DETECTING FAKES NEW WITH ML

SI 206 FINAL PROJECT

TEAM NEWSY

CHASE GOLDMAN <gochase@umich.edu>
GRANT HO <grantho@umich.edu>

GitHub Repository: https://github.com/grantyho/SI206-FINAL-PROJECT

GUIDING QUESTION:

People are bad at differentiating real news from fake news. Can we use machine learning to help curb the spread of misinformation and create a safer, more informed online community?

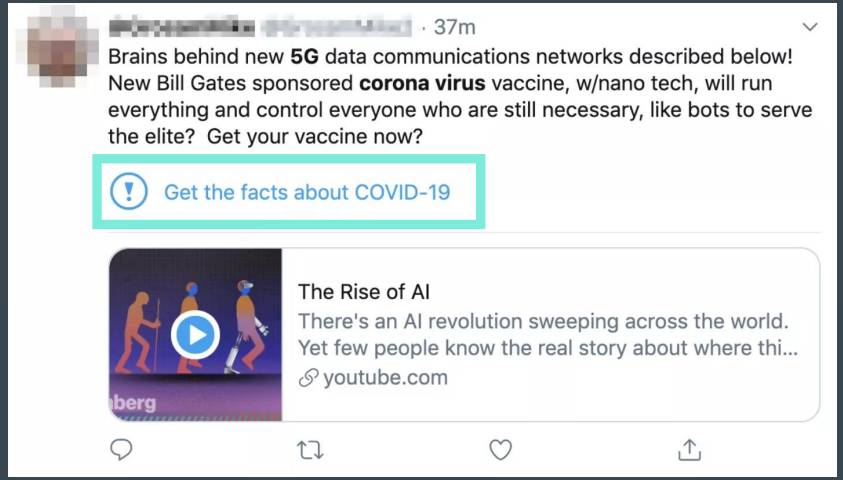
OUR INSPIRATION

Twitter to add labels and warning messages to disputed and misleading COVID-19 info

Sarah Perez @sarahintampa / 12:28 PM MDT • May 11, 2020 Comment

New labels for government and state-affiliated media accounts

By @TwitterSupport
Thursday, 6 August 2020



PROJECT GOALS

ORIGINAL GOALS

00 - API/WEB SCRAPING

Successfully integrating APIs to gather data (i.e. Tweets and Headlines from NYT, WSJ, CNN, etc.).

01 - NO STARTER CODE

Design an object oriented program without any starter code or direction. Building a project from scratch!

02 - DATABASES

Working with databases and designing databases to store and organize large amounts of information. No duplicate data!

03 - DATA VISUALIZATION

Creating data visualizations to convey information in concise and easy to understand fashion.

04 - OPEN SOURCE

Making use of online resources and the open source nature of programming by utilizing website like GeeksForGeeks and Stack Overflow.

05 - DEBUGGING

Take advantage of the debugging feature in VSCode to debug and ensure no bugs exist in code.

06 - ORGANIZATION/COMMUNICATION

Staying on top of deadlines, time management, and ensuring making sure to push to GitHub -- Communicating with partner effectively despite being virtual.

GOALS ACHIEVED?

00 - API/WEB SCRAPING

Successfully integrating APIs to gather data (i.e. Tweets and Headlines from NYT, WSJ, CNN, etc.).

01 - NO STARTER CODE

Design an object oriented program without any starter code or direction. Building a project from scratch!

02 - DATABASES

Working with databases and designing databases to store and organize large amounts of information. No duplicate data!

03 - DATA VISUALIZATION

Creating data visualizations to convey information in concise and easy to understand fashion.

04 - OPEN SOURCE

Making use of online resources and the open source nature of programming by utilizing website like GeeksForGeeks and Stack Overflow.

05 - DEBUGGING

Take advantage of the debugging feature in VSCode to debug and ensure no bugs exist in code.

06 - ORGANIZATION/COMMUNICATION

Staying on top of deadlines, time management, and ensuring making sure to push to GitHub -- Communicating with partner effectively despite being virtual.

API / SCRAPE/ DATABASE

API / WEB SCRAPING / LIBRARIES

API

- Twitter
  - Donald Trump
  - Kamala Harris
  - Joe Biden
  - Mike Pence
- News API
  - Varied sources
  - Short Articles
- The New York Times
  - 25 Articles / month from before / during Trump

WEB SCRAPING

- The New York Times
  - Given an article URL from the API, scrape article content
- The Wall Street Journal
  - Using selenium to login to account, click through archive, scrape article content

PYTHON LIBRARIES

- Pandas
  - Format Kaggle data to train ML model
- Beautiful Soup
  - Scrape content from NYT and WSJ
- Selenium
  - Automate clicking through Chrome to login to WSJ
- Plotly
  - Make visualizations using calculated data

GENERAL TABLES

SOURCES

	source_id	source_name
	Filter	Filter
1	0	CNN
2	1	The Washington Post
3	2	Sports Illustrated
4	3	POLITICO
5	4	The Hill
6	5	WFAA.com
7	6	CNBC
8	7	The New York Times

GENERAL TABLES

CALCULATION_TABLE

	source_id	article_id	ml_classification
	Filter	Filter	Filter
1	7	0	0
2	7	1	1
3	7	2	1
4	7	3	0
5	7	4	0
6	7	5	0
7	7	6	1
8	7	7	0
9	7	8	0
10	7	9	0
11	7	10	0
12	7	11	0
13	7	12	0
14	7	13	1

TWITTER TABLES

TWITTER_USERS

	UserId	Username
	Filter	Filter
1	0	JoeBiden
2	1	realDonaldTrump
3	2	KamalaHarris
4	3	Mike_Pence

TWITTER TABLES

TWITTER

	TweetId	SourceId	Tweet	Timestamp	TweetNum	UserId
	Filter	Filter	Filter	Filter	Filter	Filter
1	0	15	It's time we reward hard work in America — not just …	2020-12-04 01:39:00	1334673525655830528	0
2	1	15	On this International Day of People with Disabilities and …	2020-12-03 23:57:00	1334647856645500929	0
3	2	15	RT @Transition46: If we are going to tackle the climate …	2020-12-03 19:37:29	1334582546152235008	0
4	3	15	Once a vaccine is ready and approved, @KamalaHarris an…	2020-12-03 19:01:00	1334573366410264577	0
5	4	15	Yesterday, I spent the afternoon hearing from workers an…	2020-12-03 16:10:00	1334530332167729154	0
6	5	15	RT @Transition46: .@NeeraTanden understands the …	2020-12-03 01:20:32	1334306493328470017	0
7	6	15	With this team — and others who we'll add in the weeks …	2020-12-03 01:15:00	1334305098453815298	0
8	7	15	Let us begin the work to heal, unite, and rebuild an …	2020-12-02 22:55:00	1334269865947893760	0
9	8	15	We're facing the worst economic crisis since the Great …	2020-12-02 20:36:16	1334234952058068993	0
10	9	15	Congratulations to the newest United States Senator …	2020-12-02 19:12:00	1334213746013708293	0
11	10	15	From the most unequal economic and jobs crisis in mode…	2020-12-02 15:26:00	1334156871393968134	0
12	11	15	Today, I was proud to announce key nominations and …	2020-12-02 02:24:00	1333960074650218496	0
13	12	15	RT @Transition46: Statement by President-elect Biden on…	2020-12-02 02:12:54	1333957282502160384	0

NEWS_API TABLES

NEWS_API

	ArticleId	Title	Description	Timestamp	Url	SourceId
	Filter	Filter	Filter	Filter	Filter	Filter
1	0	Biden said he asked Fauci to stay on and be a chief medi...	President-elect Joe Biden said Thursday that he had aske...	2020-12-04T00:30:00Z	https://www.cnn.com/2020/12/03/politics/anthony-...	0
2	1	The White House liaison to the Justice Department was ...	The White House liaison to the Justice Department has ...	2020-12-04T00:29:00Z	https://www.cnn.com/2020/12/03/politics/heidi-stirru...	0
3	2	Momentum builds for bipartisan $908 billion stimulus ...	Congress faces increasing pressure to approve relief ami...	2020-12-04T00:22:41Z	https://www.washingtonpost.com/us-policy/2020/12/0...	1
4	3	Two Men and a Truck: The Last-Minute, 2,200-Mile ...	How do you get undefeated No. 13 BYU on the field agai...	2020-12-04T00:20:00Z	https://www.si.com/college/2020/12/03/byu-...	2
5	4	Trump mulls preemptive pardons for up to 20 allies, eve...	The clemency would be unprecedented, and some ...	2020-12-04T00:09:00Z	https://www.politico.com/news/2020/12/03/trump-...	3
6	5	DOJ sues Facebook, alleging it improperly hired foreign ...	The Justice Department on Thursday sued Facebook over...	2020-12-03T23:47:32Z	https://www.washingtonpost.com/technology/...	1
7	6	CNN Exclusive: Biden says he will ask Americans to wear ...	President-elect Joe Biden told CNN's Jake Tapper on ...	2020-12-03T23:45:00Z	https://www.cnn.com/2020/12/03/politics/biden-harris...	0

NEW YORK TIMES TABLES

NYT_SECTIONS

	section_id	section_name
	Filter	Filter
1	0	U.S.
2	1	New York
3	2	World
4	3	Science
5	4	Health

NEW YORK TIMES TABLES

NYT_URL_DATA

	source_id	article_id	url_extension	section_id	word_count	print_page	day	month	year
	Filter	Filter	Filter	Filter	Filter	Filter	Fil…	Filter	Filter
1	7	0	2015/01/01/nyregion/long-out-of-place-on-an-…	0	733	15	1	1	2015
2	7	1	2015/01/01/nyregion/exit-alarms-in-the-subways-are…	0	581	15	1	1	2015
3	7	2	2015/01/02/nyregion/top-2-thruway-authority-…	0	493	15	2	1	2015
4	7	3	2015/01/02/nyregion/mario-cuomo-new-york-governo…	0	4318	1	2	1	2015
5	7	4	2015/01/04/nyregion/a-review-of-the-hudson-room-i…	0	645	7	3	1	2015
6	7	5	2015/01/03/nyregion/rikers-inmate-found-dead-after-…	0	758	13	3	1	2015
7	7	6	2015/01/04/world/europe/ukraine-leader-was-defeate…	2	2576	1	4	1	2015
8	7	7	2015/01/04/nyregion/putting-life-on-pause-to-care-…	0	931	17	4	1	2015
9	7	8	2015/01/05/us/politics/steve-scalise-is-defined-and-…	1	1438	8	5	1	2015
10	7	9	2015/01/06/us/man-killed-in-dispute-with-san-…	1	515	16	6	1	2015
11	7	10	2015/01/06/us/politics/boehner-facing-dissent-and-…	1	1079	15	6	1	2015

NEW YORK TIMES TABLES

NYT_ARTICLE_CONTENT

	source_id	article_id	article_content
	Filter	Filter	Filter
1	7	0	It was an exercise in contrast, a tableau of multimillion-...
2	7	1	The aural offenses of the New York City subway are legio...
3	7	2	The top two executives of the New York State Thruway ...
4	7	3	Mario M. Cuomo, the three-term governor of New York ...
5	7	4	Does fried avocado sound like a good idea to you? It didn...
6	7	5	The warning was sent at 4:45 p.m. on New Year's Eve: A ...
7	7	6	KIEV, Ukraine — Ashen-faced after a sleepless night of ...
8	7	7	The person on the other end of the line that day last ...
9	7	8	WASHINGTON — Representative Steve Scalise of Louisian...
10	7	9	The San Francisco police shot and killed a man outside a ...
11	7	10	WASHINGTON — Speaker John A. Boehner is all but assur...

WALL STREET JOURNAL TABLES

WSJ_URL_DATA

	source_id	article_id	url_extension	day	month	year
	Filter	Filter	Filter	Fil…	Filter	Filter
1	16	0	trump-denounces-russia-bounty-intelligence-as-…	1	7	2020
2	16	1	seattle-police-dismantle-police-free-zone-11593616694	1	7	2020
3	16	2	faas-handling-of-boeing-737-max-issues-faulted-in-…	1	7	2020
4	16	3	the-coronavirus-credibility-gap-11593645643	1	7	2020
5	16	4	the-meaning-of-hong-kong-11593645526	1	7	2020
6	16	5	sheldon-whitehouses-favors-11593645437	1	7	2020
7	16	6	the-green-new-deal-in-action-11593645373	1	7	2020
8	16	7	hotel-san-francisco-11593645034	1	7	2020
9	16	8	u-s-warns-businesses-over-supply-chains-tied-to-…	1	7	2020
10	16	9	house-passes-extension-of-paycheck-protection-…	1	7	2020
11	16	10	building-collapses-in-brooklyn-11593643733	1	7	2020
12	16	11	the-trump-campaign-needs-to-hit-reset-11593642544	1	7	2020

WALL STREET JOURNAL TABLES

WSJ_ARTICLE_CONTENT

	source_id	article_id	article_content
	Filter	Filter	Filter
1	16	0	President Trump denounced as “a hoax” reports that …
2	16	1	Seattle police on Wednesday dismantled the self-styled …
3	16	2	Safety fixes after the first Boeing Co. 737 MAX crash …
4	16	3	Mandatory Credit: Photo by JIM LO SCALZO/EPA-EFE/…
5	16	4	Protesters making gestures of 1 and 5 fingers, symbolisi…
6	16	5	Senator Sheldon Whitehouse on June 11. PHOTO: POOL/…
7	16	6	Representative Peter DeFazio speaks at the unveiling of …
8	16	7	Pedestrians walk to the edge of the sidewalk to avoid …
9	16	8	WASHINGTON—The U.S. warned companies that have …

RUNNING THE CODE

INSTRUCTIONS FOR RUNNING CODE

I. GIT CLONE HTTPS://GITHUB.COM/GRANTYHO/SI206-FINAL-PROJECT.GIT

```
Chases-MacBook-Pro:final_project_demo chasegoldman$ git clone https://github.com/grantyho/SI206-FCCC
CChCCCCCChases-MacBook-Pro:final_project_demo chasegoldman$ git clone https://github.com/grantyho/SI
206-FINAL-PROJECT.git
Cloning into 'SI206-FINAL-PROJECT'...
remote: Enumerating objects: 257, done.
remote: Counting objects: 100% (257/257), done.
remote: Compressing objects: 100% (181/181), done.
remote: Total 296 (delta 129), reused 190 (delta 66), pack-reused 39
Receiving objects: 100% (296/296), 49.17 MiB | 3.29 MiB/s, done.
Resolving deltas: 100% (140/140), done.
Chases-MacBook-Pro:final_project_demo chasegoldman$
```

INSTRUCTIONS FOR RUNNING CODE CONTINUED

2A. NAVIGATE TO THE SI206-FINAL-PROJECT DIRECTORY

```
Chases-MacBook-Pro:final_project_demo chasegoldman$ ls
SI206-FINAL-PROJECT
Chases-MacBook-Pro:final_project_demo chasegoldman$ cd SI206-FINAL-PROJECT/
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ ls
Kaggle                  chromedriver_2          news_api.py             twitter.py
README.md               database.py             nytimes.py              visualizations.py
calculations.py         ml.py                   practice.sh             wsj.py
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$
```

2B. INSTALL ALL PROJECT DEPENDENCIES

1. pip3 install requests
2. pip3 install pandas
3. pip3 install tweepy
4. pip3 install selenium
5. pip3 install plotly
6. pip3 install beautifulsoup4
7. pip3 install scikit_learn

INSTRUCTIONS FOR RUNNING CODE CONTINUED

3. RUN THE WALL STREET JOURNAL SCRIPT: PYTHON WSJ.PY --MONTH 7 --DAY 1

- This command will pull 25 wsj articles for july 1 2015.
- Acceptable months are integers from 7 - 12 (inclusive)
- Acceptable days are integers 1 - 29 (inclusive)
- Example: to get 25 articles from October 10, 2015, use: python wsj.py --month 10 --day 10

RUN 1: PULLS 25 WSJ ARTICLES FOR JULY 1, 2015

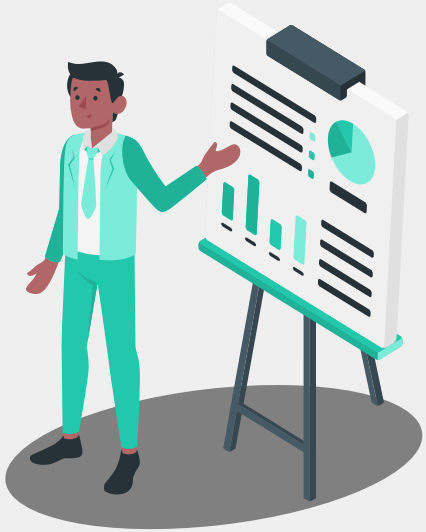
```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python wsj.py --month 7 --day 1
```

RUN 2: PULLS 25 WSJ ARTICLES FOR JULY 2, 2015

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python wsj.py --month 7 --day 1
```

RUN 3: PULLS 25 WSJ ARTICLES FOR JULY 3, 2015

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python wsj.py --month 7 --day 2
```



INSTRUCTIONS FOR RUNNING CODE CONTINUED

4. RUN THE NEW YORK TIMES SCRIPT: PYTHON NYTIMES.PY --NUMRUNS 0

- This command will pull 25 nyt articles from january 2015
- Acceptable integers for numruns are between 0 and 23 (inclusive)
- Example: to get 25 articles from february 2015, use: python nytimes.py --numruns 1
- NOTE THE CASING OF NUMRUNS IN THE TERMINAL SNIPPETS

RUN 1: PULLS 25 NYT ARTICLES FOR JANUARY 2015

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python nytimes.py --numRuns 0
```

RUN 2: PULLS 25 NYT ARTICLES FOR FEBRUARY 2015

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ nytimes.py --numRuns 1
```

RUN 3: PULLS 25 NYT ARTICLES FOR MARCH 2015

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ nytimes.py --numRuns 2
```

INSTRUCTIONS FOR RUNNING CODE CONTINUED

5. RUN THE NEWS_API SCRIPT: PYTHON NEWS_API.PY

- This command will pull the 20 most recently published articles from news API
- NOTE THIS API ONLY PROVIDES 20 HEADLINES PER APPROXIMATELY EVERY HOUR

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python news_api.py
Added 20 new headlines to News_API table!
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$
```

INSTRUCTIONS FOR RUNNING CODE CONTINUED

6. RUN THE TWITTER SCRIPT USING: PYTHON TWITTER.PY

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python twitter.py --userName JoeBiden
```

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python twitter.py --userName realDonaldTrump
```

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python twitter.py --userName KamalaHarris
```

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python twitter.py --userName Mike_Pence
```

INSTRUCTIONS FOR RUNNING CODE CONTINUED

7. RUN THE ML SCRIPT USING: PYTHON DATABASE.PY

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python database.py
```

INSTRUCTIONS FOR RUNNING CODE CONTINUED

8. MAKE THE VISUALIZATIONS USING: PYTHON VISUALIZATIONS.PY

```
[Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
 /Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/final_project_demo/SI206-FINAL-PROJECT
[Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ python visualizations.py
```

INSTRUCTIONS FOR RUNNING CODE CONTINUED

OPTIONAL: IF YOU WANT TO SKIP STEPS 3 - 8, YOU CAN USE OUR SHELL SCRIPT: BASH PRACTICE.SH

- Disclaimer: This will take roughly 24 hours to finish running!

```
# Wall Street Journal
for i in {7..11}
do
    for j in {1..29}
    do
        python3 wsj.py --month $i --day $j
        read -t 10
    done
done

# New York Times
for i in {0..23}
do
    python3 nytimes.py --numRuns $i
    read -t 10
done
```

```
# News API
for i in {0..5}
do
    python3 nytimes.py --numRuns $i
    read -t 3600
done

# Twitter
# add any valid twitter username if you want!
python twitter.py --userName JoeBiden
python twitter.py --userName realDonaldTrump
python twitter.py --userName KamalaHarris
python twitter.py --userName Mike_Pence

# ML Calculation
python3 database.py

# Visualizations and Calculation File
python3 visualizations.py
```

```
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ pwd
/Users/chasegoldman/Desktop/Michigan/Fall2020/SI206/SI206-FINAL-PROJECT
Chases-MacBook-Pro:SI206-FINAL-PROJECT chasegoldman$ bash practice.sh
```

DOCUMENTATION

DOCUMENTATION

NEWS_API.PY DOCUMENTATION

```
# Uses the NEWS API to retrieve live articles from all over the web.
# Specifically, this function gathers the Source, Title, Description, Timestamp, and URL of the article.
# newsApiData() returns a list containing ['Source', 'Title', 'Decription', 'Timestamp', 'URL']
def newsApiData()
```

```
# Uploads data retrieved from the NEW API to 'News_API' table
# newsApiTable() has data, a list, as a parameter, which is the data returned from newsApiData()
# and curr + conn (for connecting to database)
def newsApiTable(data, cur, conn)
```

```
# Connects to database and inserts data into 'News_API' table
def fillAllNewsApiTables()
```

TWITTER.PY DOCUMENTATION

```
# Uses the TWITTER API (tweepy) to retrieve Tweets from a specified user.
# This function gathers a unique Tweet ID, the Tweet text, and Timestamp of the Tweet
# twitterData() takes a username as a parameter (i.e. realDonaldTrump)
# twitterData() returns a list containing ['tweetId', 'tweetText', 'tweetTimestamp']
def twitterData(user)
```

```
# Creates table and uploads data to table called 'twitter_users' with UserId and Username as columns
# twitterUsersTable() has a list of Twitter usernams as a parameter and curr + conn (for connecting to database)
def twitterUsersTable(usernames, cur, conn)
```

```
# Creates table and adds data scraped from TWITTER API to 'twitter'
# twitterTable() has a list of Twitter usernames as a parameter and curr + conn (for connecting to database)
# The columns in the database are as follows: TweetId, Tweet, Timestamp, TweetNum, UserId
def twitterTable(usernames, cur, conn)
```

```
# Connects to database and inserts data into 'Twitter' table
def fillAllTwitterTables()
```

NYT.PY DOCUMENTATION

```
# This function takes as input an article's publication date in the format
# returned by the NYTimes API, and then parses that date into a more
# readable format to be used later in the script. Returns a tuple which
# consists of an integer day, month, and year for this publication.
def parsePubDate(pubDate):
```

DOCUMENTATION CONTINUED

```
# This function takes as inputs cur and conn so we can successfully
# connect to the database. Additionally, it takes in a list of
# NYT sections that we have gotten from the NYT API. Then, it calculates
# all of the sections that are already in the table, and finds any new sections
# that we want to add to the table because they aren't in there yet. It returns
# this new list of sections to be added to the database.
def getNewSections(cur, conn, list_sections):
```

```
# This function takes in cur and conn so we can successfully connect to
# the database. Additionally, it takes in a list of NYT sections to be
# added to the database and an integer representing the number of times
# this script has been run, parsed using the argparse library. It does
# not return anything but instead adds data into the NYT_Sections table in
# the database.
def fillNYT_Sections_Table(cur, conn, list_sections, runIteration):
```

NYT.PY DOCUMENTATION (CONT)

```
# This function takes as inputs a connection to the database along with
# a dictionary containing data about NYT articles pulled from the NYT. Additionally,
# it takes an integer representing the number of times this script has been run,
# parsed from the command line using argparse. It does
# not return anything but instead adds data into the NYT_URL_Data table in
# the database from the data_dictionary it recieves as input. Returns None.
def fillNYT_URL_Data_Table(cur, conn, data_dictionary, runIteration):
```

DOCUMENTATION CONTINUED

```
# This function takes a connection to the database and an integer
# representing the number of times this script has been run,
# parsed from the command line using argparse. It pulls all of the
# article_urls from NYT_URL_Data and uses BS to navigate to that URL
# and scrape the article content from the page. It then adds that
# article content to a table in the database. Returns None.
def fillNYTimes_ArticleContent_Table(cur, conn, runIteration):
```

```
# This is the function that gets data from the NYT API. It takes an integer
# as input representing the number of times the script has been run, parsed from the
# command line. Depending on how many times the script has been run, it pulls data about
# 25 articles from this month down from the API. Converts this JSON data to a python dict
# and returns the dictionary.
def getNYTURLDictionary(runIteration):
```

DOCUMENTATION CONTINUED

NYT.PY DOCUMENTATION (CONT)

```
# This is the driver function for the NYT section of the database, taking as input
# a connection to the database and an integer for the amount of times the script has
# been run. It calls all of the functions outlined above to make sure we collect
# and store all of the correct data in the database. Returns None.
def fillAllNYT_Tables(cur, conn, runIteration):
```

```
# This is the primary drivary for the NYT database. Gets the runIteration
# from the command line and calls all of the functions according to the
# number of times that the script has been run already. Returns None.
def driveNYT_db(runIteration):
```

WSJ.PY DOCUMENTATION

```
# Initializes this instance of Chromedriver. Takes in the path to the chromedriver
# as well as an option to make the driver headless or not. If headless, the instance
# of chrome will not physically appear on your computer, similar to the way BeautifulSoup
# operates. If not headless, the chrome window will pop up on your computer and user will
# be able to see the script moving around the page and typing things. Returns a chromedriver instance.
def getChromeDriver(path, headless = False):
```

DOCUMENTATION CONTINUED

```
# Given a chromedriver, navigates to the Wall Street Journal's login page and
# signs in with the email and password parameters. Need to sign in so we can
# access the full article content. Returns None.
def loginWSJ(driver, base_url, email, password):
```

```
# Takes in a connection to the database as input, along with a chromedriver instance
# and an integer for the month and day we are trying to scrape.
# Creates a new table in the database to hold data about each of the WSJ articles
# from July 2020 to November 2020 we will eventually be scraping. Table has columns source_id
# (unique integer for the WSJ), article_id (unique integer for each WSJ article), url_extension
# (url to get to each article), day, month, and year that the article was published.
# Fills this table with five articles from each day in 7/2020 through 11/2020. Returns None.
def fillWSJ_URL_Table(cur, conn, driver, month, day):
```

DOCUMENTATION CONTINUED

WSJ.PY DOCUMENTATION (CONT)

```
# Takes in a connection to the database as input, along with a chromedriver instance
# and an integer for the month and day we are trying to scrape.
# Creates a second table to store the actual content of each WSJ article.
# Fetches the article_id and url_extension from WSJ_URL_Data table. Uses that
# url to go to the article's link, scrape the article, and fill a new table in the
# database which contains the article_id as well as the article_content. Returns None, modifies the datbase.
def fillWSJArticleContentTable(cur, conn, driver, month, day):
```

```
# Drives all functions pertaining to the Wall Street Journal. Takes in a month and day
# as an integer so it knows what data to get using selenium.
def driveWSJ_db(month, day):
```

DATABASE.PY DOCUMENTATION

```
# Connects this file to the database
# setUpDataBase() takes db_name, a string, as a parameter
# creating a database with db_name
def setUpDatabase(db_name)
```

```
# Retrieves the source ID from the 'Sources' table in the database and
# checks if Source exists in table already
# setSourceID() takes curr + conn (for connecting to database)
# and source_name, a string, the name of a table in a database
# returns the source_id associated with that source
def getSourceID(cur, conn, source_name)
```

```
# Retrieves the highest ID for a respective table to ensure ID is unique
# getHighestID() takes curr + conn (for connecting to database) as a parameter
# and takes column_name, a string, (of a table) and table_name, a string
# returns the highest ID for a column
def getHighestId(cur, conn, column_name, table_name)
```

```
# Creates 'Calculation_Table'
# Grabs source_id, article/tweet_id, and content from columns in NYT, Twitter, New API, and WSJ table
# Inputs data into the 'Calculation_Table' from tables and machine learning calculation based on content
# takes curr + conn (for connecting to database) as a parameter
# The columns in the table are as follows: source_id, article_id, ml_classification
def compileCalculationTable(cur, conn)
```

```
# Connects to finalProject.db and inserts data into 'Calculation_Table' table
def main()
```

```
# classifier() takes text, a list of strings, as a parameter
# This function classifies text as 'Fake News' or 'True News'
# Return 0 for Fake News and 1 for True News
def classifier(text)
```

DOCUMENTATION CONTINUED

VISUALIZATION.PY DOCUMENTATION

```
# Connects to database to pull data from tables in database
# Has db_name as a parameter, a string, for the database file name
# returns cur and conn
def setUpDatabase(db_name)
```

```
# Utilizing the plotly library, this function creates a pie chart
# with the percentage of Fake News and True News
# Takes a dictionary as a paramater
def visualizations(dictionary)
```

```
# Compiles data from the 'Calculation_Table' into a dictionary which is returned:
    # {'Fake News': #, 'True News': #}
# Takes curr + conn (for connecting to database)
def mlClassificationData(cur, conn)
```

```
# Compiles data from the 'Calculation_Table' where source is from Twitter into a dictionary which is returned:
    # {'Fake News': #, 'True News': #}
# Takes curr + conn (for connecting to database)
def mlClassificationTwitterData(cur, conn)
```

```
# Compiles data from the 'Calculation_Table' where source is from NYT into a dictionary which is returned:
    # {'Fake News': #, 'True News': #}
# Takes curr + conn (for connecting to database)
```

```
# Compiles data from the 'Calculation_Table' where source is from WSJ into a dictionary which is returned:
    # {'Fake News': #, 'True News': #}
# Takes curr + conn (for connecting to database)
def mlClassificationWSJData(cur, conn)
```

VISUALIZATIONS.PY DOCUMENTATION (CONT)

```
# This function takes in a connection to the database. It then uses a JOIN to create a new table for NYT
# articles and pulls the section_name and page that each article was printed on. For each unique section
# in our database table, it calculates the average page that an article of that section is printed on.
# The function returns a dictionary containing this information.
def calculateNYTPrintPageAvg(cur, conn):
```

```
# This function takes as input the data calculated in calculateNYTPrintPageAvg() and creates a
# bar chart using plotly.
def visualizeNYTPrintPageAvg(section_average_dict):
```

```
# This function takes a connection to the database as input. For each of the 4 tables in the database
# containing article content, it does a JOIN with the sources table to get the source name and then adds
# one to an accumulator for that source, tracking the total number of articles in the database for each source.
# It then returns a dictionary containing all sources that have more than 2 articles.
def countNumArticlesPerSource(cur, conn):
```

```
# Takes as input a dictionary calculated in countNumArticlesPerSource() and puts all of this
# information into a bar chart using plotly
def visualizeNumArticlesPerSource(source_count_dict):
```

```
# This function takes as input a connection to the database. For each headline URL in the WSJ table
# it checks wether the string 'trump' was in that headline. It returns a dictionary with two keys, one is
# Trump Headlines and the other is Non-Trump Headlines, where each key has an associated value which
# is a count.
def countPercentageTrumpWSJHeadlines(cur, conn):
```

```
# Takes an output file name as input along with a connection to the database.
# Writes all of the calculations neatly formatted into the output_file.txt
def writeCalculations(output_file, cur, conn):
```

DOCUMENTATION CONTINUED

CALCULATIONS AND VISUALIZATIONS

```python
# TRAINING OUR MODEL
df = pd.read_csv('cleaned_kaggle_news.csv')

# Split the data
DV = 'fake_news' # The dependent variable, text is the independent variable here

X = df.drop([DV], axis = 1) # Drop from our X array because this is the text data that gets trained
y = df[DV]

# Training on 75% of the data, test on the rest
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.25)

count_vect = CountVectorizer(max_features = 10000) # limiting to 5000, but room to play with this here!
X_train_counts = count_vect.fit_transform(X_train['text'])
# print(count_vect.vocabulary_) # here is our bag of words!
X_test = count_vect.transform(X_test['text']) # note: we don't fit it to the model! Or else this is all useless


# Fit the training dataset on the NB classifier
Naive = MultinomialNB()
Naive.fit(X_train_counts, y_train)


# Predict the labels on validation dataset
predictions_NB = Naive.predict(X_test)

# classifier() takes text, a list of strings, as a parameter
# This function classifies text as 'Fake News' or 'True News'
# Return 0 for Fake News and 1 for True News
def classifier(text):
    Naive = MultinomialNB()
    Naive.fit(X_train_counts, y_train)

    word_vec = count_vect.transform(text)

    predict = Naive.predict(word_vec)
    return 0 if predict[0] else 1
```

CALCULATIONS

FUNCTION + TRAINING TO DO MACHINE LEARNING CALCULATIONS (located in database.py)

source_id	article_id	ml_classification
Filter	Filter	Filter
0	0	0
0	1	1
0	2	1
0	3	0
0	4	0
0	5	0
0	6	0
0	7	0
0	8	0
0	9	0
0	10	0
0	11	0
0	12	0
0	13	1
0	14	1
0	15	0
0	17	1
0	18	1
0	19	0
0	20	1
0	21	0
0	23	0
0	24	0
0	25	1
0	26	0
0	27	1
0	28	0
0	30	0
0	31	1
0	32	0

source_id	article_id	ml_classification
Filter	Filter	Filter
0	33	1
0	34	0
0	35	1
0	36	0
0	37	1
0	38	1
0	39	1
0	40	1
0	41	0
0	42	0
0	43	1
0	44	0
0	45	1
0	46	1
0	47	1
0	48	0
0	49	0
0	50	1
0	51	1
0	52	0
0	53	0
0	54	1
0	55	1
0	56	0
0	57	1
0	58	0
0	59	1
0	60	0
0	61	1
0	62	1

...

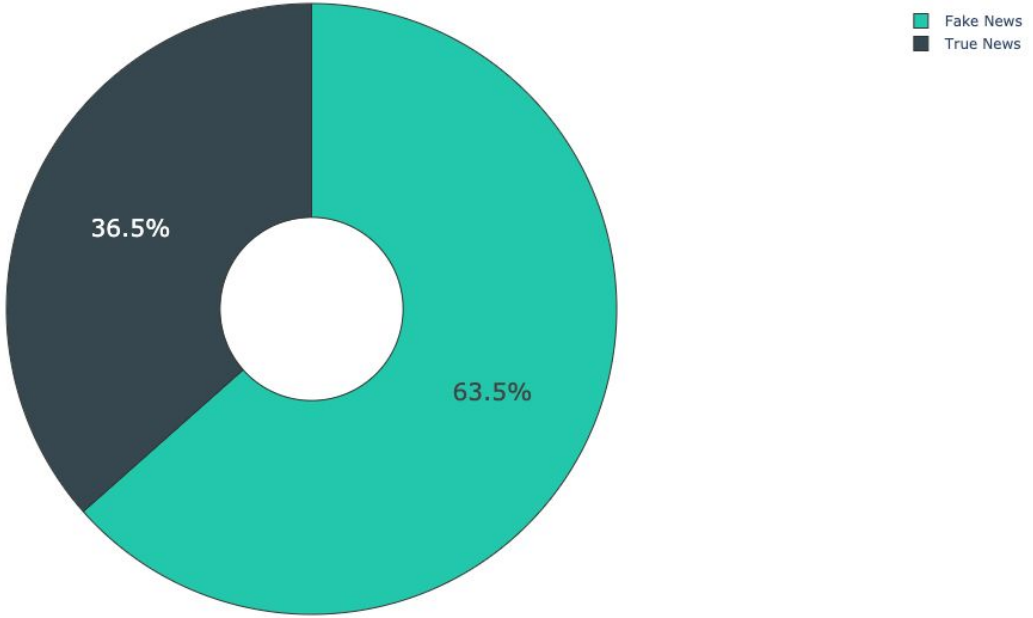
source_id	article_id	ml_classification
Filter	Filter	Filter
1	65	1
1	66	1
1	67	1
1	68	1
1	69	1
1	70	1
1	71	0
1	72	1
1	73	0
1	74	0
1	75	1
1	76	0
1	78	1
1	79	0
1	80	1
1	81	0
1	82	1
1	83	0
1	84	0
1	85	0
1	86	0
1	87	0
1	88	1
1	89	0
1	90	0
1	91	1
1	92	1
1	93	1
1	94	1
1	95	1

CALCULATIONS CONTINUED

OUTPUT OF CALCULATION TO ‘CALCULATION_TABLE’ IN DATABASE

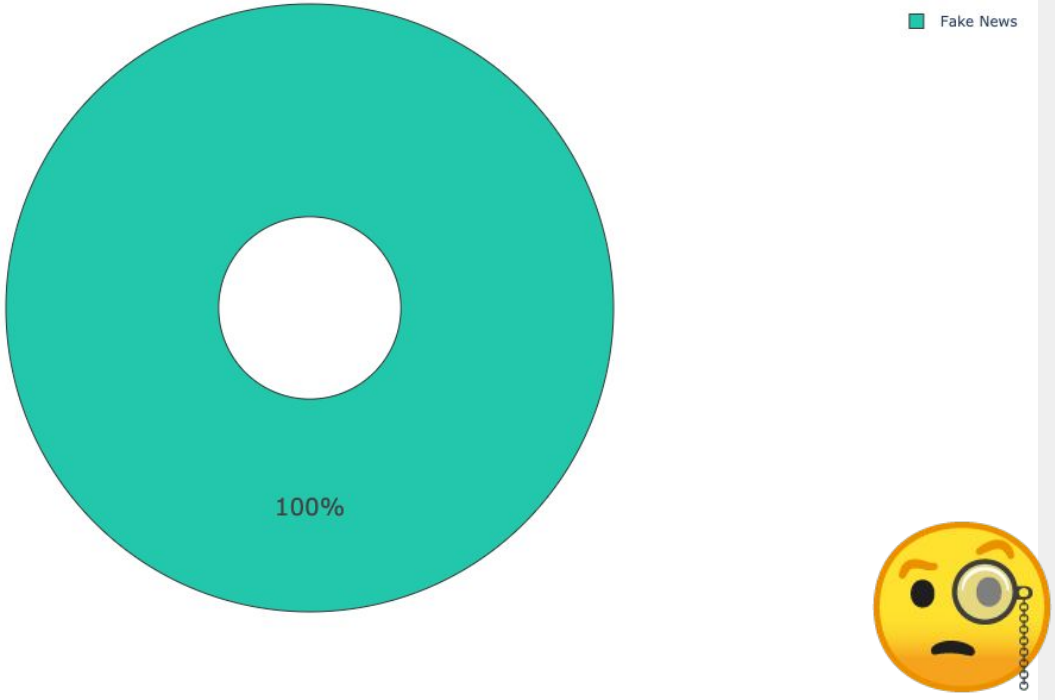
VISUALIZATIONS / CALCULATIONS

Fake News vs. True News for All Data Collected (12/4 - 12:47 PM EST)



VISUALIZATIONS / CALCULATIONS

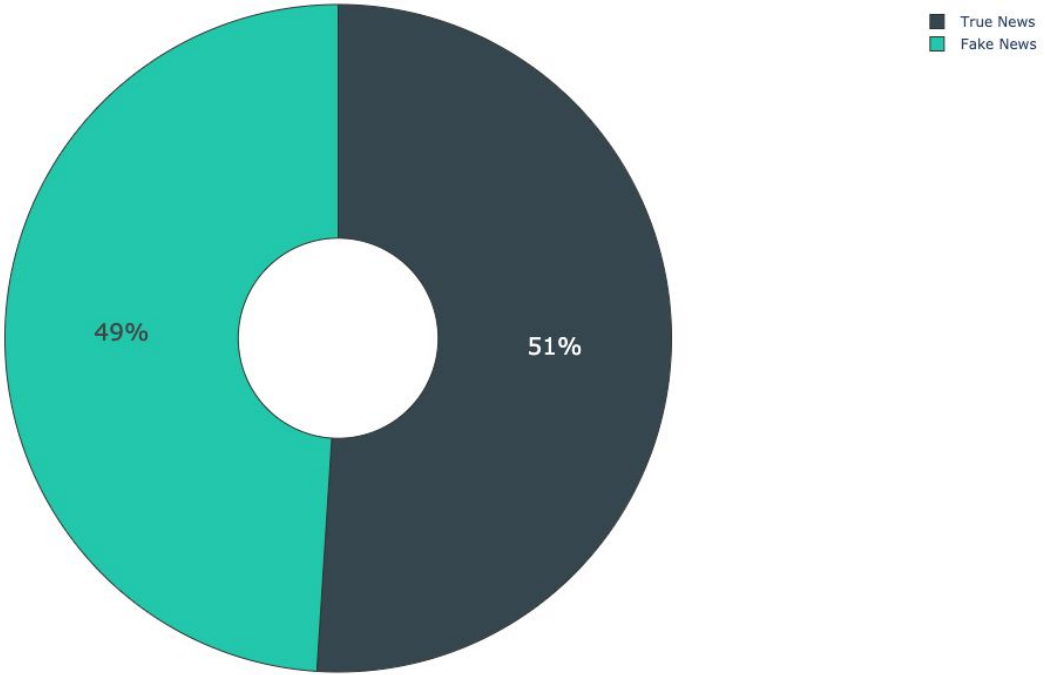
Fake News vs. True News for Twitter Data Collected (12/4 - 1:10 PM EST)



****Twitter data collected from @JoeBiden, @KamalaHarris, @realDonaldTrump & @Mike_Pence*

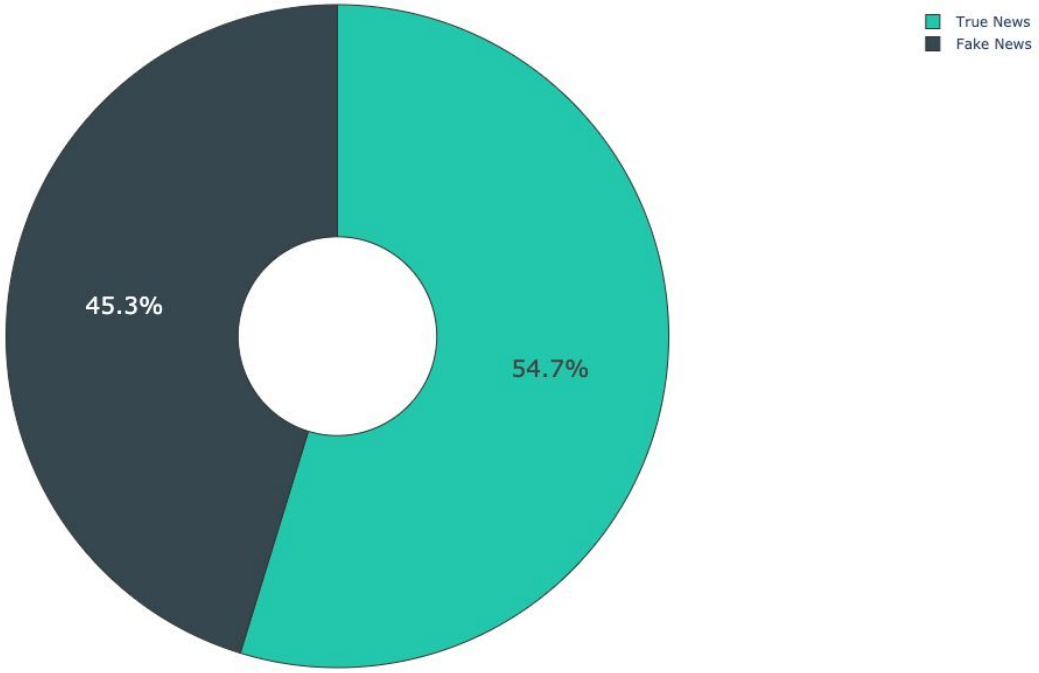
VISUALIZATIONS / CALCULATIONS

Fake News vs. True News for New York Times Data Collected (12/4 - 1:34 PM EST)

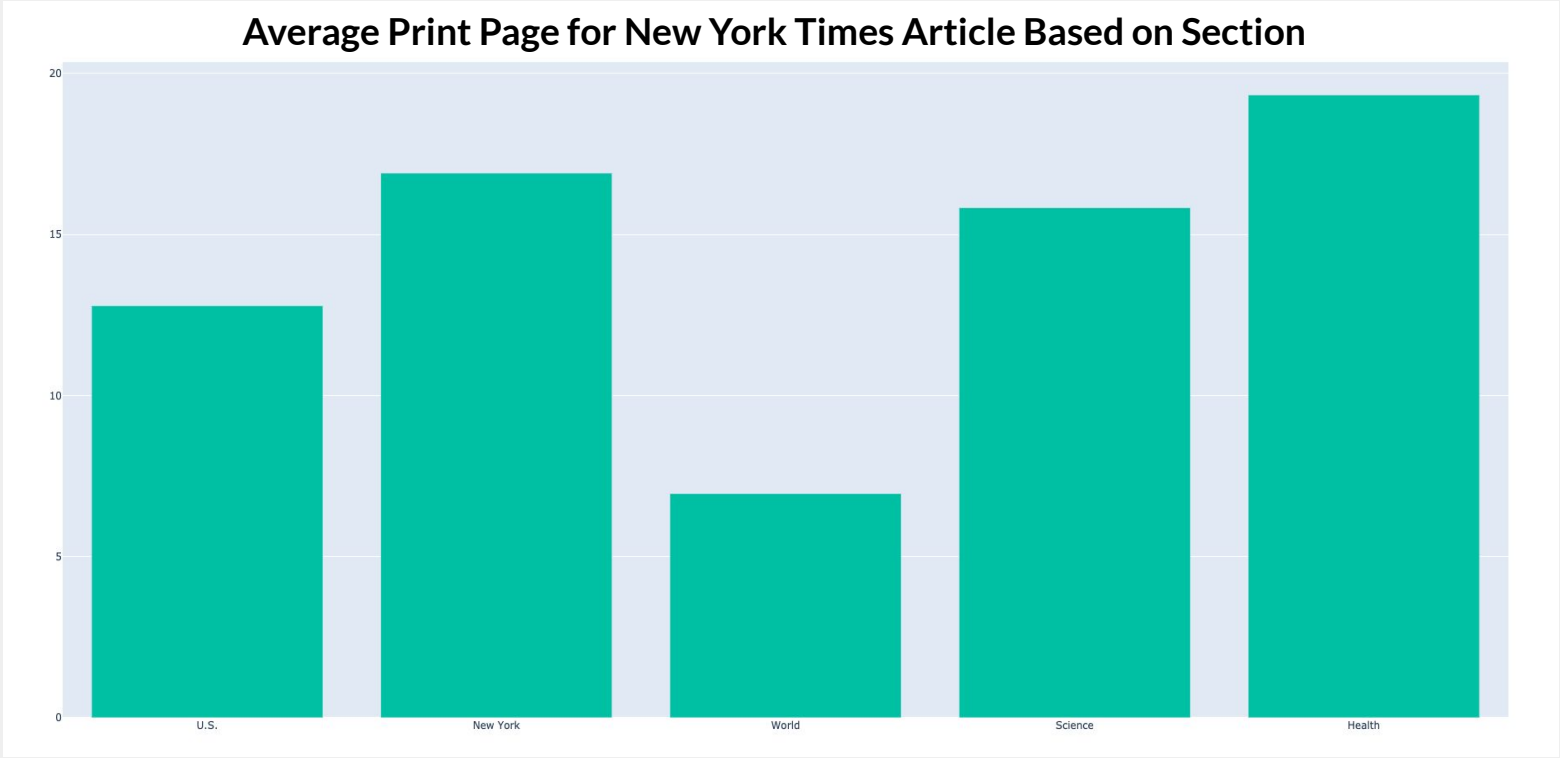


VISUALIZATIONS / CALCULATIONS

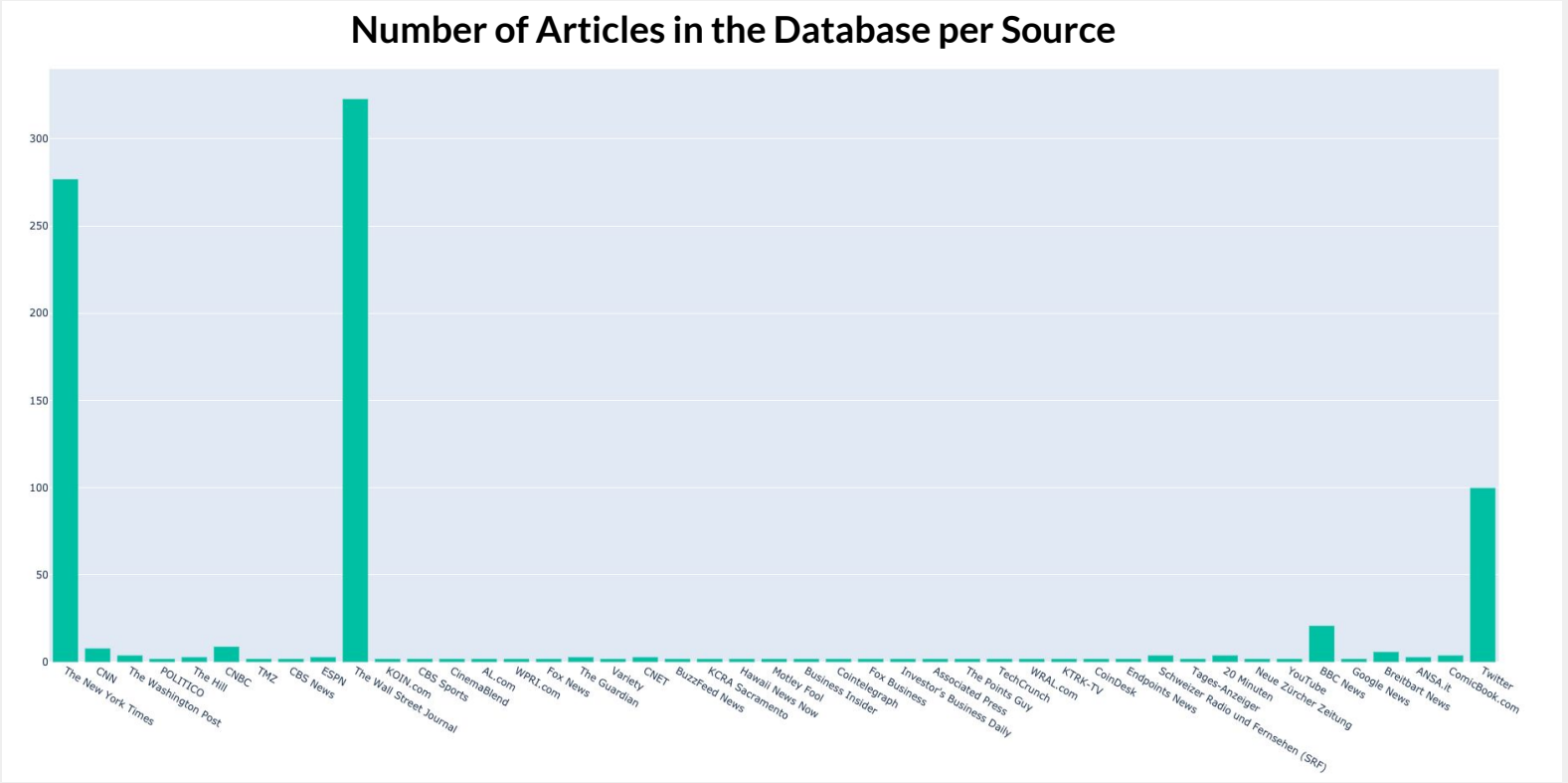
Fake News vs. True News for Wall Street Journal Data Collected (12/4 - 1:53 PM EST)



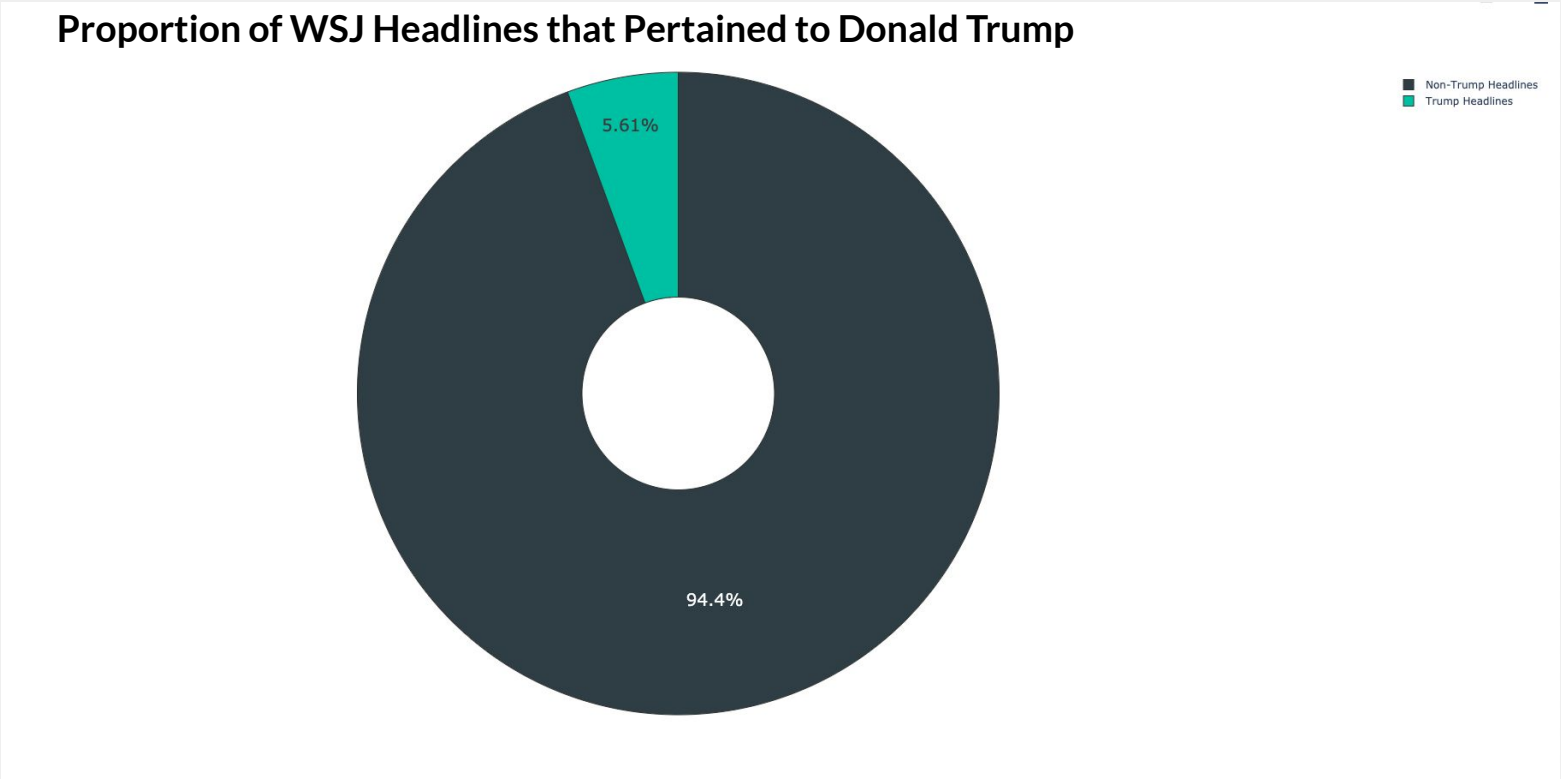
VISUALIZATIONS / CALCULATIONS



VISUALIZATIONS / CALCULATIONS



VISUALIZATIONS / CALCULATIONS



CALCULATION OUTPUT FILE PREVIEW

```
# Takes an output file name as input along with a connection to the database.
# Writes all of the calculations neatly formatted into the output_file.txt
def writeCalculations(output_file, cur, conn):
```

```
FINAL PROJECT CALCULATION
Grant Ho and Chase Goldman



CALCULATION 1: For each article in our database, is it real news or fake news?

FAKE NEWS / REAL NEWS CLASSIFICATION FOR ALL ARTICLES:
CNN Article Number: 0
    Classification: REAL NEWS

CNN Article Number: 1
    Classification: REAL NEWS

CNN Article Number: 6
    Classification: FAKE NEWS

CNN Article Number: 18
    Classification: REAL NEWS
```

GitHub Link: https://github.com/grantyho/SI206-FINAL-PROJECT/blob/main/finalOutput.txt

CALCULATION OUTPUT FILE PREVIEW (CONT)

```
CALCULATION 2: On average, what page of the newspaper was each article from the New York Times sections in our database printed on?

On average, a New York Times article belonging to the U.S. Section was printed on page 12.787878787878787 of the newspaper.
On average, a New York Times article belonging to the New York Section was printed on page 16.907894736842106 of the newspaper.
On average, a New York Times article belonging to the World Section was printed on page 6.956043956043956 of the newspaper.
On average, a New York Times article belonging to the Science Section was printed on page 15.833333333333334 of the newspaper.
On average, a New York Times article belonging to the Health Section was printed on page 19.333333333333332 of the newspaper.
```

```
CALCULATION 3: For each source in the database with more than 2 articles, how many articles did we have from that source?

Source Name: The New York Times
Number of articles from The New York Times: 277

Source Name: CNN
Number of articles from CNN: 8

Source Name: The Washington Post
Number of articles from The Washington Post: 4

Source Name: POLITICO
Number of articles from POLITICO: 2

Source Name: The Hill
Number of articles from The Hill: 3

Source Name: CNBC
Number of articles from CNBC: 9
```

GitHub Link: https://github.com/grantyho/SI206-FINAL-PROJECT/blob/main/finalOutput.txt

CALCULATION OUTPUT FILE PREVIEW (CONT)

```
CALCULATION 4: What percentage of Wall Street Journal Headlines involved Donald Trump?

Wall Street Journal Article URL: 10th-house-lawmaker-tests-positive-for-covid-19-11596306537
    Does the URL headline include Trump: NO

Wall Street Journal Article URL: a-first-step-toward-loving-our-enemies-11599174418
    Does the URL headline include Trump: NO

Wall Street Journal Article URL: a-latin-leader-copes-with-covid-19-11596396676
    Does the URL headline include Trump: NO

Wall Street Journal Article URL: a-lynching-false-alarm-in-california-11596495473
    Does the URL headline include Trump: NO
```

```
FINAL PERCENTAGE: 5.607476635514018

Of all the WSJ headlines in our database, 5.607476635514018\% of them were about Donald Trump
```

GitHub Link: https://github.com/grantyho/SI206-FINAL-PROJECT/blob/main/finalOutput.txt

PROBLEMS / RESOURCES

PROBLEMS WE FACED

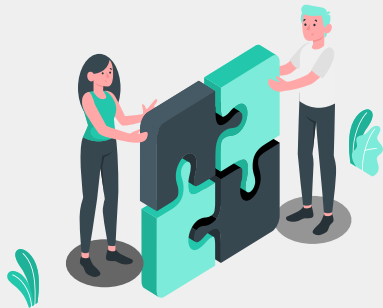
- Accessing WSJ and NYT data
- Designing our database
- Ensuring no duplicates in tables/database
  - News API and Twitter
- Understanding/Using Machine Learning
- Properly classifying articles
- Creating meaningful/comprehensive data visualizations
- Combining multiple tables into one database

RESOURCES

Date	Issue of Description	Location of Resource	Result
11/25	Creating a machine learning model	https://enlight.nyc/projects/build-a-naive-bayes-classifier	✔/❌
11/25	Using the News API	https://newsapi.org/docs	✔
11/26	Using Twitter API	https://developer.twitter.com/en/docs	✔
11/26	Using New York Times API	https://developer.nytimes.com/apis	✔
11/26	Creating a bot to login into a website	https://www.selenium.dev/selenium/docs/api/java/index.html?overview-summary.html	✔

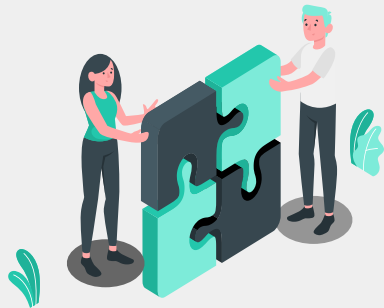
RESOURCES CONTINUED

Date	Issue of Description	Location of Resource	Result
11/26	Ignoring files when pushing to GitHub, file are too large	https://stackoverflow.com/questions/19573031/cant-push-to-github-because-of-large-file-which-i-already-deleted	✔
11/27	How to scrape Twitter tweets?	https://towardsdatascience.com/how-to-scrape-tweets-from-twitter-59287e20f0f1	✔
11/28	Creating optimal databases	https://www.sqlservertutorial.net/sql-server-basics/sql-server-create-database/	✔
12/1	Combining Two Tables Based on ID	Lecture Slides	✘
12/1	Combining Two Tables Based on ID	Piazza/Office Hours	✔



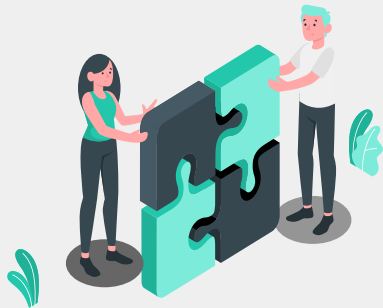
RESOURCES CONTINUED

Date	Issue of Description	Location of Resource	Result
12/1	Generating Unique ID every time Python script is run	https://stackoverflow.com/questions/29086705/sqlite-get-max-id-not-working	✓
12/1	How to select highest ID in a database?	https://stackoverflow.com/questions/6881424/how-can-i-select-the-row-with-the-highest-id-in-mysql	✓
12/2	Running multiple scripts at once	https://problemsolvingwithpython.com/07-Functions-and-Modules/07.05-Calling-Functions-from-Other-Files/	✓
12/3	Combining multiple tables from different data sources into one database	https://www.sqlshack.com/different-approaches-to-sql-join-multiple-tables/	✓



RESOURCES CONTINUED

Date	Issue of Description	Location of Resource	Result
12/3	From file_name import * go up a level in directory python	https://stackoverflow.com/questions/9856683/using-pythons-os-path-how-do-i-go-up-one-directory	✓
12/4	How to create pie chart with plotly?	https://plotly.com/python/pie-charts/	✓



CONCLUSION

CONCLUSION:

- ML IS CLEARLY NOT THE END ALL BE ALL
- HUMANS ARE NECESSARY
- MORE TRAINING ON OUR MODEL?

FOR THE FUTURE

- Train model with better data
- Account for Tweets that are just Links
- Give weighting to more reputable sources
- Take EECS 445?

THANKS!

Questions?

CHASE

GRANT