Objective

The Objective of this project is to web scrape british airways customer review data from the web and perform sentiment analysis using Azure AI APIfor Python and present Insights .

Checking and setting up environment variables

To set the environment variable for your Language resource key, open a console window, and follow the instructions for your operating system and development environment.

os.environ[***'LANGUAGE_KEY'***]= '*key*'
os.environ[**'LANGUAGE_ENDPOINT'**]= '*endpoint*'

Importing important libraries

```
!pip install beautifulsoup4
!pip install azure-ai-textanalytics==5.2.0
from azure.ai.textanalytics import TextAnalyticsClient
from azure.core.credentials import AzureKeyCredential
language_key = os.environ.get('LANGUAGE_KEY')
language_endpoint = os.environ.get('LANGUAGE_ENDPOINT')
import requests
from bs4 import BeautifulSoup
import pandas as pd

#initialize dataframe
df=pd.DataFrame(columns=['Date','Rating','Reviews_heading','Reviews_text','aircraft','Traveller_type'])
```

Authenticate the client using your key and endpoint

```
# Authenticate the client using your key and endpoint
def authenticate_client():
    ta_credential = AzureKeyCredential(language_key)
    text_analytics_client = TextAnalyticsClient(
            endpoint=language_endpoint,
            credential=ta_credential)
    return text_analytics_client

client = authenticate_client()
```

Extracting data through Beautiful soup

```
subsoup={}
subsoup2={}
subsoup3={}
subsoup4={}
subsoup5={}
subsoup6={}
subsoup7={}
##extract the review header
def soup_extract_header(x):
    for i in range(1,x+1):
        html = requests.get('https://www.airlinequality.com/airline-reviews/british-airways/page/'+str(i)+'/?
        bs=BeautifulSoup(html.text, 'html.parser')
        subsoup[i]=(bs.find_all('h2',class_='text_header'))
    return subsoup
## extract user review rating
def soup_extract_rating(x):
    for i in range(1,x+1):
        html = requests.get('https://www.airlinequality.com/airline-reviews/british-airways/page/'+str(i)+'/?
        #print(html)
        bs=BeautifulSoup(html.text, 'html.parser')
        subsoup2[i]=bs.find_all('span', itemprop="ratingValue")
    return subsoup2
###extract the review text
def soup_extract_content(x):
    for i in range(1,x+1):
        html = requests.get('https://www.airlinequality.com/airline-reviews/british-airways/page/'+str(i)+'/?
        #print(html)
        bs=BeautifulSoup(html.text, 'html.parser')
        subsoup3[i]=bs.find_all('div',class_='text_content',itemprop="reviewBody")
    return subsoup3
## extract the attributes of customer
def soup_extract_stats(x):
    for i in range(1,x+1):
        html = requests.get('https://www.airlinequality.com/airline-reviews/british-airways/page/'+str(i)+'/?
        #print(html)
        bs=BeautifulSoup(html.text, 'html.parser')
        subsoup4[i]= bs.find_all('div',class_='review-stats')
    return subsoup4
## extract the review date
def soup_extract_date(x):
    for i in range(1,x+1):
        html = requests.get('https://www.airlinequality.com/airline-reviews/british-airways/page/'+str(i)+'/?
        #print(html)
        bs=BeautifulSoup(html.text, 'html.parser')
        subsoup6[i]=bs.find_all('time',itemprop="datePublished")
    return subsoup6
## extract customer name
def soup_extract_name(x):
    for i in range(1,x+1):
        html = requests.get('https://www.airlinequality.com/airline-reviews/british-airways/page/'+str(i)+'/?
        #print(html)
        bs=BeautifulSoup(html.text, 'html.parser')
        subsoup7[i]=bs.find_all('span', itemprop="name")
    return subsoup7
```

Initialize the functions

```
subsoup_obj1=soup_extract_header(10)
subsoup_obj2=soup_extract_rating(10)
subsoup_obj3=soup_extract_content(10)
subsoup_obj4=soup_extract_stats(10)
subsoup_obj6=soup_extract_date(10)
subsoup_obj7=soup_extract_name(10)
```

```
import re
reviews_heading=[]
for i in range(1,len(subsoup_obj1)+1):
    for j in range(len(subsoup_obj1[i])):
        heading=" ".join(subsoup_obj1[i][j].contents)
        heading=re.findall(r'\w+',heading)
        heading=" ".join(heading)
        reviews_heading.append(heading)
        #print(reviews_heading)
reviews_heading[0:10]
df['Reviews_heading']=reviews_heading
```

```
ratings=[]
for i in range(1,len(subsoup_obj2)+1):
    #print(f'for i=',i)
    #print(f'subsoup_obj2[i]=',subsoup_obj2[i])
    #print(f'no of records',len(subsoup_obj2[i]))
    for j in range(0,100):
        #print(f'for j=',j)
        #print(f'subsoup_obj2[i][j]=',subsoup_obj2[i][j])
        #print(f'subsoup_obj2[i][j].contents=',subsoup_obj2[i][j].contents)
        reviews_ratings=" ".join(subsoup_obj2[i][j].contents)
        reviews_ratings=re.findall(r'\d+',reviews_ratings)[0]
        #print(reviews_ratings)
        ratings.append(reviews_ratings)
        #print(f'ratings',ratings)
        #print(len(ratings))
print(ratings[0:10])
df['Rating']=ratings
```

```
['5', '1', '9', '2', '1', '1', '2', '3', '3', '9']
```

```
Reviews_text=[]
for i in range(1,len(subsoup_obj3)+1):
    print(f'subsoup_obj3[i]',subsoup_obj3[i])
    for j in range(len(subsoup_obj3[i])):
          text=" ".join([tag.text for tag in subsoup3[i][j].contents])
          #Reviews_text=re.findall(r'\+',reviews_ratings)[0]
          Reviews_text.append(text)
          print(Reviews_text)
print(Reviews_text[0:10])
df['Reviews_text']=Reviews_text
```

```
IOPub data rate exceeded.
The notebook server will temporarily stop sending output
to the client in order to avoid crashing it.
To change this limit, set the config variable
`--NotebookApp.iopub_data_rate_limit`.

Current values:
NotebookApp.iopub_data_rate_limit=1000000.0 (bytes/sec)
NotebookApp.rate_limit_window=3.0 (secs)
```

```
stats=[]
aircraft=[]
traveller_type=[]
seat_type=[]
route=[]
date_flown=[]
recommended=[]
subsoup5={}
for i in range(1,len(subsoup_obj4)+1):
    #print(subsoup_obj4[i])
    for j in range(len(subsoup_obj4[i])):
         #print(subsoup_obj4[i][j].contents)
         subsoup5[j]=subsoup_obj4[i][j].contents[1].select('td.review-value')
         text1=" ".join(subsoup5[j][-1].contents)
         text2=" ".join(subsoup5[j][-3].contents)
         text3=" ".join(subsoup5[j][-2].contents)
         text4=" ".join(subsoup5[j][-4].contents)
         #print(subsoup5)
         #print(len(subsoup5))
         recommended.append(text1)
         date_flown.append(text3)
         route.append(text2)
         seat_type.append(text4)

#print(len(recommended))
print(seat_type[0:10])
print(recommended[0:10])
print(date_flown[0:10])
print(route[0:10])
#df['aircraft']=aircraft
#df['Traveller_type']=traveller_type
df['Seat_type']=seat_type
df['Route']=route
df['Date_Flown']=date_flown
df['Recommend']=recommended
#print(stats[0])
```

```
['Premium Economy', 'Economy Class', 'Economy Class', 'Business Class', 'Economy Class', 'Economy Class', 'Business Class', 'Ec
['no', 'yes', 'no', 'no', 'no', 'no', 'no', 'no', 'yes', 'no']
['January 2024', 'May 2024', 'May 2024', 'October 2023', 'May 2024', 'May 2024', 'May 2024', 'May 2024', 'May 2024', 'August 2
['Los Angeles to London', 'Hannover to London Heathrow', 'Austin to London Heathrow', 'Vienna to Johannesburg via London', 'Joh
```

```
df.head()
```

	Date object	Rating object	Reviews_heading o	Reviews_text obj...	aircraft object	Traveller_type obj...	Seat_ty
0	nan	5	extremely poor c...	Not Verified | We ...	nan	nan	Premiu
1	nan	1	a pleasant and ci...	✅ Trip Verified | ...	nan	nan	Econor
2	nan	9	the worst BA fligh...	✅ Trip Verified | ...	nan	nan	Econor
3	nan	2	Never again Britis...	✅ Trip Verified | ...	nan	nan	Busine
4	nan	1	only been offered...	✅ Trip Verified | ...	nan	nan	Econor

```
date=[]
for i in range(1,len(subsoup_obj6)+1):
    for j in range(len(subsoup_obj6[i])):
          text=subsoup6[i][j].contents[0]
          date.append(text)

print(date[0:10])
print(len(date))
df['Date']=date
```

```
['1st June 2024', '1st June 2024', '31st May 2024', '31st May 2024', '30th May 2024', '29th May 2024', '26th May 2024', '20th M
1000
```

```
name=[]
for i in range(1,len(subsoup_obj7)+1):
    for j in range(len(subsoup_obj7[i])):
          text=subsoup7[i][j].contents[0]
          name.append(text)
print(f'name=',name[0:10])
print(f'length of name:,',len(name))
df['Name']=name
```

```
name= ['Jason George', 'S Barton', 'Marvin Daugherty', 'Markus Hornek', 'V Smart', 'Isabel Mondorf', 'L Tomlinson', 'G Layne',
length of name:, 1000
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Date             1000 non-null   object
 1   Rating           1000 non-null   object
 2   Reviews_heading  1000 non-null   object
 3   Reviews_text     1000 non-null   object
 4   aircraft         0 non-null      object
 5   Traveller_type   0 non-null      object
 6   Seat_type        1000 non-null   object
 7   Route            1000 non-null   object
 8   Date_Flown       1000 non-null   object
 9   Recommend        1000 non-null   object
 10  Name             1000 non-null   object
dtypes: object(11)
memory usage: 86.1+ KB
```

```
df.head()
```

	Date object	**Rating** object	**Reviews_heading** o	**Reviews_text** obj...	**aircraft** object	**Traveller_type** obj...	**Seat_ty**
0	1st June 2024	5	extremely poor c...	Not Verified | We ...	nan	nan	Premiu
1	1st June 2024	1	a pleasant and ci...	✅ Trip Verified | ...	nan	nan	Econor
2	31st May 2024	9	the worst BA fligh...	✅ Trip Verified | ...	nan	nan	Econor
3	31st May 2024	2	Never again Britis...	✅ Trip Verified | ...	nan	nan	Busine
4	30th May 2024	1	only been offered...	✅ Trip Verified | ...	nan	nan	Econor

```
### cnverting date to datetime object
df['Date_new']=pd.to_datetime(df['Date'])
```

```
df.drop('Date',axis=1,inplace=True)
```

```
df.drop(['aircraft','Traveller_type'],axis=1,inplace=True)
```

```
df.shape
```

```
(1000, 9)
```

```
#extract year
df['Year']=df['Date_new'].dt.year
```

```
## extract month
df['Month']=df['Date_new'].dt.month
```

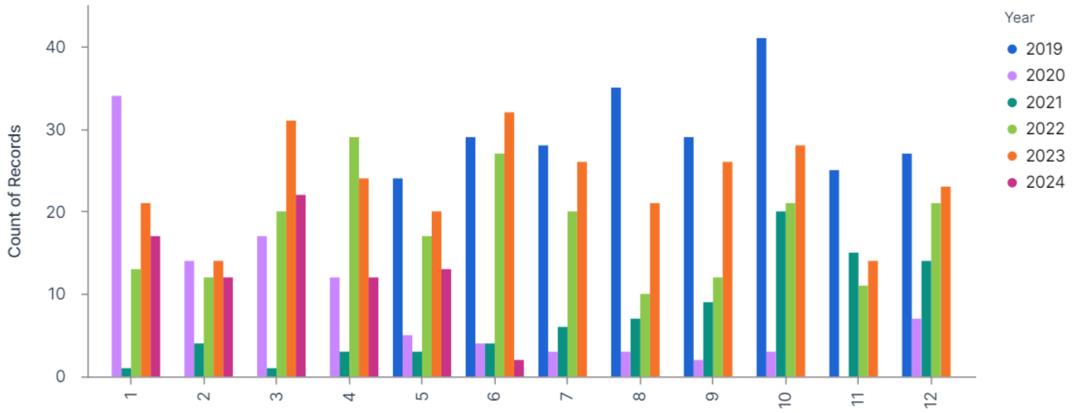
```
##extract day
df['Day']=df['Date_new'].dt.day
```

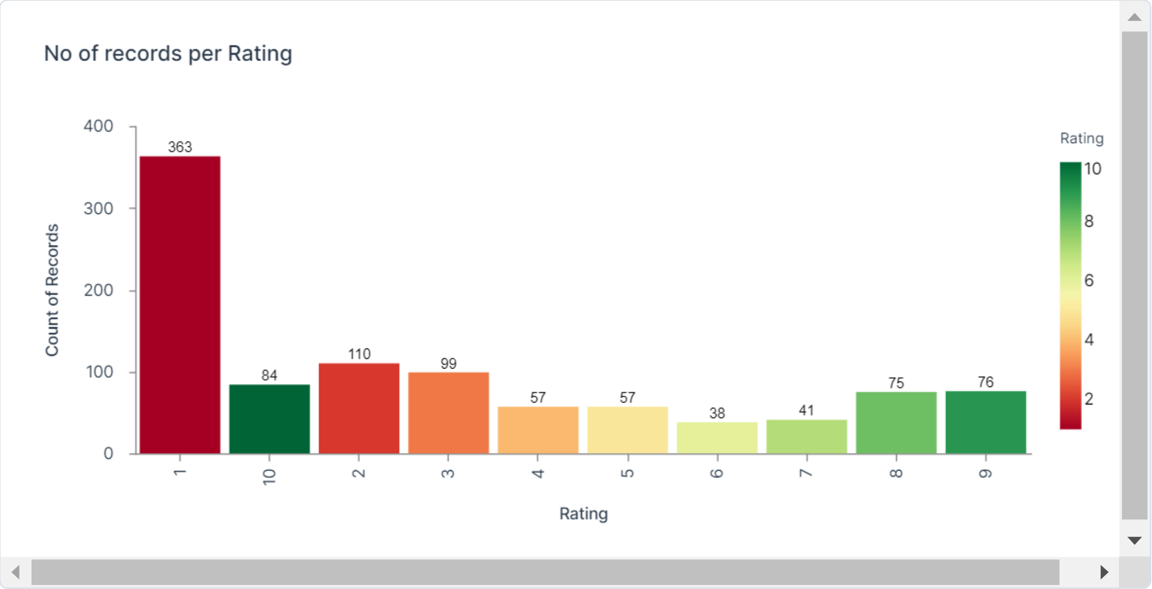
```
df.set_index('Date_new',inplace=True)
```

```
df.head()
```

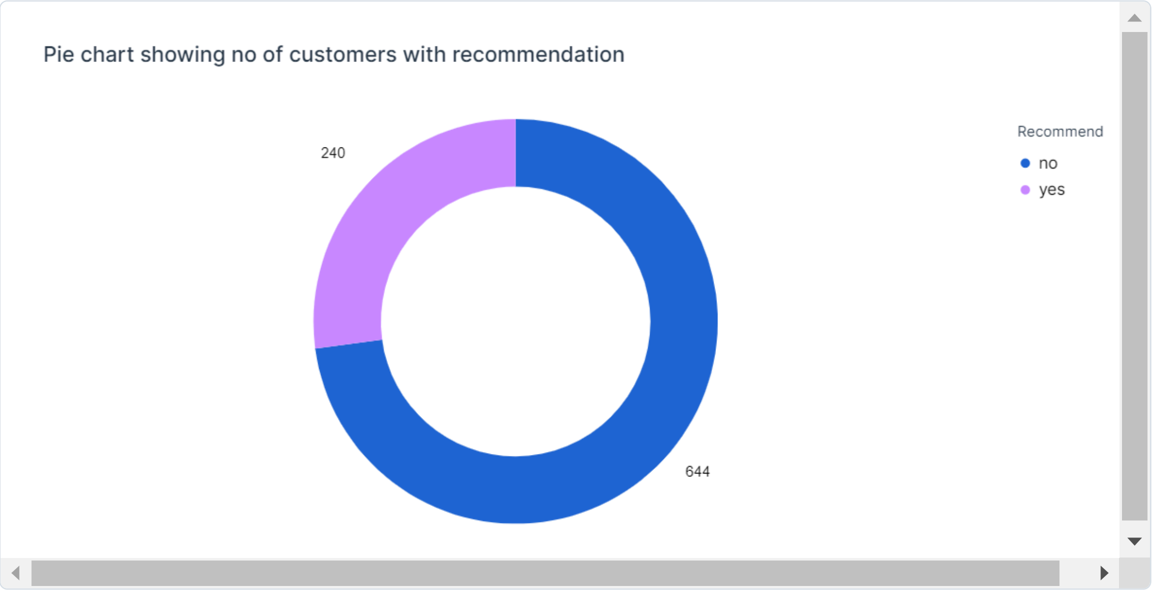
	Rating object	Reviews_heading o	Reviews_text obj...	Seat_type object	Route object	Date_Flown object	Recom
20...	5	extremely poor c...	Not Verified | We ...	Premium Economy	Los Angeles to L...	January 2024	no
20...	1	a pleasant and ci...	☑ Trip Verified | ...	Economy Class	Hannover to Lond...	May 2024	yes
20...	9	the worst BA fligh...	☑ Trip Verified | ...	Economy Class	Austin to London ...	May 2024	no
20...	2	Never again Britis...	☑ Trip Verified | ...	Business Class	Vienna to Johann...	October 2023	no
20...	1	only been offered...	☑ Trip Verified | ...	Economy Class	Johannesburg to ...	May 2024	no

No of records each month

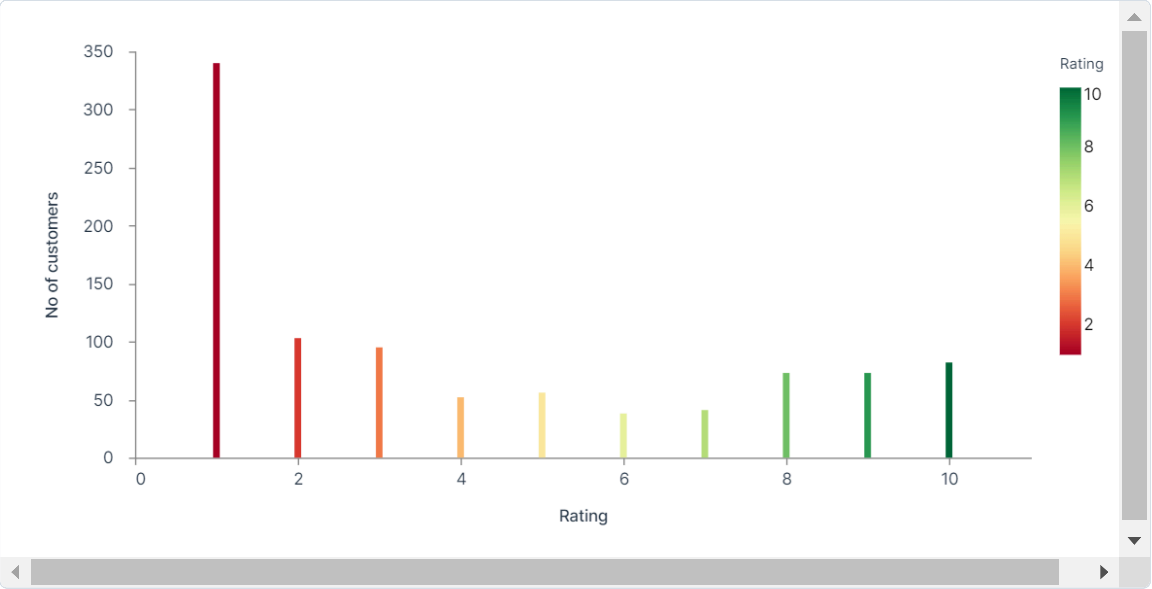




No of records per Rating



Pie chart showing no of customers with recommendation



Detecting sentiments for each record

```
positive_score={}
negative_score={}
neutral_score={}
def sentiment_analysis_with_opinion_mining_example(client):
    for  i in range(0,len(df)):
            text=[]
            text.append(df.iloc[i,2])
            print(text)
            result = client.analyze_sentiment(text, show_opinion_mining=False)
            doc_result = [doc for doc in result if not doc.is_error]
            positive_reviews = [doc for doc in doc_result if doc.sentiment == "positive"]
            print(positive_reviews)
            negative_reviews = [doc for doc in doc_result if doc.sentiment == "negative"]
            print(negative_reviews)
            for document in doc_result:
                            print("Document Sentiment: {}".format(document.sentiment))
                            print("Overall scores: positive={0:.2f}; neutral={1:.2f}; negative={2:.2f} \n".fo
                                 document.confidence_scores.positive,
                                 document.confidence_scores.neutral,
                                 document.confidence_scores.negative))
                            positive_score[i]=document.confidence_scores.positive
                            neutral_score[i]=document.confidence_scores.neutral
                            negative_score[i]=document.confidence_scores.negative
```

```
sentiment_analysis_with_opinion_mining_example(client)
```

```
Document Sentiment: positive
Overall scores: positive=0.99; neutral=0.01; negative=0.00

["✅  Trip Verified  | London to Athens. British Airways is a glorified budget airline. A 3.5-hour flight and back to Ather
[]
[]
Document Sentiment: mixed
Overall scores: positive=0.13; neutral=0.19; negative=0.68

['Not Verified  |  Terrible lack of any leg and body room in economy. This was easily the most cramped space I have ever fl
[]
[AnalyzeSentimentResult(id=0, sentiment=negative, warnings=[], statistics=None, confidence_scores=SentimentConfidenceScores
Document Sentiment: negative
Overall scores: positive=0.04; neutral=0.11; negative=0.84

['✅  Trip Verified  |  Buenos Aires to London Heathrow rwturn. The aircraft is very old, cabin configuration is very old a
[]
[AnalyzeSentimentResult(id=0, sentiment=negative, warnings=[], statistics=None, confidence_scores=SentimentConfidenceScores
Document Sentiment: negative
Overall scores: positive=0.00; neutral=0.03; negative=0.97

['✅  Trip Verified  | Mexico City to Barcelona via London Heathrow. The B787 is an incredible plane. The legroom is quite
[]
[]
Document Sentiment: mixed
Overall scores: positive=0.66; neutral=0.05; negative=0.28

['✅  Trip Verified  | Great all round. BA2591, 11 October. Good price, easy boarding, lovely cabin crew, great iced coffee
[]
[]
```

```
sentiment_df=pd.DataFrame(index=positive_score.keys(),data=positive_score.values(),columns=['Postive_score'])
sentiment_df2=pd.DataFrame(index=neutral_score.keys(),data=neutral_score.values(),columns=['Neutral_score'])
sentiment_df3=pd.DataFrame(index=negative_score.keys(),data=negative_score.values(),columns=['Negative_score'
sentiment_df=pd.concat([sentiment_df,sentiment_df2,sentiment_df3],axis=1)
```

```
df.reset_index(drop=True,inplace=True)
```

```
df_final=pd.concat([df,sentiment_df],axis=1)
```

Exporting to csv file

```
df_final.to_csv('sentiments.csv')
```

```
df_final.head()
```

	Rating object	Reviews_heading o	Reviews_text obj...	Seat_type object	Route object	Date_Flown object	Recom
0	5	extremely poor c...	Not Verified | We ...	Premium Economy	Los Angeles to L...	January 2024	no
1	1	a pleasant and ci...	☑ Trip Verified | ...	Economy Class	Hannover to Lond...	May 2024	yes
2	9	the worst BA fligh...	☑ Trip Verified | ...	Economy Class	Austin to London ...	May 2024	no
3	2	Never again Britis...	☑ Trip Verified | ...	Business Class	Vienna to Johann...	October 2023	no
4	1	only been offered...	☑ Trip Verified | ...	Economy Class	Johannesburg to ...	May 2024	no