Expedia Hotel Recommendations
Project Description

Business Problem

Planning your dream vacation, or even a weekend escape, can be an overwhelming affair. With hundreds, even thousands, of hotels to choose from at every destination, it's difficult to know which will suit your personal preferences. Should you go with an old standby with those pillow mints you like, or risk a new hotel with a trendy pool bar?

Expedia wants to take the proverbial rabbit hole out of hotel search by providing personalized hotel recommendations to their users. This is no small task for a site with hundreds of millions of visitors every month!
Currently, Expedia uses search parameters to adjust their hotel recommendations, but there aren't enough customer specific data to personalize them for each user.

Data Description

Expedia has provided us logs of customer behavior. These include what customers searched for, how they interacted with search results (click/book), whether or not the search result was a travel package.

Expedia is interested in predicting which hotel group a user is going to book. Expedia has in-house algorithms to form hotel clusters, where similar hotels for a search (based on historical price, customer star ratings, geographical locations relative to city center, etc) are grouped together. These hotel clusters serve as good identifiers to which types of hotels people are going to book, while avoiding outliers such as new hotels that don't have historical data.

Our goal is to predict the booking outcome (hotel cluster) for a user event, based on their search and other attributes associated with that user event

The following are the different fields of the dataset

1. date_time - Timestamp
2. site_name - ID of the Expedia point of sale (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)
3. posa_continent - ID of continent associated with site_name
4. user_location_country - The ID of the country the customer is located
5. user_location_region - The ID of the region the customer is located
6. user_location_city - The ID of the city the customer is located

7. orig_destination_distance - Physical distance between a hotel and a customer at the time of search. A null means the distance could not be calculated
8. user_id - ID of user
9. is_mobile - 1 when a user connected from a mobile device, 0 otherwise
10. is_package - 1 if the click/booking was generated as a part of a package (i.e. combined with a flight), 0 otherwise
11. channel - ID of a marketing channel
12. srch_ci - Check in date
13. srch_co - Checkout date
14. srch_adults_cnt - The number of adults specified in the hotel room
15. srch_children_cnt - The number of (extra occupancy) children specified in the hotel room
16. srch_rm_cnt - The number of hotel rooms specified in the search
17. srch_destination_id - ID of the destination where the hotel search was performed
18. srch_destination_type_id - Type of destination
19. hotel_continent - Hotel continent
20. hotel_country - Hotel country
21. hotel_market - Hotel market
22. is_booking - 1 if a booking, 0 if a click
23. cnt - Number of similar events in the context of the same user session
24. hotel_cluster - ID of a hotel cluster

Src : https://www.kaggle.com/c/expedia-hotel-recommendations/data

Aim

In this project, Expedia is challenging us to contextualize customer data and predict the likelihood a user will stay at 100 different hotel groups.

Approach

1. **Exploratory Data Analysis**
   Exploratory data analysis is the process of analysing the dataset to understand its characteristics. In this step, we will figure out the following.
   a. days stayed by customer
   b. Are there any NAN values?
   c. Find outliers
   d. Correlations
   e. Check for duplicate values
2. **Imputation**
   Imputation is the process of filling the missing values in appropriate ways so that we don't lose much data. The detected missing values are filled appropriately
3. **Further EDA and Visualizations to answer few questions**
   Further Exploratory data analysis includes raising a few questions on the dataset and finding out answers for the same using various visualization and data manipulation techniques. This helps us to understand more about the dataset
4. **Baseline accuracy calculation**

Baseline accuracy score provides the required point of comparison when evaluating all other machine learning algorithms.

5. **Model Building and Evaluation**
   Since this is a classification problem various classification algorithms are tried and the one that is best suited for our data is considered. The models that we build for our dataset are
   a. Random Forest classification
   b. Guassian Naive Bayes classification
   c. Logistic Regression
   d. KNN classification
   e. XGBoost classification
   f. Decision Tree classification
6. **Solutionization**
   In this step, the results of the best suited model is mended a little further to produce better results.

Modularized code

The ipython notebook is modularized into different functions so that the user can use those functions instantly whenever needed. The modularized code folder is structured in the following way.

```
input
  |__train.zip
  |__train.csv
  |__test.csv

src
  |__engine.py
  |__ML_pipeline
            |__Utils.py
            |__Date_time_conversion.py
            |__Impute.py
            |__Baseline_accuracy.py
            |__Train_model.py
            |__Evaluate_results.py
            |__Hyperparameter_tuning.py
            |__Solutionising.py

lib
  |__Expedia Hotel Recommendations.ipynb

output
```

Once you unzip the modular_code.zip file you can find the following folders within it.

1. input
2. src
3. output
4. lib

1. The input folder contains all the data that we have for analysis. In our case, it will contain a three csv files which are
    a. train.zip (zip file which contains train.csv)
    b. test.csv
    c. train.csv
2. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further contains the following.
    a. ML_pipeline
    b. engine.py

The ML_pipeline is a folder that contains all the functions put into different python files which are appropriately named. These python functions are then called inside the engine.py file

3. The output folder contains all the models that we trained for this data saved as .pkl files. These models can be easily loaded and used for future use and the user need not have to train all the models from the beginning.
4. The lib folder is a reference folder. It contains the original ipython notebook that we saw in the videos.

Project Takeaways

1. Approach to the problem statement
2. Data Exploration
3. Handling missing values
4. Creating heat maps
5. Data exploration Visualisations
6. Feature engineering
7. Visualising the engineered features
8. Data Cleaning
9. Baseline accuracy
10. Implementation using random forests
11. Implementation using Naive Bayes
12. Implementation using Logistic regression
13. Implementation using KNN
14. Hyperparameter tuning
15. Comparison of algorithms
16. How to approach & solutionize a problem statement