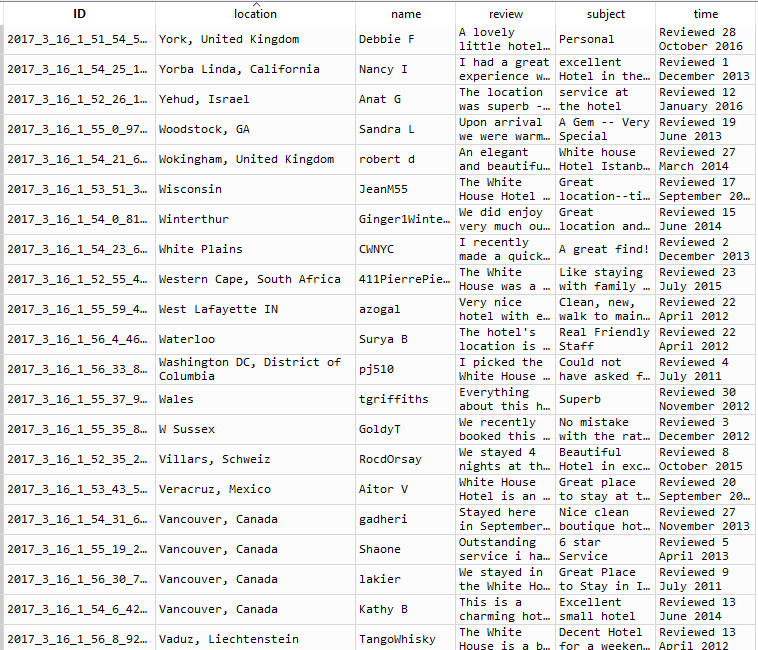
Q1. Write a program to scrape data from Tripadvisor for hotel ratings. The o/p of this data should be as shown below

Ans: Have written a python script to scrape data from Tripadvisor. For the script user needs to pass three parameters

1. Baseurl : Specify the home url of the website. (Eg: <https://www.tripadvisor.com.sg>)
2. Starturl : Specify the url from where the script needs to scrape the data (<https://www.tripadvisor.com.sg/Hotel_Review-g293974-d1604061-Reviews-White_House_Hotel_Istanbul-Istanbul.html#REVIEWS>)
3. no\_of\_pages: specify number of pages needs to be scraped

Once the script “TripAdvisor.py” is executed it will collect all the data in the pandas dataframe named “review\_collection”

Output:



2. Write the steps with scripts of how we can schedule this scraper to run every day and bring in the

recent reviews and load it into a database.

Assumption:

Have a table called “scraped\_data “ in MySQL database “webscrape”

Lets create the table, only for the first time,

create table scraped\_data

(subject\_content varchar(2000),

user\_name varchar(2000),

review varchar(2000),

review\_time varchar(2000),

user\_location varchar(2000),

scrape\_ID varchar(50) primary key not null

);

The scrapper can be scheduled using a bat file, which runs as a service all the time. It start scraping process only if the specific scheduled time reached. (Eg: 4.00 PM everyday)

Refer: scheduler.bat

Once we execute “scheduler.bat” this will automatically scrape the data everyday 8 PM and load it into the database mentioned above.

Q4. Write a text analytics algorithm on this data to determine the following

a. category of

i. Food Quality

ii. Restaurant

iii. Rooms

iv. Service

v. Price

vi. General

Since we don’t have labelling for each review we can make use of clustering technique to categorize the reviews and to get the sentiment score for each category.

1. Remove the punctuations and stop words from the review
2. After removing the stopwords perform tokenization and stemming for each review
3. Then form the Tfidf matrix (term frequency Inverse document frequency)
4. Then cluster the document by using K Means cluster
5. Once the clustering in done we can categorize the review based on the cluster results by plotting the word cloud for each cluster

Even we can use Latent Dirichlet allocation to categorize the reviews based on the content.

Refer: SentimentAnalysis.Py

In the script the variable “frame” contains the clustering results. After clustering, plotted the word cloud for each cluster as below,



From the above it is evident that the all the reviews in cluster 1 talks about the hotel room so we can categorize this cluster as “Rooms”



This cluster contains review about how good about their Stay in the hotel, can be categorized as “Service”



From this cluster result its is evident that the how close this hotel is located to the Blue Mosque Istanbul. Can be categorized as reviews based on location.



Can categorize the cluster as reviews based on staff service.

4.b) Sentiment Analysis:

Sentiment of each review can be obtained by counting the number of words used in positive intent and number of words used in the negative intent.

Refer: SentimentAnalysis.Py

In that script I have written script to find the sentiment of each review by making use of “opinion\_lexicon” from nltk.

**Analyzesentiment(sentence) :** This function accepts review as parameter for which sentiment needs to obtained.

Returns: Sentiment of the review and number of Positive/Negative/Neutral words in the sentence.

**Keywords**: All the positive/negative words in the **Analyzesentiment()** can be considered as keywords which defines the sentiment of the review.

Refer to the variable : keyword (return value of **Analyzesentiment**)

Q5. Write an overall scoring method in SQL to take the individual sentiment and get the aggregate

sentiment score for each of the categories by other dimensions – Date, Location of poster

Assuming that the “scraped\_data” table contains column called score

Select avg(score) as Score from scraped\_data group by user\_location

Q6.

Answer: Created a view as a union of all month and queried the view for required sales person.

The same can be obtained by PIVOT and UNPIVOT in MSSQL.

CREATE VIEW sales\_view

AS

select Company, Department, Salesperson, 'Jan' AS MonthName,1 as monthOrder,jan as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Feb' AS MonthName,2 as monthOrder,Feb as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Mar' AS MonthName,3 as monthOrder,mar as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Apr' AS MonthName,4 as monthOrder,Apr as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'May' AS MonthName,5 as monthOrder,May as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Jun' AS MonthName,6 as monthOrder,Jun as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Jul' AS MonthName,7 as monthOrder,Jul as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Aug' AS MonthName,8 as monthOrder,Aug as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Sep' AS MonthName,9 as monthOrder,Sep as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Oct' AS MonthName,10 as monthOrder,Oct as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Nov' AS MonthName,11 as monthOrder,Nov as Sales\_sum from sales

UNION ALL

select Company, Department, Salesperson, 'Dec' AS MonthName,12 as monthOrder,Dece as Sales\_sum from sales

then,

**select Company,Department,Salesperson, MonthName as Month, Sales\_sum as Sales from sales\_view where Salesperson='hemanta' order by monthOrder asc**

