BA data collection

September 17, 2024

[1]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
```

[2]:
```
#create an empty list to collect reviews
reviews = []
#create empty list to collect stars
stars = []
#create empty list to collect the date
date = []
#create empty list to collect the country the reviewer is from
country = []
```

[3]:
```
base_url = "https://www.airlinequality.com/airline-reviews/british-airways"
pages = 39
page_size = 100

# for i in range(1, pages + 1):
for i in range(1, pages + 1):

    #print(f"Scraping page {i}")

    # Create URL to collect links from paginated data
    url = f"{base_url}/page/{i}/?sortby=post_date%3ADesc&pagesize={page_size}"

    # Collect HTML data from this page
    response = requests.get(url)

    # Parse content
    soup = BeautifulSoup(response.content, 'html.parser')

    # Extract review containers
    review_containers = soup.find_all("article", {"itemprop": "review"})

    for container in review_containers:
        # Extract review text
        review_text = container.find("div", {"class": "text_content"})
```

```
        if review_text:
            reviews.append(review_text.text.strip())
        else:
            reviews.append("None")

        # Extract star rating
        rating = container.find("div", {"class": "rating-10"})
        if rating:
            rating_value = rating.find("span", {"itemprop": "ratingValue"})
            if rating_value:
                stars.append(rating_value.text.strip())
            else:
                stars.append("None")
        else:
            stars.append("None")


        # Extract date
        review_date = container.find("time")
        if review_date:
            date.append(review_date.text.strip())
        else:
            date.append("None")

        # Extract country
        reviewer_info = container.find("h3", {"class": "text_sub_header␣
↪userStatusWrapper"})
        if reviewer_info:
            country_text = reviewer_info.find("span").next_sibling
            if country_text:
                country.append(country_text.strip(" ()").strip())
            else:
                country.append("None")
        else:
            country.append("None")

    # Check if the number of reviews is less than the page size
    if len(review_containers) < page_size:
        print("Last page detected.")
        break  # Stop if the number of reviews is less than the expected page␣
↪size
    if not review_containers:
        print("No reviews found, possibly end of pages.")
        break  # Stop if no reviews are found (end of data)
```

```
Last page detected.
```

[4]:
```
len(reviews)
```

[4]: 3857

[5]:
```
len(stars)
```

[5]: 3857

[6]:
```
len(country)
```

[6]: 3857

[7]:
```
len(date)
```

[7]: 3857

[8]:
```
#create a DataFrame for these collected lists of data
df = pd.DataFrame({"reviews": reviews, "rating" : stars, "date" : date,␣
 ↪"country" : country})
```

[9]:
```
df.head()
```

[9]:

	reviews	rating	date	country
0	Not Verified | A nightmare journey courtesy o…	1	8th September 2024	United Kingdom
1	Trip Verified | Absolutely atrocious. LHR-OR…	1	6th September 2024	United Kingdom
2	Trip Verified | As someone who flies relentl…	4	2nd September 2024	United Kingdom
3	Trip Verified | Flew with British Airways …	2	1st September 2024	United Kingdom
4	Trip Verified | Straightforward check in T…	8	30th August 2024	United Kingdom

[10]:
```
df.shape
```

[10]: (3857, 4)

[13]:
```
df.to_csv("C:/Users/bendh/Desktop/data science/JN/BA_2.csv")
```

[ ]:
```

```