CS249 Project Summary
Personalize Expedia Hotel Searches - ICDM 2013

Chenxiao Ma, Xu Wu, Zibing Huang, and Zhaoying Yao

1 Dataset Used

We used the cleaned dataset provided by Kaggle. The total dataset contains two csv files. One is 2.25GB 'train.csv' as training dataset, the other is 1.43GB 'test.csv' as testing dataset.
Originally, the training set has 9999 rows and 54 columns, while the test dataset has 999 rows and 55 rows. The features contained in them can be classified into four groups,

1. Hotel characteristics
2. Location attractiveness of hotels
3. User's aggregate purchase history
4. Competitive OTA information

The user response is provided as a click on a hotel or/and a purchase of a hotel room.

2 Running Environment

Most of our code is written in *Python*. We run our code in **Amazon Web Service**, with a m3.2xlarge instance on Ubuntu 14.0, having 8vCPUs and 30GiB of memory.
The github address for the project is as follows,

`https://github.com/shawnhero/cs249_Parker_Proj1`

To run the code, follow the steps below,

1. Download the zip file or clone the git project to a local repository.
2. In the root directory the file is located, add three new folders names *data*, *saved_model* and *results* respectively. The relative file path is managed in *./pythoncode/SETTINGS.json*. One can also modify this file to change the file paths.
3. Download the dataset file from Kaggle.

   `https://www.kaggle.com/c/expedia-personalized-sort/data`

4. Run the following code in *./pythoncode*. Note here the $sample_size$ is a integer indicating the sample size down sampled. In our project we choose this to be $1,000,000$-$3,000,000$.

   `python train2.py [sample_size]`

5. Now the model should have been created and saved in the path

```
../../saved_model/book_model.pickle
../../saved_model/click_model.pickle
```

6. Predict the test dataset using the following code.

   ```
   python predict.py
   ```

7. After this the result should be automatically saved in the path

   ```
   ../../results/submission.csv
   ```

3 Objective Description

Our objective is to based on the training data set, train a model and predict the results of the test dataset. We don't directly predict the bool value of the $click_bool$ or $booking_bool$. Instead, we generate a list of sorted hotels that the user is mostly likely to hit given a certain search query.
The effectiveness of our model is evaluated in nDCG(normalized Discounted Cumulative Gain). It is further described in the notebook.

4 Results

Now we have ranked around 30th in kaggle's leaderboard. Considering the time budget and the difficulties brought by the size of the dataset, it is well satisfying for us.

5 Lessons and Insights

1. A big challenge in this project we face is that the the GB level size of the dataset is too large for an ordinary PC or MAC to process or it takes too long. Therefore, in the early stage of exploratory analysis, we take partial dataset as the object, and obtain valuable insights about the whole dataset.

2. The second challenge is the code efficiency. Initially all our code is in R. It took us 30 minutes to load the train.csv to the memory using our Mac. Even in the AWS it would take 4 minutes. Later on we rewrote all the code in Python and we were surprised to find a leap of efficiency. Reading table only takes 1minute. Also training the model takes no more than 3 hours, which is good compared to other teams' results mentioned in Kaggle's forum.

3. Feature engineering plays a very important role here. The 54 raw features of the training dataset are not equally relevant to the customer?s click and booking behavior, thus an investigation into which features is more significant than others will benefit the later model training process.

4. Later we utilize the computing capacity of Amazon Web Service EC2 to work on the whole training dataset. The results of feature analysis matches what we have found in partial dataset. As for finding a suitable model, some literature indicates that for the recommendation system, there are several ways to perform this learn-to-rank task.

5. Predictive models including linear ones such as linear regression and nonlinear ones includes the Random Forest, Gradient Boosting Machine, Extreme Randomized Trees.

6 Team Division

Xu Wu In charge of the feature engineering part.

Zibing Huang In charge of the prediction model part.

Chenxiao Ma In charge of the data visualization and model testing part.

Zhaoying Yao In charge of data preprocessing part and the project summary.