Machine Learning und Prognosen

Umsetzung mit R im SQLServer 2017 anhand von Taxifahrten in New York

Bachelorthesis

Studiengang *Angewandte Informatik*

Duale Hochschule Baden-Württemberg Mannheim

von

Leonhard Applis

Abgabedatum:	26.09.2018
Matrikelnummer, Kurs:	2086307, TINF15/AI-BI
Ausbildungsfirma:	Atos Information Technology GmbH
Betreuer der Dualen Hochschule:	Prof. Tobi Chosen

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Bachelorthesis mit dem Thema

> *Machine Learning und Prognosen Umsetzung mit R im SQLServer 2017 anhand von Taxifahrten in New York*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Fürth, den 28. Juni 2018

_____

Leonhard Applis

Abstract

Englisch Abstract to be done

title:	Machine-Learning and Prognosis
author:	Leonhard Applis
matriculation number:	2086307
class:	TINF15/AI-BI
supervisor DHBW:	???
supervisor Atos:	Jonas Mauer

Kurzfassung

Deutscher Abstract muss gemacht werden

Titel:	Machine Learning und Prognosen
Author:	Leonhard Applis
Matrikelnummer:	2086307
Kurs:	TINF15/AI-BI
Betreuer der Dualen Hochschule:	Prof. Tobi Chosen
Betreuer der Firma:	Jonas Mauer

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

DBMS	Database Management System
DHBW	Duale Hochschule Baden-Württemberg
SQL	Structured Query Language
ETL	Extract-Transform-Load

1 Einleitung

> "Hasta la Vista, Baby!"
>
> *Arnold Schwarzenegger* in Terminator 2

Dieses Zitat zählt wohl zu den bekanntesten der Filmgeschichte, und markiert einen der ersten bühnenreifen Auftritte *künstlicher Intelligenz*. Neben österreichischen Bodybuildern beschäftigt dieses Thema seit bald einem Jahrhundert Wissenschaftler, Ethiker und Science-Fiction-Fans gleichermaßen. Was vor zwei Jahrzehnten noch genauso fantasievoll wie schwebende Autos klang, wird in den Softwareschmieden des 21. Jahrhunderts Wirklichkeit:

Künstliche Intelligenzen besiegen Schachprofis, organisieren unsere Kalender, analysieren Bilder und helfen Pandemien einzudämmen. Neben diesen bahnbrechenden Erfolgen gibt es auch weiterhin vielversprechende Forschung auf diesem Themengebiet, zum Beispiel computergesteuerte Autos. Aber was ist künstliche Intelligenz eigentlich?

Der Begriff der künstlichen Intelligenz ist sehr weit gefächert - ein Kernelement davon stellt das *Machine Learning* dar. Dieser Bereich, der sich auf die Erstellung von Modellen anhand von Trainingsdaten stützt, hat in den letzten Jahren durch *Neuronale Netze* stark an Bedeutung gewonnen. Die Gründe hierfür sind vielseitig, dennoch sind Zwei ins Besondere zu nennen: Zum Einen sind Computer deutlich Leistungsfähiger geworden, und Aufgaben die früher einen Supercomputer benötigten, sind heute durch ein Smartphone umsetzbar. Zum Anderen sind deutlich mehr Bereiche digitalisiert, und die gewonnenen Daten detaillierter.

Genau diesem Themengebiet widmet sich diese Bachelorarbeit: Machine-Learning und explizit Neuronalen Netzen.

1.1 Ziel der Arbeit

Ziel dieser Arbeit ist es, ein Grundverständnis für Machine-Learning Algorithmen zu schaffen und dem Leser die Möglichkeit zu geben, diese mit dem SQL-Server 2017 selbst umzusetzen.

Hierfür werden die Theorie verschiedener Algorithmen detailliert vorgestellt und in R umgesetzt.

Ebenfalls wird ein detailliertes Fallbeispiel mit Versuchsaufbau und Ergebnissen erarbeitet, damit der Leser eine Einschätzung der Algorithmen vornehmen kann ohne selbst Experimente durchzuführen.

Es ist **nicht** Ziel dieser Arbeit, einen Vergleich zwischen unterschiedlichen Machine-Learning Ansätzen und Frameworks zu ziehen. Auch wird ausschließlich mit R und dem SQL-Server gearbeitet.

Zudem werden weder Grundlagen der Sprachen SQL und R, noch die Vorbereitung des Fallbeispiels geschildert.

1.2 Aufbau der Arbeit

Kapitel 2 dieser Arbeit bildet die Theorie zu modernen Ansätzen des Machine Learnings. Es werden die Algorithmen für lineare Regression, logistische Regression sowie Neuronale Netzwerke detailliert vorgestellt (In Reihenfolge der Nennung). Dieses Kapitel stellt einen rein theoretischen Teil der Arbeit dar, und beinhaltet keine Umsetzung der Algorithmen als Programme.

Darauf aufbauend werden in Kapitel 3 zunächst Grundlagen zu Microsofts SQL-Server 2017 und R geklärt, anschließend liegt der Schwerpunkt des Kapitels auf der Umsetzung von Machine-Learning Algorithmen in R. Innerhalb des Abschnittes 3.3 finden sich allgemeine Programme in T-SQL und R.

Kapitel 4 widmet sich der Umsetzung eines Fallbeispiels eines Taxiunternehmens. Zunächst werden in Abschnitt 4.1 die Ausgangslage der Daten sowie die Ziele des Fallbeispiels exakt definiert.

In Abschnitt 4.2 werden die Stammdaten des Taxiunternehmens und die Wetterdaten in Eigenschaften, Umfang und Bedeutung für Machine Learning dargestellt.

Der Abschnitt 2.4 behandelt die Verwendung der Programme aus 3.3 unter Bezugnahme auf das Fallbeispiel. Abschluss dieses Abschnittes bildet die Erzeugung eines Modells.

Abschnitt 4.4 benutzt das in Abschnitt 2.4 erzeugte Modell, um die in Abschnitt 4.1 vorgestellten Anforderungen zu bearbeiten. Hier befinden sich die eigentlichen Ergebnisse des Experiments sowie Programme die Prognosen durchführen.

Abschluss des Kapitels bildet Abschnitt 4.5 in dem Methoden zum Test eines Modells vorgestellt und durchgeführt werden.

Abschluss der Arbeit bildet in Kapitel 5 ein Fazit über die Qualität der Prognosen unter Berücksichtigung der Komplexität einzelner Teilaufgaben.

1.3 Voraussetzungen an den Leser

Innerhalb dieses Punktes werden die Kenntnisse abgesteckt, die der Leser für das Verständnis der Arbeit benötigt, welche **nicht** im Rahmen dieser Arbeit vorgestellt werden.

- **Mehrdimensionale Algebra:** Im Rahmen dieser Arbeit werden komplexe Algorithmen und Konzepte der mehrdimensionalen Algebra benötigt.

  Schwerpunkte liegen hier v.A. auf dem Lösen von mehrdimensionalen Gleichungen und Matrixoperationen. Der Umfang hierbei entspricht dem Besuch der Vorlesung *Mathematik II*. Literaturempfehlung Lin.Alg

- **Stochastik:** Zur Bewertung der Algorithmen werden tiefere Kenntnisse der Stochastik und Statistik benötigt. Die benötigten Schwerpunktthemen sind Verteilungsfunktionen, Hypothesentests und Korrelation. Literaturempfehlung Stochastik/-Statistik

- **R:** Die Programmiersprache R muss dem Leser im Umfang eines Basiskurses bekannt sein. Sie wird im Zuge der Arbeit verwendet, allerdings werden grundlegende Elemente nicht vorgestellt.

- **SQL:** Die Konzepte von SQL und der Dialekt von T-SQL sind in fortgeschrittenen Zügen benötigt. Die Verwendung von R innerhalb des SQL-Servers wird im Zuge der Arbeit vorgestellt.

2 Grundlagen zu Machine-Learning

In diesem Kapitel werden die theoretischen Grundlagen moderner Machine-Learning Algorithmen vorgestellt.

Allgemeine Umsetzungen dieser Algorithmen finden sich im Abschnitt 3.2 zu R sowie konkret anhand des Fallbeispiels in Kapitel 4.

2.1 Bias

In diese Abschnitt werden kurz verschiedene Formen von *Bias* (dt. Abweichung, Verzerrung) vorgestellt. Diese Abweichungen spielen in allen Formen des Machine-Learnings und in der Auswahl der Trainingsdaten eine wichtige Rolle (vgl. [Pfe] Absatz 1) und werden in den entsprechenden Algorithmen berücksichtigt. Die nachfolgenden Arten von Bias stellen Überbegriffe dar - v.A. im Bereich der Psychologie wird deutlich genauer unterschieden.

Natürliche Varianz Je nach Art und Gestalt der Erhebung können systematische Schwankungen der Werte auftreten. Diese stellen natürliche Verhältnisse dar, da kein perfektes Modell erfasst werden kann.

Als Beispiel sei die Messung der Zimmertemperatur genannt: Zwei Thermometer können im selben Raum unterschiedliche Ergebnisse liefern - etwa weil sie auf unterschiedlichen Höhen befestigt sind oder eines im Windzug liegt. Es ist im Allgemeinen nicht möglich, ein perfektes Modell zu erstellen welches alle Faktoren berücksichtigt.

Die Natürliche Varianz ist als Hauptgrund zu nennen, warum in jedem (modernen) Machine-Learning Algorithmus eine Abweichung berücksichtigt ist.

Insbesondere ist zu betonen, das die Genauigkeit eines Models, welches auf Machine-

Learning beruht, nie höher sein kann als die Varianz der zugrunde liegenden Trainings-Daten.

Selection Bias Unter der Selektionsverzerrung versteht man einen Fehler der Ergebnisse, welcher durch die Auswahl einer **nicht repräsentativen** Stichprobe entsteht (vgl. [Ins] Definition). Ein Beispiel einer Selektionsverzerrung tritt auf [1], wenn Anhand der Umfragen auf einer Messe für vegane Ernährung die Ernährungsgewohnheiten aller Deutscher interpretiert wird.

Im Gegensatz dazu wäre diese Stichprobe sehr wohl geeignet, die Ernährung deutscher Veganer zu beurteilen.

Confirmation Bias Unter dem *Confirmation Bias* (dt. Bestätigungsfehler) versteht man mehrere psychologische Aspekte die zu einer Verzerrung der Ergebnisse durch den Prüfer führen (vgl. [Dar84] S. 21 Absatz 5 und S. 22 Absatz 1f). Im Wesentlichen bezieht sich diese Abweichung darauf, das unbewusst Ergebnisse so interpretiert werden um bestehende Meinungen zu bestätigen. Dies wird hauptsächlich über zwei Mechanismen erreicht: Die Interpretation nicht-übereinstimmender Ergebnisse und Daten als Fehlerhaft, sowie eine überproportionale Gewichtung übereinstimmender Ergebnisse. Hierzu gehört ebenfalls die explizite Suche nach Ergebnissen welche eine Hypothese bestätigen, ohne dieselbe Sorgfalt der Gegenhypothese zukommen zu lassen.

2.2 Lineare Regression

Als erster Machine-Learning-Algorithmus soll die Lineare Regression vorgestellt werden.

Stroetmann und oder Formelsammlungen zitieren

Auch wenn lineare Regression nicht mehr Bestandteil aktueller Forschung ist, sind viele Konzepte für weitere Erklärungen nützlich. Zudem können mithilfe linearer Regression sehr gute Ergebnisse erzielt werden.

[1] Es handelt sich hierbei um eine Vermutung

2.2.1 Konzept und Ziele linearer Regression

Als Beispiel für die einfache lineare Regression dient uns das Abschätzen des Bremsweges von PKWs. Hierfür benötigen wir eine Tabelle der Gestalt

Geschwindigkeit in km/h	Gewicht in kg	Bremsweg in m
50	1500	20
60	1400	30
90	1000	60
...	...	...

Es ist hierbei offensichtlich, das diese Messwerte zusammenhängen - lediglich die zugrunde liegende Formel ist uns Unbekannt.

Mithilfe linearer Regression wollen wir eine modellhafte Formel finden, die das beste Ergebnis anhand unserer Daten liefert. Die Werte müssen für

2.2.2 Einfache Lineare Regression

Zunächst gehen wir zur Vereinfachung davon aus, dass der Bremsweg lediglich von der Geschwindigkeit abhängt. Bezeichnen wir x_i als die Geschwindigkeit des i-ten Datensatzes der Tabelle 2.2.1 und y_i als den zugehörigen Bremsweg, kann man ein lineares Modell der Form

$$y_i := \vartheta_1 \cdot x_i + \vartheta_0 \tag{2.1}$$

herleiten. Wir wollen ϑ_1 und ϑ_0 so berechnen,dass der **Mean-Squared-Error** minimal ist.

$$\texttt{MSE}(\vartheta_0, \vartheta_1) := \frac{1}{m-1} \cdot \sum_{i=1}^{m} \left(\vartheta_1 \cdot x_i + \vartheta_0 - y_i\right)^2 \tag{2.2}$$

Die optimalen Ergebnisse des MSE liefern die Variablen:

$$\vartheta_1 = r_{x,y} \cdot \frac{s_y}{s_x} \quad \text{und} \quad \vartheta_0 = \bar{\mathbf{y}} - \vartheta_1 \cdot \bar{\mathbf{x}}. \tag{2.3}$$

Wobei $\mathbf{x}$ und $\mathbf{y}$ das arithmetische Mittel der beiden Variablen darstellt, sowie

s_x und s_y die Standart-Abweichungen. bei $r_{x,y}$ handelt es sich um den **Pearson-Korrelationskoeffizienten**.

Nach der Berechnung unserer *Gewichte* besitzen wir ein Modell, welches für jeden beliebigen Geschwindigkeitswert den Bremsweg berechnet.

Dennoch können wir davon ausgehen, das ein lineares Modell für komplexere Sachverhalte keine zufriedenstellenden Ergebnisse liefert. Deswegen wird nun die lineare Regression unter Berücksichtigung mehrerer unabhängiger Variablen behandelt.

2.2.3 Allgemeine Lineare Regression

Das Prinzip der allgemeinen linearen Regression ist das Gleiche: Wir suchen eine Funktion, welche uns den abhängigen Wert schätzt. Diese Funktion hat im Allgemeineren die Form $F : \mathbb{R}^m -> \mathbb{R}^1$, und bildet ein m-Eigenschaften umfassendes Tupel x_i auf einen Wert y_i ab.

Bezogen auf unser Beispiel 2.2.1 haben wir ein 2-Tupel x_i der Form <Geschwindigkeit,Gewicht> und weiterhin einen dazugehörigen Bremsweg y_i. Wir suchen eine Funktion $F(x_i) \approx y_i$.

Diese Funktion können wir ebenfalls durch ein lineares Modell ausdrücken. Sie hat die Gestalt:

$$F(x_i) = \vartheta_2 \cdot x_i^2 + \vartheta_1 \cdot x_i^1 + \vartheta_0 \cdot x_i^0 \tag{2.4}$$

Wobei x_i^n die n-te Komponente des i-ten Elementes darstellt. x^0 ist eine Erweiterung um den Bias. Interpretiert man die Tupel (erweitert um den Bias) als transponierten Vektor x^T so kann man die allgemeine Formel für größere Tupel zusammenfassen als:

$$F(x) = \sum_{n=0}^{m} x^n \cdot \vartheta_n = x^T \cdot \vec{w} \tag{2.5}$$

wobei $\vec{w} = \begin{pmatrix} \vartheta_0 \\ ... \\ \vartheta_n \end{pmatrix}$ der Gewichtsvektor ist. Nun lässt sich unsere Funktion um den Fehler zu berechnen definieren als:

$$\texttt{MSE} := \frac{1}{m-1} \cdot \sum_{i=1}^{m} \left(F\left(\mathbf{x}^{(i)}\right) - y^{(i)} \right)^2 = \frac{1}{m-1} \cdot \sum_{i=1}^{m} \left(\left(\mathbf{x}^{(i)}\right)^\top \cdot \vec{w} - y^{(i)} \right)^2 \tag{2.6}$$

Vektoren summieren und normalengleichung

2.2.4 Bewertung der Linearen Regression

Wie berechne ich die statistische Signifikanz meines Linearen Modells?

2.3 Klassifizerung

Hier vielleicht auch Stroetmann, oder etwas leichtgewichtigeres?

2.3.1 Konzept und Ziele von Klassifizierung

Hier Beispiel bringen, vllt Binäre Klassifizierung

2.3.2 Definitionen und Notationen

Features

Label & Klassen

Model

Accuracy

Supervised Learning

Unsupervised Learning

2.3.3 Digression: Gradientenanstieg

Erklärung was Stochastic Gradient Ascent ist, kurzes Vorgreifen warum man es braucht

2.3.4 Logistische Regression

Aktivierungsfunktion Hier wird kurz erklärt was es für Aktivierungsfunktionen gibt, der Bezug zur Stochastic/Wahrscheinlichkeit und kurzes Vorgreifen warum man es braucht

Sigmoid Was ist das, was macht die, Eigenschaften beim Ableiten (Siehe Stroetmann)

Bild zur Sigmoidfunktion als Plot, ArcTang und Gauss daneben

Kurze Erklärung warum man nicht die anderen benutzt

ReLU Was ist das?? Wieso ist das SSo super gut"

Modell der Logistischen Regression Wie spielen Aktivierungsfunktion, Optimierung und Lernen innerhalb der Logistischen Regression zusammen bei der Klassifizierung

2.4 Neuronale Netzwerke

Hier Aufbereitung, Anreicherung und Übersetzung von Selby [Sel]!

2.4.1 Modell künstlicher neuronalen Netzen

Historisches Konzept Hier sagen, wie das mit dem Gehirn zusammen hängt, woher kommt der Name und warum spricht man heute von künstlichen NN

Aufbau des Modells Klassisches Bild mit Input-Layers, Hidden Layers, Output Layers

Bezug auf Logistische und Lineare Regression, das diese NN ohne Hidden Layers sind

Zuordnung der Begriffe?

2.4.2 Hidden Layers

Wie wird das Modell der logistischen Regression erweitert?

Deep Learning = Mehr als 1 Hidden Layer

Manchmal: Deep Learning = Unsupervised Learning + viele Hidden Layers

2.4.3 Forward Propagation

Reinschmeißen eines Trainingsbeispiels und messen, wie schlimm es daneben liegt bzw. ob es daneben liegt

2.4.4 Backward Propagation

Nach dem Forward-Prograpagation nachjustieren der Gewichte in Matrix

Optimierungsfunktion und Gewichte

2.4.5 Training

Wie geht Training allgemein

worauf muss man bei Trainingsdaten achten, welche Größenordnungen sind notwendig

Anmerkung zu teilweise Export des Modells

3 SQLServer 2017 und R

In diesem Kapitel werden zunächst die Umgebung des SQL-Servers 2017 sowie die Programmierprache R kurz vorgestellt, bevor in Abschnitt 3.3 eine konkrete Umsetzung der unter Kapitel 2 gezeigten Algorithmen mit R erfolgt.

3.1 SQL-Server 2017

3.2 Programmiersprache R

3.3 Machine Learning im SQL-Server 2017

Innerhalb dieses Abschnittes befinden sich Code-Beispiele zur Umsetzung der in Kapitel 2 vorgestellten Algorithmen.

Es werden im Folgenden kurz die Einbindung der R-Skripte in TSQL behandelt, anschließend werden nur die R-Skripte für die einzelnen Punkte erläutert.

Möglichkeiten in R Die Sprache R besitzt verschiedene Optionen Machine-Learning Modelle zu erzeugen. Neben der Implementation *von Grund auf* gibt es eine Vielzahl von Paketen und Bibliotheken.

Für die lineare und logistische Regression werden die Bibliotheken *RevoscaleR* und *MicrosoftML* von Microsoft benutzt (Die Dokumentation findet sich unter [Mar]). Sie wird bereits mit dem SQL-Server geliefert. Hauptargument für diese Umsetzung waren die gründliche Dokumentation von Microsoft, die eine Benutzung innerhalb des SQL-Servers bereits behandelt, sowie die gemeinsame Produktfamilie die einen einheitlichen Technik-Stack ergibt.

Möglichkeiten in Python Der SQL-Server 2017 unterstützt neben einem R-Server ebenfalls eine Instanz des Microsoft ML-Servers. Dieses Open Source Projekt zu finden auf Github [?] stellt eine Alternative zu den in R vorgestellten Methoden dar.

Der ML-Server ist in Python implementiert und die Benutzung verhält sich ähnlich zu den Ansätzen von Tensorflow in Python.

Verwendung von R im SQL-Server Um R im SQL-Server zu benutzen wird die Stored Procedure *sp_execute_external_script* benötigt. Im Folgenden ein einfaches Beispiel:

```
1 EXECUTE sp_execute_external_script
2 @language = N'R',
3 @script = N'
4   mytextvariable <- c("hello", " ", input_data);
5   OutputDataSet <- as.data.frame(mytextvariable);',
6 @input_data = N' SELECT name FROM readers'
7 WITH RESULT SETS (([Greetings] char(20) NOT NULL));
```

Hierbei wird in Zeile 2 zunächst die Sprache als Parameter übergeben, in Zeile 4 wird innerhalb des R Skriptes ein Begrüßungs-String erstellt, welcher in Zeile 5 als Ausgabe wiedergeben wird.

In Zeile 6 wird die Inputvariable definiert, an dieser Stelle sind SQL Befehle und gültige T-SQL Variablen möglich. Es können beliebig viele Inputvariablen definiert werden.

In Zeile 7 wird die Ausgabe in Tabellenform normiert. Diese Zeile ist nicht notwendig.

Dieses Schema bleibt allen Skript-Aufrufen gleich. Im Folgenden werden nur die R-Skripte vorgestellt.

3.3.1 Lineare Regression

Für die diese Form der Regression gelten innerhalb des Paketes MicrosoftML folgende Bedingungen:

1. Alle Eingabewerte des Modells müssen (reelle)[1] Zahlen sein
2. Texteingabewerte müssen zuvor über einen Faktor realisiert werden. Dieser Faktor muss eine festgesetzte Anzahl an Leveln besitzen.
3. Der Ausgabewerte ist eine reelle Zahl

Um ein Modell für die lineare Regression zu erstellen, sind in R nur wenige Zeilen notwendig:

```
1 formel <- C ~ A+B;
2 model <- rxLinMod(formula=formel, data=TrainingsData);
3 serializedModel <- data.frame(payload = as.raw(serialize(model,
      connection=null)));
```

In der ersten Zeile wird zunächst eine allgemeine Formel definiert. Diese Formel ist zu interpretieren als $f : (A\ x\ B) \rightarrow C$, das '+' ist hierbei nicht als Addition zu verstehen.

In Zeile 2 wird das Modell mithilfe der Bibliothek RevoscaleR und dem Methodenaufruf rxLinMod erstellt **und** Trainiert. Als Parameter werden die Formel und die Trainingsdaten benötigt.

In der dritten Zeile findet eine Serialisierung des Modells statt - dies ist nicht notwendig für eine direkte Verwendung, ermöglicht allerdings das speichern des Modells innerhalb des SQL-Servers als Blob.

Um das Modell zu benutzen reichen ebenfalls wenige Zeilen R-Skript:

```
1 model <- unserialize(as.raw(serializedModel));
2 C <- rxPredict(model,data.frame(TestData));
```

[1] Es gibt andere Pakete, die komplexe Zahlen unterstützen

Hierbei wird zunächst in Zeile 1 das serialisierte Modell wieder nutzbar gemacht.

In Zeile 2 wird die Methode *rxPredict* der RevoScaleR-Bibliothek aufgerufen, welche aus den zu testenden Daten und dem Model eine Prognose erstellt.

3.3.2 Klassifikation

Für die Klassifikation mit RevoscaleR gelten folgende Bedingungen:

1. Alle Eingabewerte des Modells sind reelle Zahlen.
2. Texteingabewerte müssen zuvor über einen Faktor realisiert werden. Dieser Faktor muss eine festgesetzte Anzahl an Leveln besitzen.
3. Die Klasse stellt einen Faktor mit Level 2 dar.
4. Der Ausgabewerte ist eine Wahrscheinlichkeit, mit der die Ausprägung positiv ausfällt
5. Es kann gleichzeitig nur eine Klasse überprüft werden

Der R-Code verhält sich parallel zum Code der linearen Regression:

```
1 formel <- rain ~ temperature+humidity;
2 logitmodel <- rxLogit(formula = form, data = TrainingsData);
3 rainPropability <- rxPredict(model, data.frame(TestData));
```

Als Beispiel wurde hierbei die Voraussage gewählt, ob es regnet anhand von Temperatur und Luftfeuchtigkeit.

3.3.3 Neuronale Netze

Es ist Möglich, die im Abschnitt 2.4 vorgestellten Konzepte direkt in R umzusetzen. Ein gutes Tutorial liefert hierbei [Sel], welcher eine Schritt-Für-Schritt Anleitung und Erklärung bietet ein eigenes Neuronales Netz zu entwerfen. Das Tutorial von Selby setzt einen ähnlichen Blogeintrag von [**?**] in R um.

Innerhalb dieser Arbeit wird allerdings das Paket *MicrosoftML* verwendet.

4 Fallbeispiel: Prognose von Taxifahrten

Innerhalb dieses Kapitels wird das Fallbeispiel der Taxidaten behandelt. Zunächst erfolgt eine Zielsetzung, anschließend in den Abschnitten 4.2 und 4.3 eine Beschreibung des Versuchsaufbaus und Zuletzt in Abschnitt 4.4 eine systematische Durchführung der Prognosen zuzüglich Test dieser in Abschnitt 4.5.

4.1 Ziele und Anforderungen

Ziel des Fallbeispieles ist es, *lohnenswerte* Prognosen anhand von realistischen Daten zu erheben und die Qualität der verwendeten Algorithmen objektiv zu bewerten.

User-Stories

Als lohnenswert werden hierbei Fragestellungen bezeichnet, welche für ein Unternehmen einen Mehrwert darstellen. Konkret werden folgende User-Stories behandelt:

- Wie viele Taxis brauche ich kommenden Samstagmittag am Time-Square, wenn es sonnig wird?
- Wie viel Umsatz werde ich am ersten Oktoberwochenende machen?
- Zu welchem Ort wird eine Person an einem regnerischen Morgen aus Manhatten fahren?
- Am 23.12 um 01:00 endet die Weihnachtsfeier im Trump-Tower. Wieviele Passagiere wird das Taxi haben?

- Gibt es ein Muster, nach welchem mehr Trinkgeld gegeben wird?
- Wie viel Trinkgeld werden 3 Fahrgäste geben, wenn eine relativ kurze Strecke vom JFK-Airport gefahren wird?
- Wie lange wird ein Fahrgast brauchen, wenn er dem Taxi an der Freiheitsstatue sagt *kurz zu warten*?
- Am 21.Juni um 14:30 stehen zwei Personen am Central Park bei Nebel. Werden Sie ein grünes oder ein gelbes Taxi nehmen?

Es ist anzunehmen, das einige Prognosen deutlich bessere Ergebnisse liefern als andere. Dennoch sollen bewusst auch die Grenzen von Machine-Learning gezeigt werden.

Die vorgestellten User Stories werden in dieser Reihenfolge in den folgenden Abschnitten behandelt.

Anforderungen

Um eine Objektive Bewertung vorzunehmen, werden folgende Kriterien an die Durchführung der Experimente gestellt:

- **Harte Kriterien:** Die Tests liefern als Resultat eine Genauigkeit.

  Eine Bewertung dieser Genauigkeit findet lediglich im Fazit statt.
- **Wiederholbarkeit:** Eine Wiederholung der Tests muss dieselben Resultate liefern
- **Nachstellbarkeit:** Mithilfe dieses Experimentes muss der Leser im Stande sein, die gezeigten Ergebnisse selbst nachstellen zu können

4.2 Eigenschaften der Daten

Innerhalb dieses Abschnittes werden zunächst die Daten vorgestellt, die dem Fallbeispiel zugrunde liegen.

Die vorgestellten Daten haben bereits einen ETL-Prozess durchlaufen. Dieser besteht im Wesentlichen darin, die CSV-Dateien dahingehend aufzubereiten, das amerikanische Nummerierungen (z.B. Angabe von Dezimalzahlen mit '.' anstelle von ',') auf europäische Normen gebracht werden. Prinzipiell entfällt dieser Schritt für eine rein amerikanische Umgebung.

4.2.1 Taxifahrten

Zunächst werden die Daten der Taxifahrten erläutert.

Diese stammen von der Stadt New York [Gova] und wurde von der *Taxi and Limousine Commission* (Kurz: TLC) bereitgestellt.

Die TLC stellt einen Dachverband mehrerer Taxiunternehmen dar und veröffentlicht die Daten nur - die Erhebung erfolgt in einzelnen, anonymisierten Kleinunternehmen.

Zusätzlich teilen sich die Fahrten in zwei Kategorien auf: *Green* und *Yellow*. Bei grünen Fahrten handelt es sich um Fahrzeuge mit einer anderen Lizenzierung (vgl. [Giu] Absatz 5ff) und besonderen Auflagen. Im Allgemeinen verhalten sich die Fahrten allerdings gleich, insofern werden lediglich Unterschiede aufgelistet falls diese bestehen.

Attribute und Datentypen

Die folgende Übersicht entspricht der von der NYC bereitgestellten [Govb], die Beschreibung wurde übersetzt und eine Spalte für den Datentyp [1] ergänzt.

[1] Wie sie innerhalb des SQL-Servers bezeichnet werden

Name	Beschreibung	Datentyp
VendorID	Ein Code für das Taxiunternehmen, welches die Daten bereitstellt	smallint
pickup_datetime	Uhrzeit und Datum, wann die Fahrt begann	datetime
dropOff_datetime	Uhrzeit und Datum, wann die Fahrt endete	datetime
Passenger_count	Anzahl der Fahrgäste	smallint
store_and_fwd_flag	Angabe, ob die Fahrt direkt hochgeladen wurde, oder ob die Fahrt zwischengespeichert wurden vor einem Upload	bit
RatecodeID	Ein Code für die Rate, welche für die Taxifahrt bezahlt wurde	smallint
PULocationID	Ein Code für die Zone, in welcher die Fahrt begann	smallint
DOLocationID	Ein Code für die Zone, in welcher die Fahrt endete	smallint
trip_distance	Distanzangabe des Taximeters	real
fare_amount	Der Fahrpreis berechnet aus Zeit und Distanz	
extra	Verschiedene Zuschläge auf den Fahrpreis	real
MTA_tax	Aufschlag, automatisch erhoben bei entsprechender Rate	real
improvement_surcharge	Aufschlag, automatisch erhoben in bestimmten Zonen	real
payment_type	Angabe des Zahlungsmittels als Code	smallint
tip_amount	Höhe des Trinkgeldes	real
tolls_amount	Summierter Betrag von Zuschlägen dieser Fahrt	real
total_amount	Gesamtbetrag der Fahrt **ohne Trinkgeld**	real

Die Daten der Grünen Taxis sind erweitert um einen Code für den *Trip_Type* (Ob eine Fahrt von einem Taxistand begann oder ob die Gäste an der Straße abgeholt

wurden).

Alle Distanz-Angaben entsprechen amerikanischen Meilen (1 mile→1,6 km), alle Währungsangaben Dollar.

Für die Angaben der Codes sind ebenfalls Dictionaries bereitgestellt, diese spielen allerdings für den Machine-Learning-Aspekt dieser Arbeit keine Rolle und sind daher vernachlässigt worden.

Umfang

Aus Ressourcengründen wurde ausschließlich das Jahr 2017 betrachtet.

Es gibt **113 Millionen** Einträge für gelbe Fahrten, welche insgesamt knapp **8,1 GB Speicher** benötigen. Zusätzlich wurden Indizes angelegt mit weiteren 9,4 GB Speicher (Um schnelle Anfragen auf Uhrzeiten und Orte zu ermöglichen).

Es gibt **11,7 Millionen** Einträge für grüne Fahrten, mit insgesamt **900 MB Speicherplatz**. Es wurden zusätzlich Indizes mit 1,1 GB Speicher erstellt.

Zusammen gibt es aus dem Jahr 2017 also fast **125 Millionen Einträge** welche insgesamt 19,5 GB Speicher belegen.

Anomalien

Innerhalb der Daten traten einige Ungewöhnlichkeiten auf - zum Beispiel gibt es Fahrten, die von Ort A nach Ort A gingen und keine Strecke zurückgelegt haben. Bei einer genaueren Untersuchung ergab sich allerdings, dass diese Fahrten meist wenige Minuten dauerten und ebenfalls keinen Passagier hatten.

Es ist anzunehmen, das die Taxis an dieser Stelle auf ihre Passagiere gewartet haben. Aufgrund dieser Erkenntnis wurden alle Anomalien in die Machine-Learning Algorithmen übernommen, um die Daten und somit auch die Ergebnisse nicht zu verfälschen.

Es wurden außerdem weitere Anomalien gefunden, welche kurz genannt werden:

- Fahrten mit Negativkosten
- Fahrten außerhalb von 2017
- Fahrten mit extrem hohen Trinkgeld ($\sim$ 100$) oder extrem hohen kosten ($\sim$ 300$)
- Fahrten, die wenige Sekunden gedauert haben
- Fahrten, welche mehrere Stunden gedauert haben und dabei nur kurze Strecken zurücklegen

4.2.2 Wetteraufzeichnungen

In diesem Unterabschnitt werden die Wetterdaten sowie ihr Umfang vorgestellt.

Die Wetterdaten stammen von der *National Oceanic and atmospheric Administration* [NOA] (Kurz: NOAA), welche verschiedene Klimadaten sammelt. Für dieses Fallbeispiel wurden die Wetterdaten der Wetterstation des JFK-Airports für das Jahr 2017 abgefragt.

Attribute und Datentypen

Im Gegensatz zu den Taxidaten werden in diesem Paragraphen lediglich die verwendeten Attribute vorgestellt. Es werden ebenfalls der Bezeichner, eine kurze Beschreibung und der Datentyp innerhalb des SQL Server vorgestellt.

Name	Beschreibung	Datentyp
Date	Der Tag, an welchem der Datensatz erhoben wurde	date
Hour	Die Stunde, an welcher der Datensatz erhoben wurde	smallint
DryBulbTemp	Die Trockenkugeltemperatur	real
WetBulbTemp	Die Feuchtkugeltemperatur (Messung unter Berücksichtigung von Verdunstungskälte)	real
DewPointTemp	Die Höhe des Taupunktes	real
RelativeHumidity	Die gemessene Luftfeuchtigkeit	real
Visbility	Sichtweite in Meilen	real
WindSpeed	Windgeschwindigkeit	real
WindDirection	Windrichtung in Grad	int
Sunrise	Uhrzeit des Sonnenaufgangs	datetime
Sunset	Uhrzeit des Sonnenuntergangs	datetime

Die Windgeschwindigkeiten sind hierbei in Meilen/Stunde angegeben, die Temperaturen in Grad Celcius auf eine Nachkommastelle gerundet. Die Windrichtung ist als Grad angegeben, wobei 360° Norden und 180° Süden entsprechen.

Es gab keine nennenswerten Anomalien.

Umfang

Es gibt **13351 Datensätze** die 1,4 MB Speicher benötigen. Zusätzlich gibt es 0,6 MB Indizes.

4.2.3 Machine-Learning-Sicht und Rich-Sicht

Die in den vorhergehenden Unterabschnitten vorgestellten Daten sind für die Verwendung als Sicht zusammengefasst, so das am Ende jede Taxifahrt erweitert wird um die Wetterdaten.

Insgesamt wurden vier Sichten erstellt, je zwei für die grünen und gelben Taxis:

Rich-View Enthält alle Daten in lesbarer Form, Locations wurden nach *Borough* und *Zone* aufgeteilt. In gleichem Maße sind die HändlerID's, Zahlungsmittel und Raten als Text aufgelöst.

Diese Sicht wurde erstellt, um Anomalien zu erkennen und den Import der Daten zu überprüfen. Für Machine Learning ist Sie im Allgemeinen Unbrauchbar.

Machine-Learning-View Enthält ebenfalls alle Daten, verwendet allerdings Dictionary-ID's für jeden nicht-numerischen Wert.

Ein Datensatz dieses Views entspricht einem einzelnen Vektor. Dies stellt für viele Bibliotheken von R eine Voraussetzung dar[2].

Ausschnitt-Tabellen Neben den Views werden des Weiteren kleinere Auschnitte entnommen. Dies liegt daran, dass das zufällige Entnehmen einer Stichprobe v.A. von den Gelben Taxidaten mehrere Minuten benötigt.

Die Ausschnitte wurden aus den ML-Sichten erstellt, indem jeder Datensatz einen zufälligen neuen Hashwert bekam nach welchem er sortiert wurde.

4.3 Erstellen eines Neuronalen Netzwerkes mit R

Hier packe ich meine Daten in die vorgestellten NN Algorithmen und Trainiere fleißig.

Quasi Umsetzung der vorgestellten Möglichkeiten

[2] Es findet keine implizite Konvertierung von Strings in ein Dictionary statt

4.4 Prognosen mithilfe des Neuronalen Netzes

4.5 Test des Modells

4.5.1 Kriterien

4.5.2 Testfall

4.5.3 Ergebnisse

5 Fazit

Literaturverzeichnis

[Ber] BERGMEIR, Christoph: *Package 'RSNNS'*

[Dar84] DARLEY, John M.: A Hypothesis-Confirming Bias in Labeling Effects. 44 (1984), S. 20–33

[Giu] GIUFFO, John: *NYC's New Green Taxis: What You Should Know*. https://www.forbes.com/sites/johngiuffo/2013/09/30/nycs-new-green-taxis-what-you-should-know/#5ca3d25732a2. – Erklärung was es mit den Grünen Taxis auf sich hat

[Gova] GOV, NYC: *TLC Trip Record Data*. http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. – Quelle der Taxidaten, angegebenes Date = Letztes Errata

[Govb] GOV, NYC: *TLC Trip Record Data*. http://www.nyc.gov/html/tlc/downloads/pdf/data_dictionary_trip_records_yellow.pdf. – Data-Dictionary der gelben Taxidaten, angegebenes Date = Letztes Errata

[Ins] INSTITUTE, National C.: *NCI Dictionary of Cancer Terms*. https://www.cancer.gov/publications/dictionaries/cancer-terms/def/selection-bias

[Mar] MARTINS, Heidi Steen J.: *RevoScaleR package*

[NOA] NOAA: *Land-Based Station Data*. https://www.ncdc.noaa.gov/data-access/land-based-station-data. – Quelle der Wetterdaten, angegebenes Date = Letztes Update

[Pfe] PFEIFER, Stella: *B wie Bias*. https://blog.eoda.de/2018/05/08/b-wie-bias/#more-4991

[Sel] SELBY, David: *Building a neural Network from scratch in R*. https://selbydavid.com/2018/01/09/neural-network/. – Schritt für Schritt Erklärung von NN's + Codebeispiele in R ohne Package