Hotel Recommendation System with Machine Learning using NLP

We all plan trips and the first thing to do when planning a trip is finding a hotel. There are so many websites recommending the best hotel for our trip. In this project, I'm going to walk you through how to build a hotel recommendation system with Machine Learning with Python.

A hotel recommendation system aims to predict which hotel a user is most likely to choose from among all hotels. So to build this type of system which will help the user to book the best hotel out of all the other hotels. We can do this using customer reviews.

For example, suppose you want to go on a business trip, so the hotel recommendation system should show you the hotels that other customers have rated best for business travel. It is therefore also our approach to build a recommendation system based on customer reviews and ratings.

In the section below, I will take you through a project on Hotel Recommendation System with Machine Learning using Python programming language.

The dataset that I am using here is downloaded from Kaggle. As we are going to build a recommendation system according to the user ratings so here I will be using Natural Language Processing. Now let's import the necessary Python libraries and the dataset to get started with the task of creating a hotel recommendation system:

```
## Installing the NLTK ( neural language tool kit )
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\lenovo\anaconda3\lib\
site-packages (3.7)
Requirement already satisfied: click in c:\users\lenovo\anaconda3\lib\
site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in c:\users\lenovo\anaconda3\
lib\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\lenovo\
anaconda3\lib\site-packages (from nltk) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\lenovo\anaconda3\lib\
site-packages (from nltk) (4.64.1)
Requirement already satisfied: colorama in c:\users\lenovo\anaconda3\
lib\site-packages (from click->nltk) (0.4.5)
```

Importing Required Libraries

```
import nltk
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')
import numpy as np
```

```
import pandas as pd
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem.wordnet import WordNetLemmatizer
from ast import literal_eval
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\lenovo\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\lenovo\AppData\Roaming\nltk_data...
```

Importing the DataSet

```
data = pd.read_csv(r"C:\Users\lenovo\Downloads\Hotel_Reviews.csv")
data.head()
```

```
                                        Hotel_Address  \
0   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...
1   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...
2   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...
3   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...
4   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...

   Additional_Number_of_Scoring Review_Date  Average_Score
Hotel_Name  \
0                           194    8/3/2017            7.7  Hotel
Arena
1                           194    8/3/2017            7.7  Hotel
Arena
2                           194   7/31/2017            7.7  Hotel
Arena
3                           194   7/31/2017            7.7  Hotel
Arena
4                           194   7/24/2017            7.7  Hotel
Arena

  Reviewer_Nationality
Negative_Review  \
0               Russia    I am so angry that i made this post
available...
1              Ireland                                         No
Negative
2           Australia    Rooms are nice but for elderly a bit
difficul...
3      United Kingdom    My room was dirty and I was afraid to walk
ba...
```

```
4         New Zealand    You When I booked with your company on line
y...

   Review_Total_Negative_Word_Counts  Total_Number_of_Reviews  \
0                                397                     1403
1                                  0                     1403
2                                 42                     1403
3                                210                     1403
4                                140                     1403

                                      Positive_Review  \
0   Only the park outside of the hotel was beauti...
1   No real complaints the hotel was great great ...
2   Location was good and staff were ok It is cut...
3   Great location in nice surroundings the bar a...
4    Amazing location and building Romantic setting

   Review_Total_Positive_Word_Counts  \
0                                 11
1                                105
2                                 21
3                                 26
4                                  8

   Total_Number_of_Reviews_Reviewer_Has_Given  Reviewer_Score  \
0                                           7             2.9
1                                           7             7.5
2                                           9             7.1
3                                           1             3.8
4                                           3             6.7

                                                Tags days_since_review
\
0  [' Leisure trip ', ' Couple ', ' Duplex Double...              0 days

1  [' Leisure trip ', ' Couple ', ' Duplex Double...              0 days

2  [' Leisure trip ', ' Family with young childre...              3 days

3  [' Leisure trip ', ' Solo traveler ', ' Duplex...              3 days

4  [' Leisure trip ', ' Couple ', ' Suite ', ' St...             10 days


         lat       lng
0  52.360576  4.915968
1  52.360576  4.915968
2  52.360576  4.915968
3  52.360576  4.915968
4  52.360576  4.915968
```

This dataset contains hotel data from 6 countries, namely:

- Netherlands
- United Kingdom
- France
- Spain
- Italy
- Austria

So for simplicity, I will change the name from "United Kingdom" to "UK. I can also see that there is no column as "Country" to specify the destination of the hotel but in the "Hotel_Address" column the last word mentioned is the name of the country. So I will extract the names of the countries from that column and store the name in a new column:

```
# Replacing "United Kingdom with "UK"
data.Hotel_Address = data.Hotel_Address.str.replace("United Kingdom",
"UK")

# Now I will split the address and pick the last word in the address
to identify the country
data["countries"] = data.Hotel_Address.apply(lambda x: x.split(' ')[-
1])
print(data.countries.unique())
```

```
['Netherlands' 'UK' 'France' 'Spain' 'Italy' 'Austria']
```

Now I will drop the unnecessary columns that we don't need for the task of creating a hotel recommendation system:

```
## Checking for the columns names in dataSet
data.columns.values
```

```
array(['Hotel_Address', 'Additional_Number_of_Scoring', 'Review_Date',
       'Average_Score', 'Hotel_Name', 'Reviewer_Nationality',
       'Negative_Review', 'Review_Total_Negative_Word_Counts',
       'Total_Number_of_Reviews', 'Positive_Review',
       'Review_Total_Positive_Word_Counts',
       'Total_Number_of_Reviews_Reviewer_Has_Given', 'Reviewer_Score',
       'Tags', 'days_since_review', 'lat', 'lng', 'countries'],
      dtype=object)
```

```
## Dropping the unwanted columns
data.drop(['Additional_Number_of_Scoring',
       'Review_Date','Reviewer_Nationality',
       'Negative_Review', 'Review_Total_Negative_Word_Counts',
       'Total_Number_of_Reviews', 'Positive_Review',
       'Review_Total_Positive_Word_Counts',
       'Total_Number_of_Reviews_Reviewer_Has_Given', 'Reviewer_Score',
       'days_since_review', 'lat', 'lng'],1,inplace=True)
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_22428\2106392702.py:2:
FutureWarning: In a future version of pandas all arguments of
DataFrame.drop except for the argument 'labels' will be keyword-only.
  data.drop(['Additional_Number_of_Scoring',
```

Now I will create a function to convert the strings of list into a normal list and then apply it to the “Tags” column in the dataset:

```
def impute(column):
    column = column[0]
    if (type(column) != list):
        return "".join(literal_eval(column))
    else:
        return column

data["Tags"] = data[["Tags"]].apply(impute, axis=1)
data.head()
```

```
                                        Hotel_Address  Average_Score  \
0   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...            7.7
1   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...            7.7
2   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...            7.7
3   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...            7.7
4   s Gravesandestraat 55 Oost 1092 AA Amsterdam ...            7.7

    Hotel_Name                                                 Tags
countries
0  Hotel Arena   Leisure trip  Couple  Duplex Double Room  Sta...
Netherlands
1  Hotel Arena   Leisure trip  Couple  Duplex Double Room  Sta...
Netherlands
2  Hotel Arena   Leisure trip  Family with young children  Dup...
Netherlands
3  Hotel Arena   Leisure trip  Solo traveler  Duplex Double Ro...
Netherlands
4  Hotel Arena   Leisure trip  Couple  Suite  Stayed 2 nights ...
Netherlands
```

Now I will lowercase the “Tags” and “Countries” column for simplicity:

```
## LowerCasing the "Tags" & "Countries" Column
data['countries'] = data['countries'].str.lower()
data['Tags'] = data['Tags'].str.lower()
```

Now let’s define a function to recommend the names of hotels according to the location and the description provided by the user. Here our aim is not just to recommend the name of the hotel but also rank it according to the user ratings:

```
def recommend_hotel(location, description):
    description = description.lower()
    word_tokenize(description)
    stop_words = stopwords.words('english')
    lemm = WordNetLemmatizer()
    filtered  = {word for word in description if not word in
stop_words}
    filtered_set = set()
    for fs in filtered:
        filtered_set.add(lemm.lemmatize(fs))

    country = data[data['countries']==location.lower()]
    country = country.set_index(np.arange(country.shape[0]))
    list1 = []; list2 = []; cos = [];
    for i in range(country.shape[0]):
        temp_token = word_tokenize(country["Tags"][i])
        temp_set = [word for word in temp_token if not word in
stop_words]
        temp2_set = set()
        for s in temp_set:
            temp2_set.add(lemm.lemmatize(s))
        vector = temp2_set.intersection(filtered_set)
        cos.append(len(vector))
    country['similarity']=cos
    country = country.sort_values(by='similarity', ascending=False)
    country.drop_duplicates(subset='Hotel_Name', keep='first',
inplace=True)
    country.sort_values('Average_Score', ascending=False,
inplace=True)
    country.reset_index(inplace=True)
    return country[["Hotel_Name", "Average_Score",
"Hotel_Address"]].head()
```

NOW Its time to check How It Works 😀

Now let's test this function by selection any country out of the 6 countries mentioned in the dataset and describing the purpose of our trip and see how it works:

```
recommend_hotel('Italy', 'I am going for a business trip')
```

```
                                        Hotel_Name  Average_Score  \
0  Excelsior Hotel Gallia Luxury Collection Hotel            9.4
1           Palazzo Parigi Hotel Grand Spa Milano            9.3
2                         Hotel Spadari Al Duomo            9.3
3                              Room Mate Giulia             9.3
4                             UNA Maison Milano             9.3

                                        Hotel_Address
0  Piazza Duca D Aosta 9 Central Station 20124 Mi...
```

```
1  Corso Di Porta Nuova 1 Milan City Center 20121...
2  Via Spadari 11 Milan City Center 20123 Milan I...
3  Silvio Pellico 4 Milan City Center 20121 Milan...
4  Via Mazzini 4 Milan City Center 20123 Milan Italy
```

```
recommend_hotel('UK','I am going on a honeymoon, I need a honeymoon
suite room for 3 nights')
```

```
                                    Hotel_Name  Average_Score  \
0                              Haymarket Hotel            9.6
1                                           41            9.6
2  Taj 51 Buckingham Gate Suites and Residences            9.5
3                         Charlotte Street Hotel            9.5
4                                 Ham Yard Hotel            9.5

                                      Hotel_Address
0  1 Suffolk Place Westminster Borough London SW1...
1  41 Buckingham Palace Road Westminster Borough ...
2  Buckingham Gate Westminster Borough London SW1...
3  15 17 Charlotte Street Hotel Westminster Borou...
4  One Ham Yard Westminster Borough London W1D 7D...
```

So we can see interesting results by the recommendation system.

```
                    - - - - - - - - X X X X X X X X - - - - - - - - -
```