Bonuspunkteaufgabe II

Bitte geben Sie das Übungsblatt bis zum 25.05. (23:59 Uhr) ab (Email an: verena.dorner@kit.edu).

Alle R-Implementierungen bitte als Skriptdateien abgeben, alle verbalen Ausführungen als pdf-Dateien. Wenn Sie mehr als eine Datei abgeben, fassen Sie alle Dateien in **einer** Zip-Datei zusammen.

Die **Benennung der Dateien** bitte folgendermaßen gestalten: Nummer der Bonuspunkteaufgabe *(z.B. B1_Dorner.r oder B1_Dorner.pdf)*

Alle Dateien sollten Name oder Matrikelnummer des Erstellers enthalten, bei R-Skripten als Kommentar.

Wir bauen einen Content-based Recommender

Ergebnis der Übungsaufgabe sei ein (in R programmierter) funktionsfähiger Content-based Recommender, welchem ein Dataframe mit den Items (Filmen inkl. Genrezugehörigketi), ein Dataframe mit den Ratings eines einzelnen Users zu den von ihm angesehenen Items, der ID des betreffenden Users sowie die Anzahl an Filmen, die der Recommender empfehlen soll, übergeben wird. Die Ausgabe des Recommenders sehe folgendermaßen aus:

```
> suggestFilms(titleFilmDF, userDF, userid, no_films)
Diese Filme koennten Ihnen auch gefallen:
 River Wild, The (1994)
 Time to Kill, A (1996)
 Emma (1996)
 Tin Cup (1996)
 Secrets & Lies (1996)
 Marvin's Room (1996)
 Fierce Creatures (1997)
 Absolute Power (1997)
 Donnie Brasco (1997)
 Breakdown (1997)
 Promesse, La (1996)
 Ulee's Gold (1997)
 Face/Off (1997)
 Hoodlum (1997)
 In & Out (1997)
>
```

- Sie können den Recommender **frei programmieren oder sich an den im Folgenden skizzierten Ablauf halten** und die in den Teilaufgaben 1-7 beschriebenen Funktionen umsetzen.
- Bonuspunkte gibt es nur für **nachvollziehbar kommentierten** Code ☺
- Aufgaben 3 bis 5 brauchen als Inputs die Ergebnisse der Funktionen 1 und 2. **Falls Sie Probleme bei der Implementierung von 1 oder 2 haben, generieren Sie sich Bei-**

spiel-Inputs, deren Struktur den Ergebnissen von 1 und 2 entspricht, und testen Sie 3 bis 5 damit.

- Der Content-based Recommender System soll mit **suggestFilms(titleFilmDF, userDF, userid, noFilms)** aufgerufen werden.
- Als Inputdaten für die Funktion suggestFilms() benötigen Sie **udata_exp.csv** (titleFilmDF) und **uitem_exp.csv** (userDF) aus Studip sowie die ID des Users (userid), dem Sie Filme empfehlen wollen, und die Anzahl an Filmen (no_films), die ihm empfohlen werden sollen. userid und no_films können Sie frei wählen.

1 Funktion clusterFilms(titleFilmDF)

Clustern Sie mit Hilfe des kmeans-Algorithmus die Filme auf Basis ihrer Genre-Zugehörigkeit. Als Entscheidungskriterium für die optimale Clusteranzahl sei gegeben, dass ein zusätzlicher Cluster die Heterogenität um weniger als 25% verringert. Der Rückgabewert ***movieCluster*** sei vom Typ „kmeans“.

2 Funktion getUserInfo(userDF, userid)

Finden Sie alle Filme sowie die zugehörigen Ratings, die Ihr User bereits gesehen hat. Der Rückgabewert ***activeUser*** sei ein nach Film-ID aufsteigend geordneter Dataframe, der die Spalten „itemid“ (entspricht Film-ID), „rating“ und „cluster“ besitzt. „Cluster“ sei immer gleich 0.

3 Funktion setUserFilmCluster(movieCluster, activeUser)

Schreiben Sie in den Dataframe activeUser zu jedem Film den Cluster, welchem er zugeordnet ist, in die Spalte „cluster“. Der Rückgabewert ist ***activeUser***.

4 Funktion getMeanClusterRating(movieCluster, activeUser)

Berechnen Sie für jeden Cluster den Mittelwert über alle Ratings der zum jeweiligen Cluster gehörigen Filme. Der Rückgabewert ***like*** sei ein Integer-Vektor, in dem alle Cluster (identifiziert durch Zahl) enthalten sind, deren Rating-Mittelwert größer als 3 ist.

5 Funktion getGoodFilms(like, movieCluster, titleFilmDF)

Falls der User die Filme mehrerer Cluster im Mittel besser als „3“ fand, finden Sie den Cluster, der ihm am besten gefallen hat. Suchen Sie alle Filme dieses Clusters (sowohl diejenigen, die der User noch nicht gesehen hat als auch die, welche er bereits kennt). Falls der User keinen Cluster besser als „3“ fand, wählen Sie zufällig *[Befehl: sample.int(..)]* 100 Filme aus. Der Rückgabewert ***recommend*** sei ein Integer-Vektor, der die gefundenen bzw. zufällig ausgewählten Film-IDs enthält.

6 Funktion getRecommendedFilms(titleFilmDF, userDF, userid)

Verwenden Sie die bisher implementierten Funktionen, um folgende Berechnungen durchzuführen:

- Bilden Sie die Filmcluster und finden Sie die relevanten Informationen zu Ihrem User.
- Ergänzen Sie sie um die Information, zu welchem Cluster jeder Film (den der User bereits gesehen hat) gehört.
- Berechnen Sie das Durchschnittsrating pro Cluster und finden Sie die Filme zu dem Cluster, welcher dem User am besten gefällt.

Suchen Sie aus diesen Filmen diejenigen heraus, die der User noch nicht gesehen hat *[Hinweis: Mengenlehre spart hier Arbeit]*. Der Rückgabewert ***recommend*** enthalte nun für diese Filme neben der Film-ID auch den Filmtitel (movtitle).

7 Funktion suggestFilms(titleFilmDF, userDF, userid, noFilms)

Diese Funktion empfiehlt einem bestimmten User (userid) eine bestimmte Anzahl (noFilms) Filme wie im ***Screenshot auf Seite 1*** *[noFilms=15]* dargestellt.