**Notizen Meeting BDA**

Aufteilen:

1. Data Pre-Processing

Daten so vorbereiten, dass sie vom Model aufgenommen werden können

Generate features using data from IMDb / Rotten Tomatoes (e.g., textual descriptions, reviews, pictures) use APIs or web scraping

1. Model Building
2. Output Generation

Output des Models in gewünschtes Format (Pandas Dataframe) konvertieren.

TODO:

Predict ratings of movies based on “a pre-defined test set of user-item combinations”

* What user-item combinations?
* Minimize root mean squared error (RMSE) on the test set

Daten (siehe auch doc\_train\_data.pdf):

* **actors.csv:**

Notable Actors of all movies cross-linked via movieID

Actor has also ‘actorID’, ‘actorName’ and ‘Ranking’

* **movieID** (int)
* **actorID** (str)
* **actorName** (str)
* **ranking** (int): order in which actors appear on the movie IMDb cast web page
* **countries.csv:**

Links ‘movieID’ and the ‘country’ where the movie was made

* **movieID** (int)
* **country** (str)
* **director.csv:**

Links movieID with ‚directorID‘ and ‘directorName’

* **movieID** (str)
* **directorID** (str)
* **directorName** (str)
* **genre.csv:**

Links ‘genre’ with ‘movieID’

* **movieID** (int)
* **genre** (str)
* **locations.csv:**

Links ‘movieID’ with all movie ‘locations’

* **movieID** (int)
* **location1 – location4** (str): up to four filming locations per movie

TODO:

* + Correct column headings! -> ‘location1/2/3/4’ to (‘country’, ‘state’, ‘city’, ‘street’)
  + Many empty fields
* **movie\_tag.csv:**

Mapping of movies to tags which were assigned by users.

Links ‘movieID’ with ‘tagID’ and ‘tagWeight’

* **movieID** (str)
* **tagID** (str)
* **tagWeight** (int): Weight / strength of tag assignment
* **movies.csv:**

Meta data about the movies

**id** (int)

**title** (str)

**imdbID** (int): IMDB ID of the movie

MDBURL: https://www.imdb.com/title/tt<imdbID>withimdbIDzero-paddedto 7 digits

E.g. ID 25464àURL https://www.imdb.com/title/tt0025464

**spanishTitle** (str)

**imdbPictureURL** (str)

**year** (int)

**rtID** (str): Rotten Tomatoes (RT) ID of the movie

**RTURL:** https://www.rottentomatoes.com/m/<rtID>

**rtPictureURL** (str)

* **ratings.csv**:

Combines ‘userID’, ‘movieID’ and ‘rating’

Ratings given by users to movies:

* **user\_id** (int)
* **movieID** (int)
* **rating** (float): Ratings in steps of 0.5 (worst: 0.5, best: 5.0)
* **tags.csv:**

User generated tags

* **id** (int)
* **value** (str)

**Modelansätze**

Matrixfactorization:

* used by Netflix
* <https://medium.com/analytics-vidhya/matrix-factorization-made-easy-recommender-systems-7e4f50504477>
* <https://medium.com/free-code-camp/svd-vs-matrix-factorization-in-recommender-systems-b1e99bc73599>

Deep Neuronal Network:

CNN:

Restircted Boltzman Machine:

* used by Netflix