sentiment_analysis

May 11, 2024

1 Sentiment Analysis on Twitter Data

Carolyn Bozin, CPSC324, Final Project

This workbench is part of a pipeline involving Twitter data related to the Russia/Ukraine conflict. The goal of the pipeline is to extract and clean the data and gain insights on public feeling about the Russia/Ukraine conflict via sentiment analysis.

1.0.1 Initial Information

The dataset comes from Kaggle in the form of multiple csv files, which were ingested into a Google Cloud Storage bucket using the gcloud CLI. The next step in the pipeline was transferring the data into BigQuery tables, also using the gcloud CLI. After some initial pruning of columns, as well as exploratory data analysis using SQL queries, the next step in the pipeline is to use the VertexAI API to perform sentiment analysis on the data.

1.1 Data Ingestion & Cleaning

The Cloud Shell was used to ingest the initial csv files into Google Cloud Storage, which took a little bit of time because of how much data there was. The data was collected periodically by the maintainer via a web scraper, and was pretty clean to start with. However, it took a considerable amount of time to load it into BigQuery. The main issue was that the csv files had inconsistent amounts of columns, so when trying to send all the data to one uniform table there were errors. I ended up having to make a python script in order to parse which files had which amount of columns, and to put them in seperate directories in GCS. After doing so, I loaded all the data into BigQuery using Cloud Shell and ended up with two seperate tables with different amounts of columns (one table has additional data on retweets). Both tables had an extra index column which I dropped using an SQL query.

1.2 Exploratory Data Analysis

The following are some queries I used to perform EDA on my data in BigQuery

[2]:

```
%%bigquery
SELECT MIN(tweetcreatedts) AS min_datetime , MAX(tweetcreatedts) AS max_datetime
FROM `final-project-324.ukraine_dataset.twitter_data_long`
```

```
Query is running:   0%|          |
```

```
Downloading:   0%|          |
[2]:                  min_datetime                max_datetime
     0 2022-04-22 00:00:00+00:00 2023-06-14 15:27:24+00:00
```

```
[3]: %%bigquery
     SELECT COUNT(*) as lang_count, language
     FROM `final-project-324.ukraine_dataset.twitter_data_long`
     GROUP BY language
     ORDER BY lang_count DESC;
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

```
[3]:     lang_count language
     0     28333945       en
     1      2936035      und
     2      2733165       de
     3      2679120       fr
     4      2261685       es
     ..         …      …
     61         322       sd
     62          80       dv
     63          69       ug
     64          13       lo
     65           6       bo

     [66 rows x 2 columns]
```

```
[4]: %%bigquery
     SELECT
     COUNT(CASE WHEN coordinates = '' THEN 1 END) AS empty_coordinates,
     COUNT(CASE WHEN coordinates != '' THEN 1 END) AS non_empty_coordinates
     FROM `final-project-324.ukraine_dataset.twitter_data_long`
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

```
[4]:    empty_coordinates  non_empty_coordinates
     0           47116603                  74903
```

Additionally, I made several views, such as a view that combined all shared attributes of both tables, a view for only english data, and a view that stratified the data by month. I used the month view in order to stratify the data I was analyzing sentiment on, so that I could then get insights about sentiment over time.

1.3 Sentiment Analysis

My data did not come with labels, (and was also not in .txt format) so after some research I decided to use the VertexAI Natural Language Processing API's built in semantic analysis tool for ease.

I did run into some issues here. The first one was that loading my data in from BigQuery to a VertexAI workbench was taking a very long time. The second was that I kept hitting my API call quota (the per minute rate). Eventually, I had to cut the data I was analyzing drastically. I went with 150 rows for each month, amounting in about 2500 documents to perform sentiment analysis on. Another issue I ran into was that many of the languages of the tweets were unsupported by the API, so I decided to stick to English data only.

[1]:
```
from google.cloud import bigquery
bq_client = bigquery.Client()
```

A few rows from my dataset:

[5]:
```
%%bigquery
SELECT *
FROM `final-project-324.ukraine_dataset.twitter_data_filter_langs`
LIMIT 5
```

```
Query is running:   0%|          |

Downloading:   0%|          |
```

[5]:
```
                userid        username  \
0  1500236364163559424  VladimirAliev5
1  1500240692823699461     VovaBobrov7
2  1498313123937366016       UWTracker
3  1468765057458835457  massoud_torabi
4  1503092602681372675      RadarPlane

                                            acctdesc location  following  \
0                                                                      0
1                                                                      0
2  Identification of military equipment and techn…                  0
3                                                                      0
4  Using the opensky api to track planes to and f…                  0

   followers  totaltweets             usercreatedts              tweetid  \
0          0           19 2022-03-05 22:26:44+00:00  1500237921374326785
1          0           25 2022-03-05 22:43:56+00:00  1500242668705751049
2        468          228 2022-02-28 15:04:38+00:00  1508469574295076864
3         57         9729 2021-12-09 02:11:36+00:00  1503938992059912192
4         30         4436 2022-03-13 19:36:27+00:00  1514844805108543494

             tweetcreatedts  retweetcount  \
0 2022-03-05 22:32:48+00:00             0
1 2022-03-05 22:51:40+00:00             0
2 2022-03-28 15:42:27+00:00             0
3 2022-03-16 03:39:32+00:00            48
4 2022-04-15 05:55:20+00:00             0
```

```
                                                text  \
0  @SenBrianSchatz #Ukraine needs weapons and hum…
1  @cem_oezdemir #Ukraine needs weapons and human…
2  The Russian Orlan-10 UAV was shot down by Ukra…
3  Kudos to brave #MarinaOvsiannikova an \nemploy…
4  icao24: #5100fc, callsign: #BRU941  \nOrigin C…

                                            hashtags language coordinates  \
0  [{'text': 'Ukraine', 'indices': [16, 24]}, {'t…       en
1  [{'text': 'Ukraine', 'indices': [14, 22]}, {'t…       en
2  [{'text': 'Ukraine', 'indices': [60, 68]}, {'t…       en
3  [{'text': 'MarinaOvsiannikova', 'indices': [26…       en
4  [{'text': '5100fc', 'indices': [8, 15]}, {'tex…       en

   favorite_count                      extractedts
0               0 2022-03-05 22:36:03.870508+00:00
1               0 2022-03-05 22:53:11.440631+00:00
2               0 2022-03-28 15:46:20.230985+00:00
3               0 2022-03-16 03:52:30.189677+00:00
4               0 2022-04-15 06:10:20.421445+00:00
```

Looking at which languages are supported by the API

[3]:

```
query = """
    SELECT *
    FROM `final-project-324.ukraine_dataset.languages`
    ORDER BY lang_count DESC
"""

supported_langs =␣
 ↪['ar','zh','zh-Hant','nl','en','fr','de','id','it','ja','ko','pt','es','th','tr','vi']
langs_to_keep = []
rows = bq_client.query(query)
for row in rows:
    print(row["language"], row["lang_count"], end=" ")
    if row["language"] not in supported_langs:
        print("UNSUPPORTED")
    else:
        print("")
```

```
en 28333945
und 2936035 UNSUPPORTED
de 2733165
fr 2679120
es 2261685
it 2248635
uk 1258017 UNSUPPORTED
ru 738317 UNSUPPORTED
```

```
tr 501876
ja 469795
pl 253135 UNSUPPORTED
nl 233475
th 227235
pt 207071
hi 186162 UNSUPPORTED
in 178440 UNSUPPORTED
ar 175092
el 155094 UNSUPPORTED
zh 135724
fi 133312 UNSUPPORTED
sv 94096 UNSUPPORTED
fa 90356 UNSUPPORTED
ro 83611 UNSUPPORTED
ur 77650 UNSUPPORTED
cs 72462 UNSUPPORTED
et 63150 UNSUPPORTED
ca 58238 UNSUPPORTED
ko 51303
da 48666 UNSUPPORTED
vi 48652
ta 40516 UNSUPPORTED
bn 40502 UNSUPPORTED
tl 36475 UNSUPPORTED
ht 35260 UNSUPPORTED
no 32829 UNSUPPORTED
iw 23776 UNSUPPORTED
lv 23749 UNSUPPORTED
eu 22165 UNSUPPORTED
gu 20422 UNSUPPORTED
sl 19091 UNSUPPORTED
bg 18843 UNSUPPORTED
lt 17367 UNSUPPORTED
te 16459 UNSUPPORTED
sr 13927 UNSUPPORTED
kn 13301 UNSUPPORTED
is 11053 UNSUPPORTED
ka 10931 UNSUPPORTED
ml 10499 UNSUPPORTED
mr 9080 UNSUPPORTED
cy 8978 UNSUPPORTED
hu 7231 UNSUPPORTED
si 5378 UNSUPPORTED
ne 4831 UNSUPPORTED
am 3835 UNSUPPORTED
pa 3482 UNSUPPORTED
or 2296 UNSUPPORTED
```

```
ps 2022 UNSUPPORTED
my 1679 UNSUPPORTED
ckb 791 UNSUPPORTED
hy 381 UNSUPPORTED
km 353 UNSUPPORTED
sd 322 UNSUPPORTED
dv 80 UNSUPPORTED
ug 69 UNSUPPORTED
lo 13 UNSUPPORTED
bo 6 UNSUPPORTED
```

To speed up the ingestion of data from BigQuery, I ended up using the bigframes.pandas library to store my data in dataframes optimized for BigQuery data. I found that it sped up the process of transferring the data by a lot.

[4]:
```
!pip install --upgrade bigframes
```

```
Requirement already satisfied: bigframes in /opt/conda/lib/python3.10/site-
packages (1.5.0)
Requirement already satisfied: cloudpickle>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (3.0.0)
Requirement already satisfied: fsspec>=2023.3.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2024.3.1)
Requirement already satisfied: gcsfs>=2023.3.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2024.3.1)
Requirement already satisfied: geopandas>=0.12.2 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (0.14.4)
Requirement already satisfied: google-auth<3.0dev,>=2.15.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2.29.0)
Requirement already satisfied: google-cloud-bigquery>=3.16.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery[bqstorage,pandas]>=3.16.0->bigframes) (3.21.0)
Requirement already satisfied: google-cloud-functions>=1.12.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (1.16.3)
Requirement already satisfied: google-cloud-bigquery-connection>=1.12.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (1.15.3)
Requirement already satisfied: google-cloud-iam>=2.12.1 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2.15.0)
Requirement already satisfied: google-cloud-resource-manager>=1.10.3 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (1.12.3)
Requirement already satisfied: google-cloud-storage>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2.14.0)
Requirement already satisfied: ibis-framework<9.0.0dev,>=8.0.0 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (8.0.0)
Requirement already satisfied: pandas>=1.5.0 in /opt/conda/lib/python3.10/site-
packages (from bigframes) (2.2.2)
Requirement already satisfied: pyarrow>=8.0.0 in /opt/conda/lib/python3.10/site-
packages (from bigframes) (15.0.2)
```

```
Requirement already satisfied: pydata-google-auth>=1.8.2 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (1.8.2)
Requirement already satisfied: requests>=2.27.1 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2.31.0)
Requirement already satisfied: scikit-learn>=1.2.2 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (1.4.2)
Requirement already satisfied: sqlalchemy<3.0dev,>=1.4 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (2.0.29)
Requirement already satisfied: sqlglot<=20.11,>=20.8.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (20.11.0)
Requirement already satisfied: tabulate>=0.9 in /opt/conda/lib/python3.10/site-
packages (from bigframes) (0.9.0)
Requirement already satisfied: ipywidgets>=7.7.1 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (8.1.2)
Requirement already satisfied: humanize>=4.6.0 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (4.9.0)
Requirement already satisfied: matplotlib>=3.7.1 in
/opt/conda/lib/python3.10/site-packages (from bigframes) (3.8.4)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in
/opt/conda/lib/python3.10/site-packages (from gcsfs>=2023.3.0->bigframes)
(3.9.5)
Requirement already satisfied: decorator>4.1.2 in
/opt/conda/lib/python3.10/site-packages (from gcsfs>=2023.3.0->bigframes)
(5.1.1)
Requirement already satisfied: google-auth-oauthlib in
/opt/conda/lib/python3.10/site-packages (from gcsfs>=2023.3.0->bigframes)
(1.2.0)
Requirement already satisfied: fiona>=1.8.21 in /opt/conda/lib/python3.10/site-
packages (from geopandas>=0.12.2->bigframes) (1.9.6)
Requirement already satisfied: numpy>=1.22 in /opt/conda/lib/python3.10/site-
packages (from geopandas>=0.12.2->bigframes) (1.26.4)
Requirement already satisfied: packaging in /opt/conda/lib/python3.10/site-
packages (from geopandas>=0.12.2->bigframes) (24.0)
Requirement already satisfied: pyproj>=3.3.0 in /opt/conda/lib/python3.10/site-
packages (from geopandas>=0.12.2->bigframes) (3.6.1)
Requirement already satisfied: shapely>=1.8.0 in /opt/conda/lib/python3.10/site-
packages (from geopandas>=0.12.2->bigframes) (2.0.4)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from google-
auth<3.0dev,>=2.15.0->bigframes) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/conda/lib/python3.10/site-packages (from google-
auth<3.0dev,>=2.15.0->bigframes) (0.4.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /opt/conda/lib/python3.10/site-
packages (from google-auth<3.0dev,>=2.15.0->bigframes) (4.9)
Requirement already satisfied: google-api-core!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!
=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0dev,>=1.34.1 in
/opt/conda/lib/python3.10/site-packages (from google-api-core[grpc]!=2.0.*,!=2.1
.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0
.0dev,>=1.34.1->google-cloud-bigquery>=3.16.0->google-cloud-
bigquery[bqstorage,pandas]>=3.16.0->bigframes) (1.34.1)
Requirement already satisfied: google-cloud-core<3.0.0dev,>=1.6.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery>=3.16.0->google-cloud-bigquery[bqstorage,pandas]>=3.16.0->bigframes)
(2.4.1)
Requirement already satisfied: google-resumable-media<3.0dev,>=0.6.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery>=3.16.0->google-cloud-bigquery[bqstorage,pandas]>=3.16.0->bigframes)
(2.7.0)
Requirement already satisfied: python-dateutil<3.0dev,>=2.7.2 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery>=3.16.0->google-cloud-bigquery[bqstorage,pandas]>=3.16.0->bigframes)
(2.9.0)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-bigquery-
connection>=1.12.0->bigframes) (1.23.0)
Requirement already satisfied: protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.2
1.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.19.5 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-bigquery-
connection>=1.12.0->bigframes) (3.20.3)
Requirement already satisfied: grpc-google-iam-v1<1.0.0dev,>=0.12.4 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-bigquery-
connection>=1.12.0->bigframes) (0.13.0)
Requirement already satisfied: db-dtypes<2.0.0dev,>=0.3.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery[bqstorage,pandas]>=3.16.0->bigframes) (1.2.0)
Requirement already satisfied: google-cloud-bigquery-storage<3.0.0dev,>=2.6.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery[bqstorage,pandas]>=3.16.0->bigframes) (2.24.0)
Requirement already satisfied: grpcio<2.0dev,>=1.47.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
bigquery[bqstorage,pandas]>=3.16.0->bigframes) (1.63.0)
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in
/opt/conda/lib/python3.10/site-packages (from google-cloud-
storage>=2.0.0->bigframes) (1.5.0)
Requirement already satisfied: atpublic<5,>=2.3 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (4.1.0)
Requirement already satisfied: bidict<1,>=0.22.1 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (0.23.1)
Requirement already satisfied: multipledispatch<2,>=0.6 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework<9.0.0dev,>=8.0.0->ibis-
```

```
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (1.0.0)
Requirement already satisfied: parsy<3,>=2 in /opt/conda/lib/python3.10/site-
packages (from ibis-framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (2.1)
Requirement already satisfied: pyarrow-hotfix<1,>=0.4 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (0.6)
Requirement already satisfied: pytz>=2022.7 in /opt/conda/lib/python3.10/site-
packages (from ibis-framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (2024.1)
Requirement already satisfied: rich<14,>=12.4.4 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (13.7.1)
Requirement already satisfied: toolz<1,>=0.11 in /opt/conda/lib/python3.10/site-
packages (from ibis-framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (0.12.1)
Requirement already satisfied: typing-extensions<5,>=4.3.0 in
/opt/conda/lib/python3.10/site-packages (from ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (4.11.0)
Requirement already satisfied: comm>=0.1.3 in /opt/conda/lib/python3.10/site-
packages (from ipywidgets>=7.7.1->bigframes) (0.2.2)
Requirement already satisfied: ipython>=6.1.0 in /opt/conda/lib/python3.10/site-
packages (from ipywidgets>=7.7.1->bigframes) (8.21.0)
Requirement already satisfied: traitlets>=4.3.1 in
/opt/conda/lib/python3.10/site-packages (from ipywidgets>=7.7.1->bigframes)
(5.14.3)
Requirement already satisfied: widgetsnbextension~=4.0.10 in
/opt/conda/lib/python3.10/site-packages (from ipywidgets>=7.7.1->bigframes)
(4.0.10)
Requirement already satisfied: jupyterlab-widgets~=3.0.10 in
/opt/conda/lib/python3.10/site-packages (from ipywidgets>=7.7.1->bigframes)
(3.0.10)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.7.1->bigframes)
(1.2.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.10/site-
packages (from matplotlib>=3.7.1->bigframes) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.7.1->bigframes)
(4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.7.1->bigframes)
(1.4.5)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.10/site-
packages (from matplotlib>=3.7.1->bigframes) (10.3.0)
```

```
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.10/site-packages (from matplotlib>=3.7.1->bigframes)
(3.1.2)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.10/site-
packages (from pandas>=1.5.0->bigframes) (2024.1)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.10/site-
packages (from pydata-google-auth>=1.8.2->bigframes) (69.5.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/conda/lib/python3.10/site-packages (from requests>=2.27.1->bigframes)
(3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.10/site-
packages (from requests>=2.27.1->bigframes) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from requests>=2.27.1->bigframes)
(1.26.18)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.10/site-packages (from requests>=2.27.1->bigframes)
(2024.2.2)
Requirement already satisfied: scipy>=1.6.0 in /opt/conda/lib/python3.10/site-
packages (from scikit-learn>=1.2.2->bigframes) (1.11.4)
Requirement already satisfied: joblib>=1.2.0 in /opt/conda/lib/python3.10/site-
packages (from scikit-learn>=1.2.2->bigframes) (1.4.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.10/site-packages (from scikit-learn>=1.2.2->bigframes)
(3.5.0)
Requirement already satisfied: greenlet!=0.4.17 in
/opt/conda/lib/python3.10/site-packages (from
sqlalchemy<3.0dev,>=1.4->bigframes) (3.0.3)
Requirement already satisfied: aiosignal>=1.1.2 in
/opt/conda/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->gcsfs>=2023.3.0->bigframes) (1.3.1)
Requirement already satisfied: attrs>=17.3.0 in /opt/conda/lib/python3.10/site-
packages (from aiohttp!=4.0.0a0,!=4.0.0a1->gcsfs>=2023.3.0->bigframes) (23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/opt/conda/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->gcsfs>=2023.3.0->bigframes) (1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/opt/conda/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->gcsfs>=2023.3.0->bigframes) (6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in /opt/conda/lib/python3.10/site-
packages (from aiohttp!=4.0.0a0,!=4.0.0a1->gcsfs>=2023.3.0->bigframes) (1.9.4)
Requirement already satisfied: async-timeout<5.0,>=4.0 in
/opt/conda/lib/python3.10/site-packages (from
aiohttp!=4.0.0a0,!=4.0.0a1->gcsfs>=2023.3.0->bigframes) (4.0.3)
Requirement already satisfied: click~=8.0 in /opt/conda/lib/python3.10/site-
packages (from fiona>=1.8.21->geopandas>=0.12.2->bigframes) (8.1.7)
Requirement already satisfied: click-plugins>=1.0 in
/opt/conda/lib/python3.10/site-packages (from
fiona>=1.8.21->geopandas>=0.12.2->bigframes) (1.1.1)
Requirement already satisfied: cligj>=0.5 in /opt/conda/lib/python3.10/site-
packages (from fiona>=1.8.21->geopandas>=0.12.2->bigframes) (0.7.2)
Requirement already satisfied: six in /opt/conda/lib/python3.10/site-packages
(from fiona>=1.8.21->geopandas>=0.12.2->bigframes) (1.16.0)
Requirement already satisfied: googleapis-common-protos<2.0dev,>=1.56.2 in
/opt/conda/lib/python3.10/site-packages (from google-api-core!=2.0.*,!=2.1.*,!=2
.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0dev,
>=1.34.1->google-api-core[grpc]!=2.0.*,!=2.1.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,
!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0.0dev,>=1.34.1->google-cloud-
bigquery>=3.16.0->google-cloud-bigquery[bqstorage,pandas]>=3.16.0->bigframes)
(1.63.0)
Requirement already satisfied: grpcio-status<2.0dev,>=1.33.2 in
/opt/conda/lib/python3.10/site-packages (from google-api-core[grpc]!=2.0.*,!=2.1
.*,!=2.10.*,!=2.2.*,!=2.3.*,!=2.4.*,!=2.5.*,!=2.6.*,!=2.7.*,!=2.8.*,!=2.9.*,<3.0
.0dev,>=1.34.1->google-cloud-bigquery>=3.16.0->google-cloud-
bigquery[bqstorage,pandas]>=3.16.0->bigframes) (1.48.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in
/opt/conda/lib/python3.10/site-packages (from google-auth-
oauthlib->gcsfs>=2023.3.0->bigframes) (2.0.0)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.10/site-
packages (from ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.19.1)
Requirement already satisfied: matplotlib-inline in
/opt/conda/lib/python3.10/site-packages (from
ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.1.7)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in
/opt/conda/lib/python3.10/site-packages (from
ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (3.0.42)
Requirement already satisfied: pygments>=2.4.0 in
/opt/conda/lib/python3.10/site-packages (from
ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (2.17.2)
Requirement already satisfied: stack-data in /opt/conda/lib/python3.10/site-
packages (from ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.6.2)
Requirement already satisfied: exceptiongroup in /opt/conda/lib/python3.10/site-
packages (from ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (1.2.0)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.10/site-
packages (from ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (4.9.0)
Requirement already satisfied: pyasn1<0.7.0,>=0.4.6 in
/opt/conda/lib/python3.10/site-packages (from pyasn1-modules>=0.2.1->google-
auth<3.0dev,>=2.15.0->bigframes) (0.6.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/opt/conda/lib/python3.10/site-packages (from rich<14,>=12.4.4->ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (3.0.0)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in
/opt/conda/lib/python3.10/site-packages (from
jedi>=0.16->ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.8.4)
Requirement already satisfied: mdurl~=0.1 in /opt/conda/lib/python3.10/site-
```

```
packages (from markdown-it-py>=2.2.0->rich<14,>=12.4.4->ibis-
framework<9.0.0dev,>=8.0.0->ibis-
framework[bigquery]<9.0.0dev,>=8.0.0->bigframes) (0.1.2)
Requirement already satisfied: ptyprocess>=0.5 in
/opt/conda/lib/python3.10/site-packages (from
pexpect>4.3->ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.10/site-
packages (from prompt-
toolkit<3.1.0,>=3.0.41->ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.2.13)
Requirement already satisfied: oauthlib>=3.0.0 in
/opt/conda/lib/python3.10/site-packages (from requests-oauthlib>=0.7.0->google-
auth-oauthlib->gcsfs>=2023.3.0->bigframes) (3.2.2)
Requirement already satisfied: executing>=1.2.0 in
/opt/conda/lib/python3.10/site-packages (from stack-
data->ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (2.0.1)
Requirement already satisfied: asttokens>=2.1.0 in
/opt/conda/lib/python3.10/site-packages (from stack-
data->ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (2.4.1)
Requirement already satisfied: pure-eval in /opt/conda/lib/python3.10/site-
packages (from stack-data->ipython>=6.1.0->ipywidgets>=7.7.1->bigframes) (0.2.2)
```

[5]:
```python
import bigframes.pandas as bpd
query = """
    SELECT tweetid, text
    FROM `final-project-324.ukraine_dataset.stratified_by_month`
"""
df = bpd.read_gbq_query(query=query)
```

```
<IPython.core.display.HTML object>
```

[8]:
```python
df.head()
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>
```

[8]:
```
               tweetid                                               text
0  1594709722590216192  #QatarWorldCup2022 #FIFAWorldCup its coming ho…
1  1521592206712315904  Kristina and her cat survived the war and were…
2  1658487745017434112  Horrible!! 150 Russian Wagner group destroyed …
3  1638423097156988928  Unfortunately, we must now ask this question: …
4  1610127080137654273  Love that for him. #RussiaIsLosing #RussiaIsCo…

[5 rows x 2 columns]
```

Here is where I actually implement the sentiment analysis. I made a function that appends sentiment and magnitude data to a list at the given indeces. The indeces are for limiting the amount of data being analyzed at a time, in order to stay at my per minute API call quota.

[9]:

```
from google.cloud import language_v2

ls_client = language_v2.LanguageServiceClient()

document_type = language_v2.Document.Type.PLAIN_TEXT
```

[10]:

```
def analyze_text_sentiment(start_idx, end_idx, lst):
    for index, row in df.iloc[start_idx:end_idx].iterrows():
        text = row['text']
        document = {
            "content": text,
            "type_": language_v2.Document.Type.PLAIN_TEXT,
            "language_code":"en"
        }
        encoding_type = language_v2.EncodingType.UTF8
        response = ls_client.analyze_sentiment(
            request={"document": document, "encoding_type": encoding_type}
        )

        lst.append({
            'tweetid': row['tweetid'],
            'sentiment_score': response.document_sentiment.score,
            'sentiment_magnitude': response.document_sentiment.magnitude
        })
    return lst
```

I called sleep() for a minute between each round (600 calls is the per minute limit) in order to avoid a time out.

[11]:

```
from time import sleep
sentiment_results = []
sentiment_results = analyze_text_sentiment(0,600,sentiment_results)
sleep(60)
sentiment_results = analyze_text_sentiment(600,1200,sentiment_results)
sleep(60)
sentiment_results = analyze_text_sentiment(1200,1800,sentiment_results)
sleep(60)
sentiment_results = analyze_text_sentiment(1800,2400,sentiment_results)
sleep(60)
sentiment_results = analyze_text_sentiment(2400,None,sentiment_results)
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>
```

To make sure, I printed out the difference between the expected amount of rows and the total amount. 0 is good.

```
[13]: print(2550 - len(sentiment_results))
```

```
0
```

Here is a look at what my sentiment list looks like. Sentiment ranges from -1 to 1, where -1 is the most negative and 1 is the most positive. Magnitude ranges from 0 to infinity and is “absolute”, regardless of negative or positive values.

```
[14]: sentiment_results[:5]
```

```
[14]: [{'tweetid': 1594709722590216192,
        'sentiment_score': 0.07199999690055847,
        'sentiment_magnitude': 0.335999995470047},
       {'tweetid': 1521592206712315904,
        'sentiment_score': 0.28200000524520874,
        'sentiment_magnitude': 0.492000013589859},
       {'tweetid': 1658487745017434112,
        'sentiment_score': -0.5519999861717224,
        'sentiment_magnitude': 1.309999942779541},
       {'tweetid': 1638423097156988928,
        'sentiment_score': -0.9449999928474426,
        'sentiment_magnitude': 0.9810000061988831},
       {'tweetid': 1610127080137654273,
        'sentiment_score': 0.004000000189989805,
        'sentiment_magnitude': 1.8839999437332153}]
```

1.4 Exporting to BigQuery

The final step in VertexAI was to send the sentiment values to a table in BigQuery.

```
[45]: dataset_ref = bq_client.dataset('ukraine_dataset')
      table_ref = dataset_ref.table('sentiment_results_stratified')

      schema = [
          bigquery.SchemaField("tweetid", "INTEGER", mode="REQUIRED"),
          bigquery.SchemaField("sentiment_score", "FLOAT"),
          bigquery.SchemaField("sentiment_magnitude", "FLOAT")
      ]

      table = bigquery.Table(table_ref, schema=schema)

      errors = bq_client.insert_rows_json(table, sentiment_results)
      if errors:
          print("Encountered errors while inserting rows:")
          for error in errors:
              print(error)
```

```python
print(f"Data uploaded to BigQuery table {table}")
```

```
Data uploaded to BigQuery table final-
project-324.ukraine_dataset.sentiment_results_stratified
```