Projekt:

„Identifizierung von Risikofaktoren in der Entstehung von Lungenkrebs mithilfe von maschinellem Lernen in R.“

Sabrina Frank

19.01.2024

Leitfaden für nachvollziehbare Schritte

„Identifizierung von Risikofaktoren in der Entstehung von Lungenkrebs mithilfe von maschinellem Lernen in R.“

1. Kurze Darstellung des Problembereichs / Aufriss des Themas

1.1 Inhaltlich

Lungenkrebs ist eine der tödlichsten und die prävalenteste Tumorerkrankungen weltweit[1] und zeichnet sich durch unkontrolliertes Zellwachstum und Tumorentstehung in dem Organsystem Lunge aus. Verschiedene Risikofaktoren können die Wahrscheinlichkeit, an Lungenkrebs zu erkranken, erhöhen, darunter unter anderem das Rauchen und, vor allem in Schwellenländern ein großes Problem, Luftverschmutzung durch Feinstaub[2]. Je länger der Tumor unentdeckt bleibt, desto maligner kann dieser werden und eventuell metastasieren und andere Organsysteme befallen. Weil die Tumore in frühen Stadien oftmals noch relativ problemlos und nachhaltig operativ entfernt werden können, profitieren Betroffene von einer frühen Erkennung und Behandlung massiv[3]. Bisher gibt es noch keine standardisierte Früherkennungsdiagnostik, meist entstehen Lungenkrebsbefunde erst in späten Stadien aufgrund der schwerwiegenden Symptome oder als Zufallsbefunde[2]. Um Personen mit erhöhtem Risiko besser identifizieren und mögliche Früherkennungsmaßnahmen engmaschiger gestalten und gezielter anwenden zu können, da eine generelle Früherkennungsdiagnostik für die Gesamtpopulation viel zu Ressourcenaufwändig wäre, soll ein Algorithmus implementiert werden, der auf der Basis normierter Risikofaktoren eine Risikoprädiktion erstellen kann.

1.2 Begründung desThemas

Darstellung der Relevanz des Themas?

Die Anzahl der neuen Fälle von Lungenkrebs nimmt global zu und wir diesen Trend beibehalten1. Weltweit ist es mit insgesamt 19.292.789 Fällen und 1,76 Millionen Toten in 2020 die häufigste und (für Männer) tödlichste maligne Krebserkrankung. Da es für Lungenkrebs in späten Stadien bis dato noch keine guten Heilungsmöglichkeiten gibt ist eine Früherkennung hier besonders wichtig.

---

[1] https://de.statista.com/statistik/daten/studie/1201305/umfrage/prognostizierte-anzahl-von-krebstodesfaellen-weltweit/ (Stand 18.01.2024, 13:31)
[2] https://aqli.epic.uchicago.edu/the-index/ (Stand 18.01.2024, 14:31), Patel, Jay, and William Song. "A Review of the Health Impacts of Air Pollutants." Authorea Preprints (2023).
[3] https://www.krebsdaten.de/Krebs/DE/Content/Krebsarten/Lungenkrebs/lungenkrebs_node.html (Stand 18.01.2024, 13:31)

Darstellung eines persönlichen Erkenntnisinteresses.

Durch mein Studium der Biomedizin und meine zeitweise Arbeit im Studienzentrum eines Krankenhauses, bei der ich Studienpatient*innen mit diversen Tumorerkrankungen und im besonderen auch Lungenkrebs betreut habe, ist es mir ein Anliegen, alle Möglichkeiten, so auch die Nutzung von Maschinellem Lernen, auszuschöpfen, um die Überlebenschancen und Lebensqualität von Patient*innen mit dieser belastenden Krankheit direkt oder indirekt verbessern zu können.

2. Nachvollziehbare Schritte

2.1 Der Stand der Forschung / Auswertung der vorhandenen Literatur / Tutorials ...

Es gab bereits einige Bemühungen in der wissenschaftlichen Community mit maschinellem Lernen und einer Vielzahl verschiedener Ansätze Prädiktions- und Risikostratifizierungsalgorithmen zu erstellen [4], unter anderem auch speziell mit Random Forest Algorithmen[5]. Die Wissen-schaftler hinter dem Paper von Ahmad et al. aus 2020 haben beispielsweise ein Random Forest Model mit dem Namen Lung Cancer Prediction Tool (LCPT) entwickelt, der eine Risikoklassifizierung von Patient*innen in die Level „Gering“, „Medium“ und „Hoch“ mit einer Sensitivität und Spezifität von 100% hervorsagen soll. Das Model wurde auf Grundlage der Lung cancer database[6] mit Daten zu den Features Alter, Geschlecht, Luftverschmutzung, Alkoholkonsum, Stauballergie, berufliche Risiken, genetisches Risiko, chronische Lungenerkrankungen, ausgewogene Ernährung, Fettleibigkeit, Raucherstatus, Passivraucherstatus, Brustschmerzen, Bluthusten, Müdigkeit, Gewichtsverlust, Kurzatmigkeit, Keuchen, Schluckbeschwerden, Klumpen der Fingernägel, häufige Erkältungen, trockener Husten und Schnarchen erstellt. Diese wurden sorgfältig recherchiert und ausgewählt. In der Medizin werden derzeit in allen Gebieten die Applikation von Machine Learning getestet, um Ärzte und Pathologen bei der Diagnostik zu entlasten. Mit der wachsenden Anzahl an Machine Learning Algorithmen und immer leistungsstärkeren Rechnern gibt es viele noch unausgeschöpfte Möglichkeiten, neue diagnostische Leitungsspitzen zu erreichen.

---

[4] Ahmad AS, Mayya AM. A new tool to predict lung cancer based on risk factors. Heliyon. 2020 Feb 26;6(2):e03402. doi: 10.1016/j.heliyon.2020.e03402. PMID: 32140577; PMCID: PMC7044659.

[5] P. Thangaraju, G. Barkavi, T. Karthikeyan, Mining lung cancer data for smokers and NonSmokers by using data mining techniques, Int. J. Adv. Res. Comput. Commun. Eng. 3 (7) (2014) 7622–7626; T. Christopher, J. Jamera, Study of classification algorithm for lung cancer prediction, Int. J. Innovat. Sci. Eng. Technol. 3 (2) (2016) 42–49; S. Durga, K. Kasturi, Lung disease prediction system using data mining techniques, J. Adv. Res. Dyn. Control Sys. 9 (5) (2017) 62–66; M. Markaki, I. Tsamardinos, A. Langhammer, V. Lagani, K. Hveem, O.D. Røe, A validated clinical risk prediction model for lung cancer in smokers of all ages and exposure types: a hunt study, EBioMedicine 31 (2018) 36–46.

[6] URL: https://data.world/cancerdatahp/lung-cancer-data (Zugriff: January 10, 2018)

2.2 Fragestellung

Kann anhand der ausgewählten Risikofaktoren verlässlich ein Risikolevel für die potentielle Entwicklung eines Lungenkrebses mit Hilfe eines Random Forest Algorithmus in R vorhergesagt werden?

2.3 Wissenslücke

Möglicherweise gibt es bessere Algorithmen und Datenpräprozessierung oder Risikofaktoren, um das Risiko für eine Lungenkrebserkrankung effektiv vorhersagen zu können. Hier müssen noch weitere klinische Studien durchgeführt werden und Modelle entwickelt, um die Diagnose dieser Krankheit zeitiger und effizienter zu gestalten.

2.4 Methode

2.4.1. Software

Git	Version 2.31.1.windows.1	
GitHub	https://github.com/	
R Studio	2023.12.0 Build 369	
R	platform	x86_64-w64-mingw32
	arch	x86_64
	os	mingw32
	crt	ucrt
	system	x86_64, mingw32
	status	
	major	4
	minor	3.2
	year	2023
	month	10
	day	31
	svn rev	85441
	language	R
	version.string	R version 4.3.2 (2023-10-31 ucrt)
	nickname	Eye Holes

2.4.2. Importieren der Bibliotheken und Packages

Zunächst müssen alle benötigten Bibliotheken und Pakete installiert und werden, deren Funktionalität (kommentiert im Code) im Verlauf des Skriptes benötigt werden.
Mit der set.seed-Funktion wird noch versichert, dass der Code reproduzierbare Ergebnisse produziert.

```r
## Pipes
# install.packages("dplyr")
library(dplyr)

## Data frame handling und Datenmanipulierung
# install.packages("data.table")
library(data.table)
# install.packages("tidyr")
library(tidyr)

## Visualisierung
# install.packages("ggplot2")
library(ggplot2)
# install.packages("GGally")
library(GGally)
# install.packages("gridExtra")
library(gridExtra)
# install.packages("lattice")
library(lattice)
# Measure of association (Korrelationsmatrix für kategorische Daten)
# install.packages("ggcorrplot")
library(ggcorrplot)

## Machine Learning
# install.packages("caret")
library(caret)

## Verwendete Packages
# install.packages("NCmisc")
library(NCmisc)
```

```r
## Reproduzierbarkeit des Codes
set.seed(1)
```

2.4.3. Der Datensatz

Nun muss der Datensatz aus der .csv Datei eingelesen und gesichtet werden.
Dies geschieht mit der fread-Funktion, um ein data.table Objekt zu erhalten. Data.table Objekte haben gegenüber data.frames einige Vorteile, wie etwa eine bessere Performance und eine kompaktere Syntax.

```r
## Import mit fread
lungcancer_raw <- fread(file = "./data/lung_cancer_patient_datasets.csv")
class(lungcancer_raw)
# [1] "data.table" "data.frame"

## Sichten des Datensatzes
lungcancer_raw
dim(lungcancer_raw)
# [1] 1000   26
```

Mit lungcancer_raw wir einmal eine abgekürzte Version des gesamten Datensatzes ausgegeben (hier nicht gezeigt, da zu groß; beinhaltet die ersten und letzten paar Zeilen eines Datensatzes).

Mit der str-Funktion erhalten wir Informationen über die Datentypen der Spalten und deren Werte:

```
str(lungcancer_raw)
# Classes 'data.table' and 'data.frame':  1000 obs. of  26 variables:
#   $ index                   : int  0 1 2 3 4 5 6 7 8 9 ...
# $ Patient Id              : chr  "P1" "P10" "P100" "P1000" ...
# $ Age                     : int  33 17 35 37 46 35 52 28 35 46 ...
# $ Gender                  : int  1 1 1 1 1 1 2 2 2 1 ...
# $ Air Pollution           : int  2 3 4 7 6 4 2 3 4 2 ...
# $ Alcohol use             : int  4 1 5 7 8 5 4 1 5 3 ...
# $ Dust Allergy            : int  5 5 6 7 7 6 5 4 6 4 ...
# $ OccuPational Hazards    : int  4 3 5 7 7 5 4 3 5 2 ...
# $ Genetic Risk            : int  3 4 5 6 7 5 3 2 6 4 ...
# $ chronic Lung Disease    : int  2 2 4 7 6 4 2 3 5 3 ...
# $ Balanced Diet           : int  2 2 6 7 7 6 2 4 5 3 ...
# $ Obesity                 : int  4 2 7 7 7 7 4 3 5 3 ...
# $ Smoking                 : int  3 2 2 7 8 2 3 1 6 2 ...
# $ Passive Smoker          : int  2 4 3 7 7 3 2 4 6 3 ...
# $ Chest Pain              : int  2 2 4 7 7 4 2 3 6 4 ...
# $ Coughing of Blood       : int  4 3 8 8 9 8 4 1 5 4 ...
# $ Fatigue                 : int  3 1 8 4 3 8 3 3 1 1 ...
# $ Weight Loss             : int  4 3 7 2 2 7 4 2 4 2 ...
# $ Shortness of Breath     : int  2 7 9 3 4 9 2 2 3 4 ...
# $ Wheezing                : int  2 8 2 1 1 2 2 4 2 6 ...
# $ Swallowing Difficulty   : int  3 6 1 4 4 1 3 2 4 5 ...
# $ Clubbing of Finger Nails: int  1 2 4 5 2 4 1 2 6 4 ...
# $ Frequent Cold           : int  2 1 6 6 4 6 2 3 2 2 ...
# $ Dry Cough               : int  3 7 7 7 2 7 3 4 4 1 ...
# $ Snoring                 : int  4 2 2 5 3 2 4 3 1 5 ...
# $ Level                   : chr  "Low" "Medium" "High" "High" ...
# - attr(*, ".internal.selfref")=<externalptr>
```

Hier ist zu sehen, dass die meisten Features ordinaler Natur sind.

Die summary-Funktion fasst die statistischen Momente der Features der data.table zusammen:

```
summary(lungcancer_raw)
# index        Patient Id             Age            Gender      Air Pollution   Alcohol use
# Min.   :  0.0   Length:1000        Min.   :14.00   Min.   :1.000   Min.   :1.00   Min.   :1.000
# 1st Qu.:249.8   Class :character   1st Qu.:27.75   1st Qu.:1.000   1st Qu.:2.00   1st Qu.:2.000
# Median :499.5   Mode  :character   Median :36.00   Median :1.000   Median :3.00   Median :5.000
# Mean   :499.5                      Mean   :37.17   Mean   :1.402   Mean   :3.84   Mean   :4.563
# 3rd Qu.:749.2                      3rd Qu.:45.00   3rd Qu.:2.000   3rd Qu.:6.00   3rd Qu.:7.000
# Max.   :999.0                      Max.   :73.00   Max.   :2.000   Max.   :8.00   Max.   :8.000
# Dust Allergy   OccuPational Hazards  Genetic Risk  chronic Lung Disease Balanced Diet
# Min.   :1.000   Min.   :1.00         Min.   :1.00   Min.   :1.00         Min.   :1.000
# 1st Qu.:4.000   1st Qu.:3.00         1st Qu.:2.00   1st Qu.:3.00         1st Qu.:2.000
# Median :6.000   Median :5.00         Median :5.00   Median :4.00         Median :4.000
# Mean   :5.165   Mean   :4.84         Mean   :4.58   Mean   :4.38         Mean   :4.491
# 3rd Qu.:7.000   3rd Qu.:7.00         3rd Qu.:7.00   3rd Qu.:6.00         3rd Qu.:7.000
# Max.   :8.000   Max.   :8.00         Max.   :7.00   Max.   :7.00         Max.   :7.000
# Obesity         Smoking      Passive Smoker    Chest Pain    Coughing of Blood    Fatigue
# Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000     Min.   :1.000
# 1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:3.000     1st Qu.:2.000
# Median :4.000   Median :3.000   Median :4.000   Median :4.000   Median :4.000     Median :3.000
# Mean   :4.465   Mean   :3.948   Mean   :4.195   Mean   :4.438   Mean   :4.859     Mean   :3.856
# 3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.000     3rd Qu.:5.000
# Max.   :7.000   Max.   :8.000   Max.   :8.000   Max.   :9.000   Max.   :9.000     Max.   :9.000
# Weight Loss    Shortness of Breath    Wheezing     Swallowing Difficulty
# Min.   :1.000   Min.   :1.00         Min.   :1.000   Min.   :1.000
# 1st Qu.:2.000   1st Qu.:2.00         1st Qu.:2.000   1st Qu.:2.000
# Median :3.000   Median :4.00         Median :4.000   Median :4.000
# Mean   :3.855   Mean   :4.24         Mean   :3.777   Mean   :3.746
# 3rd Qu.:6.000   3rd Qu.:6.00         3rd Qu.:5.000   3rd Qu.:5.000
# Max.   :8.000   Max.   :9.00         Max.   :8.000   Max.   :8.000
```

Die ordinalen Features reichen von 1 bis 7,8 oder 9. Das Level des Lungenkrebsrisikos hat 3 Stufen. Das Alter reicht von 14 bis 73 Jahren und hat einen für Krebserkrankungen recht jungen Median von 36 Jahren.

```
123  # Clubbing of Finger Nails Frequent Cold     Dry Cough        Snoring         Level
124  # Min.   :1.000           Min.   :1.000   Min.   :1.000   Min.   :1.000   Length:1000
125  # 1st Qu.:2.000           1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   Class :character
126  # Median :4.000           Median :3.000   Median :4.000   Median :3.000   Mode  :character
127  # Mean   :3.923           Mean   :3.536   Mean   :3.853   Mean   :2.926
128  # 3rd Qu.:5.000           3rd Qu.:5.000   3rd Qu.:6.000   3rd Qu.:4.000
129  # Max.   :9.000           Max.   :7.000   Max.   :7.000   Max.   :7.000
```

Die Anzahl der Null-Werte des Datensatzes belaufen sich für alle Features auf 0:

```
131  lungcancer_raw %>% sapply(function(x)sum(is.na(x)))
132  # index                  Patient Id                         Age
133  # 0                             0                           0
134  # Gender              Air Pollution                Alcohol use
135  # 0                             0                           0
136  # Dust Allergy     OccuPational Hazards                Genetic Risk
137  # 0                             0                           0
138  # chronic Lung Disease            Balanced Diet                     Obesity
139  # 0                             0                           0
140  # Smoking            Passive Smoker                 Chest Pain
141  # 0                             0                           0
142  # Coughing of Blood                   Fatigue                Weight Loss
143  # 0                             0                           0
144  # Shortness of Breath                  Wheezing    Swallowing Difficulty
145  # 0                             0                           0
146  # Clubbing of Finger Nails             Frequent Cold                  Dry Cough
147  # 0                             0                           0
148  # Snoring                       Level
149  # 0                             0
150
151  ## Kopie erstellen
152  lungcancer = copy(lungcancer_raw)
```

Es wird eine Kopie der data.table erstellt, um bei Fehlern nicht noch einmal die .csv Datei laden zu müssen.

2.4.4. Säubern des Datensatzes

```
161  ## Patient ID und Index entfernen
162  lungcancer[,(names(lungcancer)[0:2]):=NULL]
163
164  ## Spaltennamen korrigieren
165  colnames(lungcancer)[colnames(lungcancer) == "Clubbing of Finger Nails"] = "Finger Nails Clubbing"
166  colnames(lungcancer)[colnames(lungcancer) == "chronic Lung Disease"] = "Chronic Lung Disease"
167  colnames(lungcancer)[colnames(lungcancer) == "OccuPational Hazards"] = "Occupational Hazards"
168
169  ## Age in Integer umwandeln
170  lungcancer[, Age := as.integer(Age)]
171
172  ## Gender umcodieren: 1=Male, 2=Female
173  lungcancer[, Gender := as.character(Gender)][Gender == "1", Gender := "M"]
174  lungcancer[, Gender := as.character(Gender)][Gender == "2", Gender := "F"]
175
176  ## Für Visualisierung abspeichern
177  lungcancer_bar = copy(lungcancer[, !c("Age", "Gender")])
178
179  # Schauen, welche ordinalen Werte es gibt
180  unique_data <- sort(unique(as.vector(as.matrix(as.data.frame(lungcancer[,2:24])))))
181  unique_data
182
183  ## Faktoren einführen
184  lungcancer[,2:24] <- lapply(lungcancer[,2:24],as.factor)
185  lungcancer[,3:23] <- lapply(lungcancer[,3:23],ordered)
186  lungcancer[, ("Level") := ordered(get("Level"), levels = c("Low", "Medium", "High"))]
```

Zuerst werden unnötige Spalten entfernt und Rechtschreibfehler in den Spaltennamen korrigiert. Danach werden die Variablen in der Spalte Gender und Age ind respective chr und int umgewandelt. Es wird eine Zwischenkopie für spätere Visualisierungszwecke angelegt. Anschließend werden zu Übersichtszwecken noch einmal die einzelnen möglichen Werte der ordinalen Spalten ausgegeben, um sicherzeustellen, dass es keine unzulässigen Werte gibt. Die ordinalen Daten werden schließlich noch zu R-spezifischen geordneten Faktoren umgewandelt.

Nun wird die bereinigte data.table erneut überprüft. Alle Datentypen sind nun korrekt:

```
## Bereinigung checken
levels(lungcancer$Level)
# [1] "Low"    "Medium" "High"
str(lungcancer)
# Classes ‘data.table’ and 'data.frame':  1000 obs. of  24 variables:
#   $ Age                  : int  33 17 35 37 46 35 52 28 35 46 ...
# $ Gender               : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 1 1 1 2 ...
# $ Air Pollution        : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 2 3 4 7 6 4 2 3 4 2 ...
# $ Alcohol use          : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 4 1 5 7 8 5 4 1 5 3 ...
# $ Dust Allergy         : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 5 5 6 7 7 6 5 4 6 4 ...
# $ Occupational Hazards : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 4 3 5 7 7 5 4 3 5 2 ...
# $ Genetic Risk         : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 3 4 5 6 7 5 3 2 6 4 ...
# $ Chronic Lung Disease : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 2 2 4 7 6 4 2 3 5 3 ...
# $ Balanced Diet        : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 2 2 6 7 7 6 2 4 5 3 ...
# $ Obesity              : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 4 2 7 7 7 7 4 3 5 3 ...
# $ Smoking              : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 3 2 2 7 8 2 3 1 6 2 ...
# $ Passive Smoker       : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 2 4 3 7 7 3 2 4 6 3 ...
# $ Chest Pain           : Ord.factor w/ 9 levels "1"<"2"<"3"<"4"<..: 2 2 4 7 7 4 2 3 6 4 ...
# $ Coughing of Blood    : Ord.factor w/ 9 levels "1"<"2"<"3"<"4"<..: 4 3 8 8 9 8 4 1 5 4 ...
# $ Fatigue              : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 3 1 7 4 3 7 3 3 1 1 ...
# $ Weight Loss          : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 4 3 7 2 2 7 4 2 4 2 ...
# $ Shortness of Breath  : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 2 7 8 3 4 8 2 2 3 4 ...
# $ Wheezing             : Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 2 8 2 1 1 2 2 4 2 6 ...
# $ Swallowing Difficulty: Ord.factor w/ 8 levels "1"<"2"<"3"<"4"<..: 3 6 1 4 4 1 3 2 4 5 ...
# $ Finger Nails Clubbing: Ord.factor w/ 9 levels "1"<"2"<"3"<"4"<..: 1 2 4 5 2 4 1 2 6 4 ...
# $ Frequent Cold        : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 2 1 6 6 4 6 2 3 2 2 ...
# $ Dry Cough            : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 3 7 7 7 2 7 3 4 4 1 ...
# $ Snoring              : Ord.factor w/ 7 levels "1"<"2"<"3"<"4"<..: 4 2 2 5 3 2 4 3 1 5 ...
# $ Level                : Ord.factor w/ 3 levels "Low"<"Medium"<..: 1 2 3 3 3 3 1 1 2 2 ...
# - attr(*, ".internal.selfref")=<externalptr>
#   - attr(*, "index")= int(0)
```

Die saubere data.table wird in eine neue Variable kopiert.

```
## In besäuberte data.table abspeichern
lungcancer_clean = copy(lungcancer)
```

2.4.5. Visualisieren des Datensatzes

Wir fahren mit der Visualisierung des Datensatzes fort, um ein Gefühl für die Datenverteilung und -abhängigkeiten zu bekommen. Zunächst wir ein Pairplot erstellt und abgespeichert. Da diese Datei wegen der vielen Features enorm groß ist, ist es empfehlenswert diese Datei unter ./plots/pairplot.svg in z.B. einem Browser mit Zoom-Möglichkeit zu betrachten. Dank der Eigenschaft als Vektorgrafik können so auch sehr feine Strukturen klar angezeigt werden.

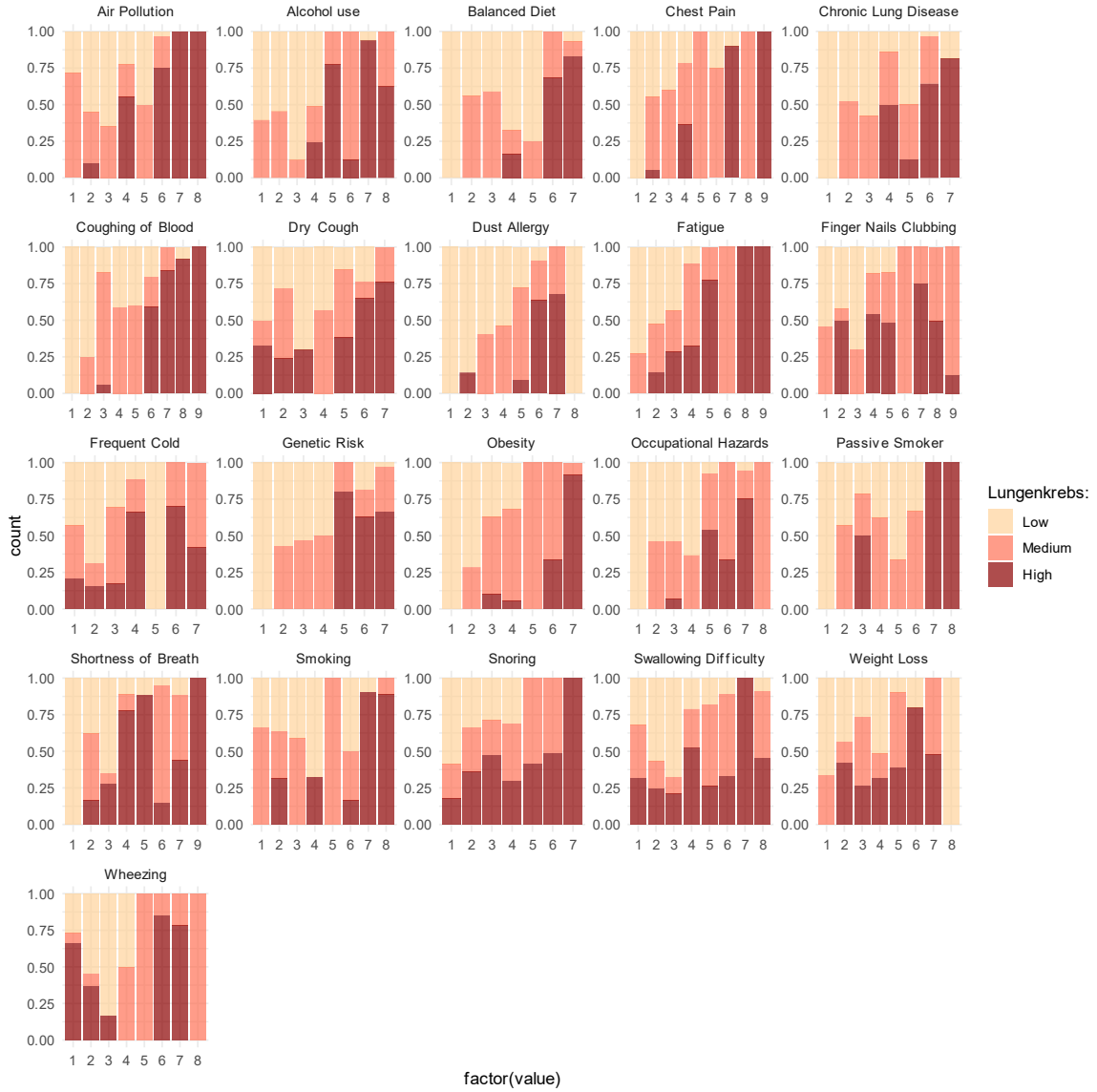
```r
## Übersichtsgrafik Pairplots aller Features
ggpairs(lungcancer_clean)
### Speichern des Plots
ggsave(filename = "pairplot.svg",
  plot = last_plot(),
  device = "svg",
  path = "./plots",
  scale = 1,
  width = 11000,
  height = 15000,
  units = "px",
  dpi = 300,
  limitsize = FALSE,
)
```

In der Darstellung fällt auf, dass die wenigsten Features Normalverteilt sind. Die Daten für „Snoring“ und „Dust Allergy“ sind schief verteilt. Die Merkmale der Features sind ebenfalls sehr heterogen über die möglichen Level verteilt (unterste Zeile). Die anderen Features zeigen keine auffälligen Verteilungen.

Als nächstes wird die Verteilung der Risikolevel innerhalb der Stufen der ordinalen Features visualisiert:

```r
## Barplots für kategorische Spalten
lungcancer_bar %>% pivot_longer(!Level, values_to = "value") %>%
                  # ggplot(aes(x = value, fill = factor(Level))) +
                  ggplot(aes(x=factor(value), fill=ordered(Level, c("Low", "Medium", "High")))) +
                  scale_fill_manual(values=c("burlywood1", "coral1", "darkred")) +
                  geom_bar(position="fill", alpha=.7)+
                  theme_minimal() +
                  labs(fill="Lungenkrebs:") +
                  facet_wrap(~name, scales="free")
### Speichern des Plots
ggsave(filename = "barplot.svg",
       plot = last_plot(),
       device = "svg",
       path = "./plots",
       scale = 1,
       width = 3000,
       height = 3000,
       units = "px",
       dpi = 300,
       limitsize = FALSE,
)
```

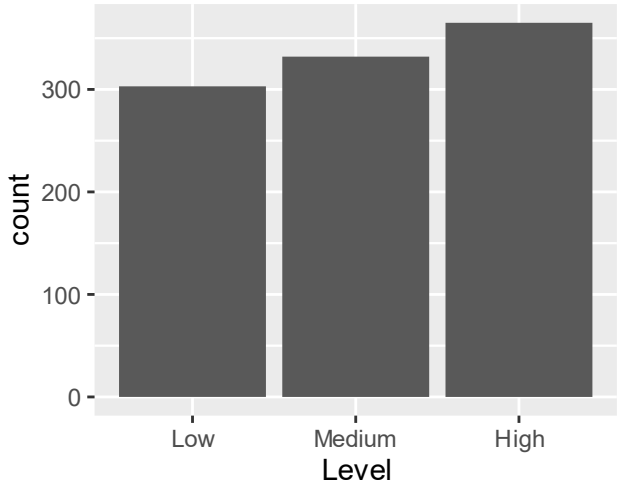
Diese Vektorgrafik ist ebenso enorm groß und unter ./plots/barplot.svg zur genaueren Betrachtung zu finden. Bei der Betrachtung der Barplots in Figur 1 fällt auf, dass die höheren Risikolevel eher bei den höheren ordinalen Werten der Features zu finden sind (helle Farbe = geringeres Risiko, dunklere Farbe = höheres Risiko). Besonders ausgeprägt ist dieses Muster bei den Faktoren „Air Pollution“, „Balanced Diet“, „Chronic Lung Disease“, „Coughing of Blood“, „Fatigue“, „Genetic Risk“, „Obesity“ und „Passive Smoker“. Ebenso sind umgekehrt die niedrigen Risikolevel eher bei den geringeren ordinalen Faktoren zu finden. Eine Ausnahme bildet hierbei das Feature „Wheezing“ und „Finger Nail Cubbing“, diese zeigen ordinal augenscheinlich beliebige Verteilungen der Risikolevel.



Figur 1: Barplots der Verteilung der Risikolevel für jede Stufe der ordinalen Features.

Ebenso wird die Verteilung der Samples pro Risikolevel untersucht, um eine mögliche Imbalance des Datensets auszuschließen. Dies wird durch den folgenden Code implementiert:

```
## Check Balance der Target-Klassen: keine große Imbalance
ggally_barDiag(lungcancer_clean,
               mapping = ggplot2::aes(x = Level),
               rescale = FALSE)
### Speichern des Plots
ggsave(filename = "target_balance.svg",
       plot = last_plot(),
       device = "svg",
       path = "./plots",
       scale = 1,
       width = 1000,
       height = 800,
       units = "px",
       dpi = 300,
       limitsize = FALSE,
)
```



Figur 2: Barplot der Risikolevelverteilung des Datensatzes. Die Daten sind annehmbar balanciert verteilt.

Tabelle 1: Tabelle der Korrelationswerte zwischen Feature und Risikolevel.

	Level
Level	1
Obesity	0.827435099588706
Coughing of Blood	0.782091675264986
Alcohol use	0.718710321764722
Dust Allergy	0.713838788275694
Balanced Diet	0.706273021135558
Passive Smoker	0.703594416182797
Genetic Risk	0.701302723149288
Occupational Hazards	0.673254877104492
Chest Pain	0.645461182604721
Air Pollution	0.636038492456584
Fatigue	0.627547100109645
Chronic Lung Disease	0.609971332919029
Smoking	0.519530145457183
Shortness of Breath	0.490907945067147
Frequent Cold	0.444016774902805
Dry Cough	0.373968359707329
Weight Loss	0.352737547070895
Snoring	0.289365954726551
Finger Nails Clubbing	0.280062851289599
Swallowing Difficulty	0.24914177369964
Wheezing	0.242793801841305
Gender	0.164985159880837
Age	0.0600478052265497

Des Weiteren wird die Korrelation der einzelnen Features mit dem Risikolevel für Lungenkrebs überprüft:

```r
## Korrelation einzelner Spalten mit dem Level des Lungenkrebs
corr_level <- lungcancer_clean %>%
              mutate_if(is.factor, as.numeric) %>%
              cor() %>%
              as.data.frame() %>%
              select(Level) %>%
              arrange(-Level)
### Plotten und speichern der Tabelle
png("./plots/correlation_table.png", height=600, width=300)
grid.table(corr_level)
dev.off()
```

```r
## Korrelationsplot für kategorische Features
model.matrix(~0+., data = lungcancer_clean) %>%
  cor(use = "pairwise.complete.obs",
      method = "spearman") %>%
  ggcorrplot(show.diag = FALSE,
             type = "lower",
             lab = TRUE,
             lab_size = 2)
### Speichern der Korrelationsmatrix
ggsave(filename = "correlation_plot_spearman.svg",
       plot = last_plot(),
       device = "svg",
       path = "./plots",
       scale = 1,
       width = 15000,
       height = 10000,
       units = "px",
       dpi = 300,
       limitsize = FALSE,
)
```

Hier wird oben im Code eine Spearman[7] Korrelationsmatrix für kategorische Features erstellt und gespeichert. Der Korrelationsplot ist wie die erste Vektorgrafik ebenfalls zu groß. Sie ist unter ./plots/correlation_plot.svg zu finden. Hier sind sowohl im positiven als auch im negativen Bereich Korrelationen von > bzw. < 0,5 zu finden. Einen Korrelationswert von unter -0,8 erreichen nur zwei Featurepaare,

Passive Smoker 6 und Passive Smoker 7

Snoring 1 und Snoring 2

wobei Korrelationswerte von größer bzw. gleich 0,8 hingegen insgesamt 8 Featurepaare besitzen:

Alcohol use 1 und Dust Allgergy 1

Alcohol use 1 und Occupational Hazards 1

Alcohol use 1 und Genetic Risk 1

Dust Allgergy 1 und Occupational Hazards 1

Dust Allergy 1 und Genetic Risk 1

Occupational Hazards 1 und Genetic Risk 1

Occupational Hazards 1 und Chronic Lung Disease 1

Genetic Risk 1 und Chest Pain 1

Auffällig ist hierbei, dass nur Featurekategorien der Stufe 1 hohe Korrelationswerte aufweisen. Die inverse Korrelation von aufeinanderfolgenden Stufen von „Snoring“ und „Passive Smoker“ sticht ebenfalls heraus.

2.5.5. Transformieren und vorbereiten des Datensatzes

Die Daten werden in Trainings- (80%) und Testdaten (20%) zerteilt und die Dimensionen sowie der Objekttyp überprüft:

```
325  ## Splitten zu 80/20 und Conversion zu data frame für preprocessing
326  split <- sample(1:nrow(lungcancer_clean), as.integer(0.8*nrow(lungcancer_clean)), F)
327  train <- as.data.frame(lungcancer_clean[split,])
328  test <- as.data.frame(lungcancer_clean[-split,])
329
330  ## Checken, ob Dimensionen erhalten sind
331  dim(train)
332  # [1] 800  24
333  dim(test)
334  # [1] 200  24
335  class(train)
336  # [1] "data.frame"
337  class(test)
338  # [1] "data.frame"
```

Training- und Testdaten wurden beim Vorgang des Aufteilens in data.frame Objekte umgewandelt, was hier aber nicht weiter stört, da die Manipulation des Datensatzes größtenteils abgeschlossen ist.
Nun wird eine Funktion für die Präprozessierung der Daten erstellt, die in diesem Fall nur die Min-Max-Skalierung der Spalte „Age“ beinhaltet:

[7] https://statsandr.com/blog/pearson-spearman-kendall-correlation-by-hand/ (Stand: 18.01.2024, 20:38)

```r
## Funktion für Preprocessing, das auf die Daten angewandt werden soll
preprocessing <- function(df){

  # Normalisierung: Min-Max-Scalierung der Spalte Age
  process <- preProcess(df["Age"], method = c("range"))
  df["Age"] <- predict(process, df["Age"])

  return(df[, names(df)!="Level"])
}

## Anwenden der preprocessing-Funktion auf Train und Test Datenset
x_train <- preprocessing(train)
x_test <- preprocessing(test)
y_train <- train[, "Level"]
y_test <- test[, "Level"]
```

Anschließend werden die präprozessierten Datensätze noch in Features (x) und Targets (y) aufgespalten.

2.5.6. Erstellen und Trainieren des Models

Jetzt wird das Random Forest Model angelegt und dessen Parameter gesetzt. Eine Kreuzvalidierung wird unter dem Parameter trControl ebenfalls implementiert.

```r
## Festlegen der Trainingseinstellungen
ctrl <- trainControl(method = "cv",
                     number = 5,
                     verboseIter = TRUE,
                     classProbs = TRUE,
                     savePredictions = TRUE)

grid <- expand.grid(mtry = seq(5, ncol(x_train),
                    by = 5))

model <- caret::train(x_train,
                      y_train,
                      method = "rf",
                      tuneGrid = grid,
                      trControl = ctrl)
```

Über den Modelnamen werden die Parameter des Models ausgegeben:

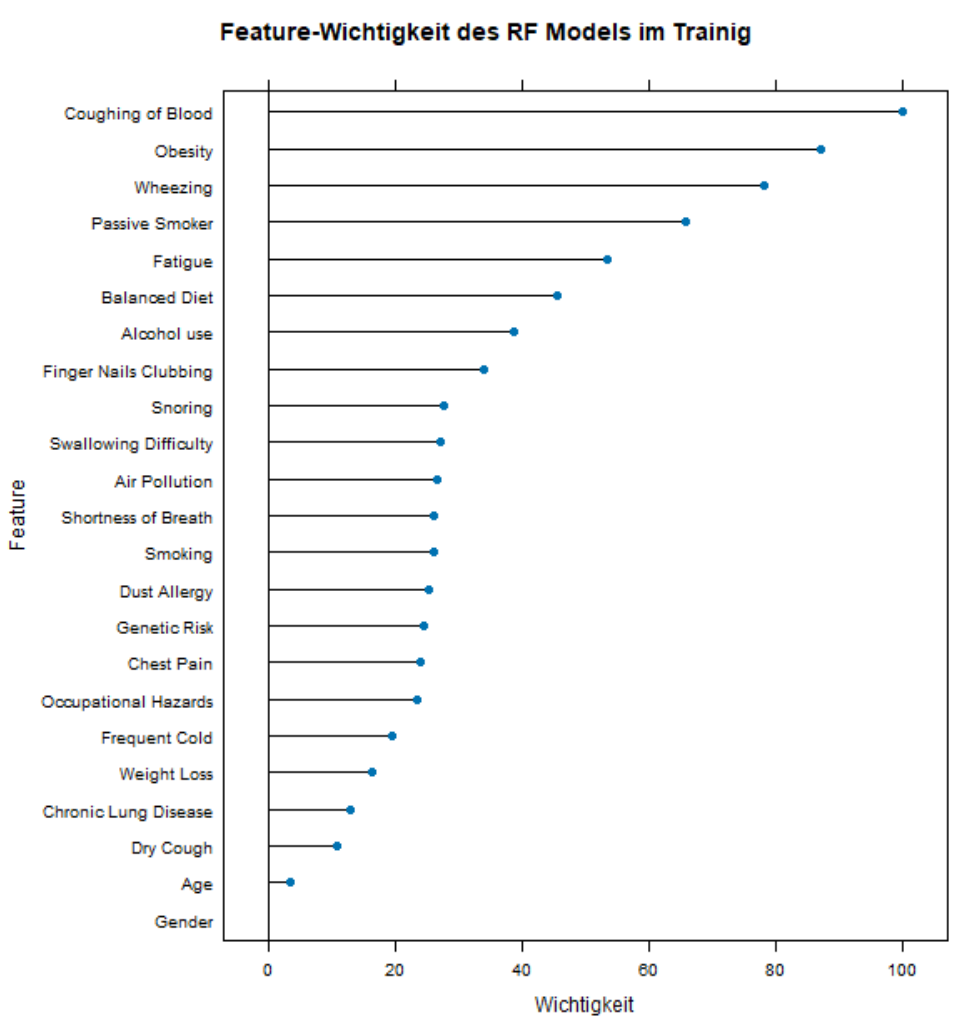
```r
model
# Random Forest
#
# 800 samples
# 23 predictor
# 3 classes: 'Low', 'Medium', 'High'
#
# No pre-processing
# Resampling: Cross-Validated (5 fold)
# Summary of sample sizes: 640, 641, 640, 640, 639
# Resampling results across tuning parameters:
```

Mit dem Trainingsdatensatz erreichen der Random Forest eine Genauigkeit von 100%:

```r
# mtry  Accuracy  Kappa
# 05    1         1
# 10    1         1
# 15    1         1
# 20    1         1
#
# Accuracy was used to select the optimal model using the largest value.
# The final value used for the model was mtry = 5.

## Plotten und speichern der Wichtigkeit der Features im RF Algorithmus
png("./plots/feature_importance.png", height=600, width=600)
plot(varImp(model),
     main = "Feature-Wichtigkeit des RF Models im Trainig",
     xlab = "Wichtigkeit",
     ylab = "Feature")
dev.off()
```

Schließlich plotten wir noch eine Reihenfolge der für den Random Forest Algorithmus wichtigsten Features für die Einschätzung des Risikolevels.



Die hier in Figur 3 zu sehende Auflistung zeigt, dass ein Abhusten von Blut der wichtigste Prädiktor für die Einschätzung des Risikolevels ist, dicht gefolgt von Übergewicht, Keuchen und passivem Rauchen. Eher unwichtig sind hingegen Alter und Geschlecht.

Figur 3: Absteigend sortierter Plot der wichtigsten Features für die Einschätzung des Risikolevels durch den Random Forest Algorithmus.

2.5.7. Testen des trainierten Models

Nun wird das Model mit unbekannten Daten getestet. Die Genauigkeit im Test liegt bei 100%

```
## Vorhersage auf Grundlage der Testdaten
prediction <- predict(model, newdata = x_test)

## Confusionsmatrix
confusionMatrix(prediction, y_test)
# Reference
# Prediction Low Medium High
# Low     63      0    0
# Medium   0     64    0
# High     0      0   73
#
# Overall Statistics
#
# Accuracy : 1
# 95% CI : (0.9817, 1)
# No Information Rate : 0.365
# P-Value [Acc > NIR] : < 2.2e-16
# Kappa : 1
# Mcnemar's Test P-Value : NA
```

Da es sich um eine medizinische Anwendung handelt, ist allerdings auch der Klassifikationsreport mit Sensitivität, Spezifität und balancierter Genauigkeit von Interesse. Diese erreichen ebenfalls alle 100%.

```
# Statistics by Class:
#                      Class: Low Class: Medium Class: High
# Class: Low Class: Medium Class: High
# Sensitivity               1.000          1.00       1.000
# Specificity               1.000          1.00       1.000
# Pos Pred Value            1.000          1.00       1.000
# Neg Pred Value            1.000          1.00       1.000
# Prevalence                0.315          0.32       0.365
# Detection Rate            0.315          0.32       0.365
# Detection Prevalence      0.315          0.32       0.365
# Balanced Accuracy         1.000          1.00       1.000
```

2.5.7. Abfragen der genutzten R Packages

Mit dem folgenden Code werden die genutzten R Packages abgefragt:

```
pkgs <- NCmisc::list.functions.in.file("Frank_Sabrina__BDR__Projektarbeit__12_2023.R")
summary(pkgs)
#                                Length      Class      Mode
# .GlobalEnv                          1     -none- character
# c(".GlobalEnv", "package:caret")    1     -none- character
# c("package:graphics", "package:base")  1     -none- character
# package:base                       30     -none- character
# package:caret                       4     -none- character
# package:data.table                  2     -none- character
# package:dplyr                       3     -none- character
# package:GGally                      2     -none- character
# package:ggcorrplot                  1     -none- character
# package:ggplot2                     9     -none- character
# package:grDevices                   2     -none- character
# package:gridExtra                   1     -none- character
# package:NCmisc                      1     -none- character
# package:stats                       3     -none- character
# package:tidyr                       1     -none- character
# package:utils                       1     -none- character
```

2.5 Ergebnisse

Das Model konnte eine hohe Genauigkeit von 100% erreichen. Da jede übersehene potentielle Erkrankung schlimmere Folgen hat, als ein zu hoch eingestuftes Risiko, ist das Augenmerk auf die Sensitivität des Tests ebenfalls von großer Bedeutung. Auch diese erreicht mit 100% einen perfekten Wert.
Das R-Projekt ist auf GitHub zu finden und öffentlich einsehbar:
https://github.com/SabiFrank/R_Projektarbeit_12_2023.git

2.7 Ausblick

Mit in Zukunft weiteren gefunden Risikofaktoren und neueren Algorithmen kann die Risikostratifizierung für Lungenkrebs sicherlich noch weiter verfeinert werden. Größere Datensätze durch größer angelegte Studien können hier natürlich noch mehr Wissen generieren.
Die nicht Normalverteilung einiger Features wurde aufgrund von Zeitmangel beim Präprozessieren unbeachtet gelassen, diese müsste bei Weiterentwicklung des Modells noch mit einbezogen werden. Des Weiteren wäre vielleicht eine Feature Selection angebracht, allerdings funktioniert das Model mit diesen Features gut. Nötig wären geeignete Ansätze für ordinale daten wie z.B. Chi-Squared test (contingency tables) oder Mutual Information[8]. Ebenfalls möglich wäre eine rekursive Feature Elimination via des bereits verwendeten caret Pakets[9].

[8] https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/ (Stand: 18.01.2024, 18:10)
[9] https://topepo.github.io/caret/recursive-feature-elimination.html#backwards-selection (Stand: 18.01.2024, 18:10)