Springboard--DSC Program

Capstone Project 2 - Project Proposal

Trivago - Hotel Recommendations

By: Ellen A. Savoye

October 08, 2020

Business problem:
One of the most important parts of traveling is having a place to stay. Ratings, reviews, images, deals, places of interest, and more influence a traveler's hotel selection. With the plethora of hotels to choose from, it can be difficult to ensure you're getting the best price for your amenities and other travel constraints. As a result, many travelers use global hotel search platforms, like *Trivago*, to find and compare hotels. Using a context-aware recommender system, is it possible to provide a list of accommodations that will match the needs of the traveler?

The goal of this capstone project is to develop a session-based, and context-aware recommender system using various input data to provide a list of accommodations that will match the needs of users.

Client:
Trivago is a global hotel search platform focused on reshaping the way travelers search for and compare hotels, while enabling advertisers of hotels to grow their businesses by providing access to a broad audience of travelers via *Trivago's* platform[1].

Data:
The data used for this project is provided by *Trivago*[1]. It consists of 2 files: user session actions (train.csv) and item metadata (item_metadata.csv). The training set contains user actions such as filter usage, search refinements, item interactions, item searches, item click-outs, as well as information about impressed items and prices at the time of a click-out.

The train dataset is comprised of the following features:

- **user_id**: identifier of the user
- **session_id**: identifier of each session
- **timestamp**: UNIX timestamp for the time of the interaction

[1] https://recsys.trivago.cloud/challenge/

- **step**: step in the sequence of actions within the session
- **action_type**: identifier of the action that has been taken by the user.
    - **clickout item**: user makes a click-out on the item and gets forwarded to a partner website. The reference value for this action is the item_id. Other items that were displayed to the user and their associated prices are listed under the ‘impressions’ and ‘prices’ column for this action.
    - **interaction item rating**: user interacts with a rating or review of an item.
        - The reference value for this action is the item id.
    - **interaction item info**: user interacts with item information.
        - The reference value for this action is the item id.
    - **interaction item image**: user interacts with an image of an item.
        - The reference value for this action is the item id.
    - **interaction item deals**: user clicks on the view more deals button.
        - The reference value for this action is the item id.
    - **change of sort order**: user changes the sort order.
        - The reference value for this action is the sort order description.
    - **filter selection**: user selects a filter.
        - The reference value for this action is the filter description.
    - **search for item**: user searches for an accommodation.
        - The reference value for this action is the item id.
    - **search for destination**: user searches for a destination.
        - The reference value for this action is the name of the destination.
    - **search for poi**: user searches for a point of interest (POI).
        - The reference value for this action is the name of the POI.
- **reference**: reference value of the action as described for the different action types
- **platform**: country platform that was used for the search, e.g. trivago.de (DE) or trivago.com (US)
- **city**: name of the current city of the search context
- **device**: device that was used for the search
- **current_filters**: list of pipe-separated filters that were active at the given timestamp
- **impressions**: list of pipe-separated items that were displayed to the user at the time of a click-out (see action_type = clickout_item)
- **prices**: list of pipe-separated prices of the items that were displayed to the user at the time of a click-out (see action_type = clickout_item)

The item ID for each property (listed in the impressions column) links the training dataset to each property’s metadata (item_metadata.csv) which consists of the item ID and the hotel’s amenities and other attributes.

Problem Approach:
The property metadata and training data sets comprise my raw corpus. After initial exploration to determine the best manner in which to combine two parts of the corpus, cleaning, or

[1] https://recsys.trivago.cloud/challenge/

pre-processing, of the two datasets can commence. The pre-processed corpus will then be used to extract features, like amenities at each hotel.

From here, we will create a probability matrix to be the input to our machine learning algorithms: Basic Markov Chain and Latent Matrix Factorization.

The output for both of these classifiers will be a “learned” system that returns a list of hotel recommendations. They will be evaluated using the mean reciprocal rank (MRR): a statistical measure that evaluates any process that produces a list of possible responses.

The “learned” system with the higher MRR will be used to provide a list of hotel recommendations for new and returning users.

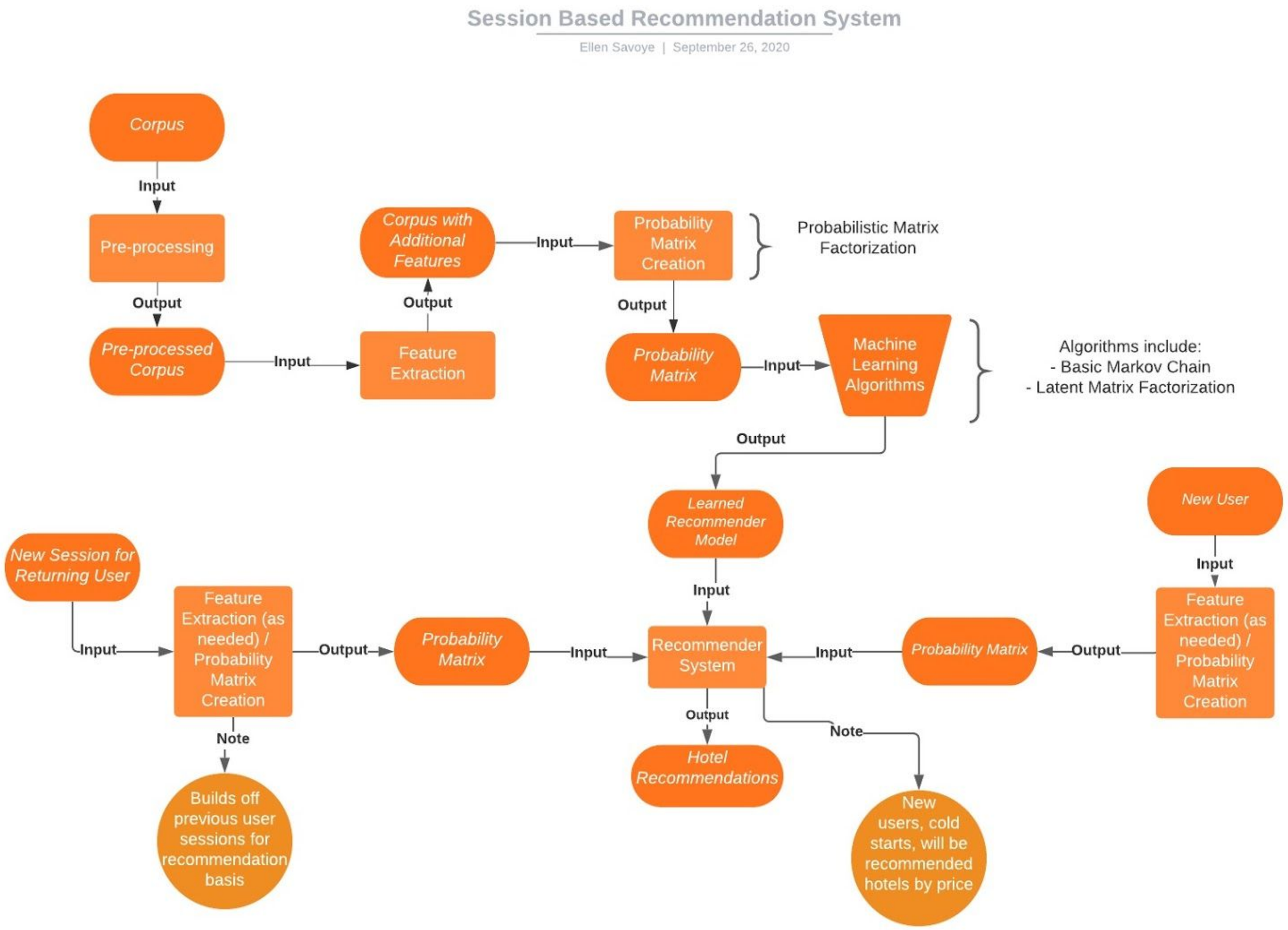
A high-level process flow can be found below. There are two use cases: returning user and new user. The two use cases share the same processes up to the point at which the recommendation is to be computed. More specifically, returning users will receive recommendations built off previous sessions; while new users, given the lack of previous session knowledge, will receive recommendations based on a list of hotels that will be filtered for specific attributes, if applicable, and sorted by price.

Deliverables:

The deliverables will include all code developed with each step contained in it’s own Jupyter Notebook, a written final report, and a written presentation slide deck.

[1] https://recsys.trivago.cloud/challenge/

Process Flow:



Session Based Recommendation System

Ellen Savoye | September 26, 2020

Corpus
Input
Pre-processing
Output
Pre-processed Corpus
Input
Feature Extraction
Output
Corpus with Additional Features
Input
Probability Matrix Creation
Probabilistic Matrix Factorization
Output
Probability Matrix
Input
Machine Learning Algorithms
Algorithms include:
- Basic Markov Chain
- Latent Matrix Factorization
Output
Learned Recommender Model
Input
New Session for Returning User
Input
Feature Extraction (as needed) / Probability Matrix Creation
Output
Probability Matrix
Input
Note
Builds off previous user sessions for recommendation basis
Recommender System
Output
Hotel Recommendations
Note
New users, cold starts, will be recommended hotels by price
New User
Input
Feature Extraction (as needed) / Probability Matrix Creation
Output
Probability Matrix
Input

[1] https://recsys.trivago.cloud/challenge/