**News Headline ETL Project**

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

21 December 2020

Table of Contents

[OVERVIEW 3](#_Toc59524243)

[ETL PROCESS 3](#_Toc59524244)

[I. Extraction 3](#_Toc59524245)

[II. Transformation 4](#_Toc59524246)

[III. Load 5](#_Toc59524247)

[Works Cited 8](#_Toc59524248)

# 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

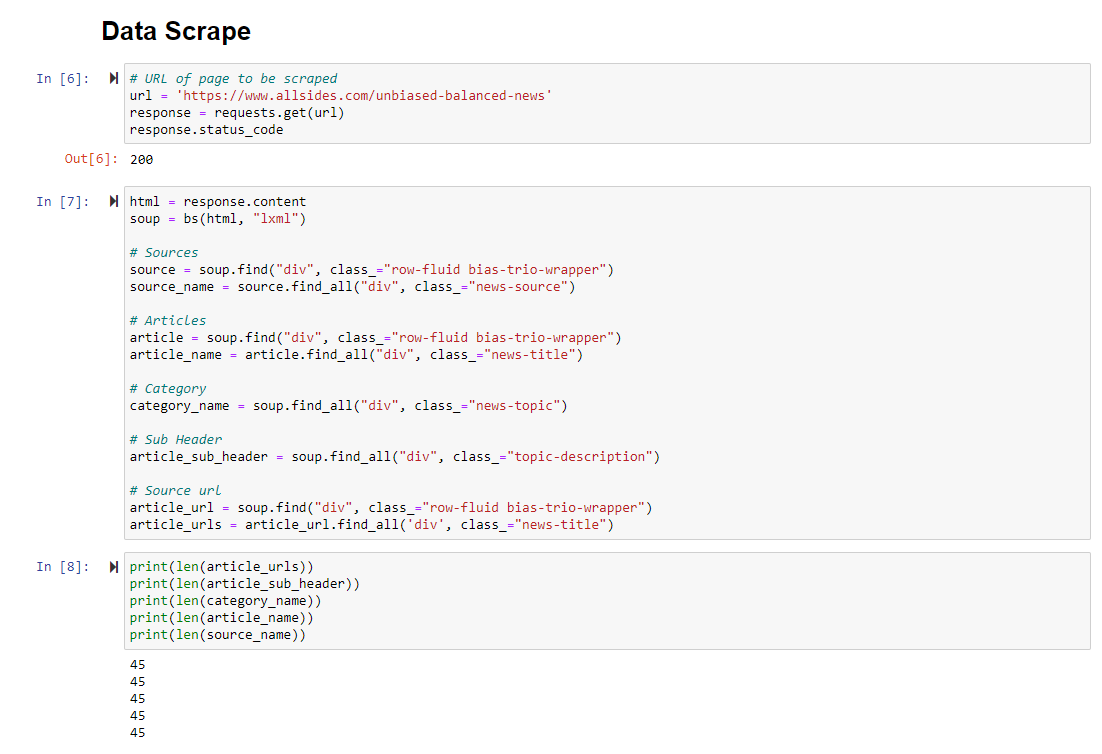


Figure 1: Data Scrape Code

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.

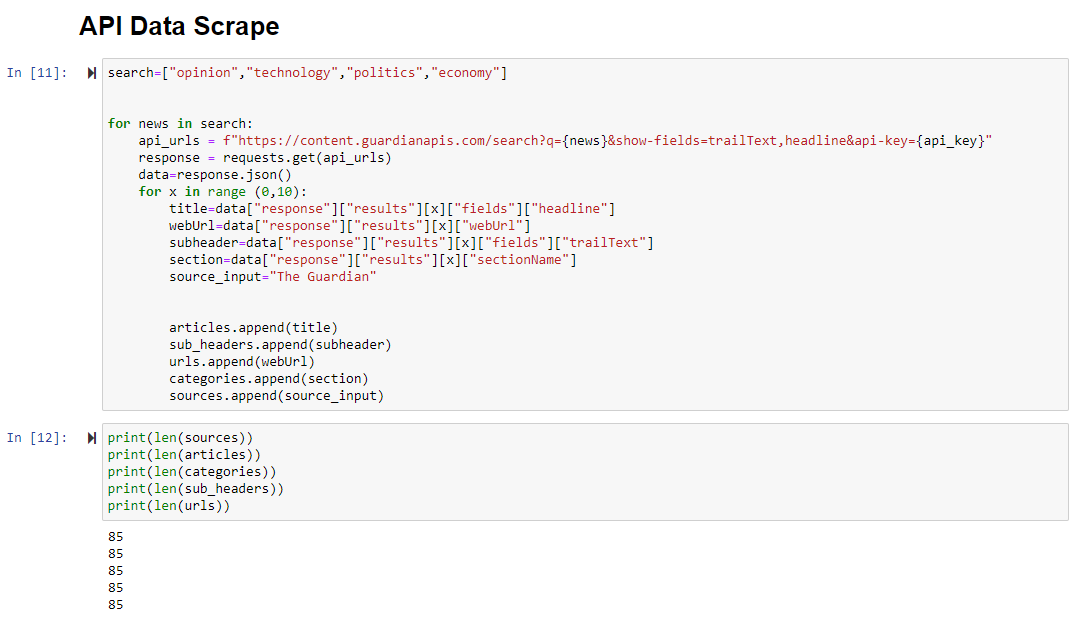


Figure 2: API Data Scrape Code

## II. Transformation

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

1. 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.
2. The Guardian API and web-scraped data were then joined together in a single dataframe. This was the “headlines” dataFrame.
3. 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.

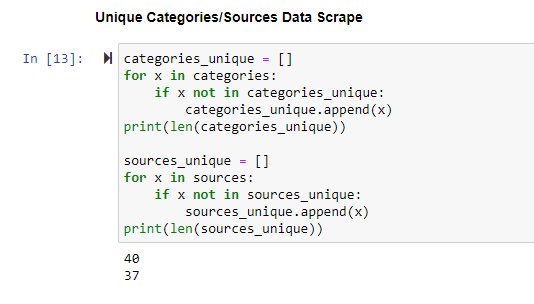


Figure 3: Unique Categories Assignment

## 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.

Graphical user interface, application

Description automatically generated

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.

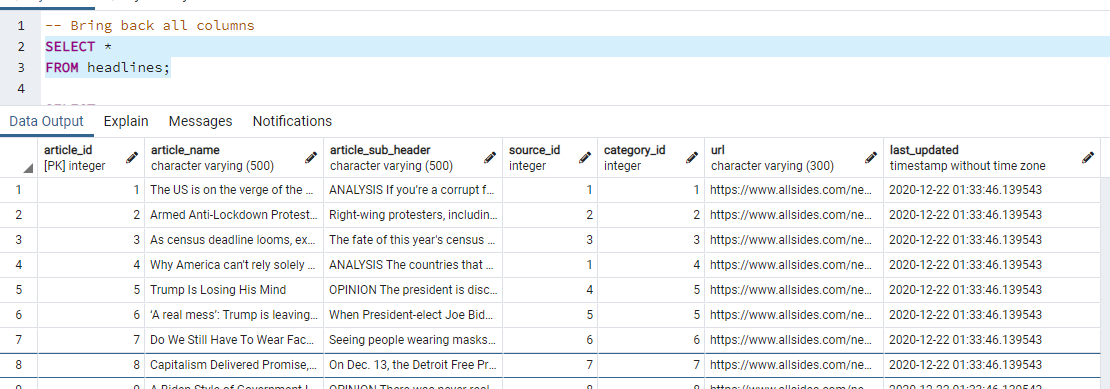


Figure 5: SQL Query #1

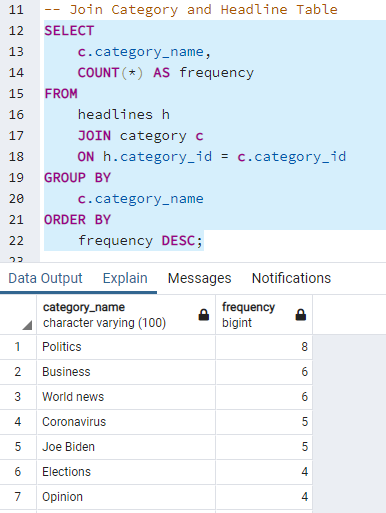


Figure 6: SQL Query #2

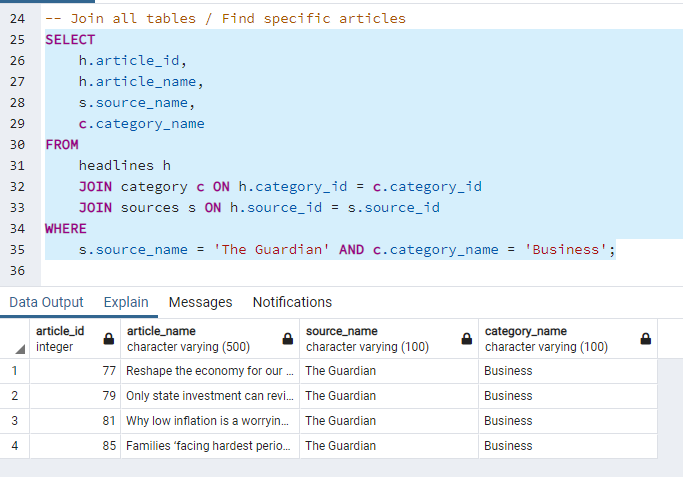


Figure 7: SQL Query #3

# 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/#/](https://app.quickdatabasediagrams.com/%23/). Accessed 21 Dec 2020.