Hochschule Bochum
Bochum University
of Applied Sciences

Fachbereich Wirtschaft
Prof. Dr. Christian Bockermann

Data Science

Wintersemester 2020/2021

Übungsblatt 4

Aufgabe 1 (Daten Exploration)

Im Verzeichnis `Vorlesung/data/` finden Sie den Datensatz `telco-churn.csv`. Der Datensatz enthält Kunden eines Telekommunikationsunternehmens und deren Vertragseigenschaften (Geschlecht, Telefon: ja/nein, Internet: ja/nein,...).
Die Spalte `Churn` gibt an, ob der Kunde im letzten Monat seinen Vertrag gekündigt hat, oder nicht. Das Thema *Churn Prediction* ist ein typischer Anwendungsfall in vertragsbasierten Geschäftsmodellen.

1. Laden Sie den Datensatz in einen DataFrame `churn`.
   Welche Spalten hat der Datensatz? Welchen Typ haben die Spalten?

2. Wenden Sie den folgenden Befehl auf dem DataFrame an:

   ```
   churn.replace({ "PhoneService": { 'Yes': 1, 'No': 0 } })
   ```

   Wie verändert dies den Datensatz? Was hat das für Vorteile?

3. Wie hoch ist die Churn-Rate? Bei welchem Geschlecht liegt die Churn-Rate höher?

4. Wie hoch ist der Anteil an weiblichen Kunden, die einen DSL-Anschluss besitzen? Welche Anschluss-Arten gibt es noch? Wie ist deren Verteilung?

Aufgabe 2 (Dataset Splitting)

Das Modul `random` bietet verschiedene Möglichkeiten, Zufallszahlen zu erzeugen und mit der Function `random.shuffle(xs)` eine Liste `xs` in eine zufällige Reihenfolge zu bringen. Ein Beispiel dafür wäre:

```
import random
xs = list(range(10))
random.shuffle(xs)
```

1. Lesen Sie mit *Pandas* den *Iris* Datensatz in einen DataFrame `df` ein und erzeugen Sie eine zufällige Liste von Index-Werten aus `df`. Den Datensatz finden Sie ebenfalls im Verzeichnis `Vorlesung/data/`.

2. Schreiben Sie eine Funktion `split_train_test`, die einen DataFrame und einen Parameter `ratio` mit einem Wert zwischen 0.0 und 1.0 übergeben bekommt und eine Aufteilung in zwei DataFrames zurückliefert, deren Größenverhältnis dem Wert von `ratio` enspricht.

   Beachten Sie dazu auch die beiden folgenden Hinweise:

Hinweis 1: Eine Funktion kann mehrere Ergebnisse zurückgeben, wenn Sie diese als Tupel mit `return` zurückgeben, z.B.:

```
def f(x)
  v = x - 1
  n = x + 1
  return v, n

vor, nach = f(5)   # ergibt:  vor=4, nach=6
```

Hinweis 2: Mit `.iloc[liste]` können Sie auf die Zeilen eines DataFrame anhand der Position zugreifen. Schauen Sie Sich dazu ggf. nochmal Folie 34 von Foliensatz 3 an - `.iloc` funktioniert sowohl mit einem Positions-Wert, als auch mit einer Liste von Positions-Werten.

3. Erzeugen Sie mit Ihrer split-Funktion Train/Test DataFrames für die Verhältnisse 60:40, 80:20 und 90:10 und schauen Sie sich z.B. mit

```
X_train, X_test = split_train_test(iris, ratio=0.6)

X_train.plot.box()
Y_test.plot.box()
```

die Ausprägungen der verschiedenen numerischen Attribute an.

Aufgabe 3 * (Modellfehler berechnen)

In der Vorlesung wurde u.a. der Klassifier `Zufall` aus dem `datascience` Modul vorgestellt. Die Klasse `Zufall` sammelt in der Trainingsphase alle Werte der Label-Spalte ein und berechnet bei der Vorhersage jeweils einen zufälligen Wert aus diesen eingesammelten Werten als Vorhersageergebnis.

1. Benutzen Sie das Pandas Modul um den *Iris* Datensatz einzulesen. Erstellen Sie aus dem Datensatz einen DataFrame `X`, der die Merkmalsspalten enthält (alle ausser `species`) und eine Series `y`, die die `species` Spalte enthält.

2. Erzeugen Sie ein neues Modell `m` der Klasse `Zufall` und trainieren Sie es auf `X` und `y`. Nutzen Sie das Modell `m` um auf dem DataFrame `X` die Vorhersagen zu berechnen und speichern Sie diese in der Variable `y_hat`.

3. Definieren Sie eine Funktion `errors`, die zwei Series Objekte als Eingabe bekommt und die Anzahl der Stellen berechnet, an denen die eingegebenen Series Objekte nicht gleich sind.

4. Die Klasse `Zufall` hat zusätzlich die Methode `describe()`, die einen Überblick über die Verteilung der Klassen aus der Trainingsphase enthält. Schauen Sie sich die Ausgabe von `describe()` an.

   Welche durchschnittliche Fehlerrate erwarten Sie, wenn Sie das Modell auf dem Trainingsdatensatz `X, y` auswerten? (Trainingsfehler)

5. Definieren Sie eine Funktion `rel_errors`, die aus zwei Series Objekten den durchschnittlichen Fehler berechnet. Benutzen Sie als Fehlerfunktion dazu die Funktion `errors`, die Sie zuvor definiert haben.