5. Aufgabenblatt
Naïve Bayes Klassifikation
Abgabe: 07.02.2018, 23:59 Uhr

Patrick Schäfer

Berlin, 22. Januar 2018

patrick.schaefer@hu-berlin.de

Vorlesung: https://hu.berlin/vl_dwhdm17
Übung: https://hu.berlin/ue_dwhdm17

Daten (Kreuzfahrt)

- Ein Veranstalter möchte für die Planung zukünftiger Kreuzfahrten vorhersagen, welche seiner Gäste Spaß haben werden
- Gegeben sind die folgenden Daten der Gäste einer Kreuzfahrt:
  - ID, LABEL, PCLASS, NAME, SEX, AGE, FARE, EMBARKED
    - LABEL: Der Gast hatte Spaß (1) oder nicht (0)
    - PLCASS: Ticket-Klasse
    - Name: Name des Gasts
    - Sex: Geschlecht
    - FARE: Ticket-Kosten
    - AGE: Alter des Gasts, falls bekannt
    - EMBARKED: Eingeschifft im Hafen

Test/Training Split

- Die Daten sind aufgeteilt in einen Training- und einen Test-Split:
  - Training: 799 Gäste mit Label
  - Test: 510 (zukünftige) Gäste ohne Label

- Euer Ziel ist es,
  - den Naive Bayes Klassifikator in PL-SQL zu implementieren und auf den Trainingsdaten zu trainieren und
  - Vorhersagen für die Testdaten zu erstellen

Beispieldaten (Auszug)

id	label	pclass	name	sex	age	fare	embarked
1	0	3	Tilman Heuer	male		7,250	S
2	0	3	Galina Weinhold	female	17	7,733	Q
3	1	1	Dietwalt Radermacher	female	30	31,000	C
4	0	3	Hasso Dittmann	male		69,550	S
5	1	1	Mary Kögler	male	56	35,500	C
6	0	3	Steffen Kracht	male		25,467	S
7	0	3	Melita Buß	male	11,5	14,500	S
8	0	3	Felizitas Scheer	male	20	9,225	S
9	1	1	Meta Graßl	female		79,200	C
10	1	2	Auguste Burkhart	female	41	19,500	S
11	1	3	Heidegret Bluhm	female		7,733	Q
12	1	3	Alwine Bohm	female	36	17,400	S
13	0	1	Lucia Siepmann	male	55	59,400	C
14	0	3	Neithard Strasser	male	28,5	16,100	S
15	0	3	Adolph Heide	male		8,050	S
16	0	1	Katherina Leder	male	47	25,588	S
17	0	3	Edelfried Keitel	male		7,750	Q
18	0	3	Christine Krone	male	23	13,900	S
19	0	3	Sigtrud Brendel	male	30	8,050	S
20	1	1	Hagen Harder	male		26,000	S

Import

- Erstellt insgesamt zwei Tabellen:
  - CRUISE_TRAIN: speichert die Trainingsdaten
  - CRUISE_TEST: speichert die Testdaten

- Importiert die Daten (von der Webseite) in die Tabellen

Naïve Bayes Klassifikation

- Implementiert den Naïve-Bayes-Klassifikator als PL-SQL-Funktion (Vorlage s. Webseite) mit folgendem Interface

```
CREATE OR REPLACE FUNCTION predict_naive_bayes_sex(
 m_sex IN cruise_train.sex%TYPE)
RETURN NUMBER AS ...
```

- Eingabe: Das Geschlecht eines Gasts
- Ausgabe: Der Gast wird Spaß haben (1) oder nicht (0)

- Beispiel:
  - Aufruf
    ```
    SELECT predict_naive_bayes_sex('female') FROM DUAL;
    ```
  - Ausgabe
    1

Naïve Bayes Klassifikation

- Implementiert nun den Naïve-Bayes-Klassifikator als PL-SQL-Funktion (Vorlage s. Webseite) auf zwei Spalten, mit dem Interface

```
CREATE OR REPLACE FUNCTION predict_naive_bayes_sex_class(
 m_sex IN cruise_train.sex%TYPE,
 m_class IN cruise_train.pclass%TYPE)
RETURN NUMBER AS ...
```

- Eingabe: Das Geschlecht und die Klasse des Tickets eines Gasts
- Ausgabe: Der Gast wird Spaß haben (1) oder nicht (0)

- Beispiel:
  - Aufruf

```
SELECT predict_naive_bayes_sex_class('female',1)
  FROM DUAL;
```

  - Ausgabe

```
1
```

Naïve Bayes Klassifikation

- Implementiert nun die beiden Methoden (Vorlage s. Webseite), die die Genauigkeit auf den Trainingsdaten berechnen

```
CREATE OR REPLACE FUNCTION eval_naive_bayes_sex
RETURN NUMBER AS ...
```

- Ausgabe: Genauigkeit (Accuracy)
- Beispiel:
  - Aufruf:
    ```
    SELECT eval_naive_bayes_sex FROM dual;
    ```
  - Ausgabe:
    ```
    0.7834...
    ```

Wettbewerb (Genauigkeit)

- Ziel des Wettbewerbs ist es, möglichst gute Vorhersagen auf den Testdaten zu treffen
- Bedingung ist, dass ihr euren Klassifikator als PL/SQL-Funktion implementiert, die die Vorhersagen erstellt

- Ideen:
  - Naive-Bayes auf beliebige Spalten-Kombinationen anwenden
  - Daten vorverarbeiten / Histogramme
  - Wert ist größer als Konstante k (Case-when-Statements)

```
SELECT CASE WHEN pclass<... THEN 1 ELSE (
          CASE WHEN age>... THEN ... END)
               END FROM cruise_test;
```

- Punkte
  1. Gruppe: 5 Punkte
  2. Gruppe: 3 Punkte
  3. Gruppe: 2 Punkte

Wettbewerb (Genauigkeit)

- Die Vorhersagen als Text-Datei an mich schicken
- Jeweils eine Vorhersage pro Zeile (sortiert nach ID des Gasts, wie in der Eingabedatei):

0
1
0
0
1
…

Abgabe

- Deadline ist Mittwoch, 07.02.2018, 23:59 Uhr
- Upload einer Zip-Datei „Gruppenname.zip“ per HU-Box: https://hu.berlin/dwhdm_ue5
- Die PL-SQL-Funktionen müssen in eurer Oracle DB liegen
- Die Abgabe (ZIP-Datei) im Einzelnen:
  - Text-Datei mit PL-SQL-Funktionen und Ergebnis der Ausführung
  - Wettbewerb (2 Dateien):
    - Text-Datei: Vorhersagen auf den Test-Daten, eine Vorhersage pro Zeile
    - Text-Datei: Die PL-SQL-Funktion zur Erstellung der Vorhersagen

Präsentation

- Ihr dürft euch aussuchen, was und wann ihr präsentieren wollt (first-come-first-served).
- Montags (Gruppe 1)
  https://dudle.inf.tu-dresden.de/dwhdm_mo_ue5/
- Mittwochs (Gruppe 2)
  https://dudle.inf.tu-dresden.de/dwhdm_mi_ue5/
- Kurzpräsentation der Lösung am 12.02. (Gruppe 1) bzw. 14.02. (Gruppe 2).

Fragen?