Recommender System for Airbnb

Personalisation and Machine Learning

Jerome Goldstein
Student Id: 21036100

Github repo: https://github.com/jeromegold/personlisation

Word Count: 1415 words

Index

Introduction

As the internet has developed and large amounts of data can be transferred at high speed, we can listen to music, watch movies and browse for fashion in real time. With the increased accessibility of the internet, it was quickly recognised that the choices and preferences of internet users could be recorded.

This stored information comprising user details and their preferences allows for bidirectional information transfer. Companies such as Meta, Google and other smaller businesses can collect data on users' profiles that can better inform product development. Additionally, users can be provided with choices that match their prior recorded preferences.

Machine learning techniques have been developed to provide the computational basis for recommender systems.

Aim

In this project the aim is to experiment with the techniques involved in developing a recommender system, specifically content based filtering and collaborative filtering, for the rental site Airbnb.

Recommender System Overview

The idea behind recommender systems is relatively simple. Data is collected on user preference and this data is then used as the basis for providing consumers and users of the internet with choices that align with their prior selections.

Recommender systems not only provide paying consumers with choices that closely match their preferences, but these systems can also be implemented to help sift through thousands and millions of potential options in spheres beyond paid consumerism. For example, there are billions of videos on Youtube and when opening Youtube's homepage there are videos tiled across the screen.

Youtube could randomly select any set of videos for any particular user. However, Youtube uses a recommender system to select specific videos to populate the home page based on viewing history. This recommender system was outlined in 2016 in an article titled, Deep Neural Networks for YouTube Recommendations, by Covington, et al.

The major branches in recommender system implementation are content based filtering, collaborative filtering and deep neural network (Figure 1).

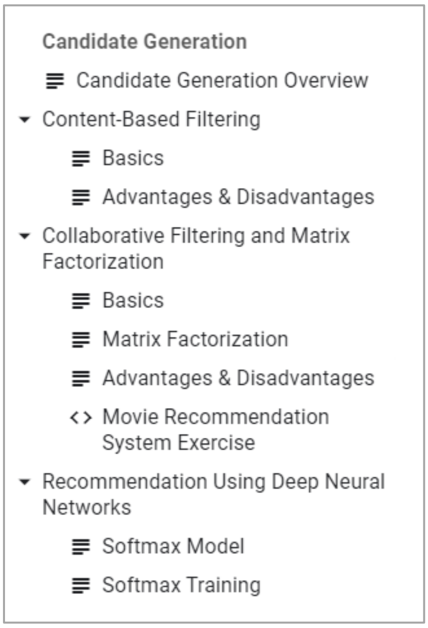

Figure 1. Recommender System types according to Google
https://developers.google.com/machine-learning/recommendation/overview

Recommender System Concepts

One of the first steps in recommending is filtering relevant options from a larger dataset. The two most common filtering options are content based filtering and collaborative filtering.

Content based filtering refers to the recommendation of a similar item based on the user's preference for that item in the past. Collaborative filtering, as the name implies, is a method of recommending items based on similarities between users/collaborators. The idea is that if two individuals have similar features such as age, gender or location then recommendations can be made based on similarity in those features.

The basis for recommenders and filtering is the concept of an embedding space. Mathematically, the embedding space is represented by E -> R^d. The general concept of a mapping is that a set can be mapped to a second set where the information housed within the second set can be used to reach judgments and infer other patterns.

Figure 2 represents an embedding space where the MNIST dataset is mapped as vectors and each integer is represented by a colour demonstrating that that location of the integers in a "space" can be used to try and identify when a similar image is close by.

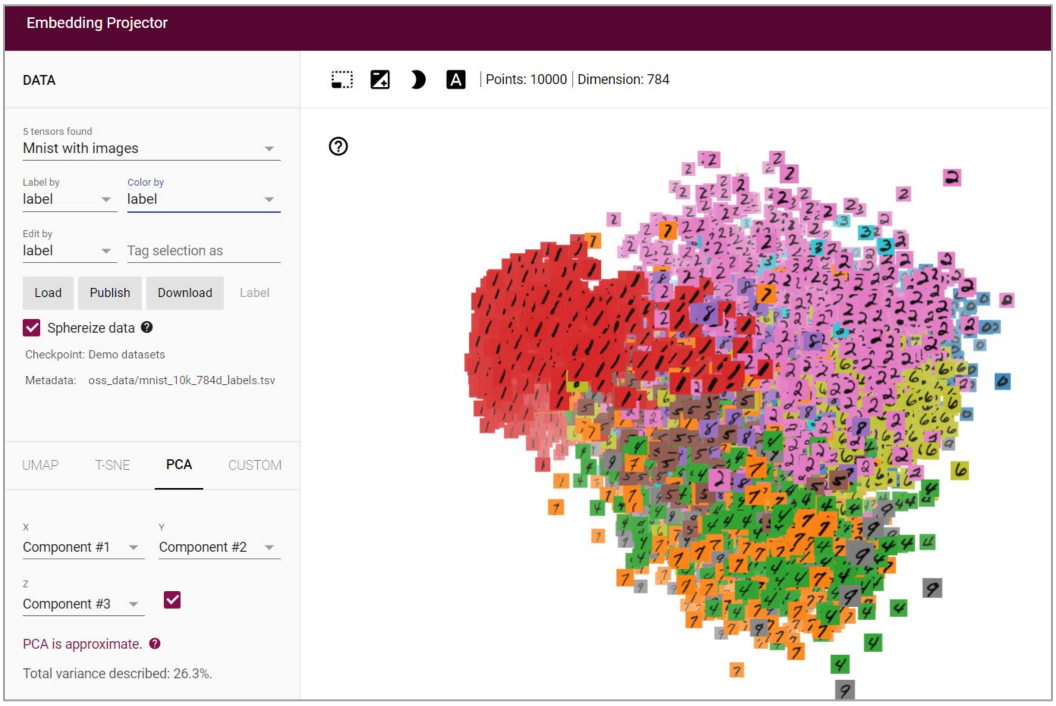

Figure 2. Visual representation of embedding space of the MNIST database with images as embedding vectors using Tensorflow's Embedding Projector https://projector.tensorflow.org/

There are several different ways to measure how close items are to each in other in an embedding space. These include using the cosine angle between two of the embedding vectors, calculating the dot product of two embeddings or determining the Euclidean distance between two embeddings.

The general idea is to determine some form of metric or “distance” between two vectors and thereby quantify similarity. When the embedding vectors are normalised the dot product, cosine and Euclidean distance have the same value.

Collaborative filtering is also associated with the concept of matrix factorisation. The matrix below is an example of a matrix factorisation where rows represent users and columns represent items.

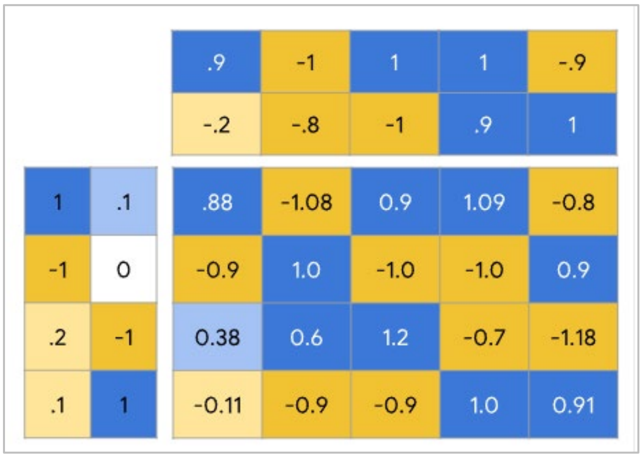

Figure 3 A matrix factorisation that represents the combination of a user matrix and item matrix. https://developers.google.com/machine-learning/recommendation/collaborative/matrix

There is an additional alternative to content and collaborative embeddings. Recommendations can also be arrived at using a deep neural network and models such as softmax. This was not implemented in this project.

Implementation of an Airbnb Recommender

Developing a recommender for Airbnb rentals requires a number of steps. The first was to find an appropriate dataset and to determine what feature and measure would be used to conduct the recommendation.

There are several datasets housing Airbnb information on Kaggle (www.kaggle.com). Some of the possible recommender options included recommending rentals based on Airbnb ratings or filtering out the various rental options based on rental description.

Irrespective of the specific nature of the recommender system, the principles remain the same. For content filtering, a set of embedding vectors needed to be established along with an embedding space. From there a method needed to be chosen (eg. cosine, dot product, etc.) to determine the "distance" between vectors and then a mechanism needs to be defined to return the recommendations to the user. And for collaborative filtering a matrix factorisation model needs to be created.

The dataset used for the recommender was the publicly available Sydney Airbnb Open Data (https://www.kaggle.com/datasets/tylerx/sydney-airbnb-open-data?select=listings_dec18.csv). The list below contains the various csv files in the dataset. The highlighted dataset "listings_dec18.csv" was the one used for this project. This file was > 130 mb and more than 30,000 rows.

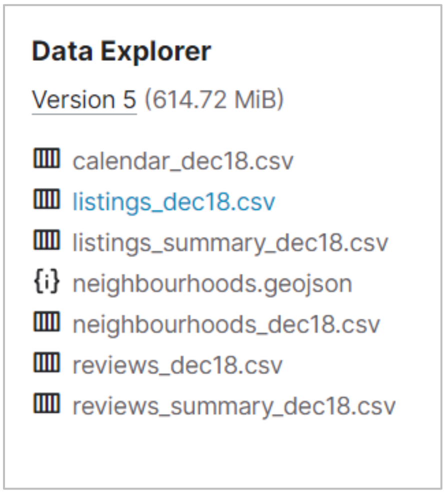

Figure 4 List of csv files from https://www.kaggle.com/datasets/tylerx/sydney-airbnb-open-data

The Jupyter Notebook available on Kaggle https://www.kaggle.com/code/rdaldian/airbnb-content-based-recommendation-system was referred to in the implementation of this recommender.

The first step was to select the required columns out of the 96 columns provided in the dataset. Using the python package "Wordcloud" the individual words were then displayed graphically to provide a relative picture of the most common words used in descriptions.

Figure 5 Graphic display of words in Airbnb descriptions using Wordcloud

Following this, sklearn's TfidfVectorizer was used to vectorise the descriptions. Once the vectors were generated than the cosine was determined for the pairs of vectors and this provided a measure to determine closeness between two vectors.

The linear kernel method is part of the sklearn package. The use of the linear kernel to provide a cosine similarity value may not be entirely accurate. The linear kernel apparently gives a result that it is the inner product (dot product) of the two vectors - but if these vectors aren't normalised then this won't be equivalent to the cosine angle. There are potentially better ways and packages to determine similarity between the vectors.

Once the cosine similarity has been determined the recommendation can be made according to the closest similarity to the selected description.

In the Jupyter Notebook cell in Figure 6 this is one example where the listing 12351 is a key in a dictionary where the value is all the cosine similarities between listing 12351 and every other listing.

Using this information the top cosine similarities for 12351 can be retrieved and then these top listings can be provided to the user.

```
In [23]: results

Out[23]: {12351: [(0.6735807244127452, 29351397),
           (0.5902515801866904, 73639),
           (0.3110520439026729, 16998238),
           (0.304159545580778, 20543279),
           (0.1534075253401848, 8559496),
           (0.15126059888306864, 5127169),
           (0.14672391284613295, 28724099),
           (0.1436210992399347, 28823085),
           (0.13947669344980887, 20172755),
           (0.11804915175853278, 23772409),
           (0.06737323314358608, 22196476),
           (0.057054410975167474, 25523595),
           (0.05390100281510371, 5885653),
           (0.04880283268851622, 5885656),
           (0.04539777572235235, 13034664),
           (0.04304806487209505, 9780677),
           (0.041755039638689935, 29015555),
           (0.04083979933965776, 13071617),
           (0.04076563814997917, 15991038),
```

Figure 6 Cosine similarities between listing 12351 and all the other listings.

Top recommendations provided to the user below for listing 12351 based on the top cosine similarities between 12351 and all the other listings in the table. The relevant score is printed and the recommendations are listed in order.

```
In [49]: print("The description of this listing is: \n \n" +  str(recommend_sub_df.iloc[[0]]["description"][0]))

The description of this listing is:

Come stay with Vinh & Stuart (Awarded as one of Australia's top hosts by Airbnb CEO Brian Chesky & key shareholder Ashton Kutch
er. We're Sydney's #1 reviewed hosts too). Find out why we've been positively reviewed 500+ times. Message us and talk first BE
FORE you make any reservation request - And please read our listing to the end (hint hint). Everything you need to know is ther
e. We're pretty relaxed hosts, and we fully appreciate staying with someone else, in their home home, is not for every-one. Thi
s is not a business, or a hotel. We're casual Airbnb hosts, not hoteliers. If you're just looking for an alternative to an expe
nsive hotel, then we're not for you. Here you'll be treated in the same way we treat family & friends when they stay. So... no
fluffy bathrobes... Please say hello and message us *BEFORE* you make your reservation request... It'll help speed things up, a
nd smooth things out... Please read our listing all the way to the end. It will make getting a confirmed reserv

In [50]: recommend(item_id = 12351, num = 5)

The
Recommending 5 products similar to Sydney City & Harbour at the door
Description: Come stay with Vinh & Stuart (Awarded as one of Australia's top hosts by Airbnb CEO Brian Chesky & key shareholder
Ashton Kutcher. We're Sydney's #1 reviewed hosts t...
---

Recommended: Sunny 2 Bedroom in trendy Surry Hills @ SYDNEY
Description: Come stay with Alana & Anthony. We are Super hosts. As frequent users of Airbnb ourselves, we wish to give back an
d be your host. We have been hosts for about 5 year...
(score:0.6735807244127452)

Recommended: Sydney City Home with Harbour Views
Description: Come stay with Vinh & Stuart (Awarded one of Australia's top hosts by Airbnb CEO Brian Chesky & key shareholder As
hton Kutcher. We're Sydney's #1 reviewed hosts too)...
(score:0.5902515801866904)
```

Figure 7 Recommendations and the associated scores relative to the original selected Airbnb rental

Conclusion

In summary, in this project I have investigated using the cosine similarity and content filtering for the implementation of a recommender system for Airbnb. This recommender system could be improved by using other features such as rental price. Additionally, whether the linear kernel method is a true representation of cosine similarity needs to be investigated further.

References:

1. Seeda, P. A Complete Guide To Recommender Systems — Tutorial with Sklearn, Surprise, Keras, Recommenders. Oct 2021. https://towardsdatascience.com/a-complete-guide-to-recommender-system-tutorial-with-sklearn-surprise-keras-recommender-5e52e8ceace1
2. Google Machine Learning Crash Course. https://developers.google.com/machine-learning/recommendation
3. Paul Covington, Jay Adams, Emre Sargin. Deep Neural Networks for YouTube Recommendations. Sep 2016. Google Mountain View, CA
4. Lecture notes and Github repo, Personlisation and Machine Learning.
5. Sato, R. Private Recommender Systems: How Can Users Build Their Own Fair Recommender Systems without Log Data? Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)