Book Recommendation System

1. Overview

This project is to build a web application with a recommendation system with book reviews and user history behavior from Amazon using Google Cloud Platform (GCP). There are two large datasets in json format, which contains interactions between users and items. Since the size of the data is large, they are processed with Big Query and Compute Engine on GCP. Weighted Alternating Least Square(WALS) algorithm is used for the recommendation system, which uses matrix factorization to predict interests of the users. The model is implemented with Tensorflow and trained on the AI-platform on GCP. Users will be asked to rate some random books on the home page. Once they press the submit button, the model will be retrained with the new data from the user on the AI-platform and the 3 recommended books for the user will be shown on the screen. The application is deployed on App Engine with Identity-Aware Proxy. Only accounts with IAP-Secured Web App User role can have access to the application.

2. Architecture

2.1 GCP Components

The components on GCP used in the application are as below.

- Google Cloud Storage Buckets
- BigQuery
- AI-Platform
- Compute Engine
- App Engine

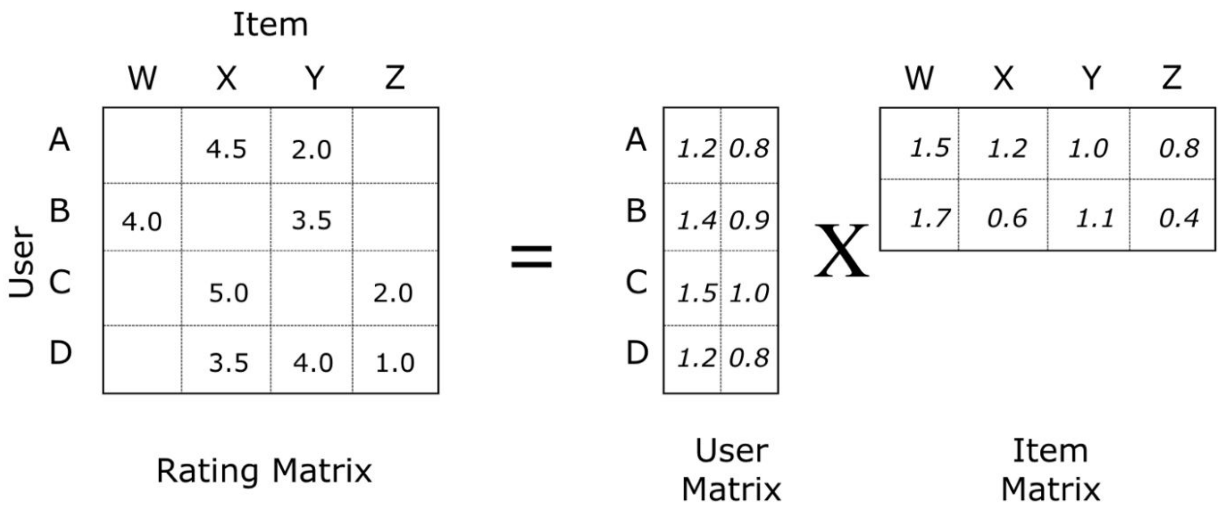
The bucket is used to store the two large datasets, the code for the model and the output after training. BigQuery is a serverless, highly scalable, and cost-effective cloud data warehouse to query data in real time, which transforms the raw data into the features that could be fed into the model. AI-Platform makes it easy to implement the algorithm , package and train the machine learning model. Compute Engine provides the virtual machine, the infrastructure.

2.2 Data Preprocessing

There are two datasets. One contains users' reviews for the books they have read, including reviews, ratings and helpfulness, size of which is 8.81G. The other contains basic information about each book, including title, writer, image url for each book, size of which is 2.37G. Both of them are in JSON format and stored in a google cloud bucket. To process the big data, the data was first imported into the BigQuery as two tables in a dataset. Then I extracted the useful features, userid, bookid and rating using BigQuery client API as a matrix consisting of user IDs and their interactions with the books. Each row corresponds to a unique user, and each column corresponds to a book. Each entry in the matrix captures a user's rating or preference for a single item. There are 603668 users, 367982 books and 8898041 interactions in total. Ratings in the dataset range from 1 to 5. Then, I group the data by users and items, which are stored into two files, "items_for_users" and "users_for_items". After that, data was transformed into Sparse Records and batched. And because when data are batched, they will be reindexed. So I remap the data to find their original index.

2.3 Model Training

In this part, I used the model Weighted Alternating Least Squares (WALS). It uses matrix factorization for sparse data problems.



$$\begin{array}{c|cccc} & W & X & Y & Z \\ \hline A & & 4.5 & 2.0 & \\ B & 4.0 & & 3.5 & \\ C & & 5.0 & & 2.0 \\ D & & 3.5 & 4.0 & 1.0 \end{array} = \begin{array}{c|cc} A & 1.2 & 0.8 \\ B & 1.4 & 0.9 \\ C & 1.5 & 1.0 \\ D & 1.2 & 0.8 \end{array} \times \begin{array}{cccc} W & X & Y & Z \\ \hline 1.5 & 1.2 & 1.0 & 0.8 \\ 1.7 & 0.6 & 1.1 & 0.4 \end{array}$$

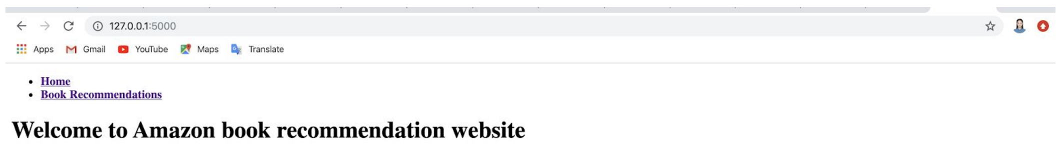
The model learns to factorize rating matrix into user and item representations, which allows the model to predict better personalized movie ratings for users. With matrix factorization, less-known movies can have rich latent representations

as much as popular movies have, which improves recommender’s ability to recommend new movies.
The alternating least squares method of matrix factorization is an iterative method for determining the optimal factors user matrix and item matrix that best approximate the rating matrix. In each iteration, one of the row or column factors is held fixed and the other is computed by minimizing the loss function with respect to the other factor. The weighted version introduces different weights for unobserved entries and the non-zero entries in the matrix. WALS is included in the contrib.factorization package of the TensorFlow code base, and is used to factorize a large matrix of user and item ratings.
The row factors and column factor tensors are created automatically by the wals model class, and are retrieved so they can be evaluated after factoring the matrix.
After implementation of the model, the model was packaged and stored into the google cloud storage bucket. When the user presses the submit button, the model will be trained on the AI platform. Once the training is done, the results are stored into the bucket.

2.4 Prediction

After the training is done, the model would choose 3 books with highest predicted scores for each user and store the results in a csv file in the storage bucket. The file will be imported into Bigquery as a table when a user presses the submit button. Then the books recommended for the user will be exported and shown on the screen.

2.5 User Interface

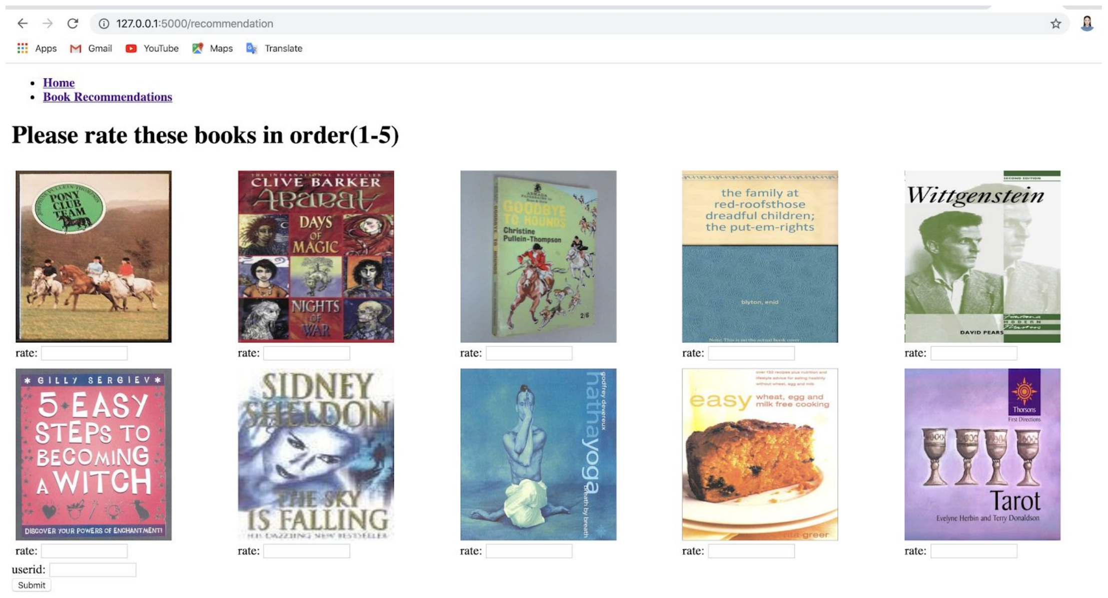

127.0.0.1:5000

Apps Gmail YouTube Maps Translate

- Home
- Book Recommendations

Welcome to Amazon book recommendation website

Fig 2. User Interface Home Page


127.0.0.1:5000/recommendation

Apps Gmail YouTube Maps Translate

- Home
- Book Recommendations

Please rate these books in order(1-5)

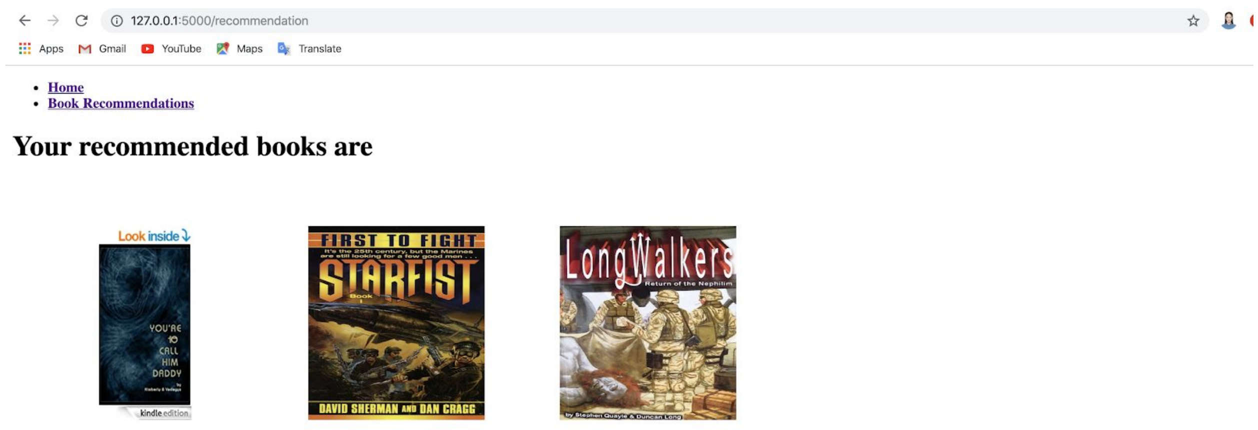
rate: rate: rate: rate: rate:

rate: rate: rate: rate: rate:

userid:

Submit

Fig 3. User Interface Recommendation Page



- Home
- Book Recommendations

Your recommended books are

Fig 4. User Interface Result Page

In the user interface, there are three pages in the user interface.

- Home Page
- Recommendation Page
- Prediction Page

Home page includes a welcome title and the link to the recommendation page, which is shown as Fig. 2. When the user clicks on the link, the website will direct to the recommendation page, which is shown as Fig. 3. On the recommendation page, there are 10 random books. The user will be asked to input user id and the ratings from 1-5 to the 10 books. After clicking on the submit button, the website will submit a job onto the AI-Platform on gcp and retrain the WALS model. After finishing the job, the results will be shown on the screen. The Fig.4 above shows the top 3 books recommended to the user.

2.6 Security

The application is secured with Identity-Aware Proxy. When a user is trying to use the application, they need to login. Only accounts with the IAP-secured Web App User role on the project will be given access.

The link of my code is https://github.com/wyx1102/BookRecommendation.git
The link of the application is
https://bookrecommendation-267223.appspot.com/