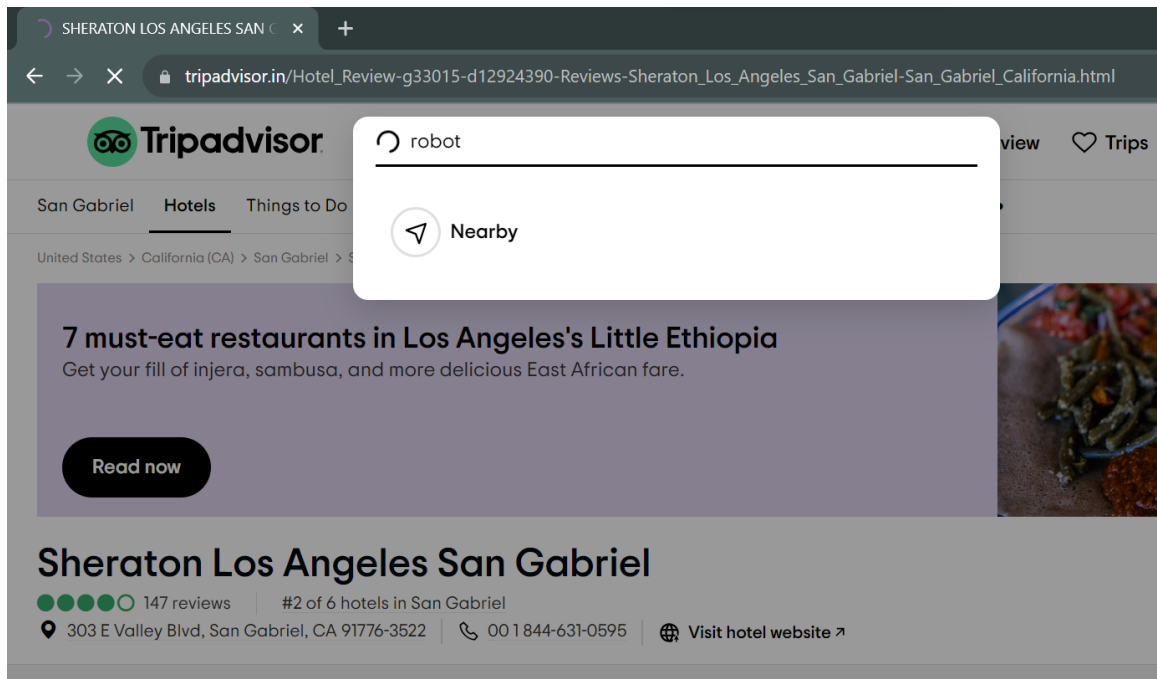


# REPORT

1. First we checked all the hotels where “robot” keyword is used (I found 900 hotels)



2. Now I need to fetch their names and links to them, so that's what I do, I used selenium to automate that process .

```
# this script is used to fetch links of hotels that have robot in any of their
comment
def new(url):
    driver =
webdriver.Edge(r"C:\Users\deesh\Downloads\edgedriver_win64\msedgedriver.exe")
    driver.get(url)
    time.sleep(2)
    driver.find_element(By.XPATH, '//*[@id="search-
filters"]/ul/li[2]/a').click()
    time.sleep(2)

    while True:
        time.sleep(1)
        try:
            #getting those name of the hotels using their css selector
            hotels_name = driver.find_elements(By.CSS_SELECTOR,"div.result-
title")

            for i in hotels_name:
                a = i.get_attribute("onclick")
                link = a.split("widgetEvCall")[1].split(",")[3]
```

```

        print(link)
        links.append(link)

    next_button = None
    try:
        next_button = driver.find_element(By.XPATH,
'//*[@id="BODY_BLOCK_JQUERY_REFLOW"]/div[2]/div/div[2]/div/div/div/div/div[1]/div/div[1]/div/div[3]/div/div[2]/div/div/div/a[2]')
    except:
        break

    if next_button and "disabled" not in
next_button.get_attribute("class"):
        next_button.click()
    else:
        break

except Exception as e:
    #handling the errors sometime loading time is too high and we need
to handle those
    print(f"Stale element error: {str(e)}")
    driver.quit()

new("https://www.tripadvisor.co.uk/Search?&q=robot")
print(links)

```

This basically print all the links and then we can create the array.

3. After getting the array of links, we can basically apply an iterator to iterate through it and pass through this function.

```

# Using this function, we are fetching near about 250 comments from the hotels
if it has that much
def extractor(url):
    global reviews_data
    try:
        driver =
webdriver.Edge(r"C:\Users\deesh\Downloads\edgedriver_win64\msedgedriver.exe")
        driver.get(r"{}".format(url))
        time.sleep(2)
        search_input = driver.find_element(By.XPATH,
'//*[@id="BODY_BLOCK_JQUERY_REFLOW"]/div[2]/div/div[2]/div/div/div/div/div[1]/div/div[1]/div/div[3]/div/div[2]/div/div/div/a[2]')
        time.sleep(1)
        search_input.send_keys('robot')
        search_input.send_keys(Keys.RETURN)
    except:
        try:driver.close()
        except:

```

```

        pass
    extractor(url)
    hotel_name = url.split("Reviews-")[1].split(".html")[0]
    count = 0
    z = []
    while True:
        try:
            time.sleep(2)
            #fetching the review text and review dates using their css
selector
            review_text = driver.find_elements(By.CSS_SELECTOR,
"div.vTVdc>div>div>span>span")
            review_dates = driver.find_elements(By.CSS_SELECTOR,
"div.cRVsd>span")
            count += len(review_dates)

            # breaking the loop if we already got near about 250 reviews
            if (count > 250):
                break
            # printing the review text and review dates for each and every
page

            print(len(review_text), len(review_dates))
            index = 0
            index2 = 0

            while (index2 < len(review_dates)):
                review_t = review_text[index].text
                if (review_t == "Helpful"):
                    index += 1
                    continue
                review_d = review_dates[index2].text.split("review ")[1]
                index += 1; index2 += 1

                # storing the data in the form of hotel name, review text and
review date

                z.append({
                    "Hotel Name": hotel_name,
                    "Review Text": review_t,
                    "Review Date": review_d
                })

        except:
            driver.close()
            extractor(url)

    next_button = None
    try:
        time.sleep(1)

```

```

        next_button = driver.find_element(By.CSS_SELECTOR,
"div.MD>div>a.next")
        time.sleep(1)
    except:
        break
    if next_button and "disabled" not in
next_button.get_attribute("class"):
        next_button.click()
    else:
        break

driver.quit()
if (len(z)>0):
    reviews_data.extend(z)
    print(hotel_name, len(reviews_data))

```

4. After that we get an array of data in the form of

```

z.append({
    "Hotel Name": hotel_name,
    "Review Text": review_t,
    "Review Date": review_d
})

```

5. Now I will make a dataframe and convert it into excel using **Pandas**.