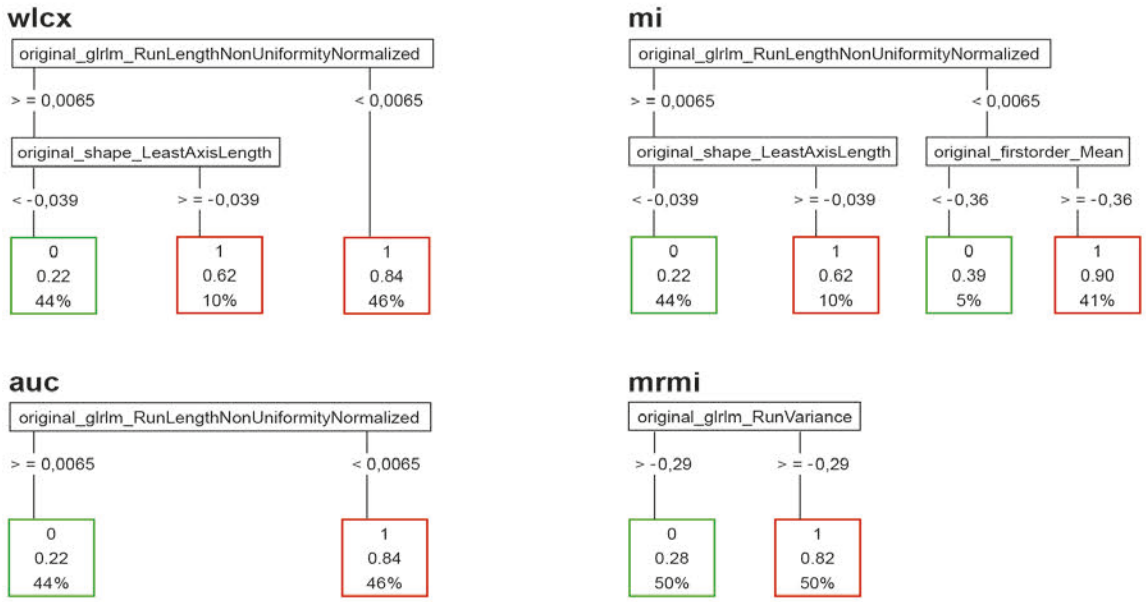


Abbildung 32: Die Klassifikationsbäume nach Beendigung des Trainings. Den vier Klassifikationsbäumen gehen jeweils andere Feature-Selektionsmethoden voraus: WLCX (Wilxocon), AUC (Area Under The Curve), MI (Mutual Information), MRMI (Maximum Relevance Minimum Redundancy). Das Feature, welches die Stichprobe kategorisiert wird in dem weißen Kasten benannt. Die Zahlen darunter geben den Schwellenwert des Features an, bei dem die Einteilung in eine der zwei Kategorien erfolgt. Erfolgt keine weitere Unterteilung enthält der letzte Knoten die final kategorisierten Stichproben. Die erste Zahl des Knotens beschreibt die binäre Klassifizierung des Lymphknotens (0=negativ, 1=positiv). Die zweite Zahl beschreibt die Wahrscheinlichkeit, dass die Lymphknoten, die in diesen Knoten unterteilt wurden, tatsächlich positiv sind. Der dritte Wert gibt die prozentualen Anteile der Stichprobe in den einzelnen Untergruppen an.

Bei den ersten drei Varianten des Klassifikationsbaums hat das *radiomic feature Run Length Non-Uniformity Normalized* die beste Kategorisierungsleistung. Diese Feature misst die Ähnlichkeit der Längen von Voxelreihen mit identischen Grauwerten innerhalb des segmentierten Lymphknotens, wobei ein niedriger Wert eine große Ähnlichkeit der Länge der Voxelabfolgen beschreibt. Ein hoher Wert quantifiziert ein großes Maß an Unterschiedlichkeit der einzelnen Längen der Voxelreihen.

Hier ein konkretes Beispiel für den Klassifikationsbaum mit Wilcoxon (WLCX): Hat das Feature *Run Length Non-Uniformity Normalized* einen Wert kleiner als 0,0065, liegt die Wahrscheinlichkeit für das korrekte Klassifizieren eines Lymphknotens als positiv bei 84%.

Hier ein konkretes Beispiel für den Klassifikationsbaum mit Maximum Relevance Minimum Redundancy (MRMI): Hat das Feature *Run Variance* einen Wert unter -0,29, liegt die Wahrscheinlichkeit, dass die Klassifikation eines Lymphknotens als negativ falsch ist, bei 28%.

Bei der Kombination aus der Feature-Selektionsmethode MRMI mit einem Klassifikationsbaum, ist das Feature *Run Variance* am aussagekräftigsten. Es misst die Varianz der Länge einzelner Voxelreihen innerhalb des segmentierten Lymphknotens, ein hoher Wert kennzeichnet hier eine starke Varianz.

Beide Features entstammen der Gruppe der *Gray Level Run Length Matrix* (GLRLM)-Features, welche im Kapitel *Material und Methoden des Klassifizierungsalgorithmus*, im Absatz *Radiomic-Feature-Set* erläutert wurde.

Ein weiteres *radiomic feature*, das in zwei Klassifikationsbäumen (KB mit WLCX, KB mit MI) eine Unterteilung der Stichprobe ermöglicht, ist die *Least Axis Length* (deutsch: Länge der kleinsten Achse) aus der Gruppe der *Shape Features (3D)*. Hier gilt: je größer der Wert des Features, desto länger die kürzeste Achse des segmentierten Lymphknotens. Der Schwellenwert, ab dem ein Lymphknoten als positiv klassifiziert wird liegt in beiden Klassifikationsbäumen bei einem Wert größer, gleich -0,039.

Das *radiomic feature First Oder Mean* aus der Gruppe der *First Order Features* kann im Klassifikationsbaum KB mit MI die Lymphknoten weiter klassifizieren. Das Feature berechnet die durchschnittliche Grauwertintensität aller Voxel des segmentierten Bereichs. Hier liegt der Schwellenwert für die Klassifikation eines Lymphknotens als positiv bei einer Grauwertintensität von größer, gleich -0.36.

5.3 Die Klassifikationsleistungen der Radiologen

Nach der Aggregation der vier Klassifikations-Kategorien der Radiologen zu zwei Gruppen entstand eine binäre Klassifikation der Nodalstatus der Lymphknoten (siehe Kapitel *Material und Methoden des Klassifizierungsalgorithmus*, Absatz *Vergleich der Klassifikationsergebnisse*).

Für die binäre Klassifikation der Lymphknoten des Testdatensatzes durch die Radiologen ergeben sich folgende Leistungsparameter:

Befunder	Treffergenauigkeit	Sensitivität	Spezifität	Positiver Prädiktiver Wert	Negativer Prädiktiver Wert
Radiologe 1	0,8	0,48	0,92	0,68	0,83
Radiologe 2	0,8	0,5	0,96	0,83	0,84

Abbildung 33: Leistungsparameter der Radiologen für die Klassifizierung des Nodalstatus eines Lymphknotens im Testdatensatz. Die Leistungsparameter entstehen nach Aggregation der vier Vorhersagekategorien der Radiologen zu insgesamt zwei Kategorien. Es entsteht also ein binäres Klassifikationssystem, das den Nodalstatus entweder als positiv oder negativ klassifiziert.

5.4 Zusatznutzen der integrierten Vorhersagemodelle in unsicheren Kategorien der Radiologen

Die folgende Abbildung enthält alle Leistungsparameter der integrierten Vorhersagemodelle des ersten Radiologen:

Radiologe	Klassifikator	Feature Selektions Methode	Genauigkeit	Sensitivität	Spezifität	Positiver Prädiktiver Wert	Negativer Prädiktiver Wert
Rad 1	LDA	WLCX	0,79	0,71	0,82	0,59	0,89
Rad 1	LDA	AUC	0,79	0,7	0,82	0,58	0,88
Rad 1	LDA	MI	0,8	0,7	0,84	0,61	0,89
Rad 1	LDA	MRMI	0,79	0,7	0,82	0,58	0,88
Rad 1	Log Reg	WLCX	0,8	0,68	0,84	0,6	0,88
Rad 1	Log Reg	AUC	0,8	0,68	0,84	0,6	0,88
Rad 1	Log Reg	MI	0,79	0,68	0,83	0,58	0,88
Rad 1	Log Reg	MRMI	0,77	0,7	0,79	0,55	0,88
Rad 1	PLS	WLCX	0,79	0,71	0,82	0,59	0,89
Rad 1	PLS	AUC	0,79	0,7	0,83	0,59	0,88
Rad 1	PLS	MI	0,8	0,71	0,84	0,61	0,89
Rad 1	PLS	MRMI	0,79	0,7	0,83	0,59	0,88
Rad 1	SVM	WLCX	0,8	0,7	0,84	0,61	0,89
Rad 1	SVM	AUC	0,78	0,7	0,82	0,57	0,88
Rad 1	SVM	MI	0,78	0,68	0,82	0,57	0,88
Rad 1	SVM	MRMI	0,78	0,68	0,82	0,57	0,88
Rad 1	NN	WLCX	0,79	0,68	0,83	0,59	0,88
Rad 1	NN	AUC	0,79	0,68	0,83	0,59	0,88
Rad 1	NN	MI	0,77	0,73	0,79	0,55	0,89
Rad 1	NN	MRMI	0,83	0,67	0,89	0,68	0,88
Rad 1	KB	WLCX	0,78	0,76	0,79	0,57	0,9
Rad 1	KB	AUC	0,8	0,68	0,84	0,61	0,88
Rad 1	KB	MI	0,79	0,71	0,82	0,59	0,89
Rad 1	KB	MRMI	0,79	0,7	0,83	0,59	0,88

Abbildung 34: Leistungsparameter der integrierten Vorhersagemodelle. Abkürzungen: Rad1 (Radiologe 1), LDA (Lineare Diskriminanzanalyse), Log. Reg. (Logistische Regression), PLS (Partial Least Squares), SVM (Support Vecotor Machines), NN (Neural Network), KB (Klassifikationsbaum). WLCX (Wilxocon), AUC (Area Under The Curve), MI (Mutual Information), MRMI (Maximum Relevance Minimum Redundancy).

Die nachfolgende Abbildung enthält alle Leistungsparameter der integrierten Vorhersage des zweiten Radiologen:

Radiologe	Klassifikator	Feature Selektions Methode	Genauigkeit	Sensitivität	Spezifität	Positiver Prädiktiver Wert	Negativer Prädiktiver Wert
Rad 2	LDA	WLCX	0,8	0,7	0,83	0,6	0,88
Rad 2	LDA	AUC	0,79	0,7	0,82	0,58	0,88
Rad 2	LDA	MI	0,8	0,67	0,85	0,62	0,88
Rad 2	LDA	MRMI	0,8	0,68	0,84	0,61	0,88
Rad 2	Log Reg	WLCX	0,8	0,67	0,85	0,61	0,88
Rad 2	Log Reg	AUC	0,8	0,67	0,85	0,61	0,88
Rad 2	Log Reg	MI	0,81	0,65	0,86	0,63	0,87
Rad 2	Log Reg	MRMI	0,79	0,68	0,83	0,58	0,88
Rad 2	PLS	WLCX	0,79	0,7	0,83	0,59	0,88
Rad 2	PLS	AUC	0,8	0,68	0,84	0,61	0,88
Rad 2	PLS	MI	0,81	0,68	0,86	0,63	0,88
Rad 2	PLS	MRMI	0,8	0,68	0,85	0,62	0,88
Rad 2	SVM	WLCX	0,81	0,67	0,86	0,64	0,88
Rad 2	SVM	AUC	0,79	0,67	0,83	0,59	0,87
Rad 2	SVM	MI	0,8	0,65	0,85	0,61	0,87
Rad 2	SVM	MRMI	0,8	0,65	0,85	0,61	0,87
Rad 2	NN	WLCX	0,8	0,67	0,84	0,6	0,88
Rad 2	NN	AUC	0,8	0,65	0,85	0,61	0,87
Rad 2	NN	MI	0,79	0,68	0,83	0,59	0,88
Rad 2	NN	MRMI	0,82	0,64	0,88	0,66	0,87
Rad 2	KB	WLCX	0,81	0,71	0,84	0,62	0,89
Rad 2	KB	AUC	0,82	0,67	0,88	0,67	0,88
Rad 2	KB	MI	0,81	0,65	0,86	0,63	0,87
Rad 2	KB	MRMI	0,81	0,68	0,85	0,62	0,88

Abbildung 35: Leistungsparameter der integrierten Vorhersagemodelle. Abkürzungen: Rad2 (Radiologe 2), LDA (Lineare Diskriminanzanalyse), Log. Reg. (Logistische Regression), PLS (Partial Least Squares), SVM (Support Vecotor Machines), NN (Neural Network), KB (Klassifikationsbaum). WLCX (Wilxocon), AUC (Area Under The Curve), MI (Mutual Information), MRMI (Maximum Relevance Minimum Redundancy).

Bei der Integration der Vorhersage der Algorithmen in den unsicheren Kategorien der Radiologen liegt die Gesamtsensitivität bei 64-71%, bei einem Mittelwert von 69%. Die Gesamtspezifität der Vorhersage liegt bei 79-89%, bei einem Mittelwert von 84%. Der Mittelwert der Treffergenauigkeit liegt bei 80%.

5.5 Testung auf signifikante Unterschiede zwischen den Klassifizierungsleistungen der Algorithmen und der Klassifizierungsleistung der Radiologen in unsicheren Kategorien

Die Testung auf signifikante Unterschiede zwischen den Leistungsparametern der Algorithmen und denen der Radiologen in unsicheren Kategorien mittels logistischer Regression, ergab folgende Ergebnisse:

1) Alle 24 Klassifikatoren sind im Vergleich zu Radiologe 1, als auch zu Radiologe 2 signifikant sensitiver.
2) Radiologe 1 hat 23 Klassifikatoren gegenüber eine signifikant höhere Spezifität. Bei dem Klassifikator, der aus einem *Neural Network* in Kombination mit der Feature-Selektionsmethode *Maximum Relevance Minimum Redundancy* besteht, ist der Unterschied nicht signifikant.
3) Radiologe 2 hat 22 Klassifikatoren gegenüber eine signifikant höhere Spezifität. Bei dem Klassifikator, der aus einem *Neural Network* in Kombination mit der Feature-Selektionsmethode *Maximum Relevance Minimum Redundancy* besteht, ist der Unterschied nicht signifikant. Dies ist auch für den Klassifikator, der aus einem *Entscheidungsbaum* und der Feature-Selektionsmethode *AUC* besteht, der Fall.
4) Die Positiven und Negativen Prädiktiven Werte beider Klassifizierungsvarianten unterscheiden sich nicht signifikant.

Die tabellarischen Auflistungen der zugrundeliegenden Leistungswerte sowie eine grafische Darstellung der Test-Ergebnisse finden sich im Anhang, S*tatistische Modellierung mit Quellcode*.

6 Diskussion

6.1 Bewertung und Einordnung der Klassifikationsleistungen

6.1.1 Allgemeine Bewertung der Klassifikationsleistungen der Algorithmen

Die Leistungswerte der Klassifikatoren zeigen, dass es gelungen ist, reproduzierbare Klassifikationsmodelle anhand von *radiomic features* zu erstellen.

Alle binären Klassifikationsalgorithmen kommen zu ähnlichen Ergebnissen, kein Algorithmus scheint eine Sonderstellung einzunehmen. Es ist daher davon auszugehen, dass hier viele verschiedene Methoden das identische Phänomen messen: den Informationsgehalt in CT-Scans, quantifiziert durch *radiomic features*.

Die Studie zeigt, dass die Klassifikation von Lymphknoten anhand von *radiomic feature*s grundsätzlich möglich ist und in Zukunft eine sinnvolle Erweiterung der Ausbreitungs- und Rezidivdiagnostik des Lungenkarzinoms darstellen könnte. Nach verbesserter Klassifikationsleistung könnte ein valider *Radiomics*-Algorithmus das prä-bioptische PET-CT sogar ersetzen, und dem Patienten somit eine komplexe und fehleranfällige Untersuchung ersparen.

In Situationen, in denen die PET kontraindiziert oder nicht verfügbar ist, könnten die *Radiomic*s-Klassifikatoren schon heute, vor der Lymphknotenbiopsie, orientierende Bild-Informationen bereitstellen, die zuvor im CT verborgen geblieben sind.

6.1.2 Bewertung der Klassifikationsleitungen der Einzelfeatures

Ein großer Vorteil der Klassifizierungsalgorithmen ist die Möglichkeit, aus wenig aussagekräftigen Einzelfeatures Feature-Untergruppen zu bilden, die in Kombination eine höhere Aussagekraft haben. Zusätzlich können durch die Bildung eines Feature-Subsets die Beziehungen der *radiomic features* untereinander abgebildet werden.

Solange die Befundung von CT-Aufnahmen allerdings allein durch den Menschen erfolgt, ist eine Reduktion der zu betrachtenden Bildparameter auf ein Minimum notwendig. Die simultane Analyse verschiedener Bildparameter, quantifiziert durch *radiomic features* in Vorhersagealgorithmen, kann in Zukunft eine Abkehr von diesem singulären Ansatz ermöglichen.

Neben den Ergebnissen der Klassifikationsalgorithmen ist auch die Betrachtung einzelner *radiomic features* sinnvoll, da sie alte Vorstellungen überprüfen und neue Wege

in der Krebsdiagnostik aufzeigen kann. Im Folgenden sind die aussagekräftigsten Features aus dem Algorithmus *Klassifikationsbaum* unter diesen Gesichtspunkten aufgeführt. Alle Features entstammen dem *Radiomic*-Feature-Set von PyRadiomics (https://pyradiomics.readthedocs.io/en/latest/features.html#, letzter Stand: 18.06.2019).

6.1.2.1 Quantifizierte Heterogenität

Aktuell werden im CT nur wenige Merkmale, die die Tumorheterogenität abbilden, durch menschliche Befunder in der Diagnostik genutzt (33, 34). Eine Ursache dafür ist die Komplexität vieler Bildmerkmale, welche der menschliche Befunder nur unzureichend wahrnehmen und errechnen kann (Beispielsweise: Die Varianz der Länge verschiedener Voxelreihen innerhalb eines definierten Bereichs, dargestellt durch das Feature *Run Variance*).

Außerdem sind aktuell verwendete radiologische Begriffe meist qualitativer („heterogen“, „irregulär“ (40) (Übersetzung durch den Autor)) und semi-quantitativer Natur („leicht“, „mäßig“ (34) (Übersetzung durch den Autor)). Für eine standardisierte Vergleichbarkeit und Reproduzierbarkeit sind aber vorrangig quantitative Merkmale notwendig, vgl. (33, 35, 40). Diese Quantifizierung gelingt teilweise mit den im Folgenden angeführten *radiomic features*.

6.1.2.1.1 Run Variance

Da das Feature *Run Variance* die Varianz der Länge einzelner Voxelreihen innerhalb des segmentierten Lymphknotens misst, quantifiziert es einen Aspekt der Tumorheterogenität. Je höher der Wert dieses Features ist, desto größer ist die Varianz innerhalb des Lymphknotens, desto größer die Heterogenität der Längen der Voxelreihen.

Es ist also gelungen, ein *radiomic feature* (im Klassifikator KB mit MRMI) zu finden, dass den Faktor Tumorheterogenität quantifiziert und einen Lymphknoten über einem definierten Schwellenwert von -0,29 mit einer Wahrscheinlichkeit von 82% richtigerweise als positiv klassifiziert.

6.1.2.2 Quantifizierung der Lymphknoten-Durchmesser

6.1.2.2.1 Least Axis Length

Das *radiomic feature Least Axis Length* quantifiziert ein bereits bekanntes Merkmal: Die räumliche Dimension eines Lymphknotens. Hier wird jedoch nicht der maximale

Durchmesser gemessen, sondern die Länge der kleinsten Achse des Lymphknotens. Beide Messvarianten sind üblich, teilweise erfolgt keine genaue Unterscheidung, vgl. (3, 12).

Beide Ansätze des Ausmaßes des Lymphknotendurchmessers sind theoretisch vergleichbar: Je größer ein Lymphknoten im dreidimensionalen Raum ist, desto höher ist die Wahrscheinlichkeit für einen metastatischen Befall (12). In der praktischen Umsetzung könnten sich die Ansätze jedoch unterscheiden, da in einigen Verfahren der Fokus auf dem Maximum der räumlichen Lymphknotenausdehnung liegt, während im *radiomic feature Least Axis Length* das Minimum gemessen wird.

Außerdem ist anzumerken, dass mediastinale Lymphknoten, je nach anatomischer Lokalisation, auch im physiologischen Zustand unterschiedliche Größen aufweisen (70), mit der Folge, dass die alleinige Betrachtung des Lymphknotendurchmessers kein verlässlicher Parameter zur Einschätzung des Nodalstatus ist, vgl. (3, 12).

In den Klassifikationsbäumen KB mit WLCX und KB mit MI hatte die Klassifizierung der Lymphknoten mit dem Feature *Least Axis Length* jedoch einen Zusatznutzen.

Ob das *radiomic feature Least Axis Length*, integriert in einen Klassifikationsalogrithmus, aber einen signifikanten diagnostischen Mehrwert gegenüber alleinigen, konventionellen Lymphknotendurchmesser-Parametern hat, muss in weiteren Analysen geprüft werden.

6.1.2.3 Quantifizierung der Lymphknoten-Dichte

6.1.2.3.1 First Order-Mean

Das *radiomic feature First Order-Mean* quantifiziert die durchschnittlichen Grauwert-Intensitäten der Voxel in einem definierten Bereich.

Die gute Performance dieses Features im Klassifikationsbaum bestätigt aktuelle Studien von Flechsig et al. aus dem Jahr 2014 und 2017, wonach eine höhere Lymphknoten-Dichte, dargestellt durch hohe Grauwert-Intensitäten der Voxel, mit einem positiven Nodalstatus assoziiert ist (47, 51). In der Studie von Flechsig et al. ermöglicht dieser statistische Parameter eine Kategorisierung des Lymphknotens, welche durch eine rein visuelle Befundung nicht gelang (47).

Auch Giesel et. al entdeckten 2017, dass Lymphknoten, die im PET als positiv klassifiziert wurden, signifikant höhere Dichtewerte im CT aufwiesen, weswegen auch hier der

mögliche Zusatznutzen der Analyse durch Dichte-messende *radiomic features* diskutiert wurde (48).

6.1.2.4 Quantifizierte Homogenität

6.1.2.4.1 Run Length Non-Uniformity Normalized

Das *radiomic feature Run Length Non-Uniformity Normalized* zeigt in drei Klassifikationsbäumen die beste Kategorisierungsleistung.

Dieses Feature misst die Unterschiedlichkeit der Längen von Voxelreihen mit identischen Grauwerten, wobei ein hoher Wert ein großes Maß an Unterschiedlichkeit innerhalb der Längen der Voxelreihen im Lymphknoten anzeigt.

Entgegen der oben postulierten Heterogenitätshypothese, wiesen in unserer Untersuchung positive Lymphknoten eine homogenere Verteilung der Voxelreihen-Längen als negative Lymphknoten auf. Die segmentierten Lymphknoten der Stichprobe konnten anhand dieses *radiomic features* in drei Klassifikationsbäumen sogar besser kategorisiert werden, als dies durch andere Features, die beispielsweise die Heterogenität oder Lymphknoten-Form abbilden, gelang.

Eine mögliche Erklärung dieses Phänomens ist die bewiesene Abhängigkeit der Voxelkonfigurationen in tumorösen Strukturen von der zu Grunde liegenden molekularen Tumorbiologie, vgl. (39, 42). Somatische Tumormutationen können, laut Lu et al., in verschiedenen Subtypen des Lungenkarzinoms sowohl zu einer größeren Homogenität, als auch zu einer größeren Heterogenität der Genexpressions-Muster führen (71). Weitere Studien müssen zeigen, ob diese Tumor-Homogenität durch *radiomic feature* wie *Run Length Non-Uniformity Normalized* suffizient abgebildet werden kann.

Balagurunathan et al. konnten 2014 mit Hilfe eines vergleichbaren Features („run-length gray-level nonuniformity“ (52)) erfolgreich Patienten mit Nicht-Kleinzelligen Lungenkarzinomen in verschiedene Überlebensgruppen aufteilen, was die Bedeutung dieser Feature-Gruppe zusätzlich unterstreicht und eine dessen Erforschung über das mediastinale Lymphknotenstaging hinaus sinnvoll erscheinen lässt.

6.1.3 Vergleich der Leistung der Klassifikationsalgorithmen mit der Klassifikationsleistung durch die Radiologen

Da die Lymphknoten-Klassifikation durch die Radiologen auf dem gleichen Test-Datensatz durchgeführt wurde, den auch die Algorithmen klassifizierte, und die Ergebnisse des PET-CTs auch bei den Radiologen als Klassifikations-Goldstandard galten, können die ermittelten Leistungsparameter direkt miteinander verglichen werden.

	Klassifikationsalgorithmen (arithmetisches Mittel)	**Radiologen** (arithmetisches Mittel)
Treffergenauigkeit	0.77	0.80
Sensitivität	0.71	0.49
Spezifität	0.80	0.94
Positiver Prädiktiver Wert	0.56	0.76
Negativer Prädiktiver Wert	0.88	0.84

Abbildung 36: Vergleich der Leistungsparameter der Klassifikationsalgorithmen mit den Leistungsparametern der Radiologen nach Klassifikation der Lymphknoten des Test-Datensatzes. Alle Werte sind gemittelt und aus Gründen der Übersichtlichkeit bis auf zwei Nachkommastellen gerundet. Es fällt auf, dass die Treffergenauigkeiten beider Verfahren mit 0,77 und 0,8 ähnlich gut sind, wobei die Sensitivität der Radiomics-Klassifikationsalgorithmen im Vergleich zu den Radiologen höher ist. Die Spezifität des Radiomics-Verfahrens hingegen ist niedriger. Der positive prädiktive Wert der Algorithmen ist deutlich kleiner, bei einem höheren negativen prädiktiven Wert als die Radiologen.

Die Treffergenauigkeiten beider Verfahren sind mit 77% bzw. 80% ähnlich hoch, beide Verfahren liegen also mit ihrer Klassifikation eines Lymphknotens vergleichbar oft richtig.

Die Sensitivität der Radiologen liegt mit 49% deutlich unter der erzielten Sensitivität der Klassifikationsalgorithmen von 71%. Die Spezifität der Radiologen ist mit 94% etwas höher als die Spezifität der Klassifikationsalgorithmen 80%. Der negative prädiktive Wert ist mit 88% sogar leicht höher als mit 84% bei den Radiologen. Der positive prädiktive Wert hingegen ist mit 56% deutlich niedriger.

Diese Parameter verschaffen einen ersten Eindruck von der potentiellen Leistungsfähigkeit zukünftiger Klassifikationsalgorithmen. Es ist prinzipiell möglich, Algorithmen herzustellen, die automatisiert Lymphknoten kategorisieren und dabei in einigen Testbereichen sogar bessere Leistungen als Radiologen erbringen können. Die

Leistungsparameter eines Algorithmus könnte in Zukunft über dessen Anwendungsbereich entscheiden. Dabei sind Kombinationen von Vorhersagealgorithmen mit der klassischen Befundung durch Radiologen durchaus denkbar, wenn zwei verschiedene Klassifikationsvarianten teststatistisch sinnvoll ergänzen.

Beispielsweise wäre in unserem Fall ein Algorithmus mit einer hohen Sensitivität für ein Vorab-Screening relevanter Bildbereiche geeignet, um so möglichst viele positive Lymphknoten einzuschließen. Diese könnten anschließend durch Radiologen mit einer hohen Spezifität final befundet werden, um die Rate an falsch-negativen Befunden zu minimieren.

Die grafische Darstellung ermöglicht eine gute visuelle Vergleichbarkeit der Leistungsparameter der Klassifikationsalgorithmen sowie der Radiologen untereinander:

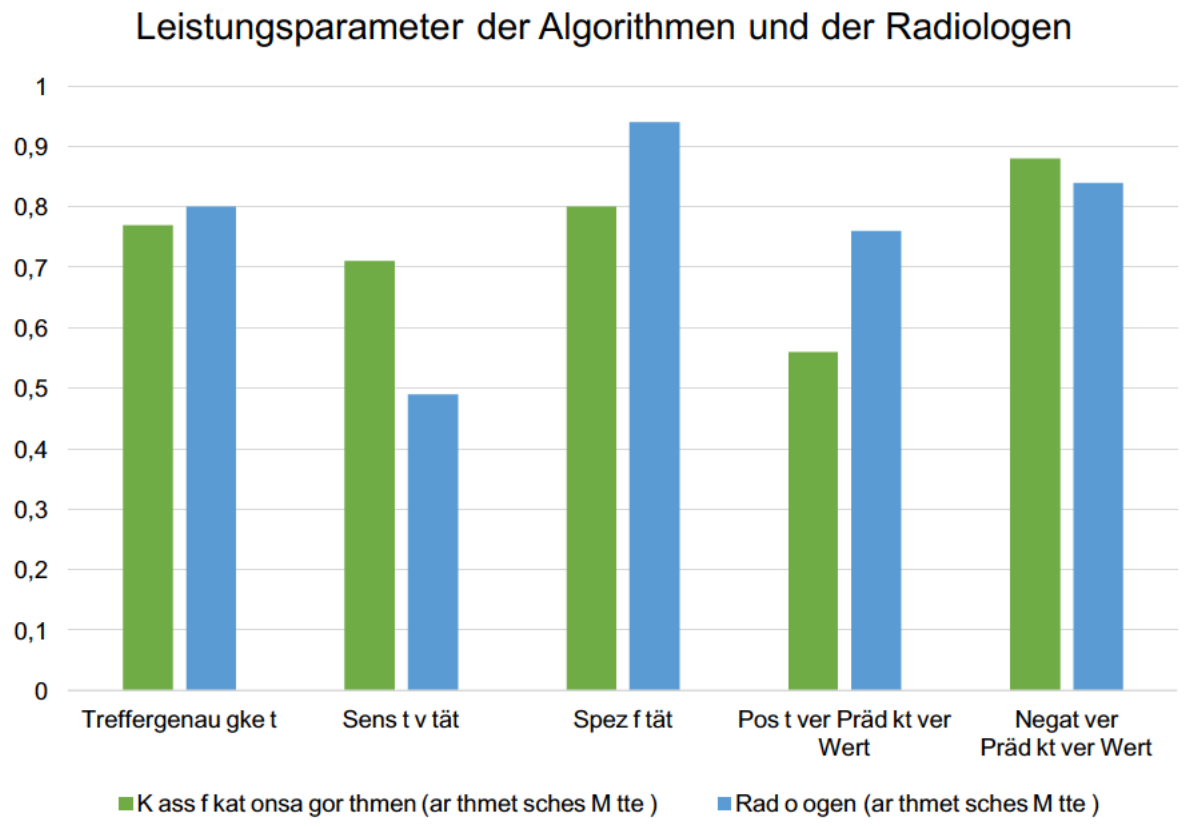


Abbildung 37: Vergleich der Leistungsparameter der Radiologen mit den Leistungsparametern der Klassifikationsalgorithmen. Die Algorithmen erzielen eine deutlich höhere Sensitivität als die Radiologen bei geringerer Spezifität, wobei die Treffergenauigkeit nahezu gleich ist.

6.1.4 Zusatznutzen der integrierten Vorhersagemodelle in unsicheren Kategorien gegenüber der Vorhersage der Radiologen

	Klassifikationsalgorithmen (arithmetisches Mittel)	**Integrierte Vorhersage** (arithmetisches Mittel)	**Vorhersage der Radiologen** (arithmetisches Mittel)
Treffergenauigkeit	0.77	0.80	0.80
Sensitivität	0.71	0.69	0.49
Spezifität	0.80	0.84	0.94
Positiver Prädiktiver Wert	0.56	0.60	0.76
Negativer Prädiktiver Wert	0.88	0.88	0.84

Abbildung 38: Vergleich der Klassifikationsalgorithmen mit der integrierten Vorhersage und der Vorhersage der Radiologen. Durch die Übernahme der Vorhersage durch die Klassifikationsalgorithmen in unsicheren Kategorien wird die Gesamtsensitivität der Radiologen von 49% auf 69% gesteigert. Die Gesamtspezifität der Vorhersage fällt von 94% auf 84%. Die Treffergenauigkeit liegt weiterhin bei 80%.

Die Integration hat mit einer Steigerung von 20% (49% auf 69%) einen diagnostischen Zusatznutzen im Bereich der Sensitivität, büßt aber dadurch 10% Spezifität (94% auf 84%) und 16% (76% auf 60%) beim positiven prädiktiven Wert ein. Hier ist zu erwähnen, dass die positiven prädiktiven Werte der Radiologen einer großen Streuung unterliegen und nur zwei Werte insgesamt vorliegen, wodurch die statistische Aussagekraft eingeschränkt wird.

Durch die Integration der Algorithmen in die Vorhersage der Radiologen in unsicheren Kategorien verändern sich die Leistungsparameter, eine absolute Verbesserung der Genauigkeit ist jedoch nicht zu beobachten (80%).

Auch hier entscheidet also der potentielle Verwendungszweck der Vorhersagemodelle, welches Modell bevorzugt werden sollte (siehe Kapitel *Bewertung der Klassifikationsleitungen der Einzelfeatures,* Absatz *Vergleich der Klassifikationsalgorithmen mit der Klassifikation durch die Radiologen*).

Insgesamt sind die Leistungsparameter der Integrierten Vorhersage vergleichbar mit den Leistungsparametern bei alleiniger Klassifikation der Lymphknoten durch die Algorithmen. Dies lässt vermuten, dass die Vorhersage der Radiologen in sicheren

Kategorien der Vorhersage der Algorithmen ähnelt, sich die Vorhersagen in unsicheren Kategorien jedoch stark unterscheiden. Diese unterschiedliche Klassifikation in unsicheren Kategorien könnte eine Ursache für die deutlich niedrigere Sensitivität der Radiologen sein, die weniger als die Hälfte aller positiven Lymphknoten als solche klassifizieren.

Die folgende Grafik visualisiert die Leistungsparameter aller drei Vorhersagevarianten:

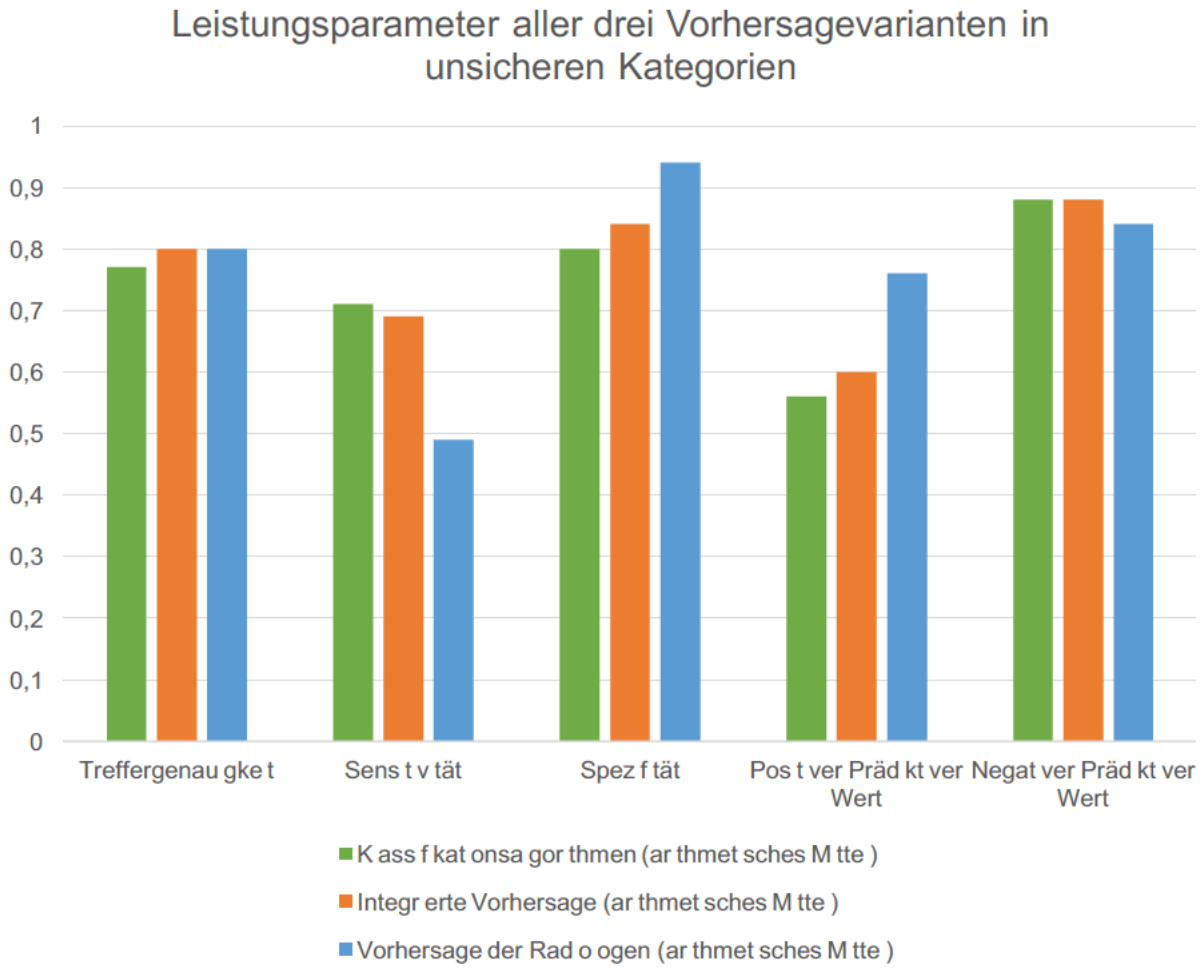


Abbildung 39: Grafischer Vergleich aller drei Vorhersagevarianten. Insgesamt nimmt die integrierte Vorhersage eine Mittelstellung zwischen der Vorhersage durch Klassifikationsalgorithmen und der Vorhersage durch Radiologen ein. Die Genauigkeiten aller drei Vorhersagevarianten ist vergleichbar. Die Sensitivität der integrierten Vorhersage sowie der Vorhersage durch die Klassifikationsalgorithmen sind nahezu identisch, jedoch größer als die Sensitivitäten der Radiologen, bei insgesamt niedrigeren Spezifitäten. Die negativ prädiktiven Werte sind vergleichbar genau. Die positiv prädiktiven Werte weisen innerhalb der Radiologen sowie den Algorithmen eine große Streuung auf, bei insgesamt höheren Werten der Radiologen.

6.1.5 Testung auf signifikante Unterschiede zwischen den Klassifizierungsleistungen der Algorithmen und der Klassifizierungsleistung der Radiologen in unsicheren Kategorien

Alle 24 Vorhersagealgorithmen waren in unsicheren Kategorien der Radiologen signifikant sensitiver. Jedoch war Radiologen 1 in unsicheren Kategorien gegenüber 23 Vorhersagealgorithmen signifikant spezifischer, Radiologen 2 erreichte gegenüber 22 Vorhersagealgorithmen eine signifikant höhere Spezifität.

Diese Ergebnisse könnten auf eine mögliche klinische Nutzbarkeit der *Radiomics*-Vorhersagealgorithmen hinweisen. Sind Radiologen bei der Klassifikation eines Nodalstatus unsicher, könnte ein Vorhersagealgorithmus einspringen und mit seiner höheren Sensitivität in einem ersten Schritt einen deutlich größeren Anteil richtig-positiver Fälle einschließen. In einem zweiten Schritt könnte diese Auswahl an Lymphknoten von Radiologen visuell klassifiziert werden, um dank ihrer höheren Spezifität, fälschlicherweise eingeschlossene, richtig-negative Lymphknoten herauszufiltern.

6.1.6 Vergleich mit externen Forschungsarbeiten

6.1.6.1 Vergleich der Radiomics-Klassifikatoren mit dem konventionellen N-Staging durch CT

Wird das N-Staging im Mediastinum mittels Computertomografie durchgeführt, ergeben sich in der Metanalyse von Toloza et. al aus dem Jahr 2003 eine durchschnittliche Sensitivität von 0,57 sowie eine durchschnittliche Spezifität von 0,82 (72).

Das entscheidende Bildmerkmal für einen positiven Nodalstatus ist die Länge der kleinsten Achse oder der maximale Lymphknotendurchmesser, wobei nach Glazer et al. eine Länge der kleinsten Achsen von 1 cm als Obergrenze für die Einordnung als negativer Lymphknoten den besten Kompromiss zwischen Sensitivität und Spezifität darstellt (70).

Da die Größe mediastinaler Lymphknoten jedoch stark variiert und sowohl größere Lymphknoten negativ, als auch kleinere Lymphknoten positiv sein können, ist die Sensitivität der alleinigen CT-Diagnostik bisher unzureichend, vgl. (3, 12, 72).

Leistungsparameter *des N-Stagings durch CT*	***Mittelwerte*** *der Metanalyse des N-Stagings durch CT (72)*
Sensitivität	0,57
Spezifität	0,82

Abbildung 40: Leistungsparameter des CT-Stagings, dargestellt als Mittelwerte aus der Metaanalyse von Toloza et al. (72).

Ein direkter Vergleich der Leistungsparameter des konventionell N-Stagings durch CT mit den *Radiomics*-Klassifikationsalgorithmen ist nicht möglich, da es methodische Unterschiede gibt: Toloza et al. verwendeten eine histologische Bestätigung oder eine klinische Langzeitbetrachtung der Lymphknoten als Goldstandard zur finalen Lymphknotenklassifikation (72), in unserer Studie hingegen dient die Lymphknotenbefundung mittels PET-CT als Goldstandard.

Auf Grund der genannten Einschränkungen des mediastinalen CT-Stagings, kann das Staging mit *radiomic features* jedoch eine sinnvolle Weiterentwicklung darstellen.

Ein möglicher Vorteil der *Radiomics*-Klassifikationsalgorithmen gegenüber des N-Stagings im CT wäre, dass die Klassifikationsalgorithmen durch die simultane Anwendung mehrerer *radiomic features* die Multidimensionalität der Bildeigenschaften des jeweiligen Lymphknotens im CT-Scan besser abbilden könnten. Durch die simultane Betrachtung mehrerer Bildparameter könnte die Fehleranfälligkeit eines einzelnen Parameters kompensiert werden, ohne auf diesen gänzlich verzichten zu müssen. So würde eine künstlich starke Gewichtung eines einzelnen Merkmals verhindert werden, welches niemals die gesamte Varianz der Lymphknotenbefunde abbilden kann.

6.1.6.2 Vergleich zum N-Staging anderer Radiomics-Arbeiten

Da sich die Methoden der meisten *Radiomics* Studien stark unterscheiden, können die Vorhersageleistungen der entsprechenden Modelle meist nicht direkt miteinander verglichen werden (siehe Absatz *Vergleich der Radiomics-Klassifikatoren mit dem konventionellen N-Staging durch CT*). Deshalb wird die Methodik aktueller Studien orientierend verglichen und richtungsweisende Ergebnisse im Kontext unserer Forschungsarbeit erläutert.

6.1.6.2.1 Radiomics als Teilaspekt in multimodalen Ansätzen bei Yang et al. 2018.

Yang et al. entwickelten im Jahr 2018 ein kombiniertes Vorhersagemodell für das Vorhandensein von Lymphknotenmetastasen bei Adenokarzinom der Lunge mit einer „Sensitivität von 91,66 % und einer Spezifität von 82,14 %“ (73) (Übersetzung durch den Autor). Das Modell bestand aus *radiomic features* und aus dem im CT festgelegten Lymphknotenstatus sowie weiteren klinischen Risikofaktoren. Da die *radiomic features* nur einen Unterteil des Modells darstellen, ist die isolierte Vorhersageleistung durch reine *Radiomic-Feature-Sets* und *Radiomic*-Einzelfeatures nicht zu ermitteln und somit der direkte Vergleich mit unseren Modellen nicht möglich. Es ist jedoch zu vermuten, dass die Kombination von *Radiomics* und klinischen Risikofaktoren in neuen Vorhersagesystemen einen Zusatznutzen aufweisen kann.

Für die Methodik ist, vergleichend zu unserer Studie, folgendes zu betonen: die Segmentierung erfolgte, bei einem relativ geringen Stichprobenumfang von 159 Patienten, automatisiert mittels *deep learning* (unbeaufsichtigtes Lernen ohne externe Modulation). Dieses Vorgehen kann besonders bei kleinen Stichproben und atypischen Lagen problematisch sein, da für eine erfolgreiche Erstellung korrekter Endpunkte im CT umfangreiche Datenmengen verarbeitet werden müssen, um physiologische, interindividuelle Schwankungen im CT-Scan nicht fälschlicherweise als pathologische Besonderheiten fehlzudeuten. Aus diesen Gründen ist bei kleineren Datenmengen die zeitaufwendigere manuelle Segmentierung in Betracht zu ziehen.

Die Kategorisierung der Lymphknoten wurde durch einen histologisch-gesicherten N-Status bestätigt, was ein Vorteil dieser Studie ist, da die Fehlerrate der histologischen Untersuchung als Goldstandard im Vergleich zum PET-CT minimal ist (74). Somit werden hier weniger Fehler in die Klassifizierung des Modells durch die Verwendung einer validen Bestätigungsdiagnostik integriert (siehe Absatz *Besonderheiten und Limitationen der eigenen Studie*).

6.1.6.2.2 Intermodaler Vergleich von machine learning – deep learning – konventioneller Lymphknotenklassifikation durch Wang et al. 2017.

Wang et al. verglichen für die Vorhersage des mediastinalen Nodalstatus bei Lungenkarzinom im Jahr 2017 fünf *machine learning* Vorhersagealgorithmen, die *Radiomics*-Texturparameter oder bekannte diagnostische Elemente aus dem PET-CT nutzen mit einer *deep learning* Methode und der Klassifikation durch Radiologen (75).

Dabei war die Vorhersage der *machine learning* Algorithmen auf Basis bekannter PET-CT Parameter der Vorhersage mittels *radiomic features* überlegen.

Die beste Vorhersageleistung durch *machine learning* Methoden unterschied sich nicht signifikant von der *deep learning* Methode oder der Lymphknotenklassifizierung durch Radiologen. Ähnlich unserer Studie wiesen alle Klassifikationsalgorithmen eine höhere Sensitivität und eine geringere Spezifität als die Befundung durch die Radiologen auf.

Auch hier ergibt sich durch die Nutzung von Vorhersagealgorithmen kein direkter Vorteil gegenüber der Befundung durch Radiologen. Dennoch wird die potentielle Nützlichkeit neuer Texturparameter weiter bestätigt.

Weitere positive Aspekte der Studie sind die histologische Bestätigung des Nodalstatus aller klassifizierten Lymphknoten und die erfolgreiche Nutzung eines *deep learning* Modells bei einer relativ umfangreichen Stichprobe von 1018 CT-Scans, welches gute Vorhersageergebnisse zeigte.

Am Beispiel beider Studien ist zu überlegen, unter welchen Studienvoraussetzungen *machine learning* durch *deep learning* abgelöst werden kann und was diese Veränderung der Methode für die Leistung der Vorhersagemodelle bedeutet.

6.2 CT-Radiomics als mögliche Alternative zum PET-CT

Den Vorteilen der FDG-PET stehen einige Limitation gegenüber, welche dessen erfolgreiche Anwendung erschweren können. Da die Genauigkeit der FDG-PET von multiplen Faktoren abhängt, ergeben sich je nach Tumor und Stoffwechsellage des Patienten Schwierigkeiten. So ist es möglich, dass gut differenzierte oder kleine Malignome mit geringem Energieumsatz kaum Glukose, und damit auch kaum FDG, aufnehmen, vgl. (18). Weiterhin wird eine ausreichende Glukoseaufnahme bei einer diabetischen Stoffwechsellage (Insulinmangel) des Patienten unter Umständen nicht gewährleistet. Beide oben genannten Konstellationen führen, laut S3-Leitlinie Lungenkarzinom, zu einer verminderten Sensitivität der Diagnostik, da vermehrt falsch-negative Ergebnisse eintreten (3).

Wird das, in der Einleitung beschriebene, PET-Protokoll nicht eingehalten, können beispielsweise durch eine zu geringe Menge injiziertes FDG oder durch eine zu kurze Wartezeit (resultiert in mangelhafter Verteilung der Glukose im Körper) weitere falsch-negative Befunde auftreten.

Als aufnahmetechnische Limitation ist die räumliche Auflösung des PET zu nennen, vgl. (16, 17). Diese ist von verschiedenen Faktoren abhängig und beträgt „etwa 6-7 mm“ (3). Laut Basu et al. (16) beeinträchtigen vor allem folgende Faktoren die räumliche Auflösung:

1) Positronenreichweite: Detektiert werde nicht der Ort der Positronenemission, sondern der Ort, an dem die Vernichtungsstrahlung (durch Interaktion eines Positrons und Elektrons) entstehe. Folglich verringert die vom Positron zuvor zurückgelegte Distanz die räumliche Auflösung zwischen „0.2 und 2.6 mm“ (Übersetzung durch den Autor) (16).
2) Nicht-Kollinearität der Vernichtungsstrahlung: Die Annihilationsphotonen laufen nicht immer in einem exakten 180° Winkel auseinander, sodass, je nach Durchmesser des Detektorringes, eine Abweichung von 1-2mm auftreten kann. (Definition der Kollinearität: Punkte, die auf einer gemeinsamen Geraden liegen.)
3) Detektorengröße: Ein Photonendetektor besteht aus einer Vielzahl kleiner Kristalle wobei deren Größe die räumliche Auflösung bedingen.

Durch die geringe räumliche Auflösung der PET sinkt die Sensitivität bei kleinen Lymphknoten, vgl. (3, 14). Zusätzlich kann eine weitere Verschlechterung der Bildqualität durch Bewegungen des Patienten verursacht werden, vgl. (13, 14). Dies ist, laut Hochhegger et al., vor allem am Herzen sowie um das Diaphragma zu erwarten und kann dazu führen, dass die im CT und PET abgebildeten Strukturen nicht deckungsgleich sind (13). Die dadurch gegeneinander verschobenen Ebenen aus anatomischer Karte (CT) und funktionellem Gewebestatus (PET) erschweren eine eindeutige Zuordnung des FDG-aufnehmenden Bereichs zu seinem anatomischen Korrelat. Als Konsequenz sind sowohl falsch-negative als auch falsch-positive Befunde zu erwarten.

Eine vermehrte Anreicherung von FDG ist bei akuten Entzündungen, die eine Erhöhung des zellulären Energieverbrauchs verursachen können, zu erwarten, vgl. (3, 16, 18, 46). Da entzündliche Prozesse auch zeitgleich mit tumorösen Veränderungen auftreten können, ist das interdisziplinäre Erkennen visueller Muster in solchen Situationen, laut Basu et al., von besonderer Bedeutung: „Die Mustererkennung ist deshalb unter diesen Umständen aus der Perspektive eines interpretierenden Radiologen oder Nuklearmediziners, sowie dem behandelnden Onkologen wichtig, um diagnostische Fehler zu vermeiden“ (16).

Außerdem sei die Beachtung des „physiologisch erhöhten FDG-Uptakes“ (16) (Übersetzung durch den Autor) von Organen wie beispielsweise des Gehirns, des Myokards sowie der Blase, für eine exakte Interpretation notwendig (16). Weitere physiologische Varianten, die eine Artefaktbildung zur Folge hätten, seien die Mehraufnahme von FDG durch braunes Fettgewebe und eine gesteigerte Aktivität des Knochenmarks, beispielsweise nach Gabe von Granulozyten-Kolonien stimulierendem Faktor (GCSF) (16).

Schließlich stellt die Nicht-Einhaltung des PET-Protokolls eine weitere Fehlerquelle dar. Durch Patientenbewegungen und mangelndes Fasten kann sich die zelluläre FDG-Aufnahme zusätzliche erhöhen, was in einem verstärkten Hintergrundrauschen resultiert und somit die Identifizierung von Strukturen mit pathologischer Tracer-Aufnahme erschwert.

Durch die daraus resultierende, erhöhte Anzahl an falsch-Positiven, senkt sich, nach S3-Leitlinie Lungenkarzinom, die Spezifität der Diagnostik (3).

Auf Grund dieser Limitationen lässt sich sagen, dass durch eine Vielzahl von Fehlerquellen im komplexen Ablauf und in der Befundung des PET-CTs, sowie der nicht flächendeckenden Verfügbarkeit von PET-CT-Untersuchungen, ein Bedarf an alternativer Diagnostik besteht.

Radiomics-Analysen im CT können eine solche Alternative darstellen: CT-Diagnostik ist nahezu überall verfügbar, einfach und schnell durchzuführen, bedarf einem geringen Personalaufwand, ist kostengünstig und kaum fehleranfällig. So geht die CT-Diagnostik nahezu allen Hindernissen, die im Rahmen der PET auftreten können, automatisch aus dem Weg. Außerdem würde der Wechsel von PET-CT auf CT-basierte Radiomics-Analysen eine geringere Untersuchungslast für Patienten bedeuten.

Die Vorteile von Radiomics werden in Kapitel 6.4 näher erläutert.

6.3 Besonderheiten und Limitationen der eigenen Studie

6.3.1 Besonderheiten

Ein Vorteil unserer Studie liegt in dem großen Stichprobenumfang von insgesamt 1799 segmentierten Lymphknoten bei 381 Lungenkrebspatienten. Mit einer solch großen

Stichprobe steigt die Wahrscheinlichkeit, relevante Zusammenhänge genau abbilden zu können.

Außerdem wurden erstmals an einem solch großen Datensatz 4 Feature-Selektionsmethoden mit 6 Algorithmen simultan getestet. Die so ermöglichte Vergleichbarkeit der 24 Klassifikationsalgorithmen untereinander zeigt, wie unterschiedliche *Radiomics*-Vorhersagesysteme einen identischen Datensatz klassifizieren können.

Schließlich ist als positiver Aspekt noch das exakte statistische Vorgehen beim Training und Testing der Algorithmen zu nennen: die ausgeglichene Balancierung der positiven und negativen Lymphknoten in beiden Datensätzen im Training sowie das anschließende Testing der erstellten Vorhersagemodelle tragen dazu bei, keine künstlich-hohen Vorhersageergebnisse zu erhalten, sondern bilden reale Klassifikationsbedingungen ab.

6.3.2 Limitationen

Das Design unserer Studie ermöglicht es maximal, die PET-CT Klassifikationsergebnisse durch *radiomic features* im CT zu reproduzieren, da die Ergebnisse des PET-CTs als Referenzstandard für die Klassifikation der *Radiomics*-Algorithmen dienen. Das Studienziel, das PET-CT durch *Radiomics* im CT zu ersetzen beziehungsweise eine diagnostische Alternative darzustellen, kann mit diesem Design dennoch erreicht werden, wenn die Klassifikationsergebisse beider Verfahren nahezu identisch sind.

Um jedoch eine objektiv bessere Klassifikationsleistung als das PET-CT zu erbringen, müssen die histopathologischen Ergebnisse aller klassifizierten Lymphknoten vorliegen und beide Klassifikationsverfahren anhand der histopathologischen Klassifizierung als Goldstandard verglichen werden.

Obwohl das PET-CT beim mediastinalen Lymphknotenstaging zwar eine gute orientierende Vorhersage treffen kann, ist die Erstellung von *Radiomics*-Klassifikationsmodellen auf dessen Basis etwas verzerrt. Dies liegt darin begründet, dass mögliche Fehler der Kategorisierung der Lymphknoten im PET-CT als fehlerfreie Response-Kategorien für das Tuning der *Radiomics*-Algorithmen übernommen werden.

Der histologisch-bestätigte Nodalstatus ist jedoch bei der Erstellung eines fehlerfreien Klassifikationsalgorithmus von großer Bedeutung da das gesamte Tuning des Modells anhand dieser Outcome-Kategorie erfolgt. Histologisch gesicherte Nodalstatus finden

sich beispielsweile in den Arbeiten von Yang et al. (73), Wang et al. (75) sowie Toney et al. (46). Je weniger Fehler die Outcome-Kategorie enthält, desto exakter klassifiziert ein Modell in klinischen Alltag unter realen Bedingungen.

Andererseits würde eine solch umfangreiche Studie mit 381 untersuchten CT-Scans und 1799 segmentierten Lymphknoten in Kombination mit einer histologischen Bestätigung durch Biopsie für jeden einzelnen Lymphknoten einen immensen Mehraufwand darstellen und wäre retrospektiv schwer zu realisieren, da nur Lymphknoten in die Studien eingeschlossen werden könnten, für die ein histopathologischer Befund vorliegt. Hier muss eine Abwägung zwischen den positiven Aspekten einer großen Stichprobe gegenüber den diagnostischen Vorteilen einer histologischen Sicherung vorgenommen werden.

6.4 Perspektiven und Limitationen von Radiomics

6.4.1.1 Perspektiven und Vorteile von Radiomics

6.4.1.1.1 Wirksamkeit und Effizienz

Laut Hosny et al. ist es vor allem das Verlangen nach „größerer Wirksamkeit und Effizienz in der klinischen Behandlung“ (35) (Übersetzung durch den Autor), welches den Einzug von künstlicher Intelligenz in die Radiologie bedingt. In den letzten Jahren ist demnach die Menge an auszuwertenden radiologischen Daten disproportional stark im Vergleich zur Anzahl an qualifizierten Befundern gestiegen, wobei diese Entwicklung von den Kliniken mit einer erhöhten Produktivität kompensiert werden müsse, was in der Folge zu Fehlern führe (35).

6.4.1.1.2 Fehlerursachen und Fehlerminimierung

Lee et al. sehen potentielle Fehler in der Radiologie in kognitiven und systemischen Ursachen, wie beispielsweise einer erhöhten Arbeitsbelastung und mangelndem Feedback, begründet (76). Als Strategie zur Fehlervermeidung nennen Lee et al. unter anderem die „Maximierung und Verfeinerung verfügbarer informationstechnologischer Werkzeuge, wie die Computer-gestützte Detektion“ (76) (Übersetzung durch den Autor).

Grundsätzlich unterscheiden Bruno et al. zwischen Wahrnehmungsfehlern und kognitiven Fehler (77). Wahrnehmungsfehler machen demnach bis zu 80% aller Fehler aus und beschrieben das Nicht-Erkennen von Abnormitäten im Bild, kognitive Fehler hingegen entstünden durch eine Fehlbewertung einer Bildbesonderheit durch den

Befunder (77). Besonders die Radiologie sei durch einen sehr variablen Befundungsprozess fehleranfällig. Dabei führen laut Bruno et al. eine hohe Fachkompetenz sowie ausreichende Bildbearbeitungszeiten nicht automatisch zu niedrigeren Fehlerquoten, die in CT-Studien bei bis zu 30% liegen (77).

Hosny et al. sprechen von einer „subjective decision matrix“ (35) (deutsch: subjektive Entscheidungsmatrix), die nach Meinung der Autoren von der Ausbildung und der Erfahrung des Radiologen sowie dessen Verständnis von einem physiologischen Befund abhängt. Die Leistungsfähigkeit der RadiologInnen im klinischen Alltag ist von vielen Faktoren abhängig und selbst innerhalb eines Individuums nicht immer ausreichend konstant: So kommt es sogar bei einer einfachen Tumorgrößenmessung im CT noch zu einer „6% intra-observer variability“ (78) (deutsch: Variabilität innerhalb eines Befunders) bei „15% inter-observer variability“ (78) (deutsch: Variabilität zwischen verschiedenen Befundern).

Einen vielversprechenden Ansatz zur Fehlerbehebung sehen Bruno et al. in „der Reduzierung der Variabilität in Arbeitsprozessen, zum Beispiel durch Standardisierung der radiologischen Arbeitsweise oder des diagnostischen Bildgebungsprotokolls“ (77) (Übersetzung durch den Autor), sowie in einem „cognitive debiasing“ (77) (deutsch: kognitive Entzerrung), sprich eine Objektivierung und neutrale Ausrichtung der Aufmerksamkeit der Befunder.

Die Implementierung von *Radiomics* kann durch die Quantifizierung objektiver Features sowohl für eine nachvollziehbare Struktur des Befundes, als auch für eine neutralere Schwerpunktlegung in der Befundung sorgen. Eine subjektiv als abnorm eingeschätzte Struktur könnte durch einen *Radiomics*-Algorithmus schnell bestätigt oder entkräftet werden und zudem Abnormitäten in Bildbereichen hervorheben, die zuvor keine Beachtung seitens der BefunderInnen erhalten haben. Somit würde dem „anchoring bias“ (77) (deutsch: Ankereffekt), also der voreiligen Fokussierung auf ein besonders hervorstechendes Merkmal zu Lasten einer vollständigen Begutachtung der radiologischen Daten, entgegengewirkt werden.

Auch Deo et al. sehen in *Radiomics* und *machine learning* einen unvoreingenommenen Ansatz, um neue Features zu entdecken: „einer der Hauptbeiträge von machine learning ist das Verfolgen eines unbefangenen Ansatzes, unerwartete informative Variablen zu identifizieren“ (33).

Da die Extraktion von *radiomic features* anhand ihrer statistischen Relevanz erfolgt und eine Vielzahl unterschiedlicher Merkmalsausprägungen auch nachträglich in die Analyse integriert werden können, ist eine umfangreichere, objektivere Prüfung möglich als bei alleiniger Fokussierung auf die in der Ausgangshypothese erwähnte Merkmale.

6.4.1.1.3 Implementierung in radiologische Prozesse

Technisch ist, nach Hosny et al. (35), *Radiomics* in die drei Bereiche der radiologischen Befundung: Detektion, Charakterisierung, Monitoring, integrierbar. Schon heute ist es technisch möglich, mit Hilfe einer Vielzahl von Computerprogrammen (CAD: Computer Aided Detection) radiologische Fragestellungen unterstützend zu beantworten, vgl. (35, 79, 80).

Die diagnostische Hardware in der Radiologie entwickelt sich schnell, sodass immer umfangreichere Bilddaten zu Verfügung stehen, vgl. (35, 44). Wenn diese Datensätze für das menschliche Auge optimiert werden (zum Beispiel durch eine Fensterung des Grauwertintensitäten-Bereiches), kommt es zu einem (temporären) Datenverlust, also einem Verlust von potentiell relevanter medizinischer Information (siehe Kapitel *Einleitung*, Absatz *Grundlagen der CT*).

Welcher Umfang an ungenutzter Information in reguläre CT-Daten bereits enthalten ist, zeigen neue Studien, die sogar molekularpathologische Eigenschaften des Tumorgewebes im CT identifizieren konnten, vgl. (39, 42). *Radiomics*-Algorithmen können in Zukunft helfen, diese ungenutzten Informationen nutzbar zu machen, um so Diagnosen zu ermöglichen, die auf einer breiteren Faktengrundlage basieren.

Durch die Nutzung dieser bereits vorhandenen Daten und dem minimalen Materialaufwand bieten *Radiomic-Features* außerdem die Chance für eine kostengünstige und schnelle Erweiterung des medizinischen Bild-Befundes, vgl. (31, 40, 42). Dies ist jedoch nur möglich, wenn in Zukunft *Radiomics*-Klassifikationsalgorithmen frei verfügbar sind und keinen teuren Lizensierungsverfahren unterliegen.

Die Geschwindigkeit der neuen Hardware in Kombination mit moderner Analysesoftware kann zu einer gesteigerten Effizienz im radiologischen Arbeitsalltag führen und somit Kosten reduzieren und die Leistung erhöhen. Zeitintensive Routineaufgabe könnten in Zukunft von *Radiomics*-Programmen übernommen werden, wobei der moderne Radiologe in diesem Kontext laut Jha et al. eher eine kontrollierende Funktion inne hat

(29). Durch die Nutzung künstlicher Intelligenz in der Radiologie verschiebt sich, laut Jha et al., das Tätigkeitsfeld der zukünftigen Radiologen weg von der einfachen, visuellen Bildanalyse hin zu „information specialists" (29) (deutsch: Informationsspezialisten), die durch Automatisierung generierte Bildbefunde überprüften und mit umfangreichen Patienteninformationen in den klinischen Kontext integrierten.

Die erfolgreiche Automatisierung und Digitalisierung medizinischer Prozesse zusammen mit detaillierteren, einheitlich strukturierten, quantifizierbaren Befunden ist ein wichtiger Schritt auf dem Weg hin zu technisch erfolgreicher und individualisierter Medizin.

Die RadiologInnen der Zukunft haben, laut Hosny et al., in dieser Entwicklung die Aufgaben, neue Prozesse kritisch zu begleiten und auf ihre inhaltliche Korrektheit hin zu überprüfen sowie Ergebnisse zu interpretieren und sinnvoll in die medizinische Behandlung zu integrieren (35).

6.4.1.2 Limitationen und Nachteile von Radiomics

Mit der fortschreitenden Entwicklung von Künstlicher Intelligenz in der Medizin entwickeln sich neue Fragestellungen, die die Bereiche Sicherheit, Verantwortung, Verständlichkeit, Umgang mit Daten und Kosten einschließen, und ein stabiles, reglementiertes System der Mensch-Maschine Interaktion erforderlich machen.

6.4.1.2.1 Standardisierung

Trotz aktueller Bemühungen von über 20 Forschungsgruppen, eine Strukturierung der Vielzahl an Features in Form der *Image biomarker standardisation initiative* (https://arxiv.org/abs/1612.07003, letzter Stand: 20.06.2019) (36) vorzunehmen, fehlt es, laut Ferreira et al., an einer einheitlichen Standardisierung der Features, an welcher die Forschung Feature-Exktraktionsprogramme validieren und kalibrieren könne (34).

Außerdem zeigt die klinische Erfahrung, dass selbst die etablierte CT-Diagnostik noch weit entfernt von einer nationalen oder globalen Standardisierung ist. Dies ist beispielsweise durch die verschiedenen Geräte- und Softwarehersteller sowie unterschiedliche Rekonstruktionsprotokolle bedingt.

Diese Vereinheitlichung ist jedoch zwingend notwendig, um eine Vergleichbarkeit der ermittelten quantitativen Parameter sowohl zwischen einzelnen Patienten als auch zwischen neuen Forschungsergebnissen unterschiedlicher Forschungsgruppen sicherzustellen.

So findet beispielsweise, laut Sollini et al., bisher weder ein standardisiertes „image denoising“ (deutsch: Entrauschen des Bildes), noch eine „partial volume correction“ (deutsch: teilweise Volumen-Korrektur) statt, welches, laut den Autoren, in beiden Fälle eine falsch-hoch gemessene Heterogenität verursacht (38). Ziel müsse es sein, die aussagekräftigsten Features standardisiert zu isolieren, Redundanz innerhalb dieser Features zu vermeiden und eine global-vergleichbare, evidenzbasierte klinische Anwendbarkeit zu ermöglichen (38). Erst dann kann der große Vorteil von *Radiomics* - die quantitative Erfassung und Vergleichbarkeit radiologischer Bildparameter - vollumfänglich genutzt werden.

6.4.1.2.2 Datensicherheit

Weitere Grundlegende Fragen werden sein, anhand welcher Datensätze KI-Algorithmen lernen, wer diese Datensätze bereitstellt und wer die Vorhersagealgorithmen programmiert. Dabei müssen die Datensicherheit und der Schutz jedes Patienten garantiert sein, besonders dann, wenn personalisierte Aufnahmen aus zentralen Datenbanken in die Erweiterung eines Deep-Learning Algorithmus integriert werden sollen. Andererseits besteht die Gefahr, durch eine restriktive Datenpolitik den notwendigen Bedarf des *machine* und *deep learnings* an großen Datenmengen nicht decken zu können.

6.4.1.2.3 Hoher Entwicklungsaufwand

Problematisch ist, neben der globalen Verfügbarkeit geeigneter Datenmengen, auch die hochkomplexe und arbeitsintensive Optimierung eines *machine* learning Algorithmus.

Features müssen vordefiniert, relevante Bildbereiche segmentiert und Ergebnisse validiert werden. Diese Aufgaben sind zeitintensiv und können nur von ausgebildeten Statistikern und medizinischem Fachpersonal durchgeführt werden, was sich in hohen Entwicklungskosten und langen Entwicklungszeiten widerspiegelt und eine, nur auf die initiale Fragestellung und deren Features begrenzte, klinische Anwendbarkeit zur Folge hat.

Schon heute sind Computer-aided-detection-Systeme (CAD) (deutsch: Computergestützte Erkennungssysteme) für bestimmte Anwendungen verfügbar, beispielsweise für die Detektion von Lungenrundherden, vgl. (79). Jedoch mindert, laut Hosny et al., eine hohe Rate an Falsch-Positiven Ergebnissen die Akzeptanz bei Befundern, da in diesen Fällen eine Einzelfallprüfung erfolgen müsse (35). Aktuell wirken

CAD-Systeme daher nur unterstützend bei der Befundung, ersetzen aber nicht die ärztliche Diagnose und sind noch nicht flächendeckend in die Diagnostik integriert.

Hier können *Radiomics* in Zukunft eine leistungsfähige Weiterentwicklung der CAD-Systeme darstellen, vlg. (34, 44).

6.4.1.2.4 Haftung der Ärzteschaft

Außerdem ist das Thema der Haftung für eventuelle Fehler neuer KI-Systeme von besonderer Bedeutung. Haftet bei einem frei-verfügbaren KI-Algorithmus der betreuende Arzt, weil er einen Fehler des Algorithmus fälschlicherweise nicht erkannt hat? Oder war in diesem Fall die Methodik des Vorhersagealgorithmus ungeeignet beziehungsweise die Datengrundlage des Algorithmus nicht ausreichend groß für eine exakte radiologische Bild-Diagnostik?

Die Kombination der drei Fachbereiche Informatik, Mathematik und Medizin erfordern eine genaue Definition der Verantwortlichkeitsbereiche untereinander sowie einen intensiven, interdisziplinären Austausch zur Behebung potentieller Fehlerquellen.

6.4.1.2.5 Nachvollziehbarkeit und Transparenz

Hosny et al. stellen zudem die Frage nach Voraussetzungen für eine erfolgreiche Zulassung durch staatliche Prüfstellen. Je umfangreicher der KI-Algorithmus, desto komplexer sei die Entscheidungsfindung und deren Nachvollziehbarkeit durch externe Prüfer der Zulassungsstellen (35).

Dies trifft besonders auf *deep learning* Strukturen zu, da in hidden-layers (deutsch: verborgene Schichten) definierte Bild-Endpunkte in großer Zahl vorhanden sind und durch Analyse umfangreicher Datenmengen automatisch generiert werden. Diese komplexen Strukturen verursachen, laut Hosny et al., ein mangelndes Verständnis des internen Ablaufes, in dem *deep learning* Algorithmen zu Schlussfolgerungen kommen, was auch als „black-box medicine“ bezeichnet wird (35).

Studien wie *Correlation Between SUVmax and CT Radiomic Analysis Using Lymph Node Density in PET/CT-Based Lymph Node Staging* von Giesel et al. (48), in denen Zusammenhänge konventioneller Parameter mit neuen *radiomic features* im *machine learning* hergestellt werden, sind daher für die Akzeptanz neuer *Radiomics*-Verfahren unverzichtbar. *Deep learning* Strukturen könnten in transparenten Validationsverfahren

auf ihre Genauigkeit untersucht werden und durch verlässliche Vorhersagen langsam das Vertrauen der Befunder gewinnen.

In Zukunft muss geklärt werden, ob ein undurchsichtiges System mit exzellenten Vorhersageleistungen einem klar-nachvollziehbaren System mit schlechterer Performance vorgezogen werden soll, oder ob eine detaillierte Nachprüfbarkeit der Entscheidungsfindung von *Radiomics*-Algorithmen das höchste Gut ist. Ziel muss sein, eine effiziente und nachvollziehbare Mensch-Maschine-Interaktion zu gewährleisten, bei der das Wohl des Patienten im Vordergrund steht.

7 Literaturverzeichnis

1. Robert Koch Institut ZfK. Bericht zum Krebsgeschehen in Deutschland 2016. 2016.
2. Thomas M, Dienemann H, Herth F, Debus J. Lungenkarzinom. In: Hiddemann W, Bartram CR, editors. Die Onkologie. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 1223 49.
3. Medizinischen LOdAdW, Fachgesellschaften e.V. (AWMF) DKeVDu, (DKH). DK. S3 Leitlinie Prävention, Diagnostik, Therapie und Nachsorge des Lungenkarzinoms. 2018.
4. Pallis AG, Syrigos KN. Lung cancer in never smokers: disease characteristics and risk factors. Critical reviews in oncology/hematology. 2013;88(3):494 503.
5. de Groot PM, Wu CC, Carter BW, Munden RF. The epidemiology of lung cancer. Translational lung cancer research. 2018;7(3):220 33.
6. Secretan B, Straif K, Baan R, Grosse Y, El Ghissassi F, Bouvard V, Benbrahim Tallaa L, Guha N, Freeman C, Galichet L, Cogliano V. A review of human carcinogens Part E: tobacco, areca nut, alcohol, coal smoke, and salted fish. The Lancet Oncology. 2009;10(11):1033 4.
7. Tobacco smoke and involuntary smoking. IARC monographs on the evaluation of carcinogenic risks to humans. 2004;83:1 1438.
8. Warth A, Endris V, Penzel R, Weichert W. Molekularpathologie des Lungenkarzinoms. Der Pathologe. 2014;35(6):565 73.
9. van der Heijden EH, Casal RF, Trisolini R, Steinfort DP, Hwangbo B, Nakajima T, Guldhammer Skov B, Rossi G, Ferretti M, Herth FF, Yung R, Krasnik M. Guideline for the acquisition and preparation of conventional and endobronchial ultrasound guided transbronchial needle aspiration specimens for the diagnosis and molecular testing of patients with known or suspected lung cancer. Respiration; international review of thoracic diseases. 2014;88(6):500 17.
10. Wittekind C. New TNM classification of lung tumors. Pathologe. 2014;35(6):578 85.
11. Detterbeck FC, Boffa DJ, Kim AW, Tanoue LT. The Eighth Edition Lung Cancer Stage Classification. Chest. 2017;151(1):193 203.
12. de Langen AJ, Raijmakers P, Riphagen I, Paul MA, Hoekstra OS. The size of mediastinal lymph nodes and its relation with metastatic involvement: a meta analysis. European journal of cardio thoracic surgery : official journal of the European Association for Cardio thoracic Surgery. 2006;29(1):26 9.
13. Hochhegger B, Alves GR, Irion KL, Fritscher CC, Fritscher LG, Concatto NH, Marchiori E. PET/CT imaging in lung cancer: indications and findings. Jornal brasileiro de pneumologia : publicacao oficial da Sociedade Brasileira de Pneumologia e Tisilogia. 2015;41(3):264 74.
14. Khiewvan B, Ziai P, Houshmand S, Salavati A, Ziai P, Alavi A. The role of PET/CT as a prognosticator and outcome predictor in lung cancer. Expert review of respiratory medicine. 2016;10(3):317 30.
15. Vansteenkiste J, Fischer BM, Dooms C, Mortensen J. Positron emission tomography in prognostic and therapeutic assessment of lung cancer: systematic review. The Lancet Oncology. 2004;5(9):531 40.
16. Basu S, Kwee TC, Surti S, Akin EA, Yoo D, Alavi A. Fundamentals of PET and PET/CT imaging. Annals of the New York Academy of Sciences. 2011;1228:1 18.
17. Goldman LW. Principles of CT and CT technology. Journal of nuclear medicine technology. 2007;35(3):115 28; quiz 29 30.
18. Harders SW, Balyasnikowa S, Fischer BM. Functional imaging in lung cancer. Clinical physiology and functional imaging. 2014;34(5):340 55.

19. Garcia Velloso MJ, Bastarrika G, de Torres JP, Lozano MD, Sanchez Salcedo P, Sancho L, Nunez Cordoba JM, Campo A, Alcaide AB, Torre W, Richter JA, Zulueta JJ. Assessment of indeterminate pulmonary nodules detected in lung cancer screening: Diagnostic accuracy of FDG PET/CT. Lung cancer (Amsterdam, Netherlands). 2016;97:81 6.
20. Madsen PH, Holdgaard PC, Christensen JB, Hoilund Carlsen PF. Clinical utility of F 18 FDG PET CT in the initial evaluation of lung cancer. European journal of nuclear medicine and molecular imaging. 2016;43(11):2084 97.
21. Takeuchi S, Khiewvan B, Fox PS, Swisher SG, Rohren EM, Bassett RL, Jr., Macapinlac HA. Impact of initial PET/CT staging in terms of clinical stage, management plan, and prognosis in 592 patients with non small cell lung cancer. European journal of nuclear medicine and molecular imaging. 2014;41(5):906 14.
22. van Tinteren H, Hoekstra OS, Smit EF, van den Bergh JH, Schreurs AJ, Stallaert RA, van Velthoven PC, Comans EF, Diepenhorst FW, Verboom P, van Mourik JC, Postmus PE, Boers M, Teule GJ. Effectiveness of positron emission tomography in the preoperative assessment of patients with suspected non small cell lung cancer: the PLUS multicentre randomised trial. Lancet (London, England). 2002;359(9315):1388 93.
23. Herder GJ, Kramer H, Hoekstra OS, Smit EF, Pruim J, van Tinteren H, Comans EF, Verboom P, Uyl de Groot CA, Welling A, Paul MA, Boers M, Postmus PE, Teule GJ, Groen HJ. Traditional versus up front [18F] fluorodeoxyglucose positron emission tomography staging of non small cell lung cancer: a Dutch cooperative randomized study. Journal of clinical oncology : official journal of the American Society of Clinical Oncology. 2006;24(12):1800 6.
24. Kay FU, Kandathil A, Batra K, Saboo SS, Abbara S, Rajiah P. Revisions to the Tumor, Node, Metastasis staging of lung cancer (8(th) edition): Rationale, radiologic findings and clinical implications. World journal of radiology. 2017;9(6):269 79.
25. Travis WD, Brambilla E, Nicholson AG, Yatabe Y, Austin JHM, Beasley MB, Chirieac LR, Dacic S, Duhig E, Flieder DB, Geisinger K, Hirsch FR, Ishikawa Y, Kerr KM, Noguchi M, Pelosi G, Powell CA, Tsao MS, Wistuba I. The 2015 World Health Organization Classification of Lung Tumors: Impact of Genetic, Clinical and Radiologic Advances Since the 2004 Classification. Journal of thoracic oncology : official publication of the International Association for the Study of Lung Cancer. 2015;10(9):1243 60.
26. Hartmaier RJ, Albacker LA, Chmielecki J, Bailey M, He J, Goldberg ME, Ramkissoon S, Suh J, Elvin JA, Chiacchia S, Frampton GM, Ross JS, Miller V, Stephens PJ, Lipson D. High Throughput Genomic Profiling of Adult Solid Tumors Reveals Novel Insights into Cancer Pathogenesis. Cancer research. 2017;77(9):2464 75.
27. Jones PA, Issa JP, Baylin S. Targeting the cancer epigenome for therapy. Nature reviews Genetics. 2016;17(10):630 41.
28. Turing AM. Computing Machinery and Intelligence. In: Epstein R, Roberts G, Beber G, editors. Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer. Dordrecht: Springer Netherlands; 1950. p. 23 65.
29. Jha S, Topol EJ. Adapting to Artificial Intelligence: Radiologists and Pathologists as Information Specialists. Jama. 2016;316(22):2353 4.
30. Ghahramani Z. Probabilistic machine learning and artificial intelligence. Nature. 2015;521(7553):452 9.
31. Lambin P, Rios Velazquez E, Leijenaar R, Carvalho S, van Stiphout RG, Granton P, Zegers CM, Gillies R, Boellard R, Dekker A, Aerts HJ. Radiomics: extracting more information from medical images using advanced feature analysis. European journal of cancer (Oxford, England : 1990). 2012;48(4):441 6.

32. Peek N, Combi C, Marin R, Bellazzi R. Thirty years of artificial intelligence in medicine (AIME) conferences: A review of research themes. Artificial intelligence in medicine. 2015;65(1):61 73.
33. Deo RC. Machine Learning in Medicine. Circulation. 2015;132(20):1920 30.
34. Ferreira Junior JR, Koenigkam Santos M, Cipriano FEG, Fabro AT, Azevedo Marques PM. Radiomics based features for pattern recognition of lung cancer histopathology and metastases. Computer methods and programs in biomedicine. 2018;159:23 30.
35. Hosny A, Parmar C, Quackenbush J, Schwartz LH, Aerts H. Artificial intelligence in radiology. Nature reviews Cancer. 2018.
36. Alex Zwanenburg SL, Martin Vallières, Steffen Löck. Image biomarker standardisation initiative. https://arxivorg/abs/161207003. 2018.
37. Chen B, Zhang R, Gan Y, Yang L, Li W. Development and clinical application of radiomics in lung cancer. Radiat Oncol. 2017;12(1):154.
38. Sollini M, Cozzi L, Antunovic L, Chiti A, Kirienko M. PET Radiomics in NSCLC: state of the art and a proposal for harmonization of methodology. Sci Rep. 2017;7(1):358.
39. Grossmann P, Stringfield O, El Hachem N, Bui MM, Rios Velazquez E, Parmar C, Leijenaar RT, Haibe Kains B, Lambin P, Gillies RJ, Aerts HJ. Defining the biological basis of radiomic phenotypes in lung cancer. eLife. 2017;6.
40. Aerts HJ, Velazquez ER, Leijenaar RT, Parmar C, Grossmann P, Carvalho S, Bussink J, Monshouwer R, Haibe Kains B, Rietveld D, Hoebers F, Rietbergen MM, Leemans CR, Dekker A, Quackenbush J, Gillies RJ, Lambin P. Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach. Nature communications. 2014;5:4006.
41. Gutierrez ME, Choi K, Lanman RB, Licitra EJ, Skrzypczak SM, Pe Benito R, Wu T, Arunajadai S, Kaur S, Harper H, Pecora AL, Schultz EV, Goldberg SL. Genomic Profiling of Advanced Non Small Cell Lung Cancer in Community Settings: Gaps and Opportunities. Clinical lung cancer. 2017;18(6):651 9.
42. Rios Velazquez E, Parmar C, Liu Y, Coroller TP, Cruz G, Stringfield O, Ye Z, Makrigiorgos M, Fennessy F, Mak RH, Gillies R, Quackenbush J, Aerts H. Somatic Mutations Drive Distinct Imaging Phenotypes in Lung Cancer. Cancer research. 2017;77(14):3922 30.
43. Hellwig D, Baum RP, Kirsch C. FDG PET, PET/CT and conventional nuclear medicine procedures in the evaluation of lung cancer: a systematic review. Nuklearmedizin Nuclear medicine. 2009;48(2):59 69, quiz N8 9.
44. Rabbani M, Kanevsky J, Kafi K, Chandelier F, Giles FJ. Role of artificial intelligence in the care of patients with nonsmall cell lung cancer. European journal of clinical investigation. 2018;48(4).
45. Huang Y, Liu Z, He L, Chen X, Pan D, Ma Z, Liang C, Tian J, Liang C. Radiomics Signature: A Potential Biomarker for the Prediction of Disease Free Survival in Early Stage (I or II) Non Small Cell Lung Cancer. Radiology. 2016;281(3):947 57.
46. Toney LK, Vesselle HJ. Neural networks for nodal staging of non small cell lung cancer with FDG PET and CT: importance of combining uptake values and sizes of nodes and primary tumor. Radiology. 2014;270(1):91 8.
47. Flechsig P, Kratochwil C, Schwartz LH, Rath D, Moltz J, Antoch G, Heussel CP, Rieser M, Warth A, Zabeck H, Kauczor HU, Haberkorn U, Giesel FL. Quantitative volumetric CT histogram analysis in N staging of 18F FDG equivocal patients with lung cancer. Journal of nuclear medicine : official publication, Society of Nuclear Medicine. 2014;55(4):559 64.
48. Giesel FL, Schneider F, Kratochwil C, Rath D, Moltz J, Holland Letz T, Kauczor HU, Schwartz LH, Haberkorn U, Flechsig P. Correlation Between SUVmax and CT Radiomic Analysis Using Lymph

Node Density in PET/CT Based Lymph Node Staging. Journal of nuclear medicine : official publication, Society of Nuclear Medicine. 2017;58(2):282 7.
49. Swensen SJ, Viggiano RW, Midthun DE, Muller NL, Sherrick A, Yamashita K, Naidich DP, Patz EF, Hartman TE, Muhm JR, Weaver AL. Lung nodule enhancement at CT: multicenter study. Radiology. 2000;214(1):73 80.
50. Hawkins S, Wang H, Liu Y, Garcia A, Stringfield O, Krewer H, Li Q, Cherezov D, Gatenby RA, Balagurunathan Y, Goldgof D, Schabath MB, Hall L, Gillies RJ. Predicting Malignant Nodules from Screening CT Scans. Journal of thoracic oncology : official publication of the International Association for the Study of Lung Cancer. 2016;11(12):2120 8.
51. Flechsig P, Frank P, Kratochwil C, Antoch G, Rath D, Moltz J, Rieser M, Warth A, Kauczor HU, Schwartz LH, Haberkorn U, Giesel FL. Radiomic Analysis using Density Threshold for FDG PET/CT Based N Staging in Lung Cancer Patients. Molecular imaging and biology : MIB : the official publication of the Academy of Molecular Imaging. 2017;19(2):315 22.
52. Balagurunathan Y, Gu Y, Wang H, Kumar V, Grove O, Hawkins S, Kim J, Goldgof DB, Hall LO, Gatenby RA, Gillies RJ. Reproducibility and Prognosis of Quantitative Features Extracted from CT Images. Translational oncology. 2014;7(1):72 87.
53. Coroller TP, Agrawal V, Huynh E, Narayan V, Lee SW, Mak RH, Aerts H. Radiomic Based Pathological Response Prediction from Primary Tumors and Lymph Nodes in NSCLC. Journal of thoracic oncology : official publication of the International Association for the Study of Lung Cancer. 2017;12(3):467 76.
54. Kumar V, Gu Y, Basu S, Berglund A, Eschrich SA, Schabath MB, Forster K, Aerts HJ, Dekker A, Fenstermacher D, Goldgof DB, Hall LO, Lambin P, Balagurunathan Y, Gatenby RA, Gillies RJ. Radiomics: the process and the challenges. Magnetic resonance imaging. 2012;30(9):1234 48.
55. Rusch VW, Asamura H, Watanabe H, Giroux DJ, Rami Porta R, Goldstraw P. The IASLC Lung Cancer Staging Project: A Proposal for a New International Lymph Node Map in the Forthcoming Seventh Edition of the TNM Classification for Lung Cancer. Journal of Thoracic Oncology. 2009;4(5):568 77.
56. R Core Team. R: A Language and Environment for Statistical

Computing. R Foundation for Statistical Computing; 2017.
57. Wickham H. Tidyverse: Easily Install and Load 'Tidyverse' Packages. 2017.
58. Xie Y. Dynamic Documents with R and Knitr. Boca Raton, Florida: Chapman; Hall/CRC; 2015.
59. Qiong W, Roland LD. The role of balanced training and testing data sets for binary classifiers in bioinformatics. PloS one. 2013;8(7):e67863.
60. Kononenko I. Machine learning and data mining : introduction to principles and algorithms. Kukar M, editor. Chichester, England: Chichester, England : Horwood Publishing; 2007. 1 online resource (475 pages) : p.
61. van Griethuysen JJM, Fedorov A, Parmar C, Hosny A, Aucoin N, Narayan V, Beets Tan RGH, Fillion Robin JC, Pieper S, Aerts H. Computational Radiomics System to Decode the Radiographic Phenotype. Cancer research. 2017;77(21):e104 e7.
62. Han J, Kamber M, Pei J. Data Preprocessing. In: Han J, Kamber M, Pei J, editors. Data Mining (Third Edition). Boston: Morgan Kaufmann; 2012. p. 83 124.
63. Guenther N, Schonlau M. Support vector machines. Stata Journal. 2016;16(4):917 37.
64. Livingstone D. A practical guide to scientific data analysis. Hoboken, N.J.: Hoboken, N.J. : Wiley; 2009. 1 online resource (359 p.) p.
65. Titterington M. Neural networks. Wiley Interdisciplinary Reviews: Computational Statistics. 2010;2(1):1 8.

66. Han J, Kamber M, Pei J. Classification: Advanced Methods. In: Han J, Kamber M, Pei J, editors. Data Mining (Third Edition). Boston: Morgan Kaufmann; 2012. p. 393 442.
67. Myles AJ, Feudale RN, Liu Y, Woody NA, Brown SD. An introduction to decision tree modeling. Journal of Chemometrics. 2004;18(6):275 85.
68. Wiest MM, Lee KJ, Carlin JB. Statistics for clinicians: An introduction to logistic regression. Journal of Paediatrics and Child Health. 2015;51(7):670 3.
69. Huber F, Herrmann A, Meyer F, Vogel J, Vollhardt K. Grundlagen zur Schätzung von komplexen Strukturgleichungen unter Verwendung von PLS. In: Huber F, Herrmann A, Meyer F, Vogel J, Vollhardt K, editors. Kausalmodellierung mit Partial Least Squares: Eine anwendungsorientierte Einführung. Wiesbaden: Gabler; 2007. p. 3 15.
70. Glazer GM, Gross BH, Quint LE, Francis IR, Bookstein FL, Orringer MB. Normal mediastinal lymph nodes: number and size according to American Thoracic Society mapping. AJR American journal of roentgenology. 1985;144(2):261 5.
71. Lu J, Wang W, Xu M, Li Y, Chen C, Wang X. A global view of regulatory networks in lung cancer: An approach to understand homogeneity and heterogeneity. Seminars in cancer biology. 2017;42:31 8.
72. Toloza EM, Harpole L, McCrory DC. Noninvasive staging of non small cell lung cancer: a review of the current evidence. Chest. 2003;123(1 Suppl):137s 46s.
73. Yang X, Pan X, Liu H, Gao D, He J, Liang W, Guan Y. A new approach to predict lymph node metastasis in solid lung adenocarcinoma: a radiomics nomogram. Journal of thoracic disease. 2018;10(Suppl 7):S807 S19.
74. Frechet B, Kazakov J, Thiffault V, Ferraro P, Liberman M. Diagnostic Accuracy of Mediastinal Lymph Node Staging Techniques in the Preoperative Assessment of Nonsmall Cell Lung Cancer Patients. Journal of bronchology & interventional pulmonology. 2018;25(1):17 24.
75. Wang H, Zhou Z, Li Y, Chen Z, Lu P, Wang W, Liu W, Yu L. Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non small cell lung cancer from (18)F FDG PET/CT images. EJNMMI research. 2017;7(1):11.
76. Lee CS, Nagy PG, Weaver SJ, Newman Toker DE. Cognitive and system factors contributing to diagnostic errors in radiology. AJR American journal of roentgenology. 2013;201(3):611 7.
77. Bruno MA, Walker EA, Abujudeh HH. Understanding and Confronting Our Mistakes: The Epidemiology of Error in Radiology and Strategies for Error Reduction. Radiographics : a review publication of the Radiological Society of North America, Inc. 2015;35(6):1668 76.
78. Fitzgerald R. Error in radiology. Clinical radiology. 2001;56(12):938 46.
79. Wang Z, Xin J, Sun P, Lin Z, Yao Y, Gao X. Improved lung nodule diagnosis accuracy using lung CT images with uncertain class. Computer methods and programs in biomedicine. 2018;162:197 209.
80. Rubin GD. Lung nodule and cancer detection in computed tomography screening. Journal of thoracic imaging. 2015;30(2):130 8.

8 Anhang

Statistische Modellierung mit Quellcode

Autor: Dr. Paul Schmidt, Berlin. Homepage: www.statistische-modellierung.de

Balancierung, Feature Selection und Klassifizierung von Lymphknoten

Dr. Paul Schmidt (paul.schmidt.mail@gmail.com (mailto:paul.schmidt.mail@gmail.com), www.statistische-modellierung.de)

7. Mai 2019

1 Einleitung

Dieser Bericht enthält die Anleitung zur Klassifikation von positiven Lymphknoten einschließlich der Schritte Balancierung, Feature Selection und Training.

Folgende Pakete werden vorausgesetzt:

```
library(knitr)
library(tidyverse); theme_set(theme_bw())
library(readxl)
library(mRMRe)
library(caret)
library(MASS)
library(rpart)
library(rpart plot)
options(OutDec = " "  width = 90  digits = 2)
```

2 Daten laden und Vorbereitung

Zunächst werden die Datensätze geladen:

```
## Data for training
df <- read_xlsx("~/201801_Makowski/02_Dokumente/20190408/20190407_luca_trainvalid_test_all_aid.xlsx"  sheet = 1)

## Data for testing
df_test <- read_xlsx("~/201801_Makowski/02_Dokumente/20190408/20190505_luca_trainvalid_test_all_wpred_aid.xlsx"  sheet = 2)
```

Als nächstes fügen wir den Datensätzen die korrekt formatierte Response-Variable (positive vs. negative Lymphknoten) hinzu:

```
df$response <- factor(df$pos_neg,
                      levels = c("n", "p"),
                      labels = 0:1)

df_test$response <- factor(df_test$pos_neg,
                           levels = c("n", "p"),
                           labels = 0:1)
```

Um eine Übermacht der negativen Lymphknoten im Trainingsprozess zu verhindern, erzeugen wir einen balancierten Trainingsdatensatz indem wir nur so viele negative Lymphknoten auswählen, wie auch positive vorhanden sind. Die Auswahl erfolgt dabei zufällig.

```
## Indicator variable for keeping
df$Keep <- 0

## Keep all positive lymphs
df$Keep[df$response == 1] <- 1

## Select as many negative lymphs as there are positives
set.seed(4828492)
s <- sample(x = which(df$response == 0),
            size = sum(df$response == 1),
            replace = TRUE)
df$Keep[s] <- 1

## Filter and delete some unnecessary variables from the data set
df <- df %>%
  filter(Keep > 0) %>%
  dplyr::select(-AID, -Keep, -pos_neg)
```

Sicherlich wird die Zufallsauswahl einen Einfluss auf das finale Ergebnis haben, da eine andere Auswahl zu anderen Ergebnissen kommen würde. Die Unterschiede fallen jedoch gering aus. Es gibt noch eine Reihe anderer Methoden, balancierte Trainingsdatensätze zu erzeugen (Subsample Majority Class, Oversample Minority Class, Adding Noise, SMOTE). Herzstück einer jeder dieser Methoden ist jedoch ein Zufallsmechanismus. Möchte man für diesen korrigieren, so kann man den gesamten Trainingsprozess (Balanceierung, Feature selection, Training) Teil einer großen Simulation machen. Die resultierenden Ergebnisse kann man mit Hilfe eines Stacking-Ansatzes zusammenfügen, was jedoch nur für nicht-Baum-basierte Methoden sinnvoll ist. An dieser Stelle wird auf eine solche große Simulation verzichtet, sie kann allerdings jederzeit nachgeholt werden.

Schließlich extrahieren wir noch die Namen der Features:

```
## Features
features <- names(df)[1:(ncol(df) - 1)]
```

3 Feature selection

Unser bisheriges Vorgehen war von Methoden geprägt, welche aus der Menge an vorhandenen, hoch kollinearen Features automatisch die relevanten Merkmale innerhalb des Trainingsprozesses erkennen (Lasso, modellbasiertes Boosting, Klassifikationsbaum). In der vorliegenden Analyse verlassen wir diesen Weg und betrachten stattdessen univariate Filtermethoden.

3.1 Die Methoden

Wir betrachten vier verschiedene Methoden für die Extrahierung der relevanten Features: Methode nach Wilcoxon, AUC-Kriterium, Maximierung der *mutual information* und der Ansatz der maximum relevance minimum redundancy.

3.1.1 Wilcoxon

Bei dieser Methode betrachten wir die metrischen Features als Zielgrößen und untersuchen diese hinsichtlich eines Unterschiedes entlang der Response-Kategorien. Bei größeren Stichprobenumfängen (wie auch hier) erhält man in fast jedem Fall einen signifikanten Unterschied zwischen den Responsekategorien. Die Größe der entsprechenden p-Werte eignet sich somit nicht für eine Auswahl der Features. Stattdessen trifft man eine Entscheidung anhand der Größe des Abstands der Lageparameter. Oftmals wird hier fälschlicherweise der Abstand der Mediane zwischen den Responsekategorien herangezogen. Wilcoxon's Test betrachtet aber stattdessen den Median der Differenzen zwischen den beiden Kategorien.

3.1.2 AUC

Bei diesem Kriterium betrachten wir die prognostische Güte eines jeden Features mit Hilfe einer ROC-Kurve. Hierbei wird entlang des gesamten Wertebereich des Features geprüft, wie hoch die Sensitivität und Spezifität in Hinblick auf eine Prognose der Response-Variable wäre, wenn der jeweilige Punkt im Wertebereich als Cutoff gewählt wird. Der Bereich unter der so entstehenden ROC-Kurve ist der AUC-Wert. Eine Auswahl der Features erfolgt dann anhand der Höhe der AUC-Werte.

3.1.3 Mutual information

Dieser Ansatz wählt die Features aus, welche mit der Response-Variable die größte mutual information aufweist. Das Konzept der mutual information ist ein sehr theoretischer Ansatz welcher sich nicht ohne weiteres auf beobachtete Daten anwenden lässt. Das in dieser Analyse verwendete Paket **mRMRe** approximiert daher die mutual information mit Hilfe von einfachen Korrelationskoeffizienten (nach Pearson, Spearman oder Cramer). Nach diesem Ansatz werden dann die Features ausgewählt, welche die höchste mutual information mit der Response-Variable aufweisen.

3.1.4 Maximum relevance minimum redundancy

Dieser Ansatz basiert auf dem der mutual information. Neben einer Maximierung der mutual information mit der Response-Variable wird aber gleichzeitig versucht, die mutual information mit den bereits ausgewählten Features zu minimieren.

3.2 Umsetzung

Für die praktische Umsetzung in R schreiben wir uns eine eigene Funktion, welcher man neben einem Datensatz eine Liste der Feature-Namen übergibt. Die Funktion führt dann die Feature Extrahierung selbständig durch.

Hier die Funktion:

```
ps_feature_selection <- function(df, features){

 ## Correct order of data frame
 df <- df %>%
  dplyr::select(response, features)

 ## Function for computing AUC
 ps_auc <- function(reference, predictor){
  ps_sensitivity <- function(tab) {
   tab[2,2] / sum(tab[2,])
  }
  ## Spezifitaet
  ps_specifity <- function(tab) {
   tab[1,1] / sum(tab[1,])
  }
  f <- function(thr, reference, predictor){
   tab <- table(reference, factor(predictor > thr, levels = c(FALSE, TRUE)))
   sens <- ps_sensitivity(tab)
   spec <- ps_specifity(tab)
   return(c(sens, spec))
  }
  f <- Vectorize(f, vectorize.args = "thr")
  s <- seq(min(predictor), max(predictor), length.out = 256)
  out <- cbind(s, t(f(thr = s, reference = reference, predictor = predictor)))
  #print(plot(1 - out[,3], out[,2], type = "l"))

  roc <- approxfun(1 - out[,3], out[,2])
  auc <- try(integrate(f = roc, lower = min(1 - out[,3]), upper = max(1 - out[,3]))$value,
             silent = TRUE)
  if(class(auc) == "try-error"){
   auc <- sum(diff(out[,3]) * (out[2:nrow(out),2] + out[1:(nrow(out) - 1),2]) / 2)
  }
  if(auc < .5){
   f <- function(thr, reference, predictor){
    tab <- table(reference, factor(predictor < thr, levels = c(FALSE, TRUE)))
    sens <- ps_sensitivity(tab)
    spec <- ps_specifity(tab)
    return(c(sens, spec))
   }
   f <- Vectorize(f, vectorize.args = "thr")
   out <- cbind(s, t(f(thr = s, reference = reference, predictor = predictor)))
   roc <- approxfun(1 - out[,3], out[,2])
   auc <- try(integrate(f = roc, lower = min(1 - out[,3]), upper = max(1 - out[,3]))$value,
              silent = TRUE)
   if(class(auc) == "try-error"){
    auc <- sum(diff(out[,3]) * (out[2:nrow(out),2] + out[1:(nrow(out) - 1),2]) / 2)
   }
  }
  tibble(
   FPR = seq(0, 1, length.out = 256),
   Sensitivity = roc(FPR),
   AUC = auc
  )
 }

 ## ======================================
 ## Feature selection by Wilcoxon and AUC
 ## ======================================

 pval_wlcx <- numeric(length(features))
 diff_wlcx <- 0 * pval_wlcx
 auc <- 0 * pval_wlcx
 for(j in 1:length(features)){
  res <- wilcox.test(df[[features[j]]] ~ df$response, conf.int = TRUE)
  pval_wlcx[j] <- res$p.value
  diff_wlcx[j] <- res$estimate
  auc[j] <- unique(ps_auc(reference = df$response, predictor = df[[features[j]]])$AUC)
 }

 ## Feature index wilcoxon
 fi_wlcx <- sort.list(abs(diff_wlcx), decreasing = TRUE)
```

```
  ## Feature index auc
  fi_auc <- sort.list(auc, decreasing = TRUE)

  ## ========================================
  ## Feature selection by mutual information
  ## ========================================

  df$response <- ordered(df$response)
  dd <- mRMR.data(data = df %>% as.data.frame)
  filter <- mRMR.classic(data = dd,
                         target_indices = c(1),
                         feature_count = length(features))

  ## Maximize mutual information
  fi_mi <- sort.list(mim(filter)[-1,1], decreasing = TRUE)

  ## maximizing the MI with y (maximum relevance) and minimizing the
  ## average MI with all the previously selected variables
  fi_mrmi <- solutions(filter)[[1]][,1] - 1

  ## ================
  ## Collect results
  ## ================

  tibble(
    features = features,
    wlcx = fi_wlcx,
    auc = fi_auc,
    mi = fi_mi,
    mrmi = fi_mrmi
  )

}
```

3.3 Standardisierung der Features

Bevor obige Funktion an einem Datensatz angewandt werden kann, müssen die Features standardisiert werden. Hierzu ziehen wir den Mittelwert ab und teilen das Ergebnis durch die Standardabweichung.

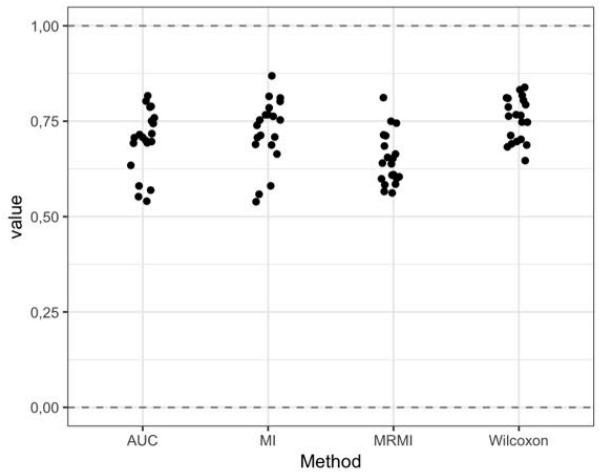
3.4 Stabilität

Es ist üblich die Feature selection Methoden auf ihre Stabilität hin zu untersuchen. Ein beliebtes Vorgehen hierbei ist es, den Datensatz in zwei gleichgroße Teile zu zerlegen, die Features zu selektieren und zu überprüfen, inwiefern die ausgewählten Variablen übereinstimmen. Im folgenden führen wir eine solche Analyse simulationsbasiert durch, d.h. wir wiederholen die zufällige Teilung des Datensatzes n_{sim}-Mal und halten jedesmal die Übereinstimmung der beiden Partitionen fest. Das Ergebnis stellen wir anschließend grafisch dar. Bereits an diesem Punkt müssen wir die Anzahl der Features festhalten (hier 20). Der gesamte Code der Simulation lautet:

```
## Number of simulations
n_sim <- 20

## Result
overlap <- matrix(nrow = n_sim, ncol = 4)

## Number of features to extract
n_features <- 20

## Main loop
for(i in 1:n_sim){

 cat(i, "\n")

 ## ===============
 ## Select subsets
 ## ===============

 indx1 <- c(sample(x = which(df$response == 0), size = 0.5 * sum(df$response == 0)),
         sample(x = which(df$response == 1), size = 0.5 * sum(df$response == 1)))
 indx2 <- (1:nrow(df))[-indx1]

 df_indx1 <- df[indx1,]
 df_indx2 <- df[indx2,]

 ## ====================
 ## Scaling of features
 ## ====================

 for(j in 1:length(features)){
  df_indx1[[features[j]]] <- (df_indx1[[features[j]]] - mean(df_indx1[[features[j]]])) /
   sd(df_indx1[[features[j]]])

  df_indx2[[features[j]]] <- (df_indx2[[features[j]]] - mean(df_indx2[[features[j]]])) /
   sd(df_indx2[[features[j]]])
 }

 ## ==================
 ## Feature selection
 ## ==================

 df_indx1_fs <- ps_feature_selection(df = df_indx1, features = features)
 df_indx2_fs <- ps_feature_selection(df = df_indx2, features = features)

 ## Overlap
 for(j in 2:5){
  overlap[i,j-1] <- mean(df_indx1_fs[[j]][1:n_features] %in% df_indx2_fs[[j]][1:n_features])
 }
}
```

```
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
```

```
colnames(overlap) <- c("Wilcoxon", "AUC", "MI", "MRMI")

## Graph
overlap <- overlap %>%
  as.tibble %>%
  gather(key = "Method", value = "value")

p <- ggplot(overlap, aes(x = Method, y = value))
p + geom_hline(yintercept = 0:1, linetype = 2, colour = grey(.5)) +
  geom_jitter(width = .1)
```



Am stabilsten scheint die Selektion nach dem Wilcoxon-Kriterium zu verlaufen. Die numerische Zusammenfassung lautet:

```
overlap %>%
  group_by(Method) %>%
  summarize(Mean = mean(value),
            SD = sd(value),
            Min = min(value),
            Max = max(value))
```

```
## # A tibble: 4 x 5
##   Method    Mean     SD   Min   Max
##   <chr>    <dbl>  <dbl> <dbl> <dbl>
## 1 AUC      0.698 0.0819  0.55  0.8
## 2 MI       0.72  0.0818  0.55  0.85
## 3 MRMI     0.648 0.0678  0.55  0.8
## 4 Wilcoxon 0.755 0.0560  0.65  0.85
```

4 Training

Wir sind nun soweit und können mit dem eigentlichen Training beginnen. Hierfür standardisieren wir zuerst die Features und wenden dann die Feature selection Methoden an:

```
## Scaling of features
## We must keep the means and standard deviations in order to correctly scale
## the test data
m <- numeric(length(features))
s <- m
for(j in 1:length(features)){
  m[j] <- mean(df[[features[j]]])
  s[j] <- sd(df[[features[j]]])
  df[[features[j]]] <- (df[[features[j]]] - m[j]) / s[j]
}

## Perform feature selection for training data
df_fs <- ps_feature_selection(df = df, features = features)
df_fs
```

```
## # A tibble: 100 x 5
##    features                          wlcx  auc   mi  mrmi
##    <chr>                            <int> <int> <int> <dbl>
##  1 original_shape_VoxelVolume          84    73    86    86
##  2 original_shape_Maximum3DDiameter    80    44    41    44
##  3 original_shape_MeshVolume           78    80    73    68
##  4 original_shape_MajorAxisLength      47    75    91    94
##  5 original_shape_Sphericity           25    45    44    49
##  6 original_shape_LeastAxisLength      45    84    80    73
##  7 original_shape_Elongation            8    41    75    95
##  8 original_shape_SurfaceVolumeRatio   26    74    45    59
##  9 original_shape_Maximum2DDiameterSlice 39  78    74    93
## 10 original_shape_Flatness             35     6    84    55
## # … with 90 more rows
```

```
write.csv2(df_fs[1:20,],
        file = "~/201801_Makowski/03_Analyse/Penzkofer/20190328_Feature_selection.csv",
        row.names = FALSE)
```

Wir müssen an dieser Stelle unbedingt die Mittelwerte und Standardabweichungen der Features speichern, da diese auf den Testdatensatz angewandt werden müssen.

Die Feature Selection setzen wir um, indem wir die selektierten Variablen in Formeln schreiben. Wir beschränken uns dabei auf die oben gewählten $n_{\text{features}} = 20$ Variablen:

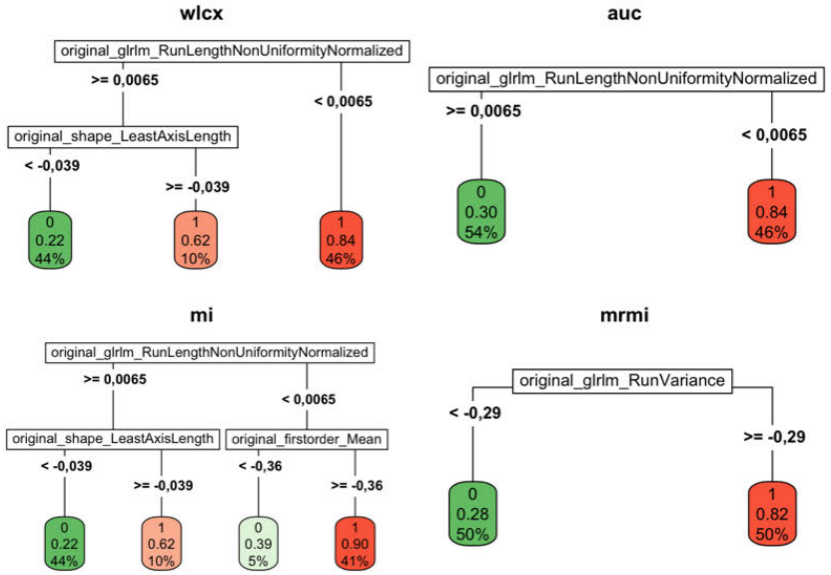
```
formulas <- list(
  ## Formula for Wilcoxon
  wlcx = as.formula(paste("response ~",
                          paste(df_fs$features[df_fs$wlcx[1:n_features]], collapse = "+"))),
  ## Formula for AUC
  auc = as.formula(paste("response ~",
                         paste(df_fs$features[df_fs$auc[1:n_features]], collapse = "+"))),
  ## Formula for MI
  mi = as.formula(paste("response ~",
                        paste(df_fs$features[df_fs$mi[1:n_features]], collapse = "+"))),
  ## Formula for MRMI
  mrmi = as.formula(paste("response ~",
                          paste(df_fs$features[df_fs$mrmi[1:n_features]], collapse = "+")))
)
```

Nun können wir mit dem eigentlichen Training beginnen. Im folgenden verwenden wir fünf Algorithmen für die Klassifikation der positiven und negativen Lymphknoten: lineare Diskriminanzanalyse (lda), logistische Regression (logistic), partial least squares (pls), support vector machines (svm), neuronal network (multilayer perceptron, mlp) und Klassifikationsbaum (rpart). Für einige dieser Algorithmen werden mit Hilfe eines Bootstrap-Ansatzes die optimalen Werte bestimmter Parameter bestimmt. Dieser Ansatz kann etwas zeitintensiv sein. Folgender Code führt das Training aus:

```
## Linear Discriminant Analysis
fit_lda <- lapply(formulas, function(f) {
  lda(formula = f, data = df)
})

## Logistic regression
fit_logistic <- lapply(formulas, function(f) {
  glm(formula = f, data = df, family = binomial())
})

## Partial least squares
fit_pls <- lapply(formulas, function(f) {
  caret::train(form = f,
               data = df,
               method = "pls",
               tuneGrid = data.frame(ncomp = 1:15),
               trControl = trainControl(number = 10))
})

## Support vector machines
fit_svm <- lapply(formulas, function(f) {
  caret::train(form = f,
               data = df,
               method = "svmLinear2",
               tuneGrid = data.frame(cost = seq(0.1, 1, by = .1)),
               trControl = trainControl(number = 10))
})

## Neuronal network (multilayer perceptron)
fit_mlp <- lapply(formulas, function(f) {
  caret::train(form = f,
               data = df,
               method = "mlp",
               tuneGrid = data.frame(size = 1:5),
               trControl = trainControl(number = 10))
})

## Recursive partition
fit_rpart <- lapply(formulas, function(f) {
  caret::train(form = f,
               data = df,
               method = "rpart",
               tuneGrid = data.frame(cp = seq(0.01, 0.15, by = .01)),
               trControl = trainControl(number = 10))
})
```

Die resultierenden Klassifikationsbäume für die vier Feature Selection Methoden sehen wie folgt aus:

```
par(mfrow = c(2, 2), mar = c(0, 0, 2, 0))
for(i in 1:4){
  fit <- rpart(formulas[[i]],
               data = df,
               control = rpart.control(cp = fit_rpart[[i]]$bestTune))
  rpart.plot(fit, type = 5, box.palette = "GnRd", main = names(formulas)[i])
}
```



5 Testing

Mit Hilfe des externen Validierungsdatensatzes sind wir in der Lage die Prognosegüte der Methoden zu bestimmen. Zunächst müssen die Variablen aber auf die gleiche Art und Weise standardisiert werden:

```
for(j in 1:length(features)){
  df_test[[features[j]]] <- (df_test[[features[j]]] - m[j]) / s[j]
}
```

Nun erzeugen wir die Vorhersagen:

```
## Prediction for LDA
pr_lda <- lapply(fit_lda, function(fit) predict(fit, newdata = df_test)$class)

## Prediction for logistic regression
pr_logistic <- lapply(fit_logistic,
                      function(fit) factor((predict(fit, newdata = df_test, type = "response") > .5) * 1))

## Prediction for Partial least squares
pr_pls <- lapply(fit_pls, function(fit) predict(fit, newdata = df_test))

## Prediction for Support vector machines
pr_svm <- lapply(fit_svm, function(fit) predict(fit, newdata = df_test))

## Prediction for neural network
pr_mlp <- lapply(fit_mlp, function(fit) predict(fit, newdata = df_test))

## Prediction for classification tree
pr_rpart <- lapply(fit_rpart, function(fit) predict(fit, newdata = df_test))
```

Und berechnen die Evaluationsmetriken:

```
## Evaluation for lda
eval_lda <- lapply(pr_lda,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for logistic
eval_logistic <- lapply(pr_logistic,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for pls
eval_pls <- lapply(pr_pls,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for svm
eval_svm <- lapply(pr_svm,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for mlp
eval_mlp <- lapply(pr_mlp,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for rpart
eval_rpart <- lapply(pr_rpart,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))
```

Diese werden nun noch etwas feiner extrahiert und schließlich zu einem Datensatz zusammengefasst:

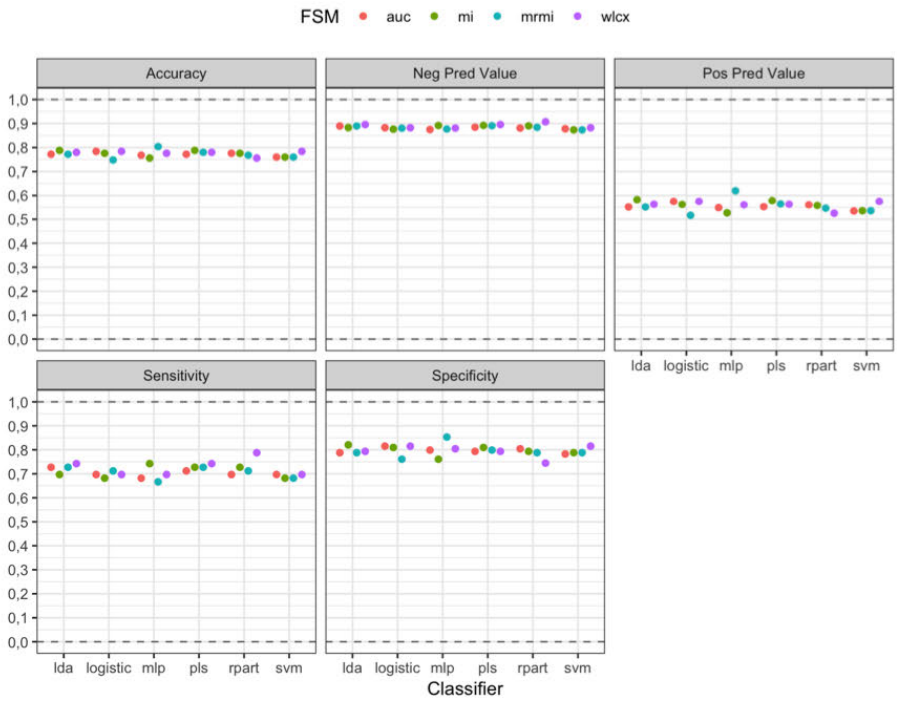
```
## lda
eval_lda <- Reduce("rbind", lapply(eval_lda, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_lda),
         Classifier = "lda")

## logistic
eval_logistic <- Reduce("rbind", lapply(eval_logistic, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_logistic),
         Classifier = "logistic")

## pls
eval_pls <- Reduce("rbind", lapply(eval_pls, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_pls),
         Classifier = "pls")

## svm
eval_svm <- Reduce("rbind", lapply(eval_svm, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_svm),
         Classifier = "svm")

## mlp
eval_mlp <- Reduce("rbind", lapply(eval_mlp, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_mlp),
         Classifier = "mlp")

## rpart
eval_rpart <- Reduce("rbind", lapply(eval_rpart, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_rpart),
         Classifier = "rpart")

## One complete data set
df_eval <- rbind(eval_lda,
                 eval_logistic,
                 eval_pls,
                 eval_svm,
                 eval_mlp,
                 eval_rpart)

## Output
kable(df_eval[,c(7,6,1:5)], digits = 2)
```

Classifier	FSM	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
lda	wlcx	0,78	0,74	0,79	0,56	0,90
lda	auc	0,77	0,73	0,79	0,55	0,89
lda	mi	0,79	0,70	0,82	0,58	0,88
lda	mrmi	0,77	0,73	0,79	0,55	0,89
logistic	wlcx	0,78	0,70	0,82	0,57	0,88
logistic	auc	0,78	0,70	0,82	0,57	0,88
logistic	mi	0,78	0,68	0,81	0,56	0,88
logistic	mrmi	0,75	0,71	0,76	0,52	0,88
pls	wlcx	0,78	0,74	0,79	0,56	0,90
pls	auc	0,77	0,71	0,79	0,55	0,88
pls	mi	0,79	0,73	0,81	0,58	0,89
pls	mrmi	0,78	0,73	0,80	0,56	0,89
svm	wlcx	0,78	0,70	0,82	0,57	0,88

Classifier	FSM	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
svm	auc	0,76	0,70	0,78	0,53	0,88
svm	mi	0,76	0,68	0,79	0,54	0,87
svm	mrmi	0,76	0,68	0,79	0,54	0,87
mlp	wlcx	0,78	0,70	0,80	0,56	0,88
mlp	auc	0,77	0,68	0,80	0,55	0,88
mlp	mi	0,76	0,74	0,76	0,53	0,89
mlp	mrmi	0,80	0,67	0,85	0,62	0,88
rpart	wlcx	0,76	0,79	0,74	0,53	0,91
rpart	auc	0,78	0,70	0,80	0,56	0,88
rpart	mi	0,78	0,73	0,79	0,56	0,89
rpart	mrmi	0,77	0,71	0,79	0,55	0,88

Obige Werte können wir auch grafisch darstellen:

```
df_eval <- df_eval %>%
  gather(-FSM, -Classifier, key = "Metrics", value = "value")

p <- ggplot(df_eval, aes(x = Classifier, y = value, colour = FSM))
p + geom_hline(yintercept = 0:1, linetype = 2, colour = grey(.5)) +
  geom_point(position = position_dodge(width = .75)) +
  facet_wrap(~ Metrics) +
  scale_y_continuous(breaks = seq(0, 1, by = .1)) +
  theme(legend.position = "top") +
  labs(y = NULL)
```



Bis auf den positiven prädiktiven Wert (Wahrscheinlichkeit, dass ein als positiv erkannter Lymphknoten tatsächlich positiv ist) erhalten wir akzeptable Werte.

5.1 Vergleich mit der Einschätzung der Radiologen

Es liegen nun die Einschätzungen von zwei Radiologen vor. Diese hatten die Aufgabe, sämtliche Beobachtungen in vier Klassen einzuordnen (0 = sicher nicht, 1 = wahrscheinlich nicht, 2 = wahrscheinlich, 3 = sicher). Mit diesen Werten soll überprüft werden, ob die Klassifikationsalgorithmen besser als die Radiologen abschneiden und ob die Klassifikationsalgorithmen in den beiden mittleren Kategorien einen Mehrwert liefern.

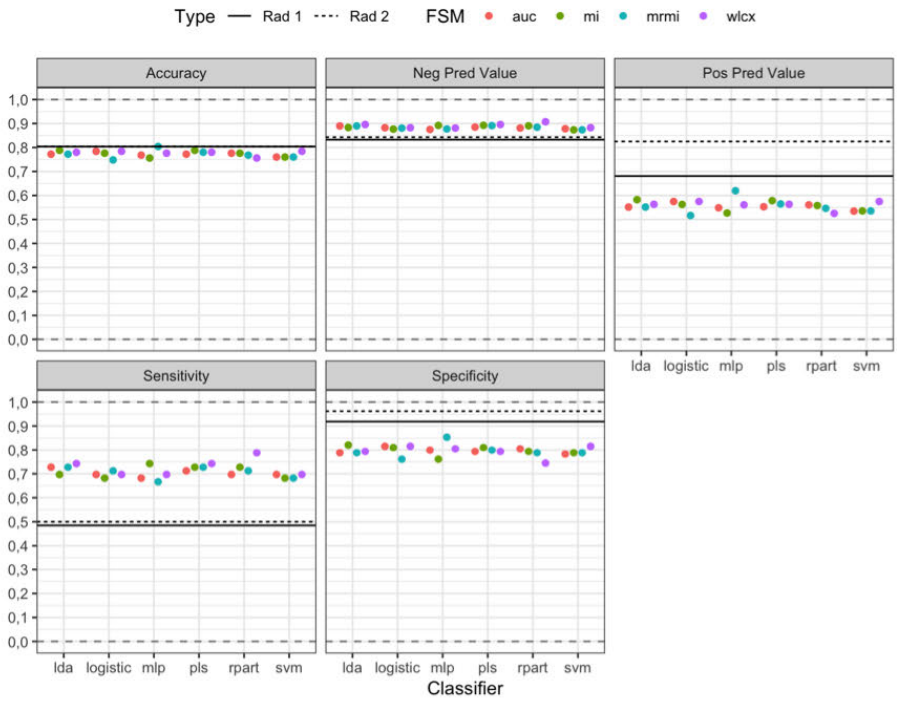
Beginnen wir mit einem Vergleich der Gesamtperformance. Um die Prognosen der Radiologen mit der Referenz vergleichen zu können müssen wir die Werte der Radiologen ebenfalls auf nur zwei Kategorien bringen. Hierzu verwenden wir den Grenzwert bei 1,5, d.h. die Klassen 0 und 1 werden zu 0 und die Klassen 2 und 3 zu Klasse 1 zusammengefasst. Damit ergeben sich folgende Performance-Werte:

```
## Binary predictions
df_test$pr_rad1 <- factor(1 * (df_test$predicted_class_rad1 > 1),
                          levels = 0:1)
df_test$pr_rad2 <- factor(1 * (df_test$predicted_class_rad2 > 1),
                          levels = 0:1)

## Confusion matrices
cm_rad1 <- confusionMatrix(data = df_test$pr_rad1, reference = df_test$response, positive = "1")
cm_rad2 <- confusionMatrix(data = df_test$pr_rad2, reference = df_test$response, positive = "1")

## Data set for graph
df_rad1_rad2 <- tibble(
  Type = c("Rad 1", "Rad 2"),
  Accuracy = c(cm_rad1$overall[1],
               cm_rad1$overall[1]),
  "Neg Pred Value" = c(cm_rad1$byClass[4],
                       cm_rad2$byClass[4]),
  "Pos Pred Value" = c(cm_rad1$byClass[3],
                       cm_rad2$byClass[3]),
  Sensitivity = c(cm_rad1$byClass[1],
                  cm_rad2$byClass[1]),
  Specificity = c(cm_rad1$byClass[2],
                  cm_rad2$byClass[2])
)

kable(df_rad1_rad2[,c(1:2,5:6,4:3)], digits = 2)
```

Type	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
Rad 1	0,8	0,48	0,92	0,68	0,83
Rad 2	0,8	0,50	0,96	0,83	0,84

Die Sensitivität der Radiologen ist um einiges schlechter, dafür ist aber die Spezifität etwas besser. Grafisch lässt sich das wie folgt veranschaulichen:

```
df_rad1_rad2_overall <- df_rad1_rad2 %>%
  gather(-Type, key = "Metrics", value = "value")

p <- ggplot(df_eval, aes(x = Classifier, y = value, colour = FSM))
p + geom_hline(yintercept = 0:1, linetype = 2, colour = grey(.5)) +
  geom_point(position = position_dodge(width = .75)) +
  facet_wrap(~ Metrics) +
  scale_y_continuous(breaks = seq(0, 1, by = .1)) +
  theme(legend.position = "top") +
  labs(y = NULL) +
  geom_hline(data = df_rad1_rad2_overall, aes(yintercept = value, linetype = Type))
```



Nun wollen wir wissen, ob die Algorithmen einen Mehrwert in den mittleren Katogorien der Radiologen liefern können. Hierzu verwenden wir für diese Werte die Prognosen der Algorithmen, d.h. wir erstellen wieder binäre Prognosen der Radiologen, für die Klassen 1 und 2 schreiten aber die Prognosen der Algorithmen ein. Das ist insgesamt etwas aufwändiger und liefert folgende Ergebnisse:

```
ps_correct_rad <- function(pr, pr_rad){
  pr_rad_tmp <- factor(1 * (pr_rad > 1), levels = 0:1)
  pr_rad_tmp[pr_rad %in% c(1,2) & pr == 1] <- 1
  pr_rad_tmp[pr_rad %in% c(1,2) & pr == 0] <- 0
  pr_rad_tmp
}

pr_logistic_rad1 <- lapply(pr_logistic,
                    function(pr) ps_correct_rad(pr = pr,
                                                pr_rad = df_test$predicted_class_rad1))
pr_lda_rad1 <- lapply(pr_lda,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad1))
pr_pls_rad1 <- lapply(pr_pls,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad1))
pr_svm_rad1 <- lapply(pr_svm,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad1))
pr_mlp_rad1 <- lapply(pr_mlp,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad1))
pr_rpart_rad1 <- lapply(pr_rpart,
                 function(pr) ps_correct_rad(pr = pr,
                                             pr_rad = df_test$predicted_class_rad1))


pr_logistic_rad2 <- lapply(pr_logistic,
                    function(pr) ps_correct_rad(pr = pr,
                                                pr_rad = df_test$predicted_class_rad2))
pr_lda_rad2 <- lapply(pr_lda,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad2))
pr_pls_rad2 <- lapply(pr_pls,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad2))
pr_svm_rad2 <- lapply(pr_svm,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad2))
pr_mlp_rad2 <- lapply(pr_mlp,
                function(pr) ps_correct_rad(pr = pr,
                                            pr_rad = df_test$predicted_class_rad2))
pr_rpart_rad2 <- lapply(pr_rpart,
                 function(pr) ps_correct_rad(pr = pr,
                                             pr_rad = df_test$predicted_class_rad2))


## Evaluation for lda
eval_lda_rad1 <- lapply(pr_lda_rad1,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for logistic
eval_logistic_rad1 <- lapply(pr_logistic_rad1,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for pls
eval_pls_rad1 <- lapply(pr_pls_rad1,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for svm
eval_svm_rad1 <- lapply(pr_svm_rad1,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for mlp
eval_mlp_rad1 <- lapply(pr_mlp_rad1,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for rpart
eval_rpart_rad1 <- lapply(pr_rpart_rad1,
            function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))
```

```
## Evaluation for lda
eval_lda_rad2 <- lapply(pr_lda_rad2,
                function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for logistic
eval_logistic_rad2 <- lapply(pr_logistic_rad2,
                   function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for pls
eval_pls_rad2 <- lapply(pr_pls_rad2,
                function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for svm
eval_svm_rad2 <- lapply(pr_svm_rad2,
                function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for mlp
eval_mlp_rad2 <- lapply(pr_mlp_rad2,
                function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))

## Evaluation for rpart
eval_rpart_rad2 <- lapply(pr_rpart_rad2,
                 function(pr) caret::confusionMatrix(data = pr, reference = df_test$response, positive = "1"))



## lda
eval_lda_rad1 <- Reduce("rbind", lapply(eval_lda_rad1, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_lda_rad1),
         Classifier = "lda")

## logistic
eval_logistic_rad1 <- Reduce("rbind", lapply(eval_logistic_rad1, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_logistic_rad1),
         Classifier = "logistic")

## pls
eval_pls_rad1 <- Reduce("rbind", lapply(eval_pls_rad1, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_pls_rad1),
         Classifier = "pls")

## svm
eval_svm_rad1 <- Reduce("rbind", lapply(eval_svm_rad1, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_svm_rad1),
         Classifier = "svm")

## mlp
eval_mlp_rad1 <- Reduce("rbind", lapply(eval_mlp_rad1, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_mlp_rad1),
         Classifier = "mlp")

## rpart
eval_rpart_rad1 <- Reduce("rbind", lapply(eval_rpart_rad1, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_rpart_rad1),
         Classifier = "rpart")

## One complete data set
df_eval_rad1 <- rbind(eval_lda_rad1,
              eval_logistic_rad1,
              eval_pls_rad1,
              eval_svm_rad1,
              eval_mlp_rad1,
              eval_rpart_rad1)
```

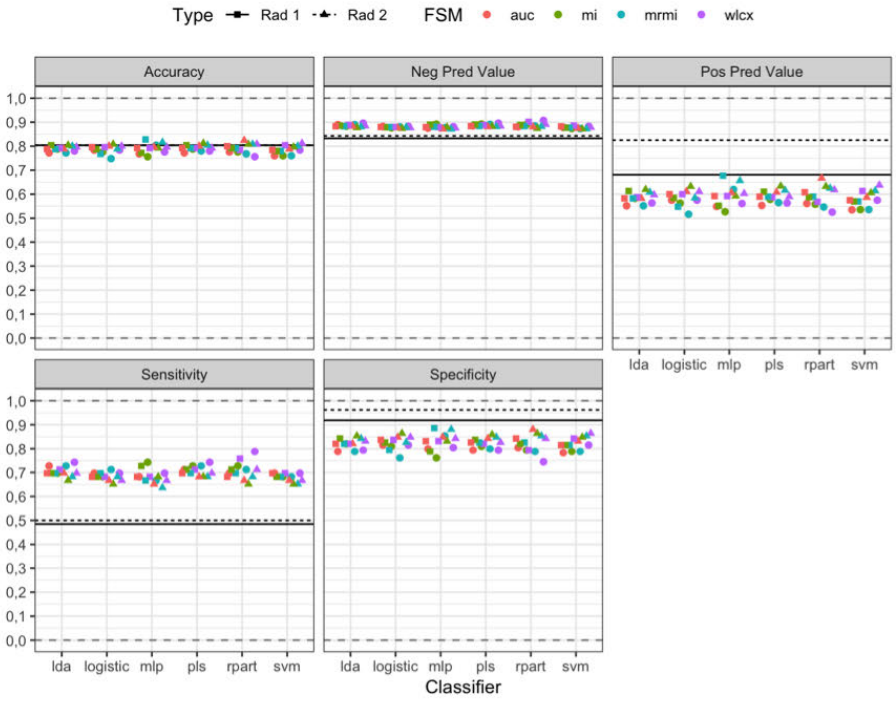
```
## lda
eval_lda_rad2 <- Reduce("rbind", lapply(eval_lda_rad2, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_lda_rad2),
         Classifier = "lda")

## logistic
eval_logistic_rad2 <- Reduce("rbind", lapply(eval_logistic_rad2, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_logistic_rad2),
         Classifier = "logistic")

## pls
eval_pls_rad2 <- Reduce("rbind", lapply(eval_pls_rad2, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_pls_rad2),
         Classifier = "pls")

## svm
eval_svm_rad2 <- Reduce("rbind", lapply(eval_svm_rad2, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_svm_rad2),
         Classifier = "svm")

## mlp
eval_mlp_rad2 <- Reduce("rbind", lapply(eval_mlp_rad2, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_mlp_rad2),
         Classifier = "mlp")

## rpart
eval_rpart_rad2 <- Reduce("rbind", lapply(eval_rpart_rad2, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(eval_rpart_rad2),
         Classifier = "rpart")

## One complete data set
df_eval_rad2 <- rbind(eval_lda_rad2,
                      eval_logistic_rad2,
                      eval_pls_rad2,
                      eval_svm_rad2,
                      eval_mlp_rad2,
                      eval_rpart_rad2)

df_eval_rad1_rad2 <- rbind(df_eval_rad1, df_eval_rad2) %>%
  mutate(Type = c(rep("Rad 1", nrow(df_eval_rad1)),
                  rep("Rad 2", nrow(df_eval_rad2))))
## Output
kable(df_eval_rad1_rad2[,c(8,7,6,1:5)], digits = 2)
```

Type	Classifier	FSM	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
Rad 1	lda	wlcx	0,79	0,71	0,82	0,59	0,89
Rad 1	lda	auc	0,79	0,70	0,82	0,58	0,88
Rad 1	lda	mi	0,80	0,70	0,84	0,61	0,89
Rad 1	lda	mrmi	0,79	0,70	0,82	0,58	0,88
Rad 1	logistic	wlcx	0,80	0,68	0,84	0,60	0,88
Rad 1	logistic	auc	0,80	0,68	0,84	0,60	0,88
Rad 1	logistic	mi	0,79	0,68	0,83	0,58	0,88
Rad 1	logistic	mrmi	0,77	0,70	0,79	0,55	0,88
Rad 1	pls	wlcx	0,79	0,71	0,82	0,59	0,89
Rad 1	pls	auc	0,79	0,70	0,83	0,59	0,88
Rad 1	pls	mi	0,80	0,71	0,84	0,61	0,89

Type	Classifier	FSM	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
Rad 1	pls	mrmi	0,79	0,70	0,83	0,59	0,88
Rad 1	svm	wlcx	0,80	0,70	0,84	0,61	0,89
Rad 1	svm	auc	0,78	0,70	0,82	0,57	0,88
Rad 1	svm	mi	0,78	0,68	0,82	0,57	0,88
Rad 1	svm	mrmi	0,78	0,68	0,82	0,57	0,88
Rad 1	mlp	wlcx	0,79	0,68	0,83	0,59	0,88
Rad 1	mlp	auc	0,79	0,68	0,83	0,59	0,88
Rad 1	mlp	mi	0,77	0,73	0,79	0,55	0,89
Rad 1	mlp	mrmi	0,83	0,67	0,89	0,68	0,88
Rad 1	rpart	wlcx	0,78	0,76	0,79	0,57	0,90
Rad 1	rpart	auc	0,80	0,68	0,84	0,61	0,88
Rad 1	rpart	mi	0,79	0,71	0,82	0,59	0,89
Rad 1	rpart	mrmi	0,79	0,70	0,83	0,59	0,88
Rad 2	lda	wlcx	0,80	0,70	0,83	0,60	0,88
Rad 2	lda	auc	0,79	0,70	0,82	0,58	0,88
Rad 2	lda	mi	0,80	0,67	0,85	0,62	0,88
Rad 2	lda	mrmi	0,80	0,68	0,84	0,61	0,88
Rad 2	logistic	wlcx	0,80	0,67	0,85	0,61	0,88
Rad 2	logistic	auc	0,80	0,67	0,85	0,61	0,88
Rad 2	logistic	mi	0,81	0,65	0,86	0,63	0,87
Rad 2	logistic	mrmi	0,79	0,68	0,83	0,58	0,88
Rad 2	pls	wlcx	0,79	0,70	0,83	0,59	0,88
Rad 2	pls	auc	0,80	0,68	0,84	0,61	0,88
Rad 2	pls	mi	0,81	0,68	0,86	0,63	0,88
Rad 2	pls	mrmi	0,80	0,68	0,85	0,62	0,88
Rad 2	svm	wlcx	0,81	0,67	0,86	0,64	0,88
Rad 2	svm	auc	0,79	0,67	0,83	0,59	0,87
Rad 2	svm	mi	0,80	0,65	0,85	0,61	0,87
Rad 2	svm	mrmi	0,80	0,65	0,85	0,61	0,87
Rad 2	mlp	wlcx	0,80	0,67	0,84	0,60	0,88
Rad 2	mlp	auc	0,80	0,65	0,85	0,61	0,87
Rad 2	mlp	mi	0,79	0,68	0,83	0,59	0,88
Rad 2	mlp	mrmi	0,82	0,64	0,88	0,66	0,87
Rad 2	rpart	wlcx	0,81	0,71	0,84	0,62	0,89
Rad 2	rpart	auc	0,82	0,67	0,88	0,67	0,88
Rad 2	rpart	mi	0,81	0,65	0,86	0,63	0,87
Rad 2	rpart	mrmi	0,81	0,68	0,85	0,62	0,88

Wir sehen mehr, wenn wir die Ergebnisse in die bekannte Grafik einzeichnen:

```
df_eval_rad1_rad2 <- df_eval_rad1_rad2 %>%
  gather(-FSM, -Classifier, -Type, key = "Metrics", value = "value")

p <- ggplot(df_eval, aes(x = Classifier, y = value, colour = FSM))
p + geom_hline(yintercept = 0:1, linetype = 2, colour = grey(.5)) +
  geom_point(position = position_dodge(width = .75)) +
  facet_wrap(~ Metrics) +
  scale_y_continuous(breaks = seq(0, 1, by = .1)) +
  theme(legend.position = "top") +
  labs(y = NULL) +
  geom_hline(data = df_rad1_rad2_overall, aes(yintercept = value, linetype = Type)) +
  geom_point(data = df_eval_rad1_rad2, aes(shape = Type), position = position_dodge(width = .75)) +
  scale_shape_manual(values = c(15, 17))
```



Die horizontalen Linien bezeichnen wieder die Performance-Maße ohne Hilfe der Algorithmen. Die Vier- und Dreiecke sind die neuen Werte. Insgesamt führt die Egänzung der Algorithmen zu einer ähnlichen Gesamtperformance, wir sind sensitiver.

5.2 Ergänzung

Ergänzend möchten wir nun überprüfen, ob es einen signifikanten Unterschied in den Performancewerten zwischen den Algorithmen und den Radiologen gibt, wenn wir uns nur auf die Beobachtungen beschränken, die von den Radiologen nicht mit voller Sicherheit klassifiziert wurden (Klassen 1 und 2). Den Nachweis eines signifikanten Unterschiedes führen wir hier über logistische Regressionsmodelle. Wählt man die Ziel- und Einflussgrößen entsprechend, so ist man mit diesem Ansatz in der Lage, Signifikanzaussagen über Sensitivität, Spezifität sowie positiv und negativen prädiktiven Wert zu treffen.

Wir betrachten zuerst die Einschätzung des ersten Radiologen. Folgender Code trägt die notwendigen Daten zusammen:

```
df_test_rad1_12 <- df_test %>%
  filter(predicted_class_rad1 %in% c(1,2))

df_test_rad1_12$pr_rad1 <- factor(1 * (df_test_rad1_12$predicted_class_rad1 > 1),
                                  levels = 0:1)

## Collect data
df_fit <- tibble(
  y = df_test_rad1_12$pr_rad1,
  x = df_test_rad1_12$response,
  type = "rad1"
)

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_rpart[[i]][df_test$predicted_class_rad1 %in% c(1,2)],
    x = df_test_rad1_12$response,
    type = paste("rpart", names(pr_rpart)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_lda[[i]][df_test$predicted_class_rad1 %in% c(1,2)],
    x = df_test_rad1_12$response,
    type = paste("lda", names(pr_lda)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_logistic[[i]][df_test$predicted_class_rad1 %in% c(1,2)],
    x = df_test_rad1_12$response,
    type = paste("logistic", names(pr_logistic)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_mlp[[i]][df_test$predicted_class_rad1 %in% c(1,2)],
    x = df_test_rad1_12$response,
    type = paste("mlp", names(pr_mlp)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))


df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_pls[[i]][df_test$predicted_class_rad1 %in% c(1,2)],
    x = df_test_rad1_12$response,
    type = paste("pls", names(pr_pls)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_svm[[i]][df_test$predicted_class_rad1 %in% c(1,2)],
    x = df_test_rad1_12$response,
    type = paste("svm", names(pr_svm)[i])
  )
)
df_fit <- rbind(df_fit,
```

```
        Reduce("rbind", df_tmp))

df_fit$type <- factor(df_fit$type,
                levels = c("rad1", unique(df_fit$type)[unique(df_fit$type) != "rad1"]))
```

Zunächst die deskriptiven Werte:

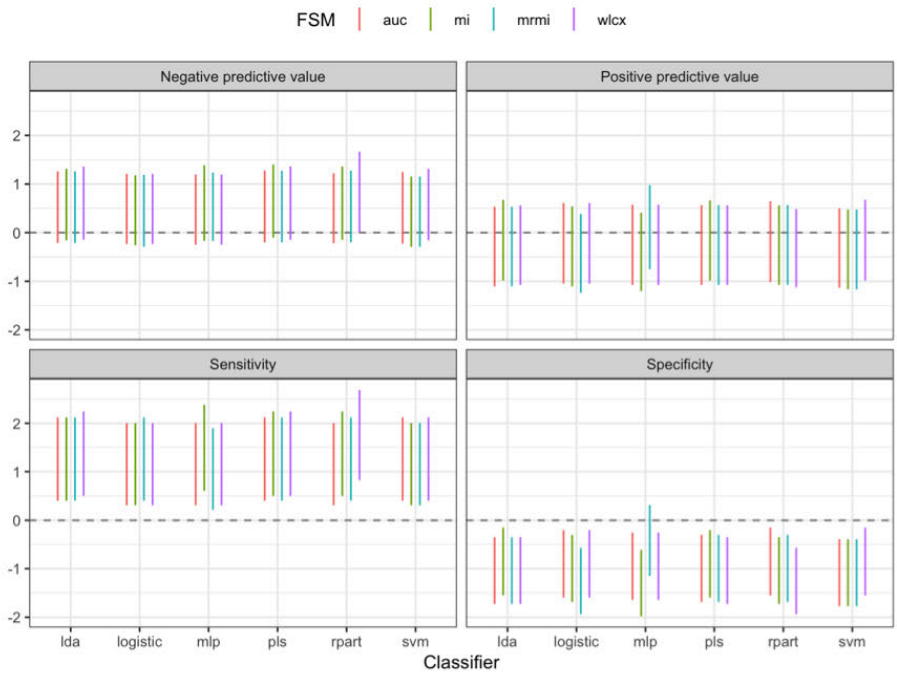
```
df_fit %>%
  group_by(type) %>%
  summarize(Sens = mean(y[x == 1] == 1),
            Spec = mean(y[x == 0] == 0),
            PPV = mean(x[y == 1] == 1),
            NPV = mean(x[y == 0] == 0)) %>%
  kable(digits = 2)
```

type	Sens	Spec	PPV	NPV
rad1	0,44	0,85	0,58	0,77
rpart wlcx	0,81	0,63	0,51	0,88
rpart auc	0,71	0,72	0,54	0,84
rpart mi	0,75	0,68	0,52	0,85
rpart mrmi	0,73	0,69	0,52	0,85
lda wlcx	0,75	0,68	0,52	0,85
lda auc	0,73	0,68	0,51	0,84
lda mi	0,73	0,72	0,55	0,85
lda mrmi	0,73	0,68	0,51	0,84
logistic wlcx	0,71	0,71	0,53	0,84
logistic auc	0,71	0,71	0,53	0,84
logistic mi	0,71	0,69	0,52	0,84
logistic mrmi	0,73	0,63	0,48	0,83
mlp wlcx	0,71	0,70	0,52	0,84
mlp auc	0,71	0,70	0,52	0,84
mlp mi	0,77	0,62	0,49	0,85
mlp mrmi	0,69	0,80	0,61	0,85
pls wlcx	0,75	0,68	0,52	0,85
pls auc	0,73	0,69	0,52	0,85
pls mi	0,75	0,71	0,55	0,86
pls mrmi	0,73	0,69	0,52	0,85
svm wlcx	0,73	0,72	0,55	0,85
svm auc	0,73	0,67	0,51	0,84
svm mi	0,71	0,67	0,50	0,83
svm mrmi	0,71	0,67	0,50	0,83

Nun schätzen wir die Modelle und stellen die Ergebnisse grafisch dar:

```
## Fit logistic regression models for Sens and Spec
fit <- glm(y ~ x * type, data = df_fit, family = binomial())
fit <- glm(y ~ x * type, data = df_fit, family = binomial())
ci <- confint(fit)

df_fit$y <- factor(df_fit$y, levels = 1:0)
df_fit$x <- factor(df_fit$x, levels = 1:0)
fit2 <- glm(y ~ x * type, data = df_fit, family = binomial())
ci2 <- confint(fit2)

## Collect results
df_res <- tibble(
  Classifier = c(gsub(pattern = "type", replacement = "", x = row.names(ci)[3:26]),
                 gsub(pattern = "x1:type", replacement = "", x = row.names(ci)[27:50])),
  Metric = c(rep("Specificity", 24),
             rep("Sensitivity", 24)),
  Lower = c(-ci[3:26,2],
            -ci2[3:26,2]),
  Upper = c(-ci[3:26,1],
            -ci2[3:26,1])
)
df_res$FSM <- lapply(strsplit(x = df_res$Classifier, split = " "), function(x) x[2]) %>% unlist
df_res$Classifier <- lapply(strsplit(x = df_res$Classifier, split = " "), function(x) x[1]) %>% unlist

## PPV and NPV

fit2 <- glm(x ~ y * type, data = df_fit, family = binomial())
ci2 <- confint(fit2)

df_fit$y <- factor(df_fit$y, levels = 0:1)
df_fit$x <- factor(df_fit$x, levels = 0:1)
fit <- glm(x ~ y * type, data = df_fit, family = binomial())
ci <- confint(fit)

df_res2 <- tibble(
  Classifier = c(gsub(pattern = "type", replacement = "", x = row.names(ci)[3:26]),
                 gsub(pattern = "y1:type", replacement = "", x = row.names(ci)[27:50])),
  Metric = c(rep("Negative predictive value", 24),
             rep("Positive predictive value", 24)),
  Lower = c(-ci[3:26,2],
            -ci2[3:26,2]),
  Upper = c(-ci[3:26,1],
            -ci2[3:26,1])
)
df_res2$FSM <- lapply(strsplit(x = df_res2$Classifier, split = " "), function(x) x[2]) %>%
  unlist
df_res2$Classifier <- lapply(strsplit(x = df_res2$Classifier, split = " "), function(x) x[1]) %>%
  unlist

df_res <- rbind(df_res, df_res2)

p <- ggplot(df_res, aes(x = Classifier, ymin = Lower, ymax = Upper, colour = FSM))
p + geom_hline(yintercept = 0, linetype = 2, colour = grey(.5)) +
  geom_linerange(position = position_dodge(width = .5)) +
  facet_wrap(~ Metric) +
  theme(legend.position = "top") +
  labs(title = "Radiologe 1")
```

Radiologe 1



Obige Grafik zeigt die 95% Konfidenzintervalle der Regressionskoeffizienzten, welche einen Unterschied zwischen den Algorithmen und Radiologen 1 zeigen. Statistisch abgesicherte Aussagen über die Unterschiede können wir für die Sensitivität und die Spezifität treffen (die KIs schneiden nicht die Null): Bezüglich der Sensitivität liefern die Algorithmen eine bessere Performance als Radiologe 1m bezüglich der Spezifität eine schlechtere Performance. Für den positiven und negativen prädiktiven Wert können wir keine statistisch abgesicherten Aussagen über die Richtung des Unterschiedes sagen (die KIs schneiden die Null).

Kommen wir zum zweiten Radiologen. Zunächst tragen wir wieder die Daten zusammen:

```
df_test_rad2_12 <- df_test %>%
  filter(predicted_class_rad2 %in% c(1,2))

df_test_rad2_12$pr_rad2 <- factor(1 * (df_test_rad2_12$predicted_class_rad2 > 1),
                                  levels = 0:1)

## Collect data
df_fit <- tibble(
  y = df_test_rad2_12$pr_rad1,
  x = df_test_rad2_12$response,
  type = "rad1"
)

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_rpart[[i]][df_test$predicted_class_rad2 %in% c(1,2)],
    x = df_test_rad2_12$response,
    type = paste("rpart", names(pr_rpart)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_lda[[i]][df_test$predicted_class_rad2 %in% c(1,2)],
    x = df_test_rad2_12$response,
    type = paste("lda", names(pr_lda)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_logistic[[i]][df_test$predicted_class_rad2 %in% c(1,2)],
    x = df_test_rad2_12$response,
    type = paste("logistic", names(pr_logistic)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_mlp[[i]][df_test$predicted_class_rad2 %in% c(1,2)],
    x = df_test_rad2_12$response,
    type = paste("mlp", names(pr_mlp)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))


df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_pls[[i]][df_test$predicted_class_rad2 %in% c(1,2)],
    x = df_test_rad2_12$response,
    type = paste("pls", names(pr_pls)[i])
  )
)
df_fit <- rbind(df_fit,
                Reduce("rbind", df_tmp))

df_tmp <- lapply(1:4, function(i)
  tibble(
    y = pr_svm[[i]][df_test$predicted_class_rad2 %in% c(1,2)],
    x = df_test_rad2_12$response,
    type = paste("svm", names(pr_svm)[i])
  )
)
df_fit <- rbind(df_fit,
```

```
        Reduce("rbind", df_tmp))

df_fit$type <- factor(df_fit$type,
                levels = c("rad1", unique(df_fit$type)[unique(df_fit$type) != "rad1"]))
```

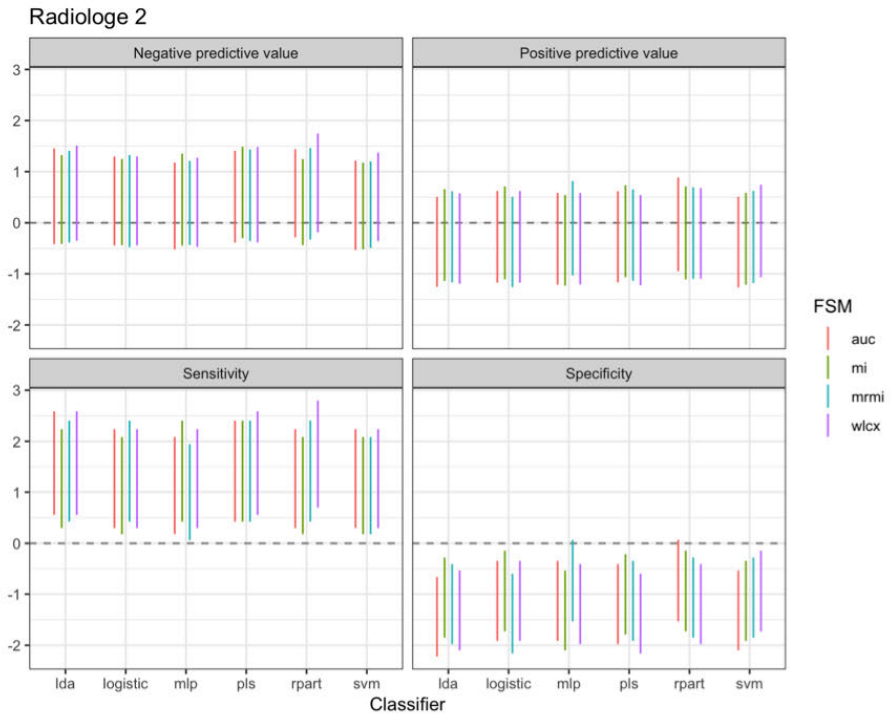
Die beobachteten Werte lauten wie folgt:

```
df_fit %>%
  group_by(type) %>%
  summarize(Sens = mean(y[x == 1] == 1),
            Spec = mean(y[x == 0] == 0),
            PPV = mean(x[y == 1] == 1),
            NPV = mean(x[y == 0] == 0)) %>%
  kable(digits = 2)
```

type	Sens	Spec	PPV	NPV
rad1	0,45	0,80	0,57	0,71
rpart wlcx	0,82	0,55	0,52	0,84
rpart auc	0,74	0,66	0,56	0,81
rpart mi	0,71	0,62	0,52	0,78
rpart mrmi	0,76	0,58	0,52	0,81
lda wlcx	0,79	0,52	0,49	0,81
lda auc	0,79	0,49	0,48	0,80
lda mi	0,74	0,58	0,51	0,79
lda mrmi	0,76	0,55	0,50	0,80
logistic wlcx	0,74	0,57	0,50	0,79
logistic auc	0,74	0,57	0,50	0,79
logistic mi	0,71	0,62	0,52	0,78
logistic mrmi	0,76	0,51	0,48	0,79
mlp wlcx	0,74	0,55	0,49	0,78
mlp auc	0,71	0,57	0,49	0,77
mlp mi	0,76	0,52	0,48	0,79
mlp mrmi	0,68	0,66	0,54	0,78
pls wlcx	0,79	0,51	0,48	0,80
pls auc	0,76	0,55	0,50	0,80
pls mi	0,76	0,60	0,53	0,81
pls mrmi	0,76	0,57	0,51	0,80
svm wlcx	0,74	0,62	0,53	0,80
svm auc	0,74	0,52	0,47	0,77
svm mi	0,71	0,57	0,49	0,77
svm mrmi	0,71	0,58	0,50	0,78

Die Modelle liefern dann folgendes Bild:

```
## Fit logistic regression models for Sens and Spec
fit <- glm(y ~ x * type, data = df_fit, family = binomial())
fit <- glm(y ~ x * type, data = df_fit, family = binomial())
ci <- confint(fit)

df_fit$y <- factor(df_fit$y, levels = 1:0)
df_fit$x <- factor(df_fit$x, levels = 1:0)
fit2 <- glm(y ~ x * type, data = df_fit, family = binomial())
ci2 <- confint(fit2)

## Collect results
df_res <- tibble(
  Classifier = c(gsub(pattern = "type", replacement = "", x = row.names(ci)[3:26]),
                 gsub(pattern = "x1:type", replacement = "", x = row.names(ci)[27:50])),
  Metric = c(rep("Specificity", 24),
             rep("Sensitivity", 24)),
  Lower = c(-ci[3:26,2],
            -ci2[3:26,2]),
  Upper = c(-ci[3:26,1],
            -ci2[3:26,1])
)
df_res$FSM <- lapply(strsplit(x = df_res$Classifier, split = " "), function(x) x[2]) %>% unlist
df_res$Classifier <- lapply(strsplit(x = df_res$Classifier, split = " "), function(x) x[1]) %>% unlist

## PPV and NPV

fit2 <- glm(x ~ y * type, data = df_fit, family = binomial())
ci2 <- confint(fit2)

df_fit$y <- factor(df_fit$y, levels = 0:1)
df_fit$x <- factor(df_fit$x, levels = 0:1)
fit <- glm(x ~ y * type, data = df_fit, family = binomial())
ci <- confint(fit)

df_res2 <- tibble(
  Classifier = c(gsub(pattern = "type", replacement = "", x = row.names(ci)[3:26]),
                 gsub(pattern = "y1:type", replacement = "", x = row.names(ci)[27:50])),
  Metric = c(rep("Negative predictive value", 24),
             rep("Positive predictive value", 24)),
  Lower = c(-ci[3:26,2],
            -ci2[3:26,2]),
  Upper = c(-ci[3:26,1],
            -ci2[3:26,1])
)
df_res2$FSM <- lapply(strsplit(x = df_res2$Classifier, split = " "), function(x) x[2]) %>%
  unlist
df_res2$Classifier <- lapply(strsplit(x = df_res2$Classifier, split = " "), function(x) x[1]) %>%
  unlist

df_res <- rbind(df_res, df_res2)

p <- ggplot(df_res, aes(x = Classifier, ymin = Lower, ymax = Upper, colour = FSM))
p + geom_hline(yintercept = 0, linetype = 2, colour = grey(.5)) +
  geom_linerange(position = position_dodge(width = .5)) +
  facet_wrap(~ Metric) +
  labs(title = "Radiologe 2")
```



Hier ergibt sich das gleiche Bild wie beim ersten Radiologen.

6 Bagging

Die Balancierung der Trainingsdaten in obiger Analyse lässt die Ergebnisse ein Stück weit vom Zufall abhängen. Auch bei anderen Balancierungstechnicken (oversampling, SMOTE, etc.) ist dies der Fall. Um diesen Zufallsfehler herauszurechnen bietet es sich an, die Balancierung mehrere Male zu wiederholen, die Algorithmen an den so entstandenen Datensätzen zu trainieren und die so trainierten Klassifikationsmethoden auf den Testdatensatz anzuwenden. Die Prognosen können dann aggregiert und mit den tatsächlichen Werten verglichen werden. Es werden also die Ergebnisse eines Bootstrap-Algorithmus aggregiert (Bootstrap + Aggregation = Bagging).

Dieses Verfahren sorgt also für eine unverfälschte Berechnung der Performance-Maße. Was hingegen *nicht* möglich ist, ist die Darstellung von aggregierten Klassifikationsmethoden, beispielsweise die aggregeation der Modellgleichung des logistischen Regressionsmodells oder die Formulierung eines aggregierten Klassifikationsbaumes. Solche Formeln oder Darstellungen müssen immer aus einer Beispieliteration des Bootstrap-Algorithmus herausgenommen werden.

Der folgende Code implementiert dieses Vorgehen. Er enthält alles notwendige und kann ohne obigen Code ausgeführt werden. Da in jeder Bootstrap-Iteration die Hyperparameter der Klassifikationsalgorithmen erneut optimiert werden müssen ist der Aufwand beträchtlich.

```
## Data for training
df <- read_xlsx("~/201801_Makowski/02_Dokumente/20190408/20190407_luca_trainvalid_test_all_aid.xlsx", sheet = 1)

## Data for testing
df_test <- read_xlsx("~/201801_Makowski/02_Dokumente/20190408/20190505_luca_trainvalid_test_all_wpred_aid.xlsx", sheet = 2)

## Correct coding of the response variable
df$response <- factor(df$pos_neg,
                      levels = c("n", "p"),
                      labels = 0:1)

df_test$response <- factor(df_test$pos_neg,
                           levels = c("n", "p"),
                           labels = 0:1)

## Function for feature selection
ps_feature_selection <- function(df, features){

 ## Correct order of data frame
 df <- df %>%
  dplyr::select(response, features)

 ## Function for computing AUC
 ps_auc <- function(reference, predictor){
  ps_sensitivity <- function(tab) {
   tab[2,2] / sum(tab[2,])
  }
  ## Spezifitaet
  ps_specifity <- function(tab) {
   tab[1,1] / sum(tab[1,])
  }
  f <- function(thr, reference, predictor){
   tab <- table(reference, factor(predictor > thr, levels = c(FALSE, TRUE)))
   sens <- ps_sensitivity(tab)
   spec <- ps_specifity(tab)
   return(c(sens, spec))
  }
  f <- Vectorize(f, vectorize.args = "thr")
  s <- seq(min(predictor), max(predictor), length.out = 256)
  out <- cbind(s, t(f(thr = s, reference = reference, predictor = predictor)))
  #print(plot(1 - out[,3], out[,2], type = "l"))

  roc <- approxfun(1 - out[,3], out[,2])
  auc <- try(integrate(f = roc, lower = min(1 - out[,3]), upper = max(1 - out[,3]))$value,
             silent = TRUE)
  if(class(auc) == "try-error"){
   auc <- sum(diff(out[,3]) * (out[2:nrow(out),2] + out[1:(nrow(out) - 1),2]) / 2)
  }
  if(auc < .5){
   f <- function(thr, reference, predictor){
    tab <- table(reference, factor(predictor < thr, levels = c(FALSE, TRUE)))
    sens <- ps_sensitivity(tab)
    spec <- ps_specifity(tab)
    return(c(sens, spec))
   }
   f <- Vectorize(f, vectorize.args = "thr")
   out <- cbind(s, t(f(thr = s, reference = reference, predictor = predictor)))
   roc <- approxfun(1 - out[,3], out[,2])
   auc <- try(integrate(f = roc, lower = min(1 - out[,3]), upper = max(1 - out[,3]))$value,
              silent = TRUE)
   if(class(auc) == "try-error"){
    auc <- sum(diff(out[,3]) * (out[2:nrow(out),2] + out[1:(nrow(out) - 1),2]) / 2)
   }
  }
  tibble(
   FPR = seq(0, 1, length.out = 256),
   Sensitivity = roc(FPR),
   AUC = auc
  )
 }

 ## ======================================
```

```
## Feature selection by Wilcoxon and AUC
## ====================================

pval_wlcx <- numeric(length(features))
diff_wlcx <- 0 * pval_wlcx
auc <- 0 * pval_wlcx
for(j in 1:length(features)){
 res <- wilcox.test(df[[features[j]]] ~ df$response, conf.int = TRUE)
 pval_wlcx[j] <- res$p.value
 diff_wlcx[j] <- res$estimate
 auc[j] <- unique(ps_auc(reference = df$response, predictor = df[[features[j]]])$AUC)
}

## Feature index wilcoxon
fi_wlcx <- sort.list(abs(diff_wlcx), decreasing = TRUE)

## Feature index auc
fi_auc <- sort.list(auc, decreasing = TRUE)

## ======================================
## Feature selection by mutual information
## ======================================

df$response <- ordered(df$response)
dd <- mRMR.data(data = df %>% as.data.frame)
filter <- mRMR.classic(data = dd,
                       target_indices = c(1),
                       feature_count = length(features))

## Maximize mutual information
fi_mi <- sort.list(mim(filter)[-1,1], decreasing = TRUE)

## maximizing the MI with y (maximum relevance) and minimizing the
## average MI with all the previously selected variables
fi_mrmi <- solutions(filter)[[1]][,1] - 1

## ===============
## Collect results
## ===============

tibble(
 features = features,
 wlcx = fi_wlcx,
 auc = fi_auc,
 mi = fi_mi,
 mrmi = fi_mrmi
)

}

## Number of Bootstrap iterations
n_sim <- 30

## Lists for saving predictions
pr_lda <- lapply(1:4, function(x) matrix(ncol = n_sim, nrow = nrow(df_test)))
pr_logistic <- pr_lda
pr_pls <- pr_lda
pr_svm <- pr_lda
pr_mlp <- pr_lda
pr_rpart <- pr_lda

## Generate random seeds for reproducibility
seeds <- sample(x = 1:1e6, size = n_sim)

## Save original data set
df_or <- df
df_test_or <- df_test

## Number of features
n_features <- 30

for(i in 1:n_sim){
```

```
message(i)
df <- df_or
df_test <- df_test_or

## Indicator variable for keeping
df$Keep <- 0

## Keep all positive lymphs
df$Keep[df$response == 1] <- 1

## Select as many negative lymphs as there are positives
set.seed(seeds[i])
s <- sample(x = which(df$response == 0),
        size = sum(df$response == 1),
        replace = TRUE)
df$Keep[s] <- 1

## Filter and delete some unnecessary variables from the data set
df <- df %>%
 filter(Keep > 0) %>%
 dplyr::select(-AID, -Keep, -pos_neg)

## Features
features <- names(df)[1:(ncol(df) - 1)]

## Scaling of features
## We must keep the means and standard deviations in order to correctly scale
## the test data
m <- numeric(length(features))
s <- m
for(j in 1:length(features)){
 m[j] <- mean(df[[features[j]]])
 s[j] <- sd(df[[features[j]]])
 df[[features[j]]] <- (df[[features[j]]] - m[j]) / s[j]
}

## Perform feature selection for training data
df_fs <- ps_feature_selection(df = df, features = features)

## Model formulas
formulas <- list(
 ## Formula for Wilcoxon
 wlcx = as.formula(paste("response ~",
                 paste(df_fs$features[df_fs$wlcx[1:n_features]], collapse = "+"))),
 ## Formula for AUC
 auc = as.formula(paste("response ~",
                  paste(df_fs$features[df_fs$auc[1:n_features]], collapse = "+"))),
 ## Formula for MI
 mi = as.formula(paste("response ~",
                paste(df_fs$features[df_fs$mi[1:n_features]], collapse = "+"))),
 ## Formula for MRMI
 mrmi = as.formula(paste("response ~",
                  paste(df_fs$features[df_fs$mrmi[1:n_features]], collapse = "+")))
)

## ========
## Training
## ========

## Linear Discriminant Analysis
fit_lda <- lapply(formulas, function(f) {
 lda(formula = f, data = df)
})

## Logistic regression
fit_logistic <- lapply(formulas, function(f) {
 glm(formula = f, data = df, family = binomial())
})

## Partial least squares
fit_pls <- lapply(formulas, function(f) {
```

```
 caret::train(form = f,
              data = df,
              method = "pls",
              tuneGrid = data.frame(ncomp = 1:15),
              trControl = trainControl(number = 10))
})

## Support vector machines
fit_svm <- lapply(formulas, function(f) {
 caret::train(form = f,
              data = df,
              method = "svmLinear2",
              tuneGrid = data.frame(cost = seq(0.1, 1, by = .1)),
              trControl = trainControl(number = 10))
})

## Neuronal network (multilayer perceptron)
fit_mlp <- lapply(formulas, function(f) {
 caret::train(form = f,
              data = df,
              method = "mlp",
              tuneGrid = data.frame(size = 1:5),
              trControl = trainControl(number = 10))
})

## Recursive partition
fit_rpart <- lapply(formulas, function(f) {
 caret::train(form = f,
              data = df,
              method = "rpart",
              tuneGrid = data.frame(cp = seq(0.01, 0.15, by = .01)),
              trControl = trainControl(number = 10))
})

## =======
## Testing
## =======

for(j in 1:length(features)){
 df_test[[features[j]]] <- (df_test[[features[j]]] - m[j]) / s[j]
}

## Prediction for LDA
pr <- lapply(fit_lda, function(fit) predict(fit, newdata = df_test)$class)
pr_lda[[1]][,i] <- pr$wlcx
pr_lda[[2]][,i] <- pr$auc
pr_lda[[3]][,i] <- pr$mi
pr_lda[[4]][,i] <- pr$mrmi

## Prediction for logistic regression
pr <- lapply(fit_logistic,
             function(fit) factor((predict(fit, newdata = df_test, type = "response") > .5) * 1))
pr_logistic[[1]][,i] <- pr$wlcx
pr_logistic[[2]][,i] <- pr$auc
pr_logistic[[3]][,i] <- pr$mi
pr_logistic[[4]][,i] <- pr$mrmi

## Prediction for Partial least squares
pr <- lapply(fit_pls, function(fit) predict(fit, newdata = df_test))
pr_pls[[1]][,i] <- pr$wlcx
pr_pls[[2]][,i] <- pr$auc
pr_pls[[3]][,i] <- pr$mi
pr_pls[[4]][,i] <- pr$mrmi

## Prediction for Support vector machines
pr <- lapply(fit_svm, function(fit) predict(fit, newdata = df_test))
pr_svm[[1]][,i] <- pr$wlcx
pr_svm[[2]][,i] <- pr$auc
pr_svm[[3]][,i] <- pr$mi
pr_svm[[4]][,i] <- pr$mrmi

## Prediction for neural network
```

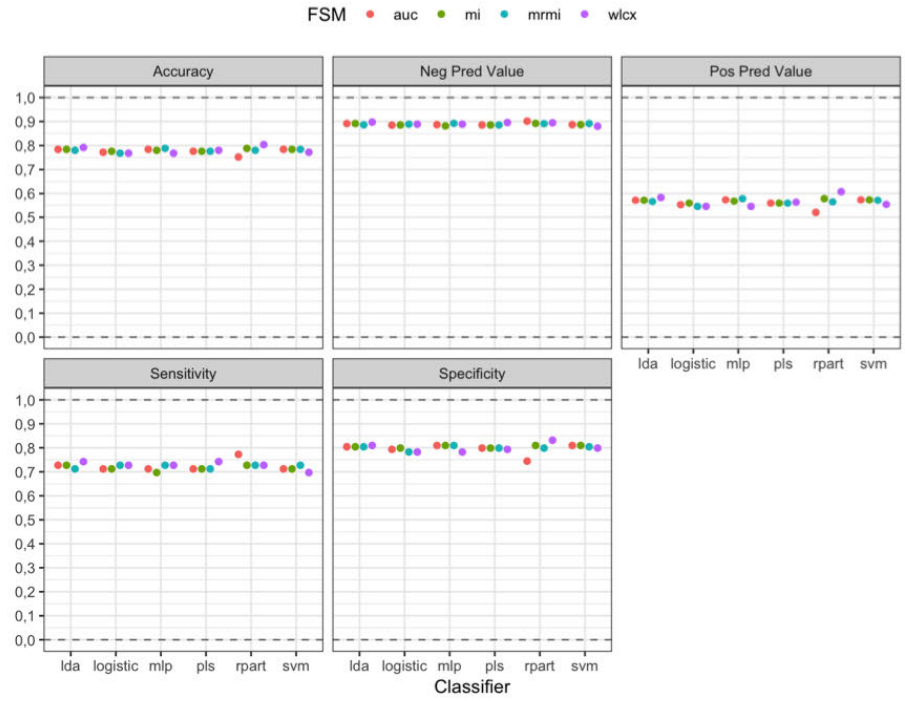
```
  pr <- lapply(fit_mlp, function(fit) predict(fit, newdata = df_test))
  pr_mlp[[1]][,i] <- pr$wlcx
  pr_mlp[[2]][,i] <- pr$auc
  pr_mlp[[3]][,i] <- pr$mi
  pr_mlp[[4]][,i] <- pr$mrmi

  ## Prediction for classification tree
  pr <- lapply(fit_rpart, function(fit) predict(fit, newdata = df_test))
  pr_rpart[[1]][,i] <- pr$wlcx
  pr_rpart[[2]][,i] <- pr$auc
  pr_rpart[[3]][,i] <- pr$mi
  pr_rpart[[4]][,i] <- pr$mrmi

}

## ===========
## Aggregation
## ===========

pr_lda <- lapply(pr_lda, function(x) factor(1 * (rowMeans(x - 1) > .5), levels = 0:1))
pr_logistic <- lapply(pr_logistic, function(x) factor(1 * (rowMeans(x - 1) > .5), levels = 0:1))
pr_pls <- lapply(pr_pls, function(x) factor(1 * (rowMeans(x - 1) > .5), levels = 0:1))
pr_svm <- lapply(pr_svm, function(x) factor(1 * (rowMeans(x - 1) > .5), levels = 0:1))
pr_mlp <- lapply(pr_mlp, function(x) factor(1 * (rowMeans(x - 1) > .5), levels = 0:1))
pr_rpart <- lapply(pr_rpart, function(x) factor(1 * (rowMeans(x - 1) > .5), levels = 0:1))

## ==========
## Evaluation
## ==========

## Evaluation for lda
eval_lda <- lapply(pr_lda,
            function(pr) caret::confusionMatrix(data = pr,
                                                reference = df_test$response,
                                                positive = "1"))

## Evaluation for logistic
eval_logistic <- lapply(pr_logistic,
                 function(pr) caret::confusionMatrix(data = pr,
                                                     reference = df_test$response,
                                                     positive = "1"))

## Evaluation for pls
eval_pls <- lapply(pr_pls,
            function(pr) caret::confusionMatrix(data = pr,
                                                reference = df_test$response,
                                                positive = "1"))

## Evaluation for svm
eval_svm <- lapply(pr_svm,
            function(pr) caret::confusionMatrix(data = pr,
                                                reference = df_test$response,
                                                positive = "1"))

## Evaluation for mlp
eval_mlp <- lapply(pr_mlp,
            function(pr) caret::confusionMatrix(data = pr,
                                                reference = df_test$response,
                                                positive = "1"))

## Evaluation for rpart
eval_rpart <- lapply(pr_rpart,
              function(pr) caret::confusionMatrix(data = pr,
                                                  reference = df_test$response,
                                                  positive = "1"))
## Collect results
## lda
eval_lda <- Reduce("rbind", lapply(eval_lda, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(pr),
         Classifier = "lda")
```

```
## logistic
eval_logistic <- Reduce("rbind", lapply(eval_logistic, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(pr),
         Classifier = "logistic")

## pls
eval_pls <- Reduce("rbind", lapply(eval_pls, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(pr),
         Classifier = "pls")

## svm
eval_svm <- Reduce("rbind", lapply(eval_svm, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(pr),
         Classifier = "svm")

## mlp
eval_mlp <- Reduce("rbind", lapply(eval_mlp, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(pr),
         Classifier = "mlp")

## rpart
eval_rpart <- Reduce("rbind", lapply(eval_rpart, function(ev) c(ev$overall[1], ev$byClass)[1:5])) %>%
  as.tibble %>%
  mutate(FSM = names(pr),
         Classifier = "rpart")

## One complete data set
df_eval <- rbind(eval_lda,
                 eval_logistic,
                 eval_pls,
                 eval_svm,
                 eval_mlp,
                 eval_rpart)

## Output
kable(df_eval[,c(7,6,1:5)], digits = 2)
```

Classifier	FSM	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
lda	wlcx	0,79	0,74	0,81	0,58	0,90
lda	auc	0,78	0,73	0,80	0,57	0,89
lda	mi	0,78	0,73	0,80	0,57	0,89
lda	mrmi	0,78	0,71	0,80	0,57	0,89
logistic	wlcx	0,77	0,73	0,78	0,55	0,89
logistic	auc	0,77	0,71	0,79	0,55	0,88
logistic	mi	0,78	0,71	0,80	0,56	0,89
logistic	mrmi	0,77	0,73	0,78	0,55	0,89
pls	wlcx	0,78	0,74	0,79	0,56	0,90
pls	auc	0,78	0,71	0,80	0,56	0,89
pls	mi	0,78	0,71	0,80	0,56	0,89
pls	mrmi	0,78	0,71	0,80	0,56	0,89
svm	wlcx	0,77	0,70	0,80	0,55	0,88
svm	auc	0,78	0,71	0,81	0,57	0,89
svm	mi	0,78	0,71	0,81	0,57	0,89
svm	mrmi	0,78	0,73	0,80	0,57	0,89
mlp	wlcx	0,77	0,73	0,78	0,55	0,89

Classifier	FSM	Accuracy	Sensitivity	Specificity	Pos Pred Value	Neg Pred Value
mlp	auc	0,78	0,71	0,81	0,57	0,89
mlp	mi	0,78	0,70	0,81	0,57	0,88
mlp	mrmi	0,79	0,73	0,81	0,58	0,89
rpart	wlcx	0,80	0,73	0,83	0,61	0,89
rpart	auc	0,75	0,77	0,74	0,52	0,90
rpart	mi	0,79	0,73	0,81	0,58	0,89
rpart	mrmi	0,78	0,73	0,80	0,56	0,89

```
## Graph
df_eval <- df_eval %>%
  gather(-FSM, -Classifier, key = "Metrics", value = "value")

p <- ggplot(df_eval, aes(x = Classifier, y = value, colour = FSM))
p + geom_hline(yintercept = 0:1, linetype = 2, colour = grey(.5)) +
  geom_point(position = position_dodge(width = .75)) +
  facet_wrap(~ Metrics) +
  scale_y_continuous(breaks = seq(0, 1, by = .1)) +
  theme(legend.position = "top") +
  labs(y = NULL)
```



Appendix

Software

Sämtliche Analysen wurden mit der statistischen Programmiersprache R (R Core Team 2017) durchgeführt (R version 3.4.4 (2018-03-15) auf einem x86_64-apple-darwin15.6.0 System unter macOS 10.14.4). Für die Datenverarbeitung und Darstellung wurde die Paket-Sammlung **tidyverse** (Wickham 2017) sowie das Paket **knitr** (Xie 2015) verwendet.

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS  10.14.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2.2   rpart.plot_2.2.0 rpart_4.1-13     MASS_7.3-49      caret_6.0-80
##  [6] lattice_0.20-35  mRMRe_2.0.7      igraph_1.2.1     survival_2.42-3  readxl_1.2.0
## [11] forcats_0.3.0    stringr_1.3.0    dplyr_0.7.6      purrr_0.2.4      readr_1.1.1
## [16] tidyr_0.8.0      tibble_2.0.1     ggplot2_3.1.0    tidyverse_1.2.1  knitr_1.20
##
## loaded via a namespace (and not attached):
##  [1] httr_1.3.1        jsonlite_1.6      splines_3.4.4     foreach_1.4.4
##  [5] prodlim_2018.04.18 modelr_0.1.1      assertthat_0.2.0  highr_0.6
##  [9] stats4_3.4.4      cellranger_1.1.0  yaml_2.1.18       ipred_0.9-6
## [13] RSNNS_0.4-10      pillar_1.3.1      backports_1.1.2   glue_1.3.0
## [17] digest_0.6.18     rvest_0.3.2       colorspace_1.3-2  recipes_0.1.4
## [21] htmltools_0.3.6   Matrix_1.2-14     plyr_1.8.4        timeDate_3043.102
## [25] pkgconfig_2.0.2   broom_0.5.1       haven_2.0.0       scales_1.0.0
## [29] gower_0.1.2       lava_1.6.1        generics_0.0.2    withr_2.1.2
## [33] nnet_7.3-12       lazyeval_0.2.1    cli_1.0.1         magrittr_1.5
## [37] crayon_1.3.4      evaluate_0.10.1   fansi_0.4.0       nlme_3.1-137
## [41] xml2_1.2.0        class_7.3-14      tools_3.4.4       hms_0.4.2
## [45] munsell_0.5.0     pls_2.6-0         compiler_3.4.4    e1071_1.6-8
## [49] rlang_0.3.1.9000  grid_3.4.4        iterators_1.0.9   rstudioapi_0.7
## [53] labeling_0.3      rmarkdown_1.9     gtable_0.2.0      ModelMetrics_1.1.0
## [57] codetools_0.2-15  reshape2_1.4.3    R6_2.3.0          lubridate_1.7.4
## [61] utf8_1.1.4        bindr_0.1.1       rprojroot_1.3-2   stringi_1.1.7
## [65] Rcpp_0.12.18      tidyselect_0.2.4
```

Literatur

R Core Team. 2017. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/ (https://www.R-project.org/).

Wickham, Hadley. 2017. *Tidyverse: Easily Install and Load 'Tidyverse' Packages*. https://CRAN.R-project.org/package=tidyverse (https://CRAN.R-project.org/package=tidyverse).

Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. http://yihui.name/knitr/ (http://yihui.name/knitr/).

9 Eidesstattliche Versicherung

„Ich, Hendrik Philipp Becker, versichere an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorgelegte Dissertation mit dem Thema: *Prädiktive Radiomics-Modelle zur Dignitätsklassifizierung mediastinaler Lymphknoten im CT bei Adeno- und Plattenepithelkarzinomen der Lunge (Englisch: Predictive radiomics models for dignity classification of mediastinal lymph nodes in CT in adeno and squamous cell carcinomas of the lungs.),* selbstständig und ohne nicht offengelegte Hilfe Dritter verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel genutzt habe.

Alle Stellen, die wörtlich oder dem Sinne nach auf Publikationen oder Vorträgen anderer Autoren/ innen beruhen, sind als solche in korrekter Zitierung kenntlich gemacht. Die Abschnitte zu Methodik (insbesondere praktische Arbeiten, Laborbestimmungen, statistische Aufarbeitung) und Resultaten (insbesondere Abbildungen, Graphiken und Tabellen) werden von mir verantwortet.

Ich versichere ferner, dass ich die in Zusammenarbeit mit anderen Personen generierten Daten, Datenauswertungen und Schlussfolgerungen korrekt gekennzeichnet und meinen eigenen Beitrag sowie die Beiträge anderer Personen korrekt kenntlich gemacht habe (siehe Anteilserklärung). Texte oder Textteile, die gemeinsam mit anderen erstellt oder verwendet wurden, habe ich korrekt kenntlich gemacht.

Meine Anteile an etwaigen Publikationen zu dieser Dissertation entsprechen denen, die in der untenstehenden gemeinsamen Erklärung mit dem Erstbetreuer, angegeben sind. Für sämtliche im Rahmen der Dissertation entstandenen Publikationen wurden die Richtlinien des ICMJE (International Committee of Medical Journal Editors; www.icmje.og) zur Autorenschaft eingehalten. Ich erkläre ferner, dass ich mich zur Einhaltung der Satzung der Charité – Universitätsmedizin Berlin zur Sicherung Guter Wissenschaftlicher Praxis verpflichte.

Weiterhin versichere ich, dass ich diese Dissertation weder in gleicher noch in ähnlicher Form bereits an einer anderen Fakultät eingereicht habe.

Die Bedeutung dieser eidesstattlichen Versicherung und die strafrechtlichen Folgen einer unwahren eidesstattlichen Versicherung (§§156, 161 des Strafgesetzbuches) sind mir bekannt und bewusst.“

21.01.2020 Hendrik Philipp Becker

10 Lebenslauf

Mein Lebenslauf wird aus datenschutzrechtlichen Gründen in der elektronischen Version meiner Arbeit nicht veröffentlicht.

11 Danksagung

Ich danke meinen Betreuern Herrn Prof. Dr. Marcus Makowski und Herrn PD Dr. Tobias Penzkofer aus dem radiologischen Institut der Charité Berlin für die konsequente Unterstützung meines Promotionsvorhabens und die motivierende Zusammenarbeit.

Ein weiterer Dank gilt Herrn Dr. med. univ. Falk Lübbe für die schnelle Einführung in die CT-Segmentierung mit MITK, Frau Dr. med Jing Zhao für die Mithilfe der Qualitätsüberprüfung der Nodalstatus sowie Frau Dr. Nina Becker für die umfangreichen Erläuterungen im Bereich des machine learnings.

Abschließend danke ich meiner Mutter Frau Dr. Heike Makoschey-Weiß, die mir nicht nur mein Studium, sondern auch diese Promotionsarbeit durch ihre Unterstützung ermöglicht hat.

12 Bescheinigung Statistik

CHARITÉ

CharitéCentrum für Human- und Gesundheitswissenschaften

Charité | Campus Charité Mitte | 10117 Berlin

Institut für Biometrie und klinische Epidemiologie (iBikE)

Direktor: Prof. Dr. Geraldine Rauch

Postantschrift:
Charitéplatz 1 | 10117 Berlin
Besucheranschrift:
Reinhardtstr. 58 | 10117 Berlin

Tel. +49 (0)30 450 562171
geraldine.rauch@charite.de
https://biometrie.charite.de/

Name, Vorname: Makoschey, Hendrik Philipp

Emailadresse: hendrik-philipp.makoschey@charite.de

Matrikelnummer:

PromotionsbetreuerIn: PD Dr. Tobias Penzkofer, Prof. Dr. Markus Makowski

Promotionsinstitution / Klinik: Institut für Radiologie und Kinderradiologie

Bescheinigung

Hiermit bescheinige ich, dass Herr Makoschey, Hendrik Philipp innerhalb der Service Unit Biometrie des Instituts für Biometrie und klinische Epidemiologie (iBikE) bei mir eine statistische Beratung zu einem Promotionsvorhaben wahrgenommen hat. Folgende Beratungstermine wurden wahrgenommen:

- Termin 1: 05. Juli 2019

Folgende wesentliche Ratschläge hinsichtlich einer sinnvollen Auswertung und Interpretation der Daten wurden während der Beratung erteilt:

- Die angewendeten Methoden wurden nachvollzogen. Die Anwendung der maschinellen Lernverfahren machte in dem Kontext der Fragestellung Sinn.
- Binäre Classifiers wurden angemessen evaluiert
- Die Abbildungen und Auswertungen entsprechen dem Standard einer statistischen Auswertung mit maschinellen Lernverfahren.

Diese Bescheinigung garantiert nicht die richtige Umsetzung der in der Beratung gemachten Vorschläge, die korrekte Durchführung der empfohlenen statistischen Verfahren und die richtige Darstellung und Interpretation der Ergebnisse. Die Verantwortung hierfür obliegt allein dem Promovierenden. Das Institut für Biometrie und klinische Epidemiologie übernimmt hierfür keine Haftung.

Datum: 14.10.2019 [redacted]

Name des Beraters/ der Beraterin:

CHARITÉ
UNIVERSITÄTSMEDIZIN BERLIN
Institut für Biometrie und Klinische Epidemiologie
Dr. Jochen Kruppa
Campus Charité Mitte
Charitéplatz 1 | D-10117 Berlin
Besucheranschrift: Rahel-Hirsch-Weg 5

Unterschrif[redacted]el