News Headline ETL Project

Group 3: Clarence Robinson, Abby Herrup, Tim Schurmann, & Brett Thompson

21 December 2020

Table of Contents

OVERVIEW

There are a lot of news sources out there, so how do you find which articles are relevant to you? Creating clean ETL pipelines and reports can ensure the clear communication of that data. This project focuses on the ETL (Extract, Transform, Load) processing of data and how it can deliver actionable insights.

ETL PROCESS

I. Extraction

We used two different sources that provided data used in our final project. The first was data from allsides.com, a news website that ranks articles based on their political leanings. Our process was creating a python script that would web-scrape relevant headlines on allsides.com based on their CSS tags in the HTML code. The fields of interest included the following:

- Article Name
- Article Sub-heading
- Category
- Source
- Article URL

Data Scrape

```
In [6]:
# URL of page to be scraped
url = 'https://www.allsides.com/unbiased-balanced-news'
response = requests.get(url)
response.status_code

Out[6]: 200

In [7]:
html = response.content
soup = bs(html, "lxml")

# Sources
source = soup.find("div", class_="row-fluid bias-trio-wrapper")
source_name = source.find_all("div", class_="news-source")

# Articles
article = soup.find("div", class_="row-fluid bias-trio-wrapper")
article_name = article.find_all("div", class_="news-title")

# Category
category_name = soup.find_all("div", class_="news-topic")

# Sub Header
article_sub_header = soup.find_all("div", class_="topic-description")

# Source url
article_url = soup.find("div", class_="row-fluid bias-trio-wrapper")
article_urls = article_url.find_all('div', class_="news-title")

In [8]:
print(len(article_urls))
print(len(article_sub_header))
print(len(category_name))
print(len(article_name))
print(len(source_name))

45
45
45
45
45
```

Figure 1: Data Scrape Code

The second data source was from The Guardian's API (Application Programming Interface). The Guardian created a useful API so to deliver the same information as listed above.

API Data Scrape

In [11]:
```
search=["opinion","technology","politics","economy"]


for news in search:
    api_urls = f"https://content.guardianapis.com/search?q={news}&show-fields=trailText,headline&api-key={api_key}"
    response = requests.get(api_urls)
    data=response.json()
    for x in range (0,10):
        title=data["response"]["results"][x]["fields"]["headline"]
        webUrl=data["response"]["results"][x]["webUrl"]
        subheader=data["response"]["results"][x]["fields"]["trailText"]
        section=data["response"]["results"][x]["sectionName"]
        source_input="The Guardian"


        articles.append(title)
        sub_headers.append(subheader)
        urls.append(webUrl)
        categories.append(section)
        sources.append(source_input)
```

In [12]:
```
print(len(sources))
print(len(articles))
print(len(categories))
print(len(sub_headers))
print(len(urls))
```

```
85
85
85
85
85
```

Figure 2: API Data Scrape Code

II. Transformation

To transform public data and use it in our study, our process was the following:

a. For the web-scraped data, we put our extracted data into lists to turn this into a dataframe. The same process was done for the data extracted through The Guardian's API.
b. The Guardian API and web-scraped data were then joined together in a single dataframe. This was the "headlines" dataFrame.
c. We created two more dataframes for categories of news and a sources dataframe (where the news came from). In order to make sure each source had a ***unique identifier*****,** we created a script that would loop through our data and assign and id number to each respective source/category. This would avoid duplicate entries of unique identifies connected to each article.

Unique Categories/Sources Data Scrape

In [13]:

```
categories_unique = []
for x in categories:
    if x not in categories_unique:
        categories_unique.append(x)
print(len(categories_unique))

sources_unique = []
for x in sources:
    if x not in sources_unique:
        sources_unique.append(x)
print(len(sources_unique))
```

```
40
37
```

Figure 3: Unique Categories Assignment

III. Load

After all data was loaded into data frame, we connected to PostgreSQL using PG admin. An ERD was created using the quick database diagrams website and the initial code to create our initial table schema in postgres was exported as well.

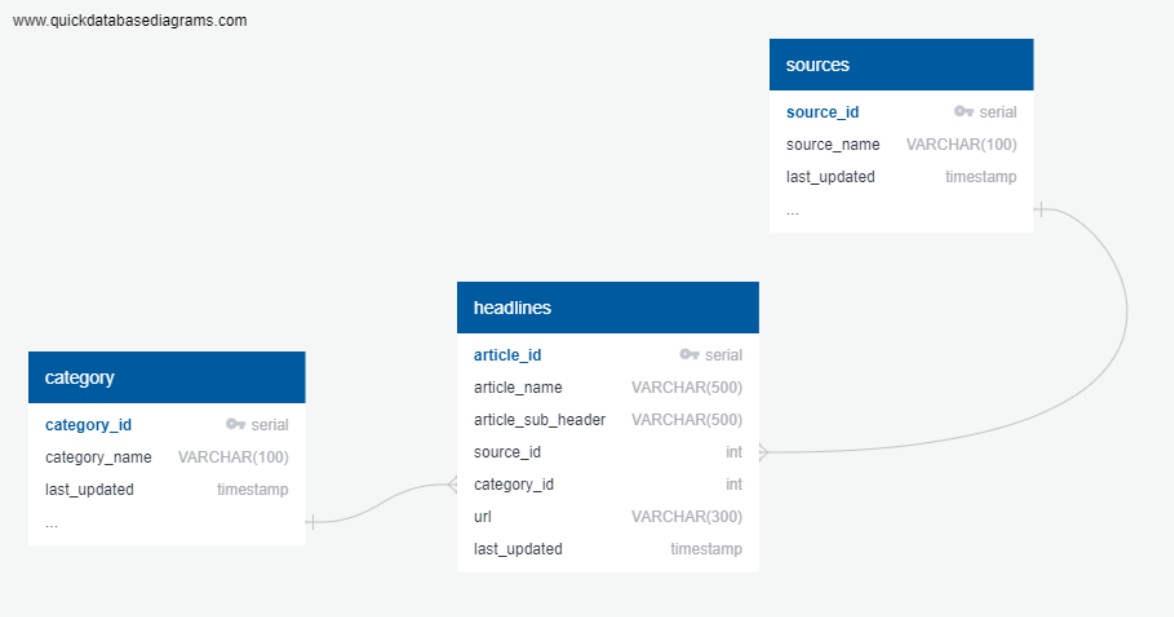

Figure 4: ERD Diagram

We then connected to our postgres database through pandas. The data was then loaded into Postgres using pandas “to_sql” code.

At this point, we could run queries in postgres to deliver the most relevant news articles, or search for a specific article faster than navigating through a website.

```
1  -- Bring back all columns
2  SELECT *
3  FROM headlines;
4
```

Data Output | Explain | Messages | Notifications

	article_id [PK] integer	article_name character varying (500)	article_sub_header character varying (500)	source_id integer	category_id integer	url character varying (300)	last_updated timestamp without time zone
1	1	The US is on the verge of the ...	ANALYSIS If you're a corrupt f...	1	1	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
2	2	Armed Anti-Lockdown Protest...	Right-wing protesters, includin...	2	2	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
3	3	As census deadline looms, ex...	The fate of this year's census ...	3	3	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
4	4	Why America can't rely solely ...	ANALYSIS The countries that ...	1	4	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
5	5	Trump Is Losing His Mind	OPINION The president is disc...	4	5	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
6	6	'A real mess': Trump is leaving...	When President-elect Joe Bid...	5	5	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
7	7	Do We Still Have To Wear Fac...	Seeing people wearing masks...	6	6	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543
8	8	Capitalism Delivered Promise,...	On Dec. 13, the Detroit Free Pr...	7	7	https://www.allsides.com/ne...	2020-12-22 01:33:46.139543

Figure 5: SQL Query #1

```
11  -- Join Category and Headline Table
12  SELECT
13      c.category_name,
14      COUNT(*) AS frequency
15  FROM
16      headlines h
17      JOIN category c
18      ON h.category_id = c.category_id
19  GROUP BY
20      c.category_name
21  ORDER BY
22      frequency DESC;
```

Data Output | Explain | Messages | Notifications

	category_name character varying (100)	frequency bigint
1	Politics	8
2	Business	6
3	World news	6
4	Coronavirus	5
5	Joe Biden	5
6	Elections	4
7	Opinion	4

Figure 6: SQL Query #2

```sql
-- Join all tables / Find specific articles
SELECT
    h.article_id,
    h.article_name,
    s.source_name,
    c.category_name
FROM
    headlines h
    JOIN category c ON h.category_id = c.category_id
    JOIN sources s ON h.source_id = s.source_id
WHERE
    s.source_name = 'The Guardian' AND c.category_name = 'Business';

```

Data Output | Explain | Messages | Notifications

	article_id integer	article_name character varying (500)	source_name character varying (100)	category_name character varying (100)
1	77	Reshape the economy for our ...	The Guardian	Business
2	79	Only state investment can revi...	The Guardian	Business
3	81	Why low inflation is a worryin...	The Guardian	Business
4	85	Families 'facing hardest perio...	The Guardian	Business

Figure 7: SQL Query #3

Works Cited

1. "Allsides | Balanced News Via Media Bias Ratings For An Unbiased News Perspective". *Allsides*, 2019, https://www.allsides.com/unbiased-balanced-news. Accessed 21 Dec 2020.

2. "Theguardian / Open Platform - Documentation / Overview". *Open-Platform.Theguardian.Com*, 2020, https://open-platform.theguardian.com/documentation/. Accessed 21 Dec 2020.

3. Ltd, Dovetail. "Quickdbd". *Quickdatabasediagrams*, 2020, https://app.quickdatabasediagrams.com/#/. Accessed 21 Dec 2020.