Hotel Recommendation System with Machine Learning using NLP

Let’s import the necessary Python libraries and the dataset to get started with the task of creating a hotel recommendation system:

In [2]:
```
import nltk
nltk.download('wordnet')
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

data = pd.read_csv("E:\Hotel Recommendation System with NLP\Hotel_Reviews.csv")
data.head()
```

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

Out[2]:

	Hotel_Address	Additional_Number_of_Scoring	Review_Date	Average_Score	Hotel_Name	Reviewer_Nationality	Negative_Review	Review_Total
0	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7.7	Hotel Arena	Russia	I am so angry that i made this post available...	
1	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	8/3/2017	7.7	Hotel Arena	Ireland	No Negative	
2	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7.7	Hotel Arena	Australia	Rooms are nice but for elderly a bit difficul...	
3	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/31/2017	7.7	Hotel Arena	United Kingdom	My room was dirty and I was afraid to walk ba...	
4	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	194	7/24/2017	7.7	Hotel Arena	New Zealand	You When I booked with your company on line y...	

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:

In [3]:
```
# 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:

In [4]:
```
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\SHREE\AppData\Local\Temp/ipykernel_2164/692062591.py:1: FutureWarning: In a future version of pandas all arg
uments 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:

In [5]:
```
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()
```

Out[5]:

	Hotel_Address	Average_Score	Hotel_Name	Tags	countries
0	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	7.7	Hotel Arena	Leisure trip Couple Duplex Double Room Sta...	Netherlands
1	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	7.7	Hotel Arena	Leisure trip Couple Duplex Double Room Sta...	Netherlands
2	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	7.7	Hotel Arena	Leisure trip Family with young children Dup...	Netherlands
3	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	7.7	Hotel Arena	Leisure trip Solo traveler Duplex Double Ro...	Netherlands
4	s Gravesandestraat 55 Oost 1092 AA Amsterdam ...	7.7	Hotel Arena	Leisure trip Couple Suite Stayed 2 nights ...	Netherlands

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

In [6]:
```
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:

In [7]:
```
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()
```

Let’s See 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:

In [8]:
```
recommend_hotel('Italy', 'I am going for a business trip')
```

Out[8]:

	Hotel_Name	Average_Score	Hotel_Address
0	Excelsior Hotel Gallia Luxury Collection Hotel	9.4	Piazza Duca D Aosta 9 Central Station 20124 Mi...
1	Palazzo Parigi Hotel Grand Spa Milano	9.3	Corso Di Porta Nuova 1 Milan City Center 20121...
2	Hotel Spadari Al Duomo	9.3	Via Spadari 11 Milan City Center 20123 Milan I...
3	Room Mate Giulia	9.3	Silvio Pellico 4 Milan City Center 20121 Milan...
4	UNA Maison Milano	9.3	Via Mazzini 4 Milan City Center 20123 Milan Italy

In [9]:
```
recommend_hotel('UK','I am going on a honeymoon, I need a honeymoon suite room for 3 nights')
```

Out[9]:

	Hotel_Name	Average_Score	Hotel_Address
0	Haymarket Hotel	9.6	1 Suffolk Place Westminster Borough London SW1...
1	41	9.6	41 Buckingham Palace Road Westminster Borough ...
2	Taj 51 Buckingham Gate Suites and Residences	9.5	Buckingham Gate Westminster Borough London SW1...
3	Charlotte Street Hotel	9.5	15 17 Charlotte Street Hotel Westminster Borou...
4	Ham Yard Hotel	9.5	One Ham Yard Westminster Borough London W1D 7D...