

**Social Media Analytics and Sentiment analysis in
Hospitality**

BY
JETHAWA JATIN

Abstract

Social media analytics is one of the trending topic now days. Whether people feel happy, sad or excited, they first post it on social media to express. In a similar way, people express their thoughts about products, services offered by hotels, experiences online. This paper reflects report on social media analytics and how we can use customer reviews for benefit of the organization in the hospitality sector.

Purpose

In the world of social media today, businesses can learn a lot of things from social media data which includes happy as well as unhappy customers. Especially the reviews that we get from social media where customers post their feedbacks through Facebook, Instagram, twitter, businesses websites, third party sites et cetera. But the bad news is businesses do not even know how to use this information. Here the purpose is, with the help of natural language processing methods such as sentiment analysis, we can extract emotions related to some raw texts.

Methodology

The research aims to determine the usefulness of data scrapped from social media via various tools such as python, octoparse which can help the business (i.e. booking.com hotels) to improve their services.

Findings

The findings have been further discussed while performing sentiment analysis and represented in tableau.

Business implications

By analyzing the positive and negative reviews, hotels will know key strengths and weaknesses based on a given review and businesses can build in house customer support team to call back customers who gave negative feedbacks, and ensuring their revisit which will maintain their brand value, ultimately results in increase of business.

Limitations

The research has been successful but there are some things such as budget, funding, time constrains that poses some limitations to this.

Introduction

There is vast amount of valuable data that the hotel booking companies such as Booking.com, TripAdvisor.com are collecting from customers which are among famous booking partners. Reviews that are given by customers are of great value to many hotels. So we will discuss the various data sources that we used for the purpose of research where we scrapped the data from various sources such as booking.com and tripadvisor.com using various tools and techniques. Additionally, we will perform sentiment analysis where we analyze the sentiments about hotels.

Depending on what part of the world you live in, you may be more familiar with one over the other, they and their huge holdings make a huge chunk of the online market. Below figure 1 shows the working model of booking.com. working is similar for tripadvisor.com

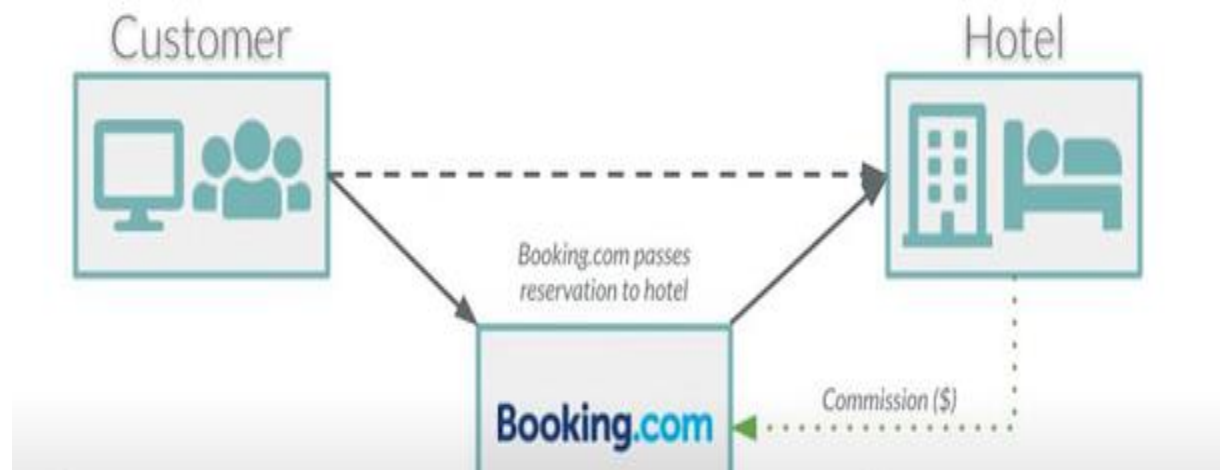


Figure 1

1. First the customer makes the booking through booking.com.
2. Then booking.com will pass that reservation to the hotel.
3. Hotel will pay commission back to the hotel, most often they will do that around the time that the customer completes their travel for the booking, but at no point in the process will booking.com see the customer money themselves.

So if the customer doesn't get the best experience as stated in the website while booking the hotel, it may affect the hotel brand value and ultimately it affects booking.com money model. So it becomes of utmost importance to analyze the reviews that are posted by customers.

Web scrapping

Some websites like booking.com and tripadvisor.com contains a large amount of invaluable customer data like customer ratings, reviews, prices, amenities provided by hotel et cetera. If one wants that information, then either you need to copy that information into a document manually or use the format what the website uses. If its small information that one wants to extract then its good, otherwise it would be very tedious to extract large amount of information. This is where web scrapping comes into play.

It refers to the technique where we extract the data from websites, social media sites such as Twitter, Facebook, API et cetera. The data extracted is exported into files such as csv or excel, which is more useful to the user. There are two ways that we will discuss by help of which we scrape data :

1. Using programming language such as Python, Java.
2. Automated tools such as octoparse, data miner.

Working of web scrapper

1. First the scrapper is provided with the url to load from which user need to scrape the information and then it loads the entire css, html and java script elements where advanced scrappers mostly render the whole information which is present in the website or any site.
2. User go through the data that is very specific according to his goal, which user wants from the page. For example, user might want to scrape the booking.com hotel page for reviews from customers but not interested in hotel prices.
3. Once the data is scraped, then it is exported into file which is probably csv or excel file as seen in below figure 2.

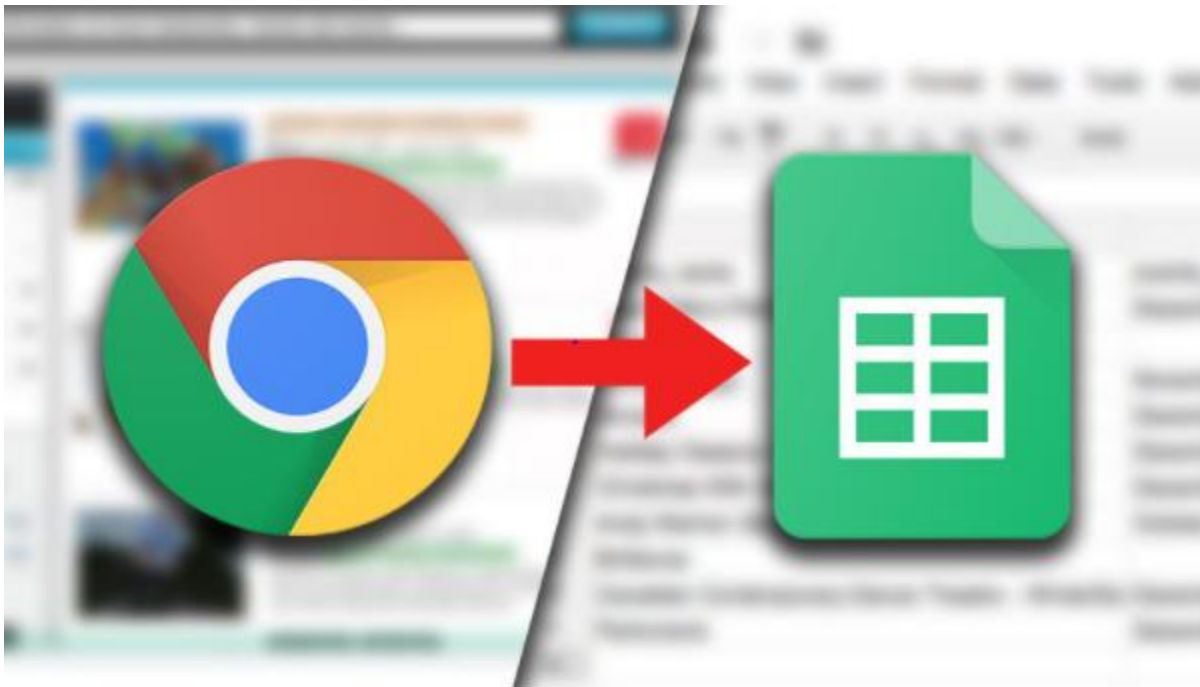


figure 2

Tools and Data sources used for web scrapping

As discussed in **Web scrapping**, there are two ways that by which user can scrape data will be discussing here:

1. Using programming language such as Python, Java.
2. Automated tools such as octoparse, data miner.

Using programming language such as Python

In this method, we will be scrapping the data from websites such as booking.com and tripadvisor.com which are hotel booking partners with the help of programming language such as python. Depending on the difficulty, one can scrape any site on the internet. User needs to go the following step for scrapping the site using python which are as follows;

Scan your source of data (Booking.com, Tripadvisor.com)


The web scrapping project starts with exploring the website (i.e. booking.com, tripadvisor.com) to understand the structure of the site so that one can extract the information that is relevant. Here we go through booking.com, then one can see listings of many hotels then if you click on that specific hotel (i.e. Radisson Blu hotel, Cardiff) then you will see that hotel with detail description with change in the url in the browser address bar as seen in below figure 3.

Cardiff: 390 properties found

Show on map

Our Top Picks | Homes & apartments first | Stars (highest first) | Stars (lowest first) | Distance From Downtown | ...

Commission paid and other benefits may affect an accommodation's ranking. [Learn more.](#) X



Radisson Blu Hotel, Cardiff ★★★★★


[Cardiff Center, Cardiff](#) · [Show on map](#) · 0.3 miles from center

Travel Sustainable property

In the heart of the city center, the 21-story Radisson Blu Hotel offers panoramic city views, air-conditioned rooms and free Wi-Fi. Cardiff Central Rail Station is close by.

Good **7.7**
5,114 reviews

Show prices

 **Looking for a space of your own?** X

Find privacy and peace of mind with an entire home or apartment to yourself

figure 3

Understand the url structure:

During scrapping the website, first we understand the url structure about how it is organized. For example, here we scrapping the data from two websites which are as follows:

1. Booking.com:

<https://www.booking.com/hotel/gb/radisson-blu.html>

2. Tripadvisor.com:

https://www.tripadvisor.co.uk/Hotel_Review-g186460-d1456219-Reviews-Radisson_Blu_Hotel_Cardiff-Cardiff_South_Wales_Wales.html

https://www.tripadvisor.co.uk/Hotel_Review-g186460-d550059-Reviews-Park_Plaza_Cardiff-Cardiff_South_Wales_Wales.html#REVIEWS

The above url is splitted into two parts in which;

1. the base url consists <https://www.booking.com/hotel/gb/> and <https://www.tripadvisor.co.uk/>.
2. the rest of the path is the site location which is specific that ends with .html.

It will use the same base url when any hotels are posted on above websites and only the site location which is specific will be changing depending on the which specific hotel you are viewing. now scan the website using developer tools which is readily available in all modern browsers to understand the html structure as seen in below figure 4.

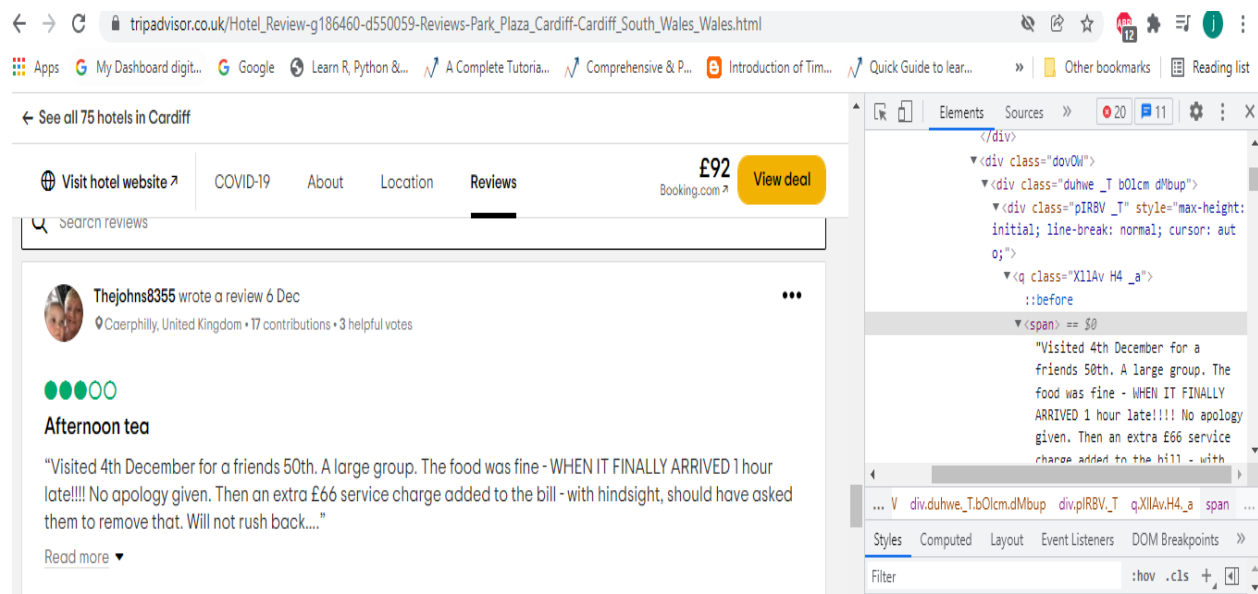


figure 4: html structure

Extract the html content from the page:

It is time to use python, we first need to install and then import requests (to send http request to retrieve content), Pandas and Numpy library to get the html content in the script using the below command:

```
import requests
import pandas as pd
import numpy as np
```

figure 5

Pandas is used to read the data and manipulating the dataframe, while Numpy is used for numerical calculation. Along with that, also need to use the agent in headers variable that tells the website that one

```
headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36'
}
source=requests.get('https://www.tripadvisor.co.uk/Hotel_Review-g186460-d1456219-Reviews-Radisson_Blu_Hotel_Cardiff-Cardiff_South_Wales-England.html')
print(source.status_code)
```

It will send get requests to the above page, server sends the html data back and stores in python variable 'source'. If we print text attribute of source variable, then we get html data as shown in below figure 7 which is unformatted. It is the static content within our python script.

figure 7

The above html content looks messy and there are large number of html elements, attributes here and there. So here beautiful soup library in python comes for rescue, which helps to parse the data in readable structured format. In figure we imported the beautiful soup library and it takes the html content which we scraped earlier as input and second argument is lxml which is appropriate parser for this problem as shown in below figure 8.

figure 8

```
print(soup.prettify())

<!DOCTYPE html>
<html lang="en-GB" xmlns:og="http://opengraphprotocol.org/schema/">
<head>
  <meta content="text/html; charset=utf-8" http-equiv="content-type"/>
  <link href="https://static.tacdn.com/favicon.ico?v2" id="favicon" rel="icon" type="image/x-icon"/>
  <link color="#000000" href="https://static.tacdn.com/img2/brand_refresh/application_icons/mask-icon.svg" rel="mask-icon" sizes="any"/>
  <meta content="#34e0a1" name="theme-color"/>
  <meta content="telephone=no" name="format-detection"/>
  <script type="text/javascript">
    window.taRollupsAreAsync = true;
  </script>
  <link crossorigin="" href="https://static.tacdn.com/css2/webfonts/TripSans/TripSans.css?v1.002" rel="stylesheet"/>
  <link as="fetch" crossorigin="anonymous" href="/static/decodeKey.txt" rel="preload"/>
  <title>
    RADISSON BLU HOTEL, CARDIFF - Updated 2021 Prices, Reviews, and Photos - Tripadvisor
  </title>
  <meta content="TripAdvisor" property="al:ios:app_name"/>

```

Extract the useful information(Reviews)

1. first look for class (i.e.” cWwQK MC R2 Gi z Z BB dXjy”) under the div tag for all the pages using find_all function.
2. We then search for div tag with class (i.e. ‘cqoFv _T’) under the step1 using find function.
3. Again search for div tag with class (i.e. ‘pIRBV _T’) under step 2, access the span tag and print the text which are our reviews.
4. Store the reviews in empty list as defined in first line named as reviews.

figure 10


```
[ 'Great location and brilliant bar staff The check in experience took over an hour and the room we booked was still not right. Even the housekeeper said we had to queue in reception to get extra bed made up. Coffee machine had no coffee. Only positive was thank goodness the bar and bar staff had a clue.',
' My husband and I came and stayed with you in Dec 2021, we stay in Cardiff for a long weekend every couple of months, this was our Christmas shopping trip, the location is perfect, both for shopping and socialization, everything you could possibly need within walking distance, the hotel is very nice and well ran, will definitely return. ',
' Nice hotel but room very cold, had to ask for a electric heater. No mirror by desk to do make up/dry hair and straighten ect which I found odd. Room was clean and spacious. Breakfast was nice, should offer a meat free alternative english breakfast, rather than cereal etc.',
' The hotel is very central in Cardiff and the rooms, bar and restaurant are all clean, tidy and well serviced. But I want to give a special mention to their Maintenance man named Milan! On our way to Cardiff we had an accident involving us and an articulated lorry. Not unsurprisingly we came off worse. Milan spent ages helping us to repair the car so we could return home the next day. He went well beyond the call of duty and ensured that the car was safe to drive. Thanks to the hotel and its staff.',
' Lovely clean hotel, helpful and friendly staff, good amenities and beds were very comfortable. Breakfast was offered at additional cost, however we chose not to have breakfast so we could get an earlier start at the shops We would definitely look to stay again soon. ',
' It could be that they were exploiting demand for rooms (there was a rugby test on in Cardiff when we went) but at over 200GBP per night for a room with two small single beds and no bar fridge overlooking the rail yards it was very disappointing. T
```

figure 11

Similarly, it follows for Park plaza hotel, we follow all the steps as shown in below figure:

Here the below code scraps the reviews given by the customer for Park Plaza Hotel which is located in Cardiff. the below code scraps the reviews from pages 1 to 80 from url shown in below figure 12. url takes the following the structure: 'https://www.tripadvisor.co.uk/Hotel_Review-g186460-d550059-Reviews-' + 'or'+str(i)+'-Park_Plaza_Cardiff-Cardiff_South_Wales_Wales.html#REVIEWS'. it loops through the pages from 0 to 80 as defined in range function. As we are interested in extracting the reviews we go through the following steps:

1. first look for class (i.e. "cWwQK MC R2 Gi z Z BB dXjiy") under the div tag for all the pages using find_all function.
2. We then search for div tag with class (i.e. 'fpMxB MC _S b S6 H5 _a') under the step1 using find function.
3. Again search for div tag with class (i.e. 'dovOW') under step 2 and print the text which are our reviews.

Store the reviews in empty list as defined in first line named as reviews1.

```
reviews1=[]
for i in range(0,80,5):
    url='https://www.tripadvisor.co.uk/Hotel_Review-g186460-d550059-Reviews-' + 'or'+str(i)+'-Park_Plaza_Cardiff-Cardiff_South_
    source=requests.get(url,headers=headers).text
    soup=BeautifulSoup(source,'lxml')
    for instance in soup.find_all('div',class_='cWwQK MC R2 Gi z Z BB dXjiy'):
        js=instance.find('div',class_='fpMxB MC _S b S6 H5 _a')
        js=instance.find('div',class_='dovOW').text
        reviews1.append(js)
    print(reviews1)
```

figure 12

The output is as follows as shown in figure 13 when we print the reviews1 variable which will list all the reviews for the hotel.

reviews1

['Visited 4th December for a friends 50th. A large group. The food was fine - WHEN IT FINALLY ARRIVED 1 hour late!!!! No apology given. Then an extra £66 service charge added to the bill - with hindsight, should have asked them to remove that. Will not rush back...Read moreDate of stay: December 2021Trip type: Travelled with friendsHelpfulShare ',
'Simon and Jackson made our trip an absolute wonder, we absolutely loved our trip here, I would definitely recommend staying here, especially because of these two, staff are lovely, rooms are immaculate, and the facilities are beautiful.Read moreDate of stay: November 2021Trip type: Travelled with friendsHelpfulShare Response from Shannon Thurlby, Marketing at Park Plaza CardiffResponded 6 days agoThank you for leaving such wonderful feedback Pearl. I have shared your comments with both Simon and Jackson and thanked them for their great customer service and for making your trip one to remember. Read more',
'Perfect girls weekend away with the daughter(13). Location was perfect for us as we were mainly here for the shopping. Hotel modern and tidy. Beds were soft and comfy. Unfortunately the TV didn't have any signal. Pool was perfect, very clean and not too busy. Breakfast provided great options(even with a fussy teen). Previously stayed in a similar priced hotel in Cardiff and park plaza met far higher standards. Will definitely be returning as we love Cardiff.Read moreDate of stay: November 2021Trip type: Travelled with familyRoom Tip: Check in a little busy at 3pm (small lobby)See more room tipsHelpfulShare Response from Shannon Thurlby, Marketing at Park Plaza CardiffResponded 6 days agoHi Danielle, thank you for leaving such glowing feedback based on your most recent visit with your daughter. We are sorry the TV did not work in your room, due to a recent storm, some of the TV signals were disconnected and have since been rectified. We look forward to welcoming you back to Cardiff and to the Park Plaza within the near future. Read more',
'Pros - lovely hotel - Friendly staff - Great location - ask reception for discount code on ncp app for greyfriars road car park - Nice use of facilities - pool and jacuzzi - Breakfast was lovely - The bar and lounge are really nice - Rooms are

figure 13

Now we scrape the reviews from booking.com also in a similar way:

Here the below code scraps the reviews given by the customer for Radisson blu Hotel which is located in Cardiff. the below code scraps the reviews from url shown in below figure 14. url takes the following the structure:"https://www.booking.com/hotel/gb/radisson-blu.html?aid=304142;label=gen173nr-1FCAEoggI46AdIM1gEaFCIAQGYATG4ARfIAQzYAQH4AQKIAgGoAgO4AqnGr40GwAIB0gIkNGRkYWEwYjEtNDYyYi00MjhmLWI2OGQtZGU5Yzg4NTQxOTc12AIF4AIB;sid=90a38df6345d1951f1c921f74d3c51a4;dest_id=2591777;dest_type=city;dist=0;group_adults=2;group_children=0;hapos=3;hpos=3;no_rooms=1;req_adults=2;req_children=0;room1=A%2CA;sb_price_type=total;sr_order=popularity;sreporch=1638654778;srpvid=00a399dc28550026;type=total;ucfs=1&#tab-reviews". As we are interested in extracting the reviews we go through the following steps:

1. first look for class (i.e." c-review__row") under the div tag for all the pages using find_all function.
2. We then search for span tag with class (i.e. 'c-review__body') under the step1 using find function and print text which are our reviews as seen in below figure 14.

```
headers = {  
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36'  
}  
source=requests.get('https://www.booking.com/hotel/gb/radisson-blu.html?aid=304142;label=gen173nr-1FCAEoggI46AdIM1gEaFCIAQGYATG4A
```

```
soup=BeautifulSoup(source,'lxml')
```

```
soup
```

```
edia="screen" rel="stylesheet" type="text/css"/>
<link href="https://cf.bstatic.com/static/css/hotel_experiments_cloudfront_sd.iq_ltr/b9c4a902c5ef747651352658419b64bf5e7abff
8.css" media="screen" rel="stylesheet" type="text/css"/>
<link href="https://cf.bstatic.com/static/css/hotel_experiments_rtrw_cloudfront_sd.iq_ltr/ca57db08e6aabfd4d1a3c36a7c9b03d73c4
33295.css" media="screen" rel="stylesheet" type="text/css"/>
<link data-defer-prefetch="" href="https://cf.bstatic.com/static/css/searchresults_cloudfront_sd.iq_ltr/45149ad5ba8f17d8211b6
6c547b6d43ff0173f32.css" rel="_prefetch"/>
<link data-defer-prefetch="" href="https://cf.bstatic.com/static/css/book_cloudfront_sd.iq_ltr/21709b3d489903fbfdb1e4bc771d75
bdefed850a.css" rel="_prefetch"/>
<link href="https://cf.bstatic.com/static/css/incentives_cloudfront_sd.iq_ltr/6ebaa1b38db75ed4aac17958c02e8dbc6a8bf09e.css" m
edia="screen" rel="stylesheet" type="text/css"/>
<link href="https://cf.bstatic.com/static/css/mlt_cloudfront_sd.iq_ltr/5993142fd8e90ad7f4cd7f00ffaa44345c5fdf46.css" rel="sty
lesheet"/>
<style> #basiclayout, .basiclayout { margin: 0; } #special_actions { margin: 3px 15px 3px 0; } .ticker_space { margin-top: 3p
x !important; } #logo_no_globe_new_logo { top: 14px; } .b_msie_6 #top, .b_msie_6 body.header_resuffle #top {height:61px !imp
ortant;} .b_msie_6 #special_actions { margin: 3px 15px 3px 0; overflow:visible; } body.header_resuffle #top { min-height: 50
px !important; height: auto !important; } .nobg { background: #fff url("https://cf.bstatic.com/static/img/nobg_all_blue_iq/b7

for revies in soup.find_all('div',class_='c-review__row'):
    reviews=reviews.find('span',class_='c-review__body').text
    print(reviews)
    print()]
```

There was a good standard of hygiene”er pressure and instant hot water.

“Staff were fantastic - very helpful-very accommodating”

“Breakfast was fantastic, so much to choose from and the restaurant and bar was also so lovely, definitely worth the money gett
ing a breakfast meal”

“Excellent hotel that is a few minutes walk from the train station, and also literally a minute walk to the city centre. I had
a large room with views of the city and could see the castle and stadium from my window. The room and bathroom was large, it ha
d a fridge, kettle, study desk, cupboard and a safe which was perfect. Lift was also modern and fast, unlike some hotels I've s
tayed in”

“The room was spacious, the bed was comfortable and the staff were welcoming on arrival. Can't fault the location, right in the
centre of Cardiff.”

“Breakfast was great location to the moot point was excellent”

“Freindly welcome and helpful staff. Comfortable beds and clean surroundings. Breakfast was out of the world, lots of choice, w

figure 14

Scrapping using Automated tools such as Octoparse

Octoparse is one of the web scrapping technique built for the one who is not technical at all but wants to scrap data. It does not require coding at all, additionally it is easy to use and effective cost wise also. Now we will build scrapper with the help of octoparse where we intend to scrape details related to hotels which include address, rating, price, name, url. The process is as follows:

1. First need to create an account on octoparse which gives one free trial with limited access.
2. Secondly, octoparse have a browser from where we load our page.
3. As we want to scrape the data from multiple pages, there is an Action tips menu at the top right corner in which we need to select “loop click the selected link”. It basically tells to go through all pages while scraping as seen in below figure 15.



figure 15

- Now we need to go one by one through each of the hotels by clicking the title on the listing page and then click on "loop click on each element" as seen in Action tips menu. Finally you arrive at the detail page of the hotel as seen in below figure 16.

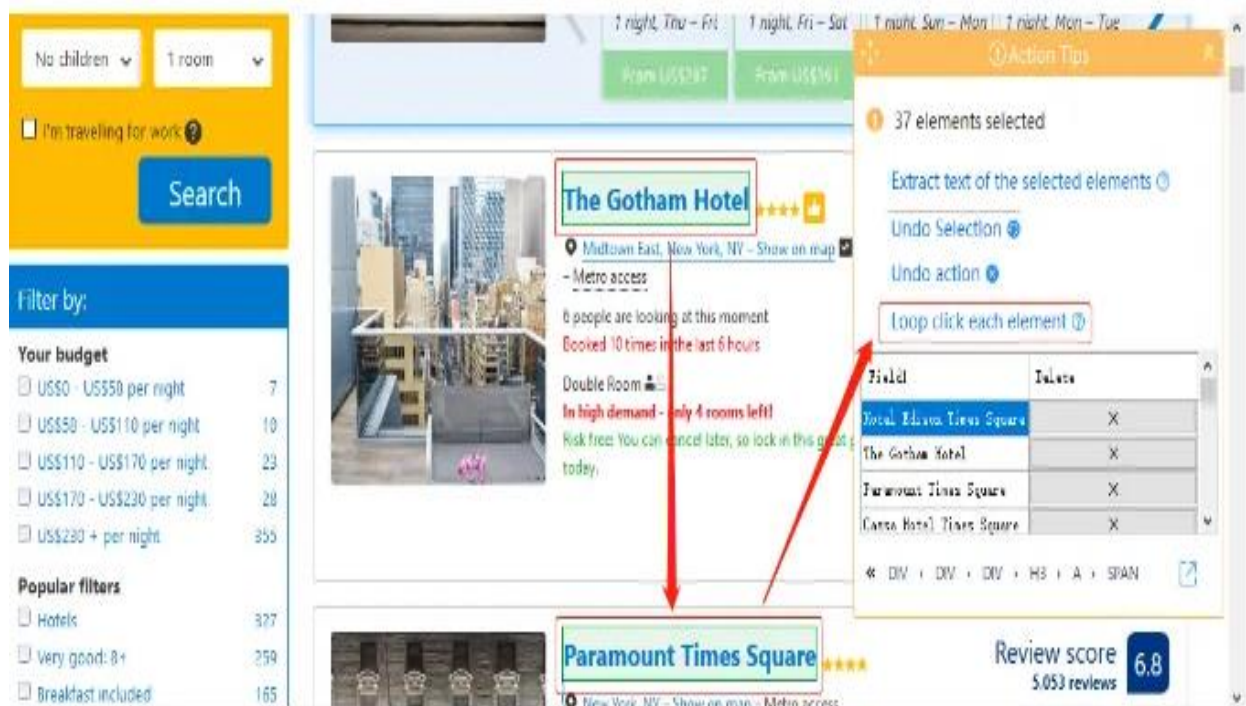


figure 16

- Whatever the columns needed, click on that (i.e. rating, price, name et cetera) as seen in below figure 17.

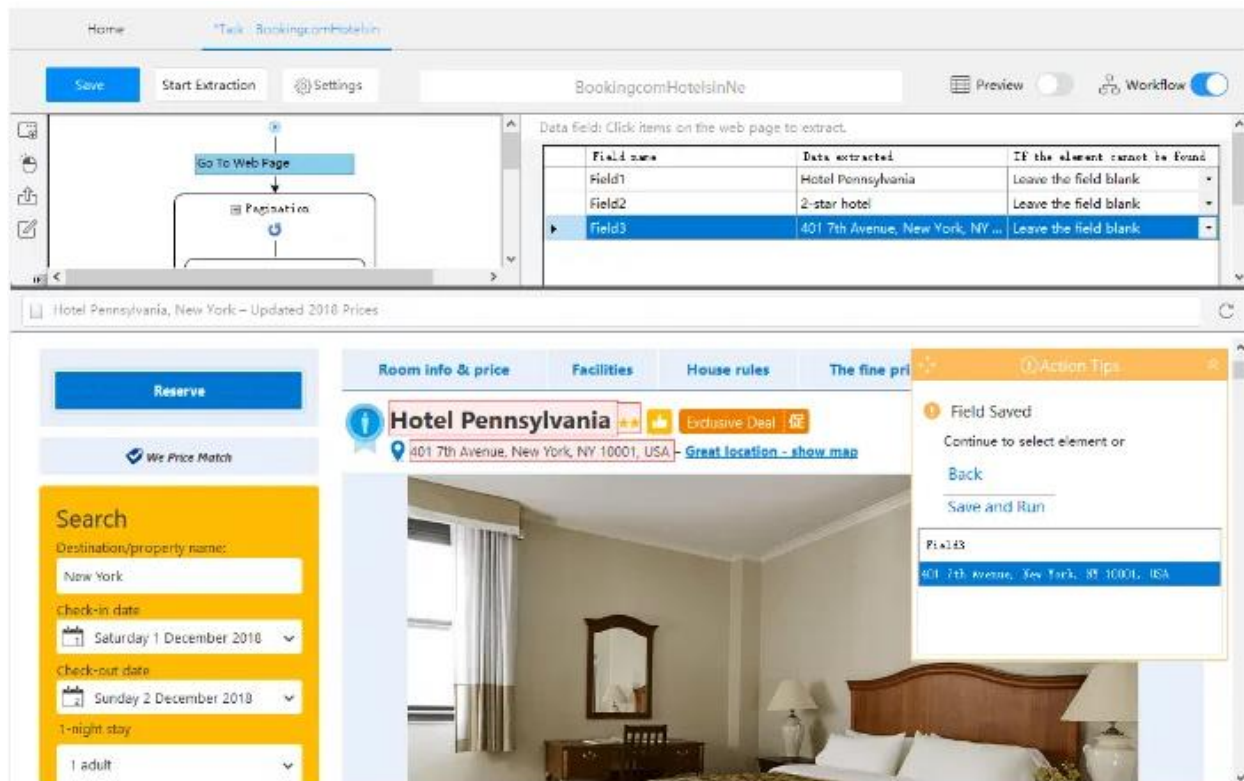


figure 17

- Run the task and the automated tool will scrap the data for you as seen in below figure:

Hotel_Ad	Additional	Review_D	Average	Hotel_Na	Reviewer	Negative_Review_T	Total_Nur	Positive_F	Review_T	Total_Nur	Reviewer	Tags	days_sinc	lat	Ing
s Gravesa	194	#####	7.7	Hotel Are	United Ki	My room	210	1403	Great loc	26	1	3.8 [' Leisure	13 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	Cleaner d	33	1403	The room	18	6	4.6 [' Leisure	17 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	Apart fro	11	1403	Good loc	19	1	10 [' Leisure	17 days	52.36058	4.915968
s Gravesa	194	7/7/2017	7.7	Hotel Are	United Ki	Nothing a	5	1403	Rooms wi	101	2	10 [' Leisure	127 days	52.36058	4.915968
s Gravesa	194	7/6/2017	7.7	Hotel Are	United Ki	The floor	28	1403	Comfy be	6	7	4.6 [' Leisure	128 days	52.36058	4.915968
s Gravesa	194	7/3/2017	7.7	Hotel Are	United Ki	Very stee	38	1403	Great ons	14	8	6.3 [' Leisure	131 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	Bed was c	40	1403	Friendly s	17	1	6.3 [' Leisure	145 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	Nothing	3	1403	Lovely ho	130	2	9.6 [' Leisure	168 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	Nothing a	51	1403	The Hotel	134	2	9.6 [' Leisure	170 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	Extensive	61	1403	Friendly s	16	14	7.1 [' Leisure	184 days	52.36058	4.915968
s Gravesa	194	#####	7.7	Hotel Are	United Ki	the only t	93	1403	The locati	186	2	8.8 [' Leisure	1105 day	52.36058	4.915968

figure 18

Performing analysis on scrapped reviews of hotels

Word cloud:

Radisson Blu Hotel

Word cloud is used for large text if you have large text and you want to present that text in a visual way by focusing on the important words in that text words that repeated multiple times and people seem interested in that phrase or word then this will come in a large text.

The size of the word indicates about how many times that word being mentioned by different people in your survey or your report or your article that gives an indication of interests and people are interested in that keywords.

Reasons to use Word cloud:

1. Make an impact: it makes an impact, you can show things and visualize that are important to the people.
2. it is easy to understand, by looking at it, you can see based on the size of the words or phrase that gives an indication that this is important and can easily be shared in an image and send it an email or post it in social media.

As seen in below figure 19 we first import important libraries such as word cloud, stopwords, and Matplotlib where stopwords represents most common words which does not contribute much in the analysis. We will eliminate it from the text before creating word cloud. The below code will show a maximum of hundred words in the cloud which has been often used.

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
```

```
stopwords= set(STOPWORDS)

def mywordcloud(data, title=None):
    wordcloud=WordCloud(
        background_color='white',
        stopwords=stopwords,
        max_words=100,
        max_font_size=40,
        scale=3,
        random_state=1).generate(str(data))

    fig=plt.figure(1,figsize=(20,20))
    plt.axis('off')
    if title:
        fig.subtitle(title, fontsize=20)
        fig.subplots_adjust(top=2.3)
    plt.imshow(wordcloud)
    plt.show()
```

figure 19



There are many words that tells what customers are saying about the Radisson blu hotel in Cardiff: some words tell about the positive experience of customers such as enjoyed, comfortable, comfortable et cetera which tells people were happy. On other hand, some words tell about the negative experience of customers such as exploiting, disappointing, bad which are small in size which tells very less customers had the bad experience.

Park Plaza Hotel

As seen in below figure 21 we first import important libraries such as word cloud, stopwords, and Matplotlib where stopwords represents most common words which does not contribute much in the analysis. We will eliminate it from the text before creating word cloud. The below code will show a maximum of hundred words in the cloud which has been often used.

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
```

```
stopwords= set(STOPWORDS)

def mywordcloud(data, title=None):
    wordcloud=WordCloud(
        background_color='white',
        stopwords=stopwords,
        max_words=100,
        max_font_size=40,
        scale=3,
        random_state=1).generate(str(data))

    fig=plt.figure(1,figsize=(20,20))
    plt.axis('off')
    if title:
        fig.subtitle(title, fontsize=20)
        fig.subplots_adjust(top=2.3)
    plt.imshow(wordcloud)
    plt.show()
```

figure 21

2. Numpy: used for performing mathematical calculation on arrays.
3. Nltk: one of the packages to perform natural language processing.
4. Matplotlib: used for visualization.
5. Seaborn: similar package as similar to seaborn which is also used for visualization.
6. Genism: used for topic and vector modelling.
7. Scikit-learn: to perform machine learning operations.

Reading and grouping the data

First we will read the hotel data which is located in downloads folder using pandas read function and then will append the positive and negative reviews into one column named as review. Additionally, we will be creating one column named as 'is_bad_review' based on reviewer score in which if score is less than 5 then its flagged as 1 otherwise 0 as seen in below figure 23.

```
import pandas as pd

# read data
reviews_df = pd.read_csv("C:/Users/DELL/Downloads/Hotel_Reviews.csv")
# append the positive and negative text reviews
reviews_df["review"] = reviews_df["Negative_Review"] + reviews_df["Positive_Review"]
# create the label
reviews_df["is_bad_review"] = reviews_df["Reviewer_Score"].apply(lambda x: 1 if x < 5 else 0)
# select only relevant columns
reviews_df = reviews_df[["review", "is_bad_review"]]
reviews_df.head()
```

	review	is_bad_review
0	I am so angry that i made this post available...	1
1	No Negative No real complaints the hotel was g...	0
2	Rooms are nice but for elderly a bit difficul...	0
3	My room was dirty and I was afraid to walk ba...	1
4	You When I booked with your company on line y...	0

figure 23

Cleaning the data

The below code removes 'no negative' and 'no positive' from text as seen in below figure 24.

```
# remove 'No Negative' or 'No Positive' from text
reviews_df["review"] = reviews_df["review"].apply(lambda x: x.replace("No Negative", "").replace("No Positive", ""))
```

figure 24

In below figure 25 we import stopwords, wordnetlemmatizer libraries where stopwords represents the basic words such as the, a, an, of, for etcetera which does not contribute while performing our analysis and lemmatizer converts various words to their root form where it represents one word only.

```
import string
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.tokenize import WhitespaceTokenizer
from nltk.stem import WordNetLemmatizer
```

figure 25

In below figure 25 we will go through various steps for cleaning the raw text which involves as follows:

1. Text are mixed of small and capital words,for simplicity we will convert all text to small.
2. Secondly we will take out punctuations and will split the text into words.
3. Remove the words that are not useful such as the one which contains numbers.
4. Take off stopwords as described earlier which does not contribute in the analysis.
5. Lemmatization

```
def clean_text(text):
    # Lower text
    text = text.lower()
    # tokenize text and remove punctuation
    text = [word.strip(string.punctuation) for word in text.split(" ")]
    # remove words that contain numbers
    text = [word for word in text if not any(c.isdigit() for c in word)]
    # remove stop words
    stop = stopwords.words('english')
    text = [x for x in text if x not in stop]
    # remove empty tokens
    text = [t for t in text if len(t) > 0]
    # pos tag text
    pos_tags = pos_tag(text)
    # Lemmatize text
    text = [WordNetLemmatizer().lemmatize(t[0], get_wordnet_pos(t[1])) for t in pos_tags]
    # remove words with only one letter
    text = [t for t in text if len(t) > 1]
    # join all
    text = " ".join(text)
    return(text)
```

figure 26

Feature engineering

Firstly, we will import Vader library from nltk which is used for sentiment analysis. It basically flags negative or positives based on list of words and context of sentences. For each row, it returns negative score, positive score, neutral score and overall score. Also we added the column that represents no. of characters and words.

```
# add sentiment analysis columns
from nltk.sentiment.vader import SentimentIntensityAnalyzer

E:\ANACONDA\lib\site-packages\nltk\twitter\__init__.py:20: UserWarning: The twython library has not been installed. Some functionality from the twitter package will not be available.
  warnings.warn("The twython library has not been installed. ")

sid = SentimentIntensityAnalyzer()
reviews_df["sentiments"] = reviews_df["review"].apply(lambda x: sid.polarity_scores(x))
reviews_df = pd.concat([reviews_df.drop(['sentiments'], axis=1), reviews_df['sentiments'].apply(pd.Series)], axis=1)

# add number of characters column
reviews_df["nb_chars"] = reviews_df["review"].apply(lambda x: len(x))

# add number of words column
reviews_df["nb_words"] = reviews_df["review"].apply(lambda x: len(x.split(" ")))
```

figure 27

Now we import gensim package which represents words in numerical vector space. It represents similar words in a way how close they are to each other. Along with that we use doc2vec, the name itself tells converting text into vectors where we use those vectors for training. We will feed the text data to our model to get representation vectors as seen in figure 28.

```
# create doc2vec vector columns
from gensim.test.utils import common_texts
from gensim.models.doc2vec import Doc2Vec, TaggedDocument

documents = [TaggedDocument(doc, [i]) for i, doc in enumerate(reviews_df["review_clean"].apply(lambda x: x.split(" ")))]

# train a Doc2Vec model with our text data
model = Doc2Vec(documents, vector_size=5, window=2, min_count=1, workers=4)

# transform each document into a vector data
doc2vec_df = reviews_df["review_clean"].apply(lambda x: model.infer_vector(x.split(" "))).apply(pd.Series)
doc2vec_df.columns = ["doc2vec_vector_" + str(x) for x in doc2vec_df.columns]
reviews_df = pd.concat([reviews_df, doc2vec_df], axis=1)

E:\ANACONDA\lib\site-packages\gensim\utils.py:1212: UserWarning: detected Windows; aliasing chunkize to chunkize_serial
  warnings.warn("detected Windows; aliasing chunkize to chunkize_serial")
```

figure 28

As seen in below figure 29, we import Tf id vectorizer from sklearn package which stands for term frequency inverse document frequency. It overcomes the disadvantages of bag of model in which we simply count the frequency of words in each document. It accounts for relative importance rather than giving weightage based on frequency of words.

```
# add tf-idfs columns
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(min_df = 10)
tfidf_result = tfidf.fit_transform(reviews_df["review_clean"]).toarray()
tfidf_df = pd.DataFrame(tfidf_result, columns = tfidf.get_feature_names())
tfidf_df.columns = ["word_" + str(x) for x in tfidf_df.columns]
tfidf_df.index = reviews_df.index
reviews_df = pd.concat([reviews_df, tfidf_df], axis=1)
```

figure 29

Exploring the data

To get more sense of the data let's explore, we can see that about 5% of our reviews are negative and it tells that it is an imbalance problem where classes are not balanced as seen in below figure 30.

```
reviews_df["is_bad_review"].value_counts(normalize = True)

0    0.956761
1    0.043239
Name: is_bad_review, dtype: float64
```

figure 30

In the below figure, we are building word cloud which will show maximum of 200 words.

```
# wordCloud function

from wordcloud import WordCloud
import matplotlib.pyplot as plt

def show_wordcloud(data, title = None):
    wordcloud = WordCloud(
        background_color = 'white',
        max_words = 200,
        max_font_size = 40,
        scale = 3,
        random_state = 42
    ).generate(str(data))

    fig = plt.figure(1, figsize = (20, 20))
    plt.axis('off')
    if title:
        fig.suptitle(title, fontsize = 20)
        fig.subplots_adjust(top = 2.3)

    plt.imshow(wordcloud)
    plt.show()
```

figure 31

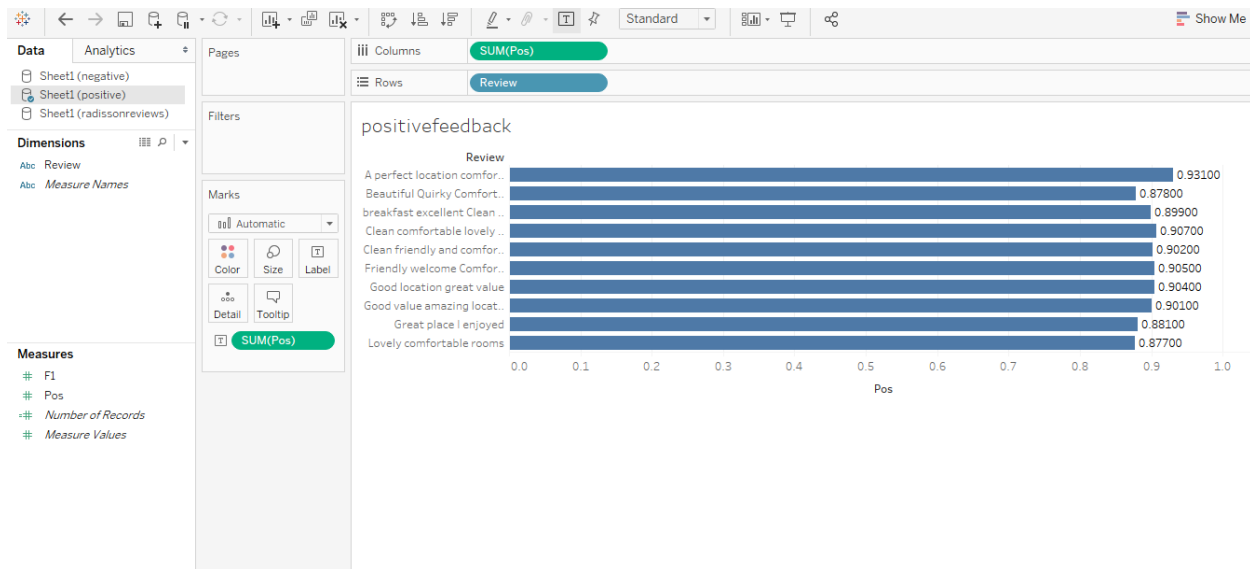


figure 34

Negative feedback visualization

As seen in below figure 35 and 36, the feedbacks that are bad which we received from the customers indeed represents negative experience. It has been similarly represented in tableau which is one of the visualization software based on drag and drop concept used for visualization. It has many capabilities which makes it one of top trending software in data science, data analytics industries used for visualization.

	review	neg
193086	No dislikes LOCATION	0.831
356368	Nothing Great helpful wonderful staff	0.812
318516	A disaster Nothing	0.804
458794	Nothing Excellent friendly helpful staff	0.799
29666	A bit noisy No	0.796
426057	Dirty hotel Smells bad	0.762
263187	Very bad service No	0.758
443796	Nothing perfect	0.750
181508	Window blind was broken	0.744
175316	Nothing Super friendly staff	0.743

figure 35

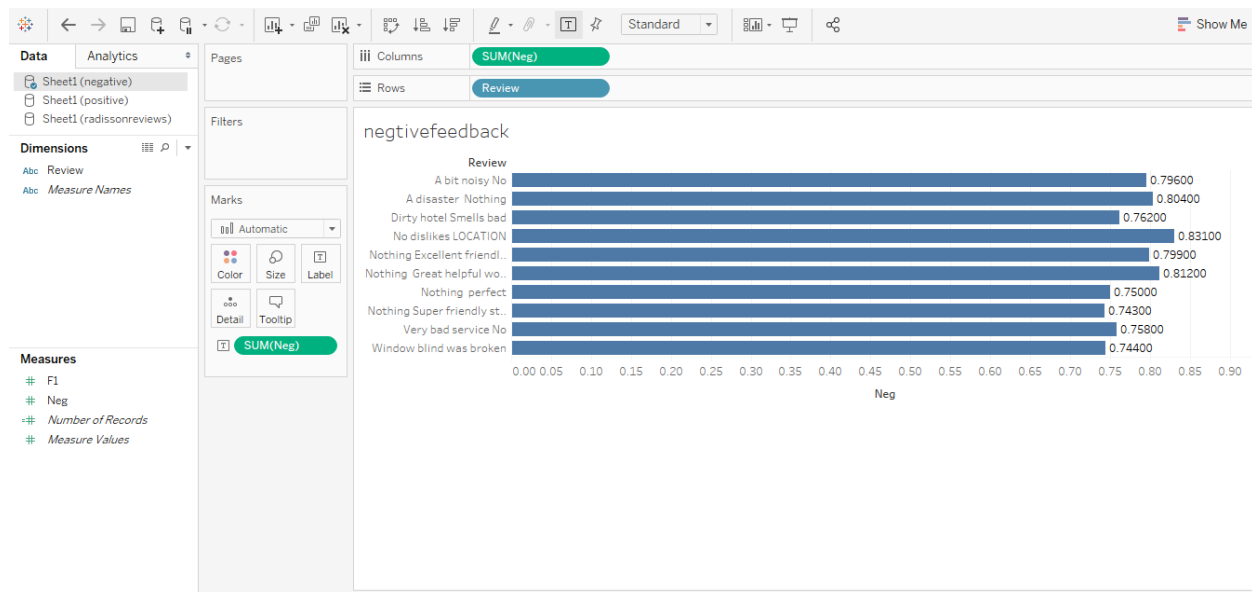


Figure 36

Training the sentiment model

Splitting the dataset in training and testing set

First, we divide whole encoded dataset into X and y variables where X represents list of independent variables and y represents dependent variable as shown in below figure 37.

Training dataset is on which we will train the model and testing dataset is on which we test our trained model. The ratio used here is 80:20 for training and testing dataset as show in below figure 35. X_train represents training dataset of list of independent variables and y_train represents training dataset of target variable whereas X_test represents testing dataset of list of independent variables and y_test represents testing dataset of target variable.

```
# feature selection
label = "is_bad_review"
ignore_cols = [label, "review", "review_clean"]
features = [c for c in reviews_df.columns if c not in ignore_cols]

# split the data into train and test
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(reviews_df[features], reviews_df[label], test_size = 0.20, random_state = 42)
```

figure 37

Random forest classifier

It is the machine learning algorithm which will be using here in below figure 38. Here the base estimator is decision tree, trained on a different bootstrap sample having the same size as the training set. It introduces the further randomization in the training of individual trees. Additionally, when a tree is trained at each node, only d features are sampled from all features without replacement. The node is then split using the sampled feature that maximizes the information gain. d defaults to square root of no. of features. The below figure gives the most important features that contributes in predicting the type of sentiment. Here the doc2vec representations have more importance, in addition some words have a descent importance as well.

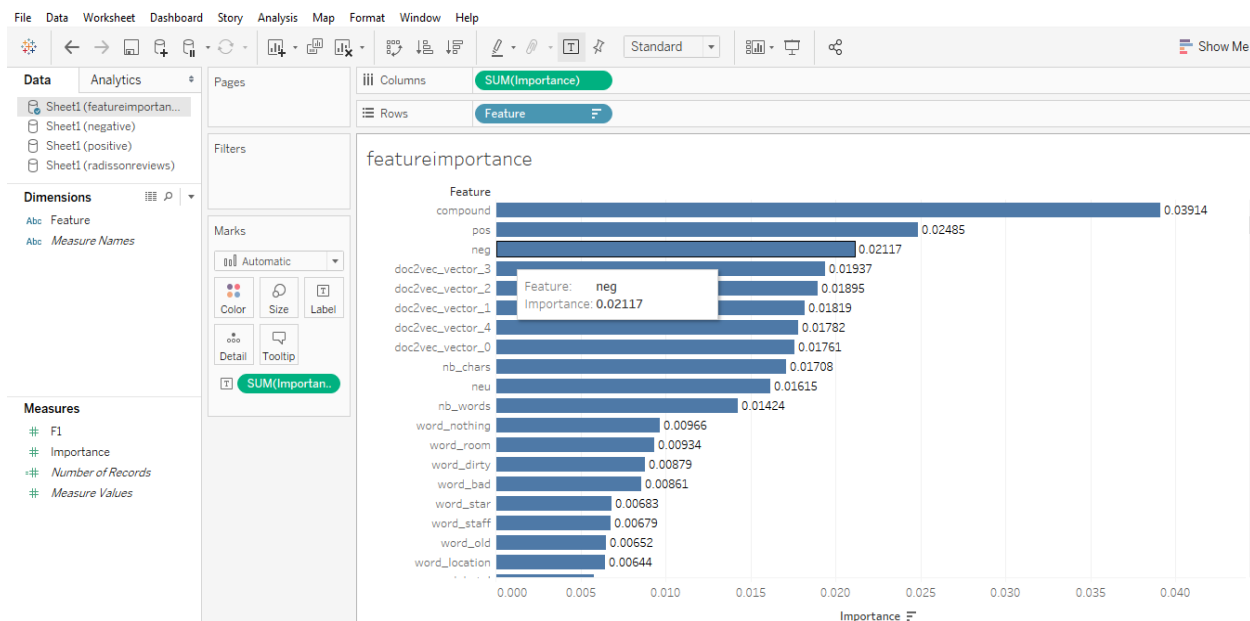
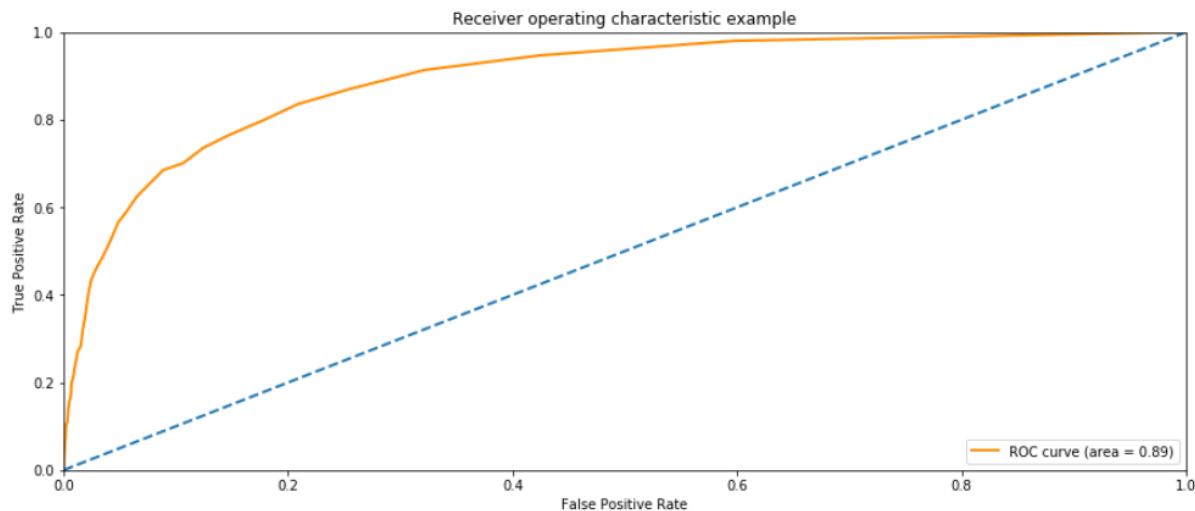


figure 38 : feature importance

Conclusion:

After training the model in previous step, we would evaluate with help of Roc curve that stands for receiver operating characteristics, which tells about how the model is performing. The more the curve to the left, the good predictions are. Area under the curve lies between 0 and 1 and it is 0.89 which is very good.



Overall, we analyzed raw text for two hotels (Radisson blu and park plaza) at first and then all hotels in Cardiff, the results were quite optimistic meaning most of the people got good experience (as represented in word clouds) as stated at the booking.com and TripAdvisor website. Along with that we also come across negative reviews, for which by help of this analysis the hotel brands can improve their service. Moreover, we trained a sentiment model that can predict whether review is good or bad based on raw text and results were represented in tableau.

Limitations

1. There are many challenges that came while scrapping as there is a limitation on no. of pages to be scrapped.
2. Server would block if we repeating hitting the server.
3. Another challenge is in terms of budget, funding and time constraint that poses some limitations to the project.
4. When it comes to big data regarding to text data, at some point tableau will be lagged in terms of processing and time complexity.

References

Weiguo Fan and Gordon, M.D. (2014). *The Power of Social Media Analytics*. [online] ResearchGate. Available at: https://www.researchgate.net/publication/259148570_The_Power_of_Social_Media_Analytics [Accessed 10 Dec. 2021].

Stieglitz, S., Mirbabaie, M., Ross, B. and Neuberger, C. (2018). Social media analytics – Challenges in topic discovery, data collection, and data preparation. *International Journal of Information Management*, [online] 39, pp.156–168. Available at: <https://www.sciencedirect.com/science/article/pii/S0268401217308526#bib0045> [Accessed 10 Dec. 2021].

Real Python (2021). *Beautiful Soup: Build a Web Scraper With Python*. [online] Realpython.com. Available at: <https://realpython.com/beautiful-soup-web-scraper-python/> [Accessed 10 Dec. 2021].

Madila, S.S., Dida, M.A. and Kaijage, S. (2021). A Review of Usage and Applications of Social Media Analytics. *Journal of Information Systems Engineering and Management*.

Oheix, J. (2018). *Detecting bad customer reviews with NLP - Towards Data Science*. [online] Medium. Available at: <https://towardsdatascience.com/detecting-bad-customer-reviews-with-nlp-d8b36134dc7e> [Accessed 10 Dec. 2021].

Octoparse.com. (2021). *Scrape Hotel Data without Writing a Single Line of Code with Octoparse*. [online] Available at: <https://www.octoparse.com/blog/how-to-build-a-hotel-data-scraper-when-you-are-not-a-techie#> [Accessed 10 Dec. 2021].

Rishan Sanjay (2021). *Sentiment Analysis using NLP on Hotel Review Dataset*. [online] Medium. Available at: <https://medium.com/analytics-vidhya/sentiment-analysis-using-nlp-on-hotel-review-dataset-fa049e23de29> [Accessed 10 Dec. 2021].