Student Id : 012419407
Student Name : Akhil Raveendran

HOTEL RECOMMENDATION SYSTEM

Objective

All online travel agencies are scrambling to meet the AI driven personalization standard that is the upcoming trend in the marketing world. In addition, the world of online travel has become a highly competitive space where brands try to capture our attention (and wallet) with recommending, comparing, matching and sharing. I am trying to build a project which tries to do the same. Which hotel is children friendly? Which hotel will my family like? Will this hotel fit my pocket? Can I find a budget hotel of my choice? Finding a perfect hotel is tough. There is a need for a tool which can help in making such decisions. This project aims to build a recommender system to predict which hotel a browsing user will click on next.

Technical Specifications

Softwares used: Jupyter Notebook(Python 2.6.6),Spyder
Frameworks used : Spark1.6.0, numpy, scipy, scikit

Methodology:

The goal is to build two types of recommender systems, **Content-Based System and Collaborative-Based System.**

a. **Content-Based System:** Content-based systems examine properties of the items recommended. For instance, if a Netflix user has watched many cowboy movies, then recommend a movie classified in the database as having the "cowboy".

b. **Collaborative-Based Systems:** Collaborative- Based Systems recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users.

Data Understanding

The initial data is collected and is verified for its quality. It is observed that our data constitutes combination of user features and hotel features, with user data portraying user demography and search preference. The hotel features mainly constitute its geological location along with its proximity from the user. The description of the data is provided as follows:

File Name	File Size	File Description

Destination.csv	16.18 mb	hotel search latent attributes
Test.csv	65.92 mb	the test set
Train.csv	511.16 mb	the training set

Training data includes all the users in the logs, including both click events and booking events. **Test data** only includes booking events.

Destinations.csv data consists of features extracted from hotel reviews text.

Data Processing

I Implemented various data cleaning, data reduction and feature selection techniques and their combinations to prepare our data. replacements such as median, minimum, maximum, zero, none, most frequent value of the attribute. The replacement option giving the highest accuracy is chosen for further data reduction. I chose zero or none to replace missing values in our data.

1. **High Correlation Filter**: Data columns with very similar trends are also likely to carry very similar information. In this case, only one of them will suffice to feed our machine learning model. Pairs of columns with correlation coefficient higher than a threshold are reduced to only one.

2. **Low Variance Filter**: Data columns with little changes in the data carry little information. Thus all data columns with variance lower than a given threshold are removed.

3. **Missing Values Ratio**: Data columns with too many missing values are unlikely to carry much useful information. Thus data columns with number of missing values greater than a given threshold can be removed. The higher the threshold, the more aggressive the reduction. Here is a screenshot in which the highlighted

column in blue shows the number of missing values some of the features have in our data.

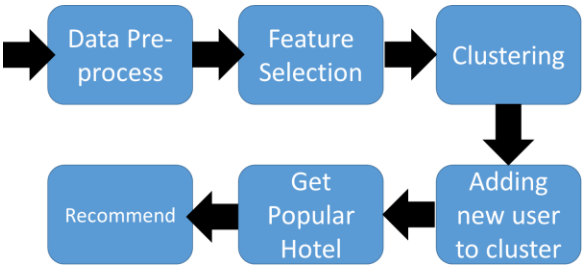
4. **Replace Missing Values:** Data columns with missing values are replaced by specified

5.**Principal Component Analysis**: Principal Component Analysis (PCA) is a statistical procedure that orthogonally transforms the original *n* coordinates of a data set into a new set of *n* coordinates called principal components. As a result of the transformation, the first principal component has the largest possible variance; each succeeding component has the highest possible variance under the constraint that it is orthogonal to the preceding components. Keeping only the first $m < n$ components reduces the data dimensionality while retaining most of the data information, i.e. the variation in the data.

Algorithms and Result

a. **Content-Based System Modelling:** Used Apache Spark framework to cluster 'Train Data' based on user feature. Initially I used K-means for clustering. K-Means clustering intends to partition *n* objects into *k* clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly *k* different clusters of greatest possible distinctions. This process was achieved by converting the categorical fields in one hot vector format. However, 'memory exception' error was obtained due to the large size of the data. To tackle this issue k-modes algorithm was used for clustering training data which converted our data into vector form. K-modes algorithm is basically used for clustering categorical variables. It defines clusters based on the number of matching categories between data points.This trained model was used to map new users from Test data. Post clustering, I calculated popular 'Hotel Clusters' for each 'User Cluster'. I then recommended 'Hotel Clusters' for the Test User based on popular hotels of the Trained cluster.

The Content Based System thus created provided 45.6% accuracy in recommendation



b. **Collaborative-Based System Modelling:** I first identified that combination of 'user_location_city' and 'orig_destination_distance' accurately defines specific hotel. Based on this finding Ibuilt our recommender model using three cases.

- First I looked for leakage data - If there existed same combination 'orig_destination_distance' in 'Test Data' as in 'Train Data' I output same 'hotel_cluster' variable associated with 'Train Data'. Further, I ranked our data based on time of searches, to rank same 'hotel_cluster' having same features.

- I then calculated the most popular 'hotel_cluster' having same 'srch_destination_id','hotel_country', 'hotel_market' and 'book_year' to recommend popular hotels in the destination with high accuracy.The popularity was calculated by assigning weights to number of clicks and booking ratio. To assign weight to booking, I first calculated that for every 6 clicks there is one successful booking on average. Using this

finding I calculated that booking constitutes 5.67 clicks.

Finally, I found the most popular 'hotel cluster' based on 'hotel country' to determine popular hotels in the country to recommend best hotels overall.

The Collaborative Based System thus created provided 49.2% accuracy in recommendation

Conclusion

For Data preparation I tried different preprocessing methods on our data and choose the best performing methods. Then, I implemented different classification models namely Boosted Decision Tree, Neural Networks, Logistic Regression and SVM Model. Then I built two recommender systems(collaborative and content based) using languages like python and spark framework. Lastly, I compared the accuracies of the models and analyzed the result. This project gave us a better understanding of Data Science concepts particularly the Crisp-DM model, Data Preprocessing methods, Data Classification and validation.

References

[1]https://en.wikipedia.org/wiki/Recommender_system
[3]http://recommender-systems.org/collaborative-filtering/