Silja Wiegmann

Hättest du die Titanic überlebt?

Eine kurze Einführung in das Data Mining mit freier Software

TYP DES DOKUMENTS | TYPE OF THE DOCUMENT
Zeitschriftenartikel / Journal Article

API

OPEN ACCESS

Publiziert im Januar 2023
DOI: 10.15460/apimagazin.2023.4.1.130

Zeitschriftenartikel

Begutachtet

Begutachtet:

Tom Alby
HAW Hamburg
Deutschland

Erhalten: *20. November 2022*

Akzeptiert: *30. November 2022*

Publiziert: *31. Januar 2023*

Empfohlene Zitierung:

WIEGMANN, Silja, 2023: Hättest du die Titanic überlebt? Eine kurze Einführung in das Data Mining mit freier Software. In: *API Magazin* **4**(1) [Online] Verfügbar unter: DOI 10.15460/apimagazin.2023.4.1.130

Hättest du die Titanic überlebt?
Eine kurze Einführung in das Data Mining mit freier Software

Silja Wiegmann[1]*

[1] Hochschule für Angewandte Wissenschaften Hamburg, Deutschland
Studentin im 3. Semester des Masterstudiengangs Digitale Transformation der Informations- und Medienwirtschaft

* Korrespondenz: redaktion-api@haw-hamburg.de

Zusammenfassung

Am 10. April 1912 ging Elisabeth Walton Allen an Bord der „Titanic“, um ihr Hab und Gut nach England zu holen. Eines Nachts wurde sie von ihrer aufgelösten Tante geweckt, deren Kajüte unter Wasser stand. Wie steht es um Elisabeths Chancen und hätte man selbst das Unglück damals überlebt? Das Titanic-Orakel ist eine algorithmusbasierte App, die entsprechende Prognosen aufstellt und im Rahmen des Kurses „Data Science“ am Department Information der HAW Hamburg entstanden ist. Dieser Beitrag zeigt Schritt für Schritt, wie die App unter Verwendung freier Software entwickelt wurde. Code und Daten werden zur Nachnutzung bereitgestellt.

Schlagwörter: Data Mining, Open Source, R, Shiny, Klassifikation, Support Vector Machines, Entscheidungsbaum

Would you have survived the Titanic?
A brief introduction to data mining using free software

Abstract

On April 10, 1912, Elisabeth Walton Allen boarded the „Titanic“ to bring her belongings to England. One night she was awakened by her distraught aunt, whose cabin was under water. What are Elisabeth's chances and would one have survived the disaster at the time? The Titanic Oracle is an algorithm-based app for predicting such outcomes that was built as part of the class „Data Science“ at the Department of Information at HAW Hamburg. This step-by-step article shows how the app was created using free software. Both code and data are provided for reuse.

Keywords: Data Mining, Open Source, R, Shiny, Classification, Support Vector Machines, Decision Trees

DOI: 10.15460/apimagazin.2023.4.1.130

1 Einleitung

1912 war ein aufregendes Jahr für Elisabeth Walton Allen: Der Liebe wegen hatte es die Amerikanerin ins Ausland verschlagen, und nun würde der aufwendige Emigrationsprozess bald abgeschlossen sein. Am 10. April ging sie im britischen Southampton an Bord des Passagierschiffs „Titanic“, um ihr Hab und Gut aus den USA nach England zu holen. Vier Tage lang genoss Elisabeth die Reise in der ersten Klasse, bis sie des Nachts von ihrer aufgelösten Tante geweckt wurde, deren Kajüte unter Wasser stand (Baxter et al. 2020). Erschrocken stellte sich Elisabeth die alles entscheidende Frage: Werde ich überleben? Das Titanic-Orakel[1] birgt Antworten (Abb. 1).

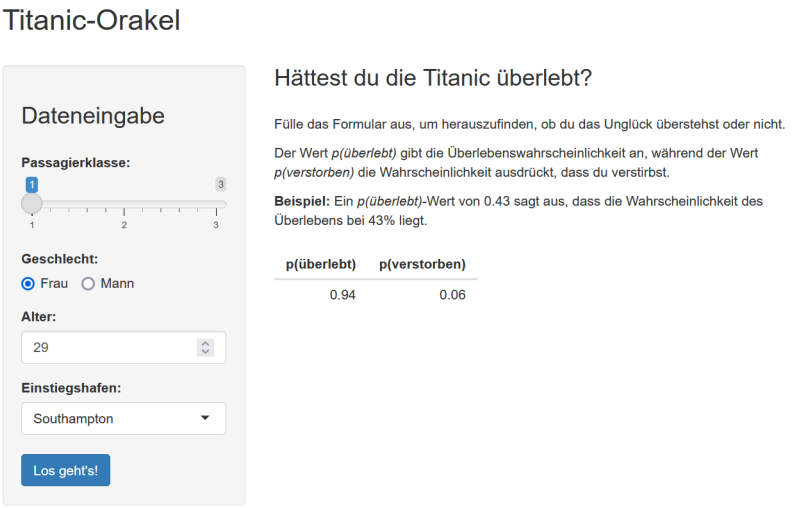

Abb. 1: Überlebenswahrscheinlichkeit von Elisabeth Walton Allen (Eigene Darstellung)

Mit einer äußerst hohen Überlebenswahrscheinlichkeit von 94% entkam Elisabeth dem sinkenden Schiff, verklagte die Reederei und feierte eine Doppelhochzeit mit ihrer Schwester Claire (Baxter et al. 2020). Wie das Titanic-Orakel zu seiner richtigen Einschätzung kam, wird in diesem Beitrag gezeigt.

2 Data Mining mithilfe von Klassifikationsalgorithmen

Die Daten der Titanic-Reisenden sind häufig Bestandteil von Klassifikations-Tutorials, die zeigen, wie sich anhand historisch überlieferter Merkmale wie Alter oder Passagierklasse Überlebensprognosen aufstellen lassen (Alby 2022, S. 210). Ein solches Durchforsten von Daten nach Mustern wird als *Data Mining* bezeichnet.

1 Aufzurufen unter: https://titanic-orakel.shinyapps.io/model/ [Online, Zugriff am 19.11.2022].

DOI: 10.15460/apimagazin.2023.4.1.130

2.1 Was ist Data Mining?

Eine Datenanalyse durchläuft insgesamt fünf Stufen, die in Abbildung 2 dargestellt sind.

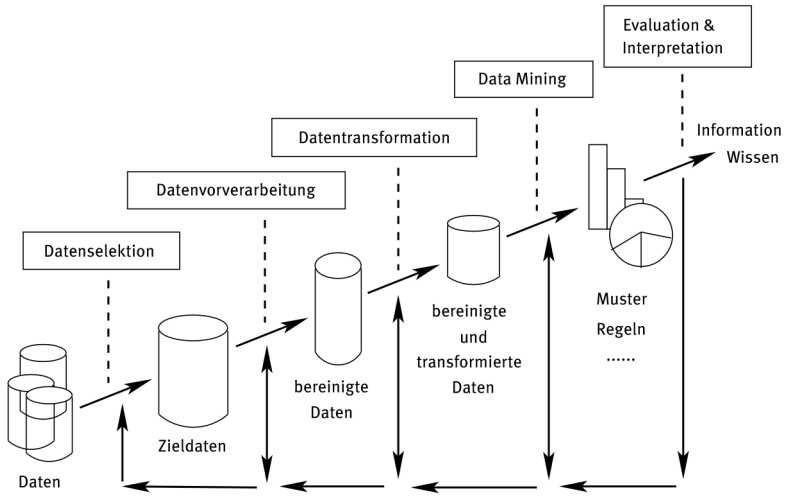

Abb. 2: Ablauf einer Datenanalyse (Cleve & Lämmel 2020, S. 3)

Eingangs werden die relevanten Daten ausgewählt und für die Analyse vorbereitet. Die bereinigten und transformierten Daten werden nach Regeln durchsucht, die es abschließend zu interpretieren gilt. Dieser Beitrag fokussiert sich auf die vierte Stufe des Analyseprozesses, also das Data Mining bzw. Datenschürfen, welches sich definieren lässt als „die Extraktion von Wissen aus Daten“ (Cleve & Lämmel 2020, S. 41 f.). Der Weg vom Datum zum Wissen lässt sich anhand des Merkmals ‚überlebt/verstorben‘ veranschaulichen: Die Zahl 1 ist ein schlichtes Datum. Im Titanic-Datensatz steht die 1 dafür, dass eine Person überlebt hat, wodurch die Zahl zu einer Information wird. Gelingt es, herauszufinden, warum eine Person überlebt hat, so wird Wissen generiert.

2.2 Wie werden Vorhersagen getroffen?

Mit Data Mining können verschiedene Ziele verfolgt werden. Eines davon ist die Klassifikation, die neue Datensätze in existierende Klassen einordnet. Voraussetzung hierfür ist ein Datensatz, der bereits klassifiziert vorliegt (Cleve & Lämmel 2020, S. 59; 61). Diese Bedingung ist im Fall der Titanic-Daten erfüllt, da sowohl die Merkmale der Reisenden als auch ihr Über- oder Ableben bekannt sind. Ziel ist es, auf Basis dieser historischen Daten Vorhersagen für die Überlebenschancen neuer Personen treffen.

Um ein solches Vorhersagemodell zu erhalten, wird das Datenset in Trainings- und Testdaten aufgeteilt. Anhand der Trainingsdaten wird ein Modell entwickelt, welches anschließend an den Testdaten geprüft und verbessert wird. Weil sowohl die Trainings- als auch die Testdaten klassifiziert vorliegen, wird diese Art des maschinellen Lernens auch als *Supervised Learning* bezeichnet: Der Algorithmus lernt, wie die

DOI: 10.15460/apimagazin.2023.4.1.130

Merkmale einer Person die Klassenzugehörigkeit beeinflussen (Alby 2022, S. 48 f.; Cleve & Lämmel 2020, S. 64 f., 258). Mit diesem gewonnenen Wissen können nun auch neue Personen, deren Schicksal unbekannt ist, in die Klassen ‚überlebt/verstorben' eingeordnet werden.

3 Kommerzielle und freie Statistik-Software

Es gibt verschiedene Wege, diese Theorie in die Praxis umzusetzen. An Universitäten kommt seit den 1970er-Jahren häufig die Statistiksoftware SPSS zum Einsatz, die seit 2009 zum IT-Konzern IBM gehört und somit in die Sparte der proprietären bzw. kommerziellen Software fällt. Wer SPSS nutzen möchte, muss Geld bezahlen und sich an strenge Lizenzbedingungen halten, die den Kreis der Nutzungsberechtigten stark einschränken (Alby 2022, S. 68; Kohn & Öztürk 2017, S. 6; Lang 2003, S. 202). Diese Restriktionen bestehen, obwohl immaterielle Güter wie Software im Gegensatz zu physischen Produkten endlos und sogar gleichzeitig genutzt werden können, ohne sich zu verbrauchen. Mit kommerziellen Lizenzen werden Menschen trotzdem vom Konsum ausgeschlossen, z.B. durch das Beschränken der Geräteanzahl, auf denen eine Software installiert werden darf (Dewenter & Rösch 2015, S. 15 f.; Lang 2003, S. 201 f.).

Eine Reaktion auf diese künstliche Verknappung ist die Free-Software- bzw. Open-Source-Bewegung.[2] Ihr Slogan lautet: „[F]ree as in freedom, not free as in beer" (Peterson 2018). Freie Software ist kostenlos, doch im Kern geht es nicht darum, Geld zu sparen. Edward Snowden, bekannter Whistleblower und überzeugter Nutzer freier Software, formulierte den zentralen Gedanken der Bewegung einst wie folgt: „We can't compete with Apple, we can't compete with Google, directly, in the field of resources. [… But] [w]e can compete on the ground of ideology because ours is better" (LibrePlanet 2016).

Um nachzuvollziehen, was diese freie Ideologie beinhaltet, bedarf es eines Rückblicks in die 1970er-Jahre, als IBM wegen Verstoßes gegen das Kartellrecht rechtlich zur Verantwortung gezogen wurde. Als Reaktion auf den Prozess spaltete sich die Hard- und Software-Entwicklung in zwei unabhängige Industriezweige auf. Zum Geschäftsmodell der neuen Software-Industrie gehört bis heute die Geheimhaltung des eigenen Programmcodes (May 2009, S. 369).

Gegen die Idee, dass man Code besitzen könne, wandte sich der Programmierer Richard Stallman, indem er 1985 die gemeinnützige *Free Software Foundation* gründete. Sein in eigenen Worten „greatest hack" ist das Formulieren der *GNU General Public License* (GPL), die das freie Ausführen, Kopieren und Bearbeiten des Quellcodes sowie die Verbreitung angepasster Versionen erlaubt – unter der Voraussetzung,

2 Streng genommen folgen diese Bewegungen unterschiedlichen Philosophien, die jedoch beide von einer „logic of openness" angetrieben (May 2009, S. 370) und daher hier zusammengefasst werden.

 DOI: 10.15460/apimagazin.2023.4.1.130

dass alle Versionen ebenfalls unter der GNU GPL stehen, was auch als *Copyleft* bezeichnet wird (Free Software Foundation 2007; May 2009, S. 369). Die Vorteile offenen Quellcodes reichen von der Fehlerkorrektur über das Aufdecken von Spionagetools bis hin zu mehr Bildungschancen und Teilhabe an der Informationsgesellschaft (Lang 2003, S. 203–205).

3.1 Die freie Programmiersprache R

Eine freie Alternative zu SPSS[3] ist die Programmiersprache R, die unter der GNU GPL verfügbar ist.[4] Das Besondere an R ist die Vielzahl an ergänzenden Paketen für statistische Berechnungen und Grafiken, die frei heruntergeladen werden können. Selbst ohne Pakete lassen sich mit wenigen Handgriffen maßgeschneiderte Plots bzw. Diagramme erstellen, die in wissenschaftlichen Publikationen verwendet werden können. Neben der Wissenschaft wird R vermehrt auch in der Wirtschaft eingesetzt (Alby 2022, S. 67 f.).

3.2 Die Entwicklungsumgebung RStudio (Posit)

Weil sich das reine R nicht intuitiv nutzen lässt, ist die zusätzliche Installation der integrierten Entwicklungsumgebung *RStudio (Posit)*[5] zu empfehlen, die in einer Open-Source-Version zur Verfügung steht.[6] Die grafische Oberfläche vereint u.a. einen Code-Editor, eine Konsole und einen Arbeitsbereich für Datenobjekte. Neben R-Skripten lassen sich in RStudio auch sogenannte R-Notebooks erstellen, die in der Auszeichnungssprache *Markdown* geschrieben werden und eine Kombination aus erläuterndem Text und ausführbaren Code-Elementen erlauben (Xie et al. 2022).

4 Erstellung des Vorhersagemodells

Nach der Installation von R und RStudio wird im nächsten Schritt ein Vorhersagemodell erstellt, das neue Personen in die Klassen ‚überlebt/verstorben' einordnet.

4.1 Datensatz

Im Internet existieren unterschiedliche Versionen des beliebten Titanic-Datensatzes.[7] Das Titanic-Orakel basiert auf einem offenen Datensatz, der von mehreren Händen stückweise ergänzt wurde und auf der Plattform *Open Machine Learning* frei zur Verfügung steht (Vanschoren 2017). Abbildung 3 zeigt einen Auszug aus dem aufbereiteten Datensatz.

3 Es existiert auch eine freie SPSS-Variante namens PSPP, die jedoch nur über einen Teil der Funktionen des Originals verfügt (Alby 2022, S. 68).
4 Download unter: https://www.r-project.org/ [Online, Zugriff am 19.11.2022].
5 RStudio wurde im Oktober 2022 in Posit umbenannt (Allaire & Wickham 2022). Bei Manuskripterstellung (November 2022) wird das Produkt auf der Website jedoch unverändert als RStudio bezeichnet.
6 Download unter: https://posit.co/download/rstudio-desktop/ [Online, Zugriff am 19.11.2022].
7 Die Suche nach Datensets mit dem Stichwort „Titanic" ergibt 1.135 Treffer bei Kaggle (Stand: November 2022).

DOI: 10.15460/apimagazin.2023.4.1.130

	survived	pclass	sex	age	embarked
1	1	1	1	29.0000	1
2	1	1	0	0.9167	1
3	0	1	1	2.0000	1
4	0	1	0	30.0000	1
5	0	1	1	25.0000	1

Abb. 3: Auszug aus dem aufbereiteten Titanic-Datensatz (Eigene Darstellung)

Die erste Zeile enthält die Daten von Elisabeth: Wie bereits bekannt ist, hat die 29 Jahre alte *(age: 29)* Frau *(sex: 1),* die in Southampton an Bord ging *(embarked: 1)* und in der ersten Passagierklasse reiste *(pclass: 1),* das Unglück überlebt *(survived: 1).* Diese fünf Merkmale wurden für insgesamt 1.044 Mitreisende lückenlos erfasst, deren Daten nun in die Analyse einfließen. Hierfür müssen zunächst die erforderlichen Pakete installiert und geladen werden. Diese Arbeit übernimmt das Paket *pacman:*

```
if (!require("pacman")) install.packages("pacman")
pacman::p_load(farff, caret, e1071, rpart, rpart.plot)
```

Code-Element 1

Anschließend wird der Datensatz, der in dem im maschinellen Lernen verbreiteten Format ARFF vorliegt, eingelesen:

```
titanic <- readARFF("titanic-orakel_data.arff")
```

Code-Element 2

4.2 Aufteilung in Trainings- und Testdaten

Mithilfe des Pakets *caret* wird eine Aufteilung in Trainings- und Testdaten vorgenommen:

```
set.seed(1234)                          # Reproduzierbarkeit gewährleisten
trainIndex <- createDataPartition(
  y = titanic$survived,                 # vorherzusagende Zielvariable
  p = .8,                               # 80% der Daten ins Trainingsset
  list = FALSE                          # Listenausgabe verhindern
)
```

Code-Element 3

Dabei achtet das Paket darauf, dass die Merkmalsattribute im Trainings- und Testset gleichmäßig verteilt sind. So wird gewährleistet, dass nicht etwa durch Zufall alle Frauen in einem Set landen (Alby 2022, S. 216).

8 Der Link zum aufbereiteten Datensatz sowie zum R-Notebook ist im Begleitmaterial (Kap. 7) zu finden.

DOI: 10.15460/apimagazin.2023.4.1.130

4.3 Der Algorithmus: Support Vector Machines

Es gibt eine Reihe von unterschiedlichen Klassifikationsalgorithmen. Das Titanic-Orakel verwendet *Support Vector Machines* (SVM), deren Funktionsweise in Abbildung 4 dargestellt ist.

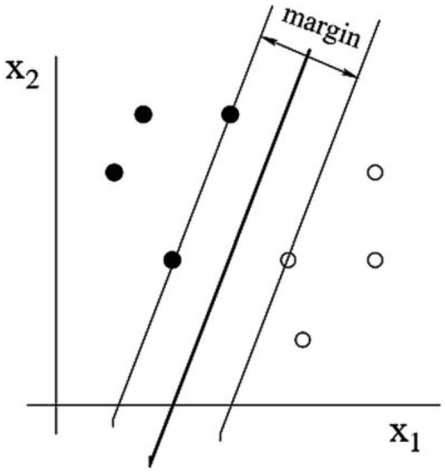

Abb. 4: Support Vector Machines (Kubat 2021, S. 86)

Wie es im maschinellen Lernen häufig der Fall ist, basieren auch SVM auf Distanzen, also dem Messen von Abständen. Aus diesem Grund kann der Algorithmus nur metrische Datentypen verarbeiten, was bei der Datenaufbereitung beachtet werden muss. Die breite Hauptlinie stellt den besten Klassifikator dar, der von zwei parallel verlaufenden Hilfsvektoren flankiert wird, welche die Distanz visualisieren. Ziel ist es, einen möglichst weiten Abstand der Hilfsvektoren von der Hauptlinie herzustellen und die Anzahl der Fehler bzw. Datenpunkte, die in die falsche Klasse eingeordnet werden, so weit wie möglich zu reduzieren (Alby 2022, S. 221f.; Cleve & Lämmel 2020, S. 139; Kubat 2021, S. 86).

SVM sind in dem Paket *e1071* enthalten. Nachdem der Algorithmus trainiert wurde, können anhand des entstandenen Modells Vorhersagen für die Testdaten getroffen werden:

```
set.seed(1234)                # Reproduzierbarkeit gewährleisten
model_svm <- svm(
  formula = survived ~ .,     # abhängige Zielvariable
  data = train_data,          # Trainingsset verwenden
  probability = TRUE          # Vorhersagen ermöglichen
  )

# Vorhersagen erstellen
pred_svm <- predict(model_svm, test_data[,-1], probability = TRUE)
```

Code-Element 4

Wie gut ist dem Algorithmus die Einteilung in Überlebende und Verstorbene gelungen? Ein Gütemaß zur Beurteilung von Klassifikatoren ist die *Confusion Matrix,* die Fehlerraten aufführt und schnell zu erstellen ist (Alby 2022, S. 52; 235; Cleve & Lämmel 2020, S. 253):

DOI: 10.15460/apimagazin.2023.4.1.130

```
confusionMatrix(pred_svm, test_data$survived)
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 116  27
         1   7  58

               Accuracy : 0.8365
            Sensitivity : 0.9431
            Specificity : 0.7059
```

Code-Element 5

Die *Accuracy* drückt aus, dass das Modell rund 84% der Beobachtungen richtig klassifizieren konnte. Insgesamt hat das Modell 34-mal eine falsche Prognose aufgestellt: In 27 Fällen hat eine totgeglaubte Person überlebt (falsch-positiv), während in sieben Fällen Reisende, die vom Algorithmus als Überlebende klassifiziert wurden, tatsächlich gestorben sind (falsch-negativ). In welchem Maße die Verstorbenen erkannt wurden, wird von der Richtig-positiv-Rate *(Sensitivity)* ausgedrückt, die 94% beträgt. Ihr Pendant für die Überlebenden ist die Richtig-negativ-Rate *(Specificity),* die bei 71% liegt. Im Vergleich dieser Kenngrößen wird deutlich, dass das Modell relativ gut darin ist, die Verstorbenen zu identifizieren. Überlebende korrekt zu klassifizieren, ist hingegen schwieriger.

5 Ergebnispräsentation

Die Confusion Matrix zeigt, wie nützlich das Modell ist, ist jedoch visuell nicht ansprechend gestaltet. Zudem lässt sich aus ihr nicht ablesen, was genau die Überlebenschancen von Reisenden wie Elisabeth erhöht. Nach welchen Kriterien entscheidet das Titanic-Orakel? Mit einem Entscheidungsbaum lässt sich offenlegen, welche Merkmale bei der Klassifikation besonders ausschlaggebend sind. Darüber hinaus lässt sich das Modell in Form einer interaktiven App darstellen.

5.1 Wissensgenerierung mit Entscheidungsbäumen

Der Entscheidungsbaum ist ein weiterer Klassifikationsalgorithmus. Modell und Grafik werden mithilfe der Pakete *rpart* und *rpart.plot* erstellt:

```
set.seed(1234)               # Reproduzierbarkeit gewährleisten
model_dt <- rpart(
  formula = survived ~ .,    # abhängige Zielvariable
  data = train_data,         # Trainingsset verwenden
  method = "class"           # Klassifikationsbaum erstellen
  )

# Grafik erstellen
plot_dt <- rpart.plot(model_dt)
```

Code-Element 6

DOI: 10.15460/apimagazin.2023.4.1.130

Die entstandene Grafik (Abb. 5) dient nicht nur der Visualisierung, sondern auch dem Data Mining, weil der Entscheidungsweg des Algorithmus Schritt für Schritt nachvollzogen werden kann.

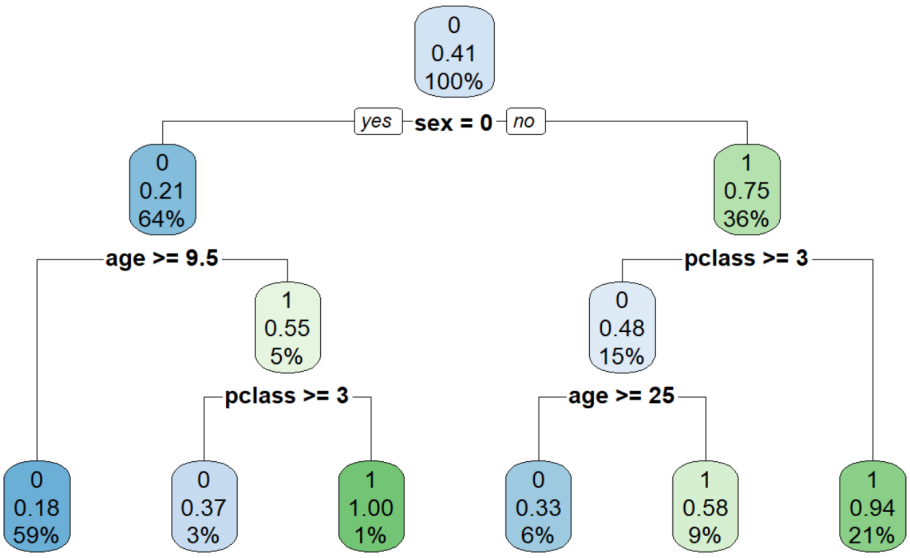

Abb. 5: Entscheidungsbaum (Eigene Darstellung)

Diese Interpretierbarkeit unterscheidet den Algorithmus von anderen Klassifikatoren. Alle Verzweigungen stellen die Ergebnisse von Bedingungen dar, aus denen Wissen in Form von logischen Regeln im *Wenn-dann*-Format abgeleitet werden kann (Cleve & Lämmel 2020, S. 75 f.; Kubat 2021, S. 93). Oben an der Baumwurzel steht das einflussreichste Merkmal, anhand dessen die Daten am besten klassifiziert werden können (Alby 2022, S. 217). Auf der Titanic war eindeutig das Geschlecht am ausschlaggebendsten: Während die Wahrscheinlichkeit, zu überleben, für Frauen bei 75% liegt, haben Männer (*sex: 0*) lediglich eine Überlebensrate von 21%.

Das zweitwichtigste Kriterium für Männer ist ihr Alter: Kleine Kinder überleben mit einer Wahrscheinlichkeit von 55%, wohingegen die Chancen für Jungen ab neuneinhalb Jahren rapide sinken. Bei den Frauen spielt wiederum die Passagierklasse eine wichtige Rolle, die zugleich ein Indikator für sozioökonomischen Status ist (Vanschoren 2017): Frauen aus der ersten oder zweiten Klasse weisen eine Überlebensrate von 94% auf, während Passagierinnen der dritten Klasse nur mit einer Wahrscheinlichkeit von 48% überleben. So erklärt der Entscheidungsbaum auch Elisabeths Schicksal: Ihr Geschlecht und ihre Passagierklasse führen in Kombination zu einer äußerst hohen Überlebenswahrscheinlichkeit.

5.2 Interaktive Shiny-App

Shiny ist ein Framework für Web-Apps, das aus demselben Produktionshaus wie RStudio stammt. Weil Shiny nicht nur ein Paket, sondern auch eine Serverumgebung ist, können die Apps mit wenigen Handgriffen livegeschaltet und fortan per URL aufgerufen werden. User:innen können mit der App interagieren, indem ihre spontanen Eingaben verarbeitet und als neue Ansicht im Browser angezeigt werden (Alby

 DOI: 10.15460/apimagazin.2023.4.1.130

2022, S. 292; Wiley & Wiley 2020, S. 357 f.). Nach diesem Prinzip funktioniert auch das Titanic-Orakel. In Anlehnung an Wiley & Wiley (2020, S. 361) zeigt das folgende Code-Element eine „bare-bones"-Version der App, um die grundlegende Funktionsweise von Shiny zu veranschaulichen:[9]

```
ui <- shinyUI(fluidPage(
  titlePanel("Titanic-Orakel"),
  sidebarLayout(
    sidebarPanel( [Dateneingabe] ),
    mainPanel( [Anzeige der Prognose] ))
  ))

server <- shinyServer(function(input, output, session) {
  observeEvent( [Erstellen der Prognose] )
})

shinyApp(ui = ui, server = server)
```

Code-Element 7

Jede Shiny-App besteht aus zwei Komponenten, dem User-Interface (UI) und dem Server (Alby 2022, S. 295). Das UI enthält die Browser-Ansicht und erhält im Fall des Titanic-Orakels ein Sidebar-Layout, welches die Anzeige in ein Sidebar- und ein Main-Panel aufteilt. Während das Main-Panel einen erläuternden Text bereitstellt und später die Prognose anzeigt, enthält das Sidebar-Panel mehrere Widgets für die Dateneingabe. Mit Klick auf den „Los geht's!"-Button wird der Server aktiv: Anhand des in Kapitel 4 entwickelten SVM-Modells wird eine Überlebensprognose aufgestellt, die an das UI gereicht und dort ausgegeben wird.

Die fertige App lässt sich mit Klick auf den „Run App"-Button in RStudio ausprobieren. Um das Titanic-Orakel livezuschalten, bietet sich der von Shiny angebotene Cloud-Hosting-Dienst *Shinyapps.io* an. Obwohl der Dienst kostenlos ist, steht er unter einer kommerziellen Lizenz (Posit Software, PBC 2022), womit der Open-Source-Ansatz dieses Beitrags an seine Grenzen kommt. Dass dies nicht problematisch sein muss, zeigt der Verweis auf den bekannten Software-Entwickler Linus Torvalds, der im Zuge der *BitKeeper*-Kontroverse[10] verkündete, kein „free software zealot" zu sein: Ist ein kommerzielles Tool besser für seine Zwecke geeignet als ein freies, so zögert er nicht, es zu verwenden (Brown 2018).

Weil das Betreiben eines eigenen Servers eine Reihe von Herausforderungen mit sich bringt, lässt sich in Bezug auf das Hosting eine ebenso pragmatische Einstellung einnehmen. Um Shinys Cloud-Hosting-Dienst nutzen zu können, muss zunächst ein Shinyapps.io-Account angelegt werden.[11] Eingeloggt kann der Reiter „Tokens" aufgerufen und das Token über den „Show"-Button kopiert werden (Abb. 6).

9 Der Link zum R-Skript ist im Begleitmaterial (Kap. 7) zu finden.
10 Als die Linux-Community eine Software zur Versionsverwaltung benötigte, wählte Torvalds ein kommerzielles Produkt aus. Die Entscheidung war so umstritten, dass er wenige Jahre später eine eigene Software namens *Git* entwickelte (The Linux Foundation 2015).
11 Sign-Up unter: https://www.shinyapps.io/admin/#/signup [Online, Zugriff am 19.11.2022].

DOI: 10.15460/apimagazin.2023.4.1.130

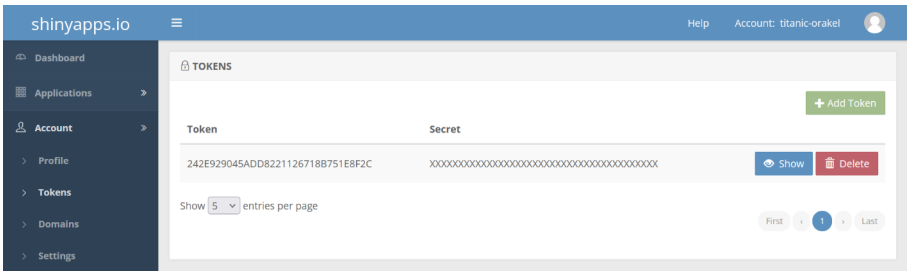

Abb. 6: Token in Shinyapps.io (Eigene Darstellung)

In RStudio lässt sich die Verbindung zum Shinyapps.io-Account über den „Publish"-Button (Abb. 7) herstellen, wofür das Token eingefügt werden muss.

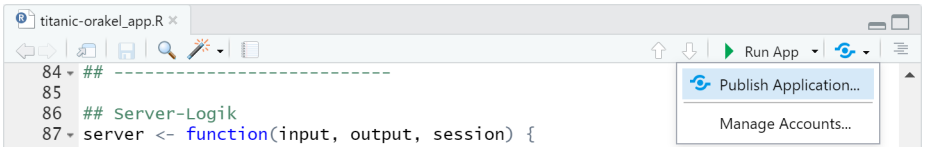

Abb. 7: Button „Publish Application" (Eigene Darstellung)

Im nächsten Schritt können das R-Skript und das SVM-Modell gemeinsam unter einem beliebigen Titel veröffentlicht werden (Abb. 8).

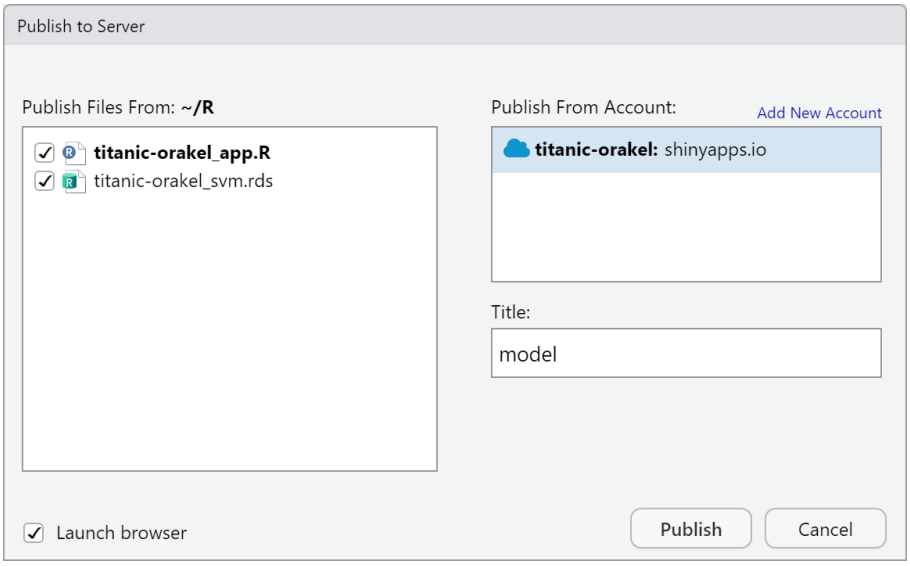

Abb. 8: Veröffentlichung der App (Eigene Darstellung)

Ab sofort steht das Titanic-Orakel frei im Netz zur Verfügung und kann unter der URL https://titanic-orakel.shinyapps.io/model/ erreicht werden.

6 Fazit

Das Wissen, das spielerisch aus dem Titanic-Datensatz geschürft wurde, hat wenig Neuigkeitswert, doch es gibt viele reale Situationen, in denen Klassifikationsalgorithmen still in das Alltagsleben der Menschen eingreifen (Alby 2022, S. 207; Cleve & Lämmel 2020, S. 19). Wer ist kreditwürdig und wer nicht? Kann ein E-Mail-Postfach Spam-Nachrichten verlässlich erkennen? Die Algorithmen, die hinter diesen Anwen-

DOI: 10.15460/apimagazin.2023.4.1.130

dungsszenarien stecken, funktionieren im Kern nicht viel anders als das Titanic-Orakel. Dieser Beitrag hat versucht, etwas Licht in die verborgene Welt der Algorithmen zu bringen und dabei auf die Potenziale freier Software aufmerksam zu machen. Weil offene Praktiken nicht nur die Produktqualität, sondern auch den Wissenstransfer erhöhen, lohnt es sich, für transparente Standards in der Software-Entwicklung einzutreten.

7 Begleitmaterial

Es gibt viele Aspekte, die sich am Titanic-Orakel verändern oder ergänzen ließen. So könnte z.B. ein anderer Algorithmus ausprobiert oder ein neues Layout für das UI entworfen werden. Zu diesem Zweck sind auf GitHub[12] der aufbereitete Titanic-Datensatz, das R-Notebook für die Modellerstellung sowie das R-Skript der App zu finden, die unter der Lizenz *CC0 1.0 Universal* steht und somit in die Gemeinfreiheit entlassen wurde.

12 Aufzurufen unter: https://github.com/titanic-orakel/model [Online, Zugriff am 19.11.2022].

DOI: 10.15460/apimagazin.2023.4.1.130

Literatur und Quellen

ALBY, Tom, 2022. *Data Science in der Praxis: Eine verständliche Einführung in alle wichtigen Verfahren*. 1. Auflage. Bonn: Rheinwerk Verlag. ISBN 978-3-8362-8462-2

ALLAIRE, J.J. und WICKHAM, Hadley, 2022. *RStudio is becoming Posit* [online]. Boston, MA: Posit Software, PBC, 27.07.2022 [Zugriff am: 20.11.2022]. Verfügbar unter: https://posit.co/blog/rstudio-is-becoming-posit/

BAXTER, Trevor, BELL, Gavin, ENGBERG-KLARSTRÖM, Peter, FINDLAY, Michael A., GOWAN, Phillip und SÖLDNER, Hermann, 2020. *Elisabeth Walton Allen* [online]. o.O.: Encyclopedia Titanica, 01.04.2020 [Zugriff am: 20.11.2022]. Verfügbar unter: https://www.encyclopedia-titanica.org/titanic-survivor/elisabeth-walton-allen.html

BROWN, Zack, 2018. A Git Origin Story. In: *The Linux Journa*l [online]. 27.07.2018 [Zugriff am: 20.11.2022]. Verfügbar unter: https://www.linuxjournal.com/content/git-origin-story

CLEVE, Jürgen und LÄMMEL, Uwe, 2020. *Data Mining* [online]. 3. Auflage. Berlin: De Gruyter [Zugriff am: 20.11.2022]. PDF E-Book. ISBN 978-3-11-067627-3. Verfügbar unter: DOI: 10.1515/9783110676273

DEWENTER, Ralf und RÖSCH, Jürgen, 2015. *Einführung in die neue Ökonomie der Medienmärkte* [online]. *Eine wettbewerbsökonomische Betrachtung aus Sicht der Theorie der zweiseitigen Märkte*. Wiesbaden: Springer Gabler [Zugriff am: 20.11.2022]. PDF E-Book. ISBN 978-3-658-04736-8. Verfügbar unter: DOI: 10.1007/978-3-658-04736-8

FREE SOFTWARE FOUNDATION, 2007. *GNU General Public License* [online]. o.O.: The Free Software Foundation, 29.06.2007 [Zugriff am: 20.11.2022]. Verfügbar unter: https://www.gnu.org/licenses/gpl-3.0.html.en

KOHN, Wolfgang und ÖZTÜRK, Riza, 2017. *Statistik für Ökonomen* [online]. *Datenanalyse mit R und SPSS*. 3., überarbeitete Auflage. Berlin: Springer Gabler [Zugriff am: 20.11.2022]. PDF E-Book. ISBN 978-3-662-50442-0. Verfügbar unter: DOI: 10.1007/978-3-662-50442-0

KUBAT, Miroslav, 2021. *An introduction to machine learning* [online]. 3. Auflage. Cham: Springer [Zugriff am: 20.11.2022]. PDF E-Book. ISBN 978-3-030-81935-4. Verfügbar unter: DOI: 10.1007/978-3-030-81935-4

LANG, Bernard, 2003. Der Kampf um die freie Software. In: *Schweizerisches Jahrbuch für Entwicklungspolitik* [online]. 10.06.2010 [Zugriff am: 20.11.2022]. Verfügbar unter: DOI: 10.4000/sjep.565

LIBREPLANET, 2016. The last lighthouse: Edward Snowden in conversation with Daniel Kahn Gillmor. In: *MediaGoblin* [online]. 23.03.2016 [Zugriff am: 20.11.2022]. Verfügbar unter: https://media.libreplanet.org/u/libreplanet/m/libreplanet-2016-the-last-lighthouse-3d51/

MAY, Christopher, 2009. Globalizing the logic of openness: Open source software and the global governance of intellectual property. In: CHADWICK, Andrew und HOWARD, Philip N., Hrsg. *Routledge handbook of internet politics*. London: Routledge. S. 364–375. ISBN 978-0-203-96254-1

PETERSON, Christine, 2018. *How I coined the term 'open source'*. o.O.: Red Hat, Inc., 01.02.2018 [Zugriff am: 20.11.2022]. Verfügbar unter: https://opensource.com/article/18/2/coining-term-open-source-software

POSIT SOFTWARE, PBC, 2022. *Get your Shiny apps online* [online]. Boston, MA: Posit Software, PBC, 2022 [Zugriff am: 20.11.2022]. Verfügbar unter: https://posit.co/products/open-source/shinyserver/

THE LINUX FOUNDATION, 2015. *10 years of Git: An interview with Git creator Linus Torvalds* [online]. San Francisco, CA: The Linux Foundation, 06.04.2015 [Zugriff am: 20.11.2022]. Verfügbar unter: https://www.linuxfoundation.org/blog/blog/10-years-of-git-an-interview-with-git-creator-linus-torvalds

VANSCHOREN, Joaquin, 2017. *Titanic* [online]. o.O.: OpenML, 16.10.2017 [Zugriff am: 20.11.2022]. Verfügbar unter: https://www.openml.org/search?type=data&sort=runs&id=40945&status=active

WILEY, Matt und WILEY, Joshua F., 2020. *Advanced R 4 data programming and the cloud* [online]. *Using PostgreSQL, AWS, and Shiny*. 2. Auflage. New York, NY: Apress [Zugriff am: 20.11.2022]. PDF E-Book. ISBN 978-1-4842-5973-3. Verfügbar unter: DOI: 10.1007/978-1-4842-5973-3

XIE, Yihui, ALLAIRE, J. J. und GROLEMUND, Garrett, 2022. *R Markdown* [online]. *The definitive guide*. o. O.: Chapman & Hall/CRC [Zugriff am: 20.11.2022]. HTML E-Book. Verfügbar unter: https://bookdown.org/yihui/rmarkdown/