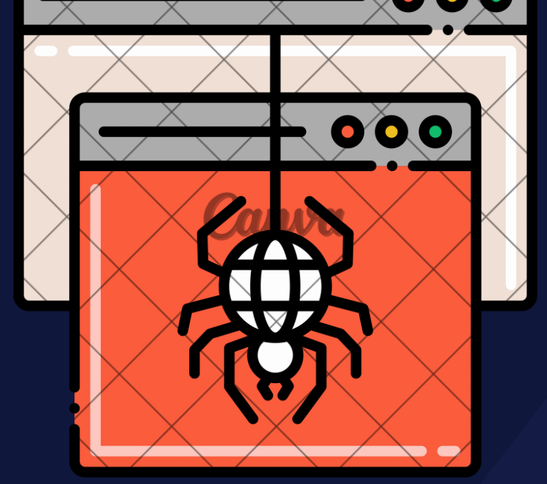
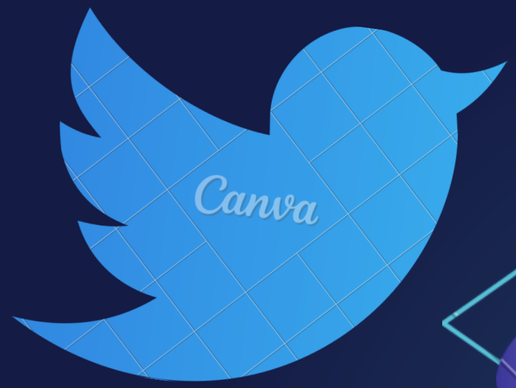




Twitter Data Crawler & Insight Tool "TweetTrack"

Introduction

Team Red

Sum Ng

Sam Ngan

Victor Chow

Introduction

About Twitter

450,000k+

Social media platforms like Twitter have become a rich source of data for businesses and researchers

500,000k+

User On Twitter are generation about half billion tweets everyday

Introduction

Twitter Data: Why It Matters

1

Reveal Insights Into Customer Needs preferences, and sentiments.

2

Monitor Brand Reputation

3

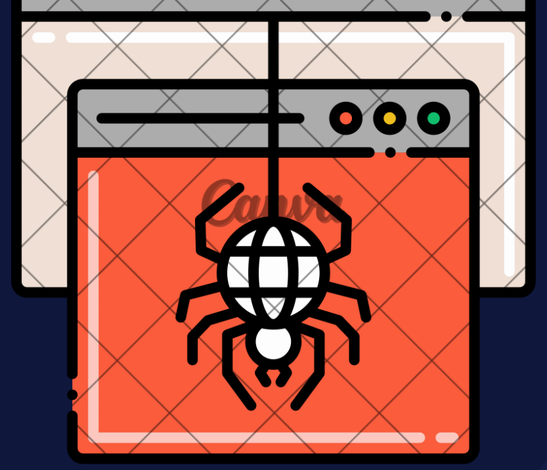
Identify Eemerging Trends

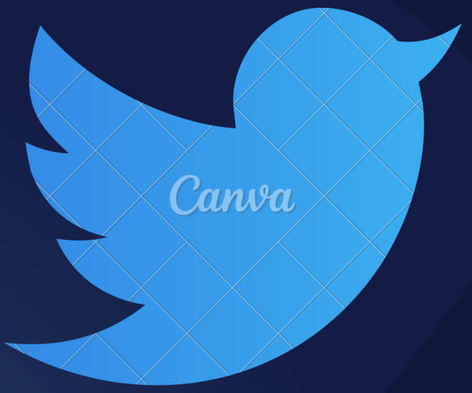
4

Study Social Phenomena

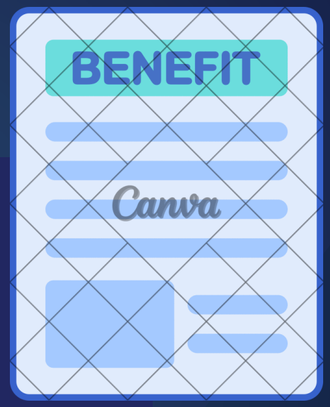
5

Analyze Public Opinion





Our Tool



TweetTrack

introduce a tool for crawling and analysing Twitter data, and show how it can provide valuable insights into user behavior, opinions, and trends."

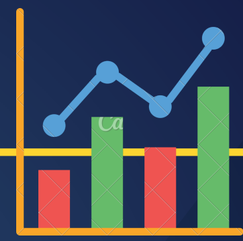
TweetTrack

Process

Collect Different Type Data From Twitter

Store Crawled Data & Data Clean

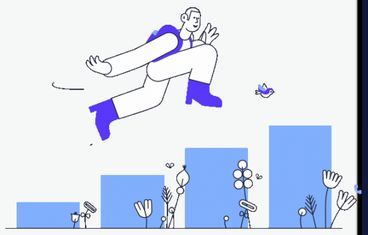
Analysis



Provide Valuable Insights

TweetTrack

Let's start



Collect Different Type Data From Twitter

Collect Different Data From Twitter

Library-Scweet (1)

```
data1 = scrape(words=keyword, since=startdate, until=enddate, from_account=None, interval=1,
               headless=True, display_type="Top", save_images=False, lang=None,
               resume=False, filter_replies=False, proximity=False, geocode=None)
```

Words

Words to search for.

Hashtag

Tweets containing #hashtag

Since

Start date for search query

Until

End date for search query

From_account

Tweets posted by this account

Interval

Interval days between each start date and end date for

Headless

Show the website or not

Display type

Display type of Twitter page (Latest or Top tweets)

Lang

Tweets language

Resume

Resume the last scraping. specify the csv file path

Proximity

Show the website or not

Geocode

Display type of Twitter page (Latest or Top tweets)

Unofficial Scrape Twitter tool
Use Google Chrome To Scrape Tweets

Collect Different Data From Twitter

Library-Scweet (2)

UserScreenName
User nickname

UserNamen
UserName

Embedded_text
embedded text written above the tweet

Emojis
emojis in the tweet

Retweets
number of retweets

Image link
link of the image in the tweet

Timestamp
timestamp of the tweet

Text
tweet text

Comments
number of comments

Likes
number of likes

Tweet URL
tweet URL

```
#   Column          Non-Null Count  Dtype
---  ------          --------------  -----
0   UserScreenName  21185 non-null  object
1   UserName        21185 non-null  object
2   Timestamp       21185 non-null  object
3   Text            21185 non-null  object
4   Embedded_text   21185 non-null  object
5   Emojis          21185 non-null  object
6   Comments        21185 non-null  object
7   Likes           21185 non-null  object
8   Retweets        21185 non-null  object
9   Image link      21185 non-null  object
10  Tweet URL       21185 non-null  object
```

Collect Different Data From Twitter

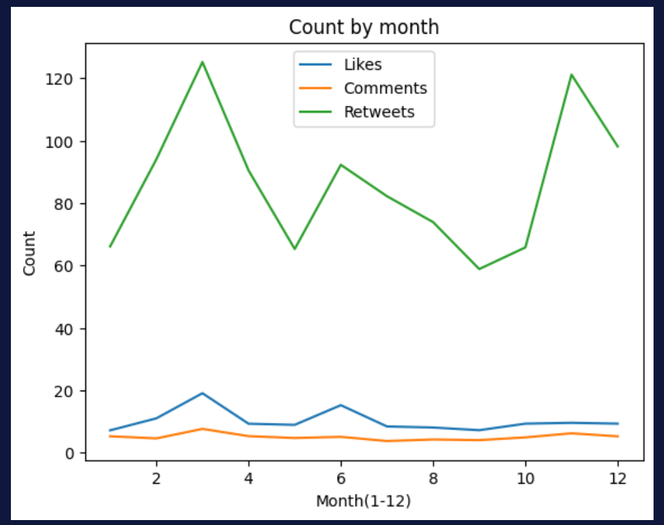
Library-pandas / Pickle

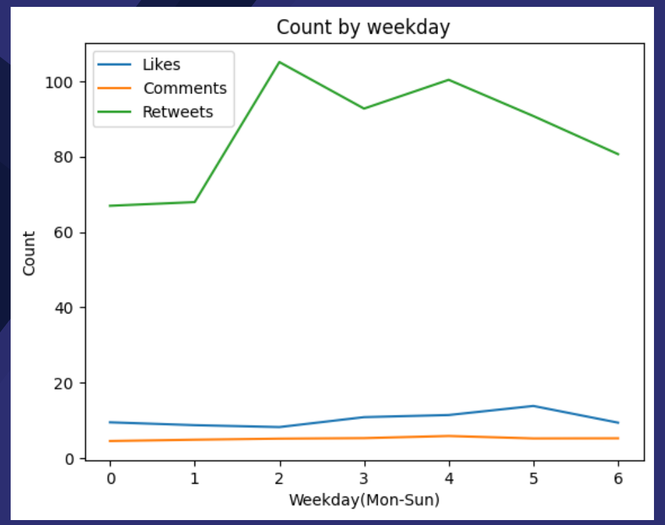
- Temporarily store the result as dataframe
- Change data type
- Save the result as pickle file

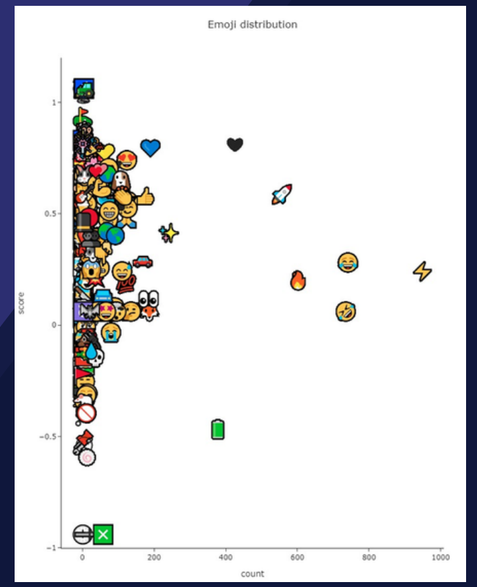
Library-Matplotlib / Plotly / scatter

```
data1.groupby(data1["Timestamp"].dt.month)['Likes','Comments','Retweets'].mean().plot(title='Count by month',xlabel='Month(1-12)',ylabel='Count')
```

- Visualize the analyzed data







Collect Different Data From Twitter

Library- Emosent

DeepMoji

In 2017, the MIT Media Lab published DeepMoji, a deep neural network sentiment analysis algorithm that was trained on 1.2 billion emoji occurrences in Twitter data from 2013 to 2017. DeepMoji was found to outperform human subjects in correctly identifying sarcasm in Tweets and other online modes of communication.

Definition

- unicode_codepoint : Unicode for the emoji
- occurrences : Showed times in "DeepMoji"
- position : Positioning score (0-1)
- negative : Use as negative targeting (0-1)
- neutral : Use as neutral targeting (0-1)
- positive : Use as positive targeting (0-1)
- unicode_name : Emoji's name
- unicode_block : Emoji's type
- sentiment_score : sentiment score

```
get_emoji_sentiment_rank('❤')
✓ 0.0s
{'unicode_codepoint': '0x2764',
 'occurrences': 8050,
 'position': 0.746943086,
 'negative': 355.0,
 'neutral': 1334.0,
 'positive': 6361.0,
 'unicode_name': 'HEAVY BLACK HEART',
 'unicode_block': 'Dingbats',
 'sentiment_score': 0.746}
```

Get Sentiment Rankings for Unicode Emojis

How To Use "TweetTrack"?

InterFace

```
Input the Keyword:
tesla
Input the hashtag#:

Input the start day(YYYY-MM-DD):
2022-1-1
Input the end day(YYYY-MM-DD), empty = cur
rent date:
2023-1-1
```

Notice:

*date format: YYYY-MM-DD
*you will be required to re-enter till to correct format

1 **Step 1 - Input data**

- Keyword / hashtag (Either one)
- The start day(must) & end day(not must) of search period

2 **Step 2 - Wait**

- The scraped data will sent to you in CSV file by email
- Two CSV files will be sent to you if you inputted one keyword & one hashtag


DEMO

Collect Different Data From Twitter

Coding - scrape (1)

```
# definite variable
keyword = ""
hashTagWord = ""
startdate = ""
enddate = ""
```

```
# Input keyword / hashTagWord
while keyword == "" and hashTagWord == "":
    # Input Keyword
    print('Input the Keyword:')
    keyword = input()

    # Input Hashtag
    print('Input the hashtag#:')
    hashTagWord = input()

    # Check if there is one of the input A/B
    if keyword == "" and hashTagWord == "":
        print('Please input keyword / hashtag for finding tweet.')
    else:
        pass
```

```
# Input startdate with specify format
isInvalidStartDate= True
while isInvalidStartDate:
    if startdate == "":
        print('Input the start day(YYYY-MM-DD):')
        startdate = input()
    try:
        # try to convert startDate String to datetime
        datetime.datetime.strptime(startdate, '%Y-%m-%d')
        isInvalidStartDate = False
    except :
        # print error message
        print('Please input valid date with correct format!!!!!!!')
        pass

# Input enddate with specify format
isInvalidStartDate= True
while isInvalidStartDate:
    if enddate == "":
        print('Input the end day(YYYY-MM-DD), empty = current date:')
        enddate = input()
    if enddate != "":
        try:
            # try to convert endDate String to datetime
            datetime.datetime.strptime(enddate, '%Y-%m-%d')
            isInvalidStartDate = False
        except :
            # print error message
            print('Please input valid date with correct format!!!!!!!')
            pass
    else:
        # convert enddate to None if did not input anything
        enddate = None
        isInvalidStartDate = False
```

Collect Different Data From Twitter

Coding - scrape (2)

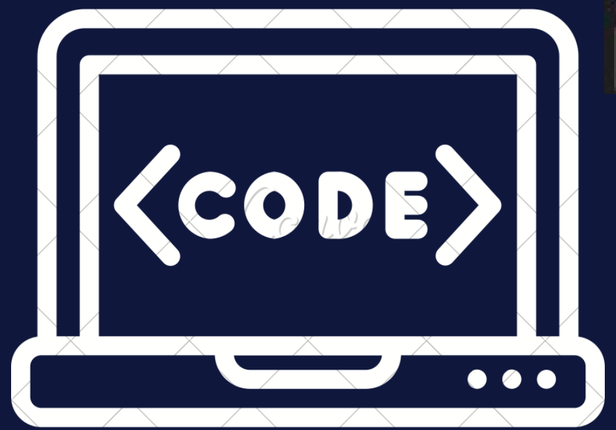
```
# Check keyword is empty or not
if keyword != '' and keyword != None:
    # find post with keyword
    data1 = scrape(words=keyword, since=startdate, until=enddate, from_account=None, interval=1,
                   headless=True, display_type="Top", save_images=False, lang=None,
                   resume=False, filter_replies=False, proximity=False, geocode=None)

    # add column 'keyword' to DF
    data1["Keyword"] = keyword
    # print(data1)

# tidy up data in column 'Likes' / 'Retweets' / 'Comments'  (data type & strange word)
data1['Timestamp'] = data1[['Timestamp']].astype('datetime64')
data1['Likes'] = data1['Likes'].apply(change_data_type.fix_numbers)
data1['Retweets'] = data1['Retweets'].apply(change_data_type.fix_numbers)
data1['Comments'] = data1['Comments'].apply(change_data_type.fix_numbers)
```

Collect Different Data From Twitter

Coding - scrape (3)

```
# save dataframe tp pkl file for back up
data1.to_pickle('twitter_keyword_'+keyword+'.pkl')
data1.to_csv('twitter_keyword_'+keyword+'.csv')
```

```
# find latest pkl file & print out
list_of_files = glob.glob('*.pkl')
latest_file = max(list_of_files, key=os.path.getctime)
df_latest_file  = pd.read_pickle(latest_file)
# print(df_latest_file)

# Connect to CockroachDB and insert record
if __name__ == "__main__":
    DB_final.main()
```




```
# Connect to CockroachDB and insert record
if __name__ == "__main__":
    DB_final.main()
```

Data Clean
Coding - Change data type

```python
def fix_numbers(x):

    if type(x) == int:
        return x

    elif type(x) == str:
        newNumber = x.replace(",", "").replace("K", "").replace("萬", "").replace(".", "")

        if "K" in x:
            if "." in x:
                newNumber = newNumber + "00"
            else:
                newNumber = newNumber + "000"

        if "萬" in x:
            if "." in x:
                newNumber = newNumber + "000"
            else:
                newNumber = newNumber + "0000"

    try:
        return int(newNumber)
    except:
        return 0
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20276 entries, 0 to 20275
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   UserScreenName  20228 non-null  object
 1   UserName        20274 non-null  object
 2   Timestamp       20276 non-null  object
 3   Text            20276 non-null  object
 4   Embedded_text   20276 non-null  object
 5   Emojis          5871 non-null   object
 6   Comments        11106 non-null  object
 7   Likes           4789 non-null   object
 8   Retweets        18685 non-null  object
 9   Image link      20276 non-null  object
 10  Tweet URL       20276 non-null  object
dtypes: object(11)
memory usage: 1.7+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20276 entries, 0 to 20275
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   UserScreenName  20228 non-null  object
 1   UserName        20274 non-null  object
 2   Timestamp       20276 non-null  datetime64[ns]
 3   Text            20276 non-null  object
 4   Embedded_text   20276 non-null  object
 5   Emojis          5871 non-null   object
 6   Comments        20276 non-null  int64
 7   Likes           20276 non-null  int64
 8   Retweets        20276 non-null  int64
 9   Image link      20276 non-null  object
 10  Tweet URL       20276 non-null  object
dtypes: datetime64[ns](1), int64(3), object(7)
memory usage: 1.7+ MB
```

Store Crawled Data

Coding - Connect to database (1)

```
#Create table
def create_table(conn):
    with conn.cursor() as cur:
        cur.execute(
            "CREATE TABLE IF NOT EXISTS twitter_scrape (Tweet_id UUID PRIMARY KEY, UserScreenName VARCHAR,
            UserName VARCHAR, Timestamp TIMESTAMP, Text VARCHAR, Embedded_text VARCHAR, Emojis VARCHAR,
            Comments VARCHAR, Likes VARCHAR, Retweets VARCHAR, Image_link VARCHAR, Tweet_URL VARCHAR, Keyword
            VARCHAR)"
        )

#Insert record data1->sql db
def insert_record(conn, data1):
    with conn.cursor() as cur:
        for index, arr in enumerate(data1.iterrows()):
            if index ==0:
                continue
            id1 = uuid.uuid4()
            record = arr[1]
            cur.execute("UPSERT INTO twitter_scrape (tweet_id, userscreenname, username, timestamp, text,
            embedded_text, emojis, comments, likes, retweets, image_link, tweet_URL, keyword) values (%s,%s
            %s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)", (id1, record[0], record[1], record[2], record[3], record[4]
            record[5], record[6], record[7], record[8], record[9], record[10], record[11])
            )
```

Store Crawled Data

Coding - Connect to database (2)

```python
#delete record <all>
def delete_accounts(conn):
    with conn.cursor() as cur:
        cur.execute("DELETE FROM twitter_scrape")
        logging.debug("delete_twitter_scrape(): status message: %s",
                      cur.statusmessage)
```

```python
#display record
def print_record(conn):
    with conn.cursor() as cur:
        print(f"record at {time.asctime()}:")
        for row in cur.execute("SELECT count(keyword) as count FROM twitter_scrape"):
            # print(row.tweet_id)
            print("count : {0}  ".format(row.count))
```

```python
#delete table!!!!!!
def del_table(conn):
    with conn.cursor() as cur:
        cur.execute('DROP TABLE twitter_scrape')
```

twitter_scrape_2

tweet_id	UUID
userscreenname	VARCHAR
username	VARCHAR
timestamp	TIMESTAMP
text	VARCHAR
embedded_text	VARCHAR
emojis	VARCHAR
comments	INT(8)
likes	INT(8)
retweets	INT(8)
image_link	VARCHAR
tweet_url	VARCHAR
keyword	VARCHAR

Store Crawled Data

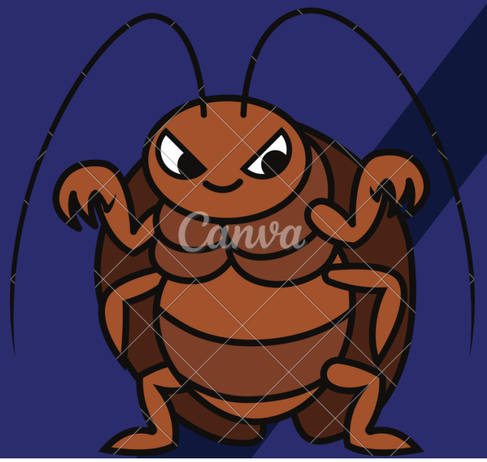
Coding - Connect to database (3)

```
#run function
def main():

    # try:
        conn = psycopg.connect("postgresql://sum:M_                    -jRAw@mean-grivet-3788.8nk.cockroachlabs.
        cloud:26257/twitter_project_red?sslmode=ver
                               application_name="$ docs_simplecrud_psycopg3",
                               row_factory=namedtuple_row)

        # print('start')

        # del_table(conn)
        # delete_accounts(conn)
        # create_table(conn)
        # insert_record(conn, data1)
        print_record(conn)
        conn.commit()

        # print('finish')
    # except Exception as e:
    #     logging.fatal("database connection failed")
    #     logging.fatal(e)
    #     return
```

```
#run which function
if __name__ == "__main__":
    main()
```

Store Crawled Data

Database



CockroachDB

CockroachDB Ratings Overview

Review weighting ⓘ ☐ Reviewed in Last 12 Months ✉ Email Page

4.6 ★★★★½ 30 Ratings (All Time)

90% Would Recommend

Rating Distribution

5 Star	67%
4 Star	27%
3 Star	7%
2 Star	0%
1 Star	0%

Customer Experience

Evaluation & Contracting	4.5
Integration & Deployment	4.4
Service & Support	4.7
Product Capabilities	4.6

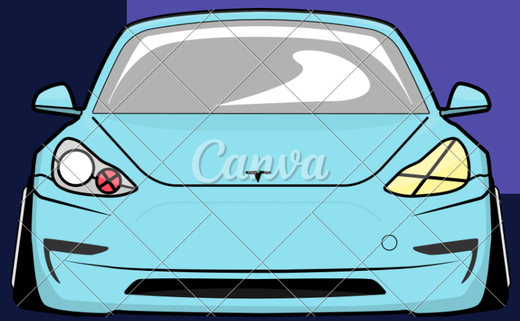
Demonstration

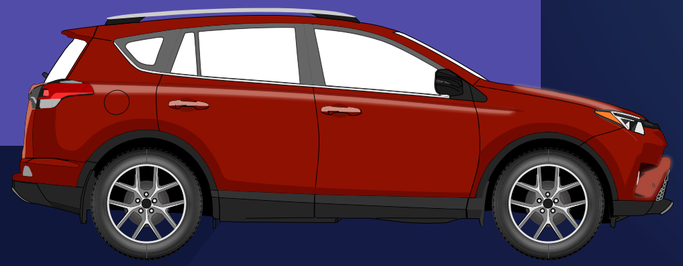
Car Brand Insight "Tesla vs Toyota"

TESLA

VS

TOYOTA



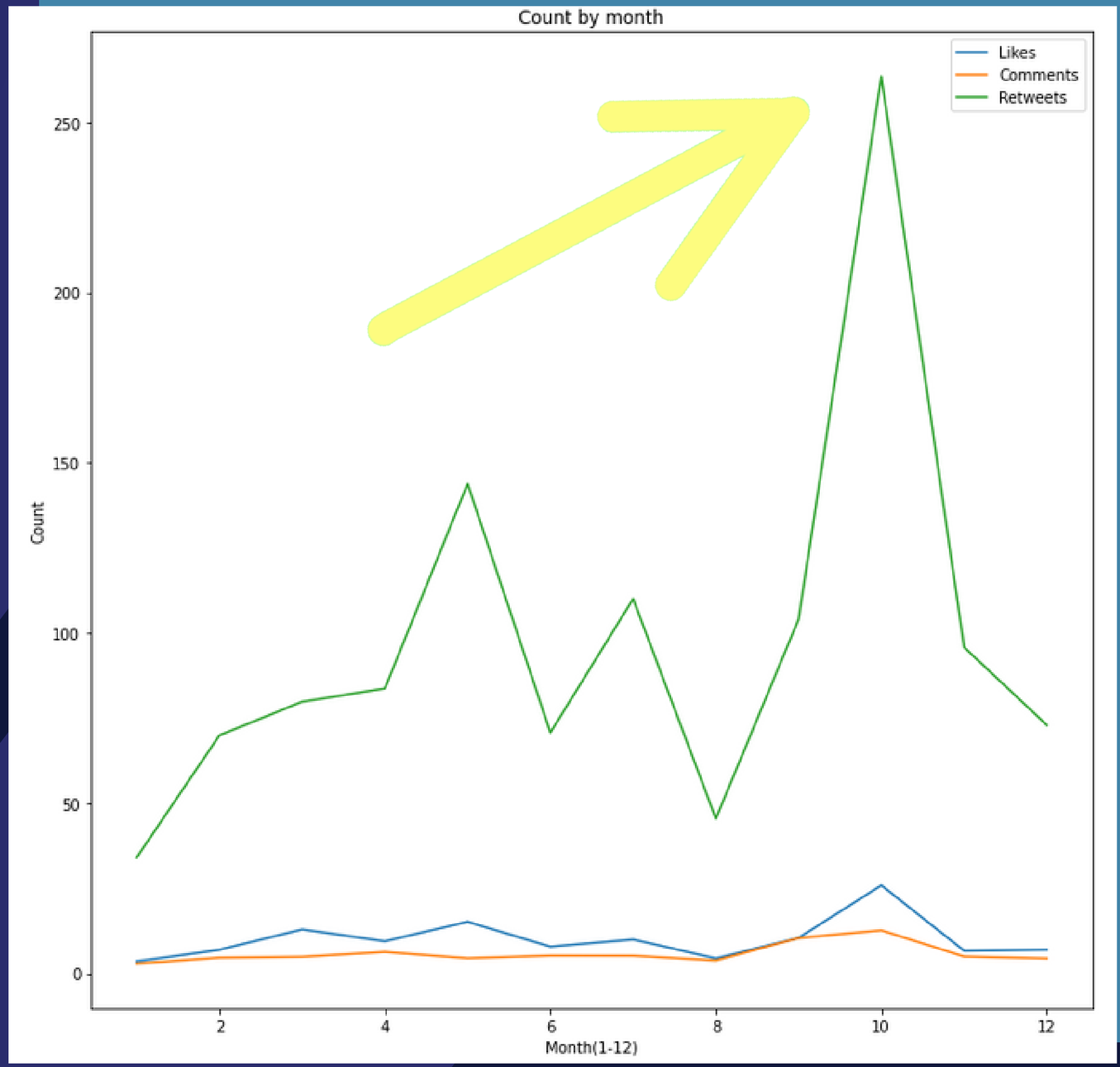


Tesla - Insight

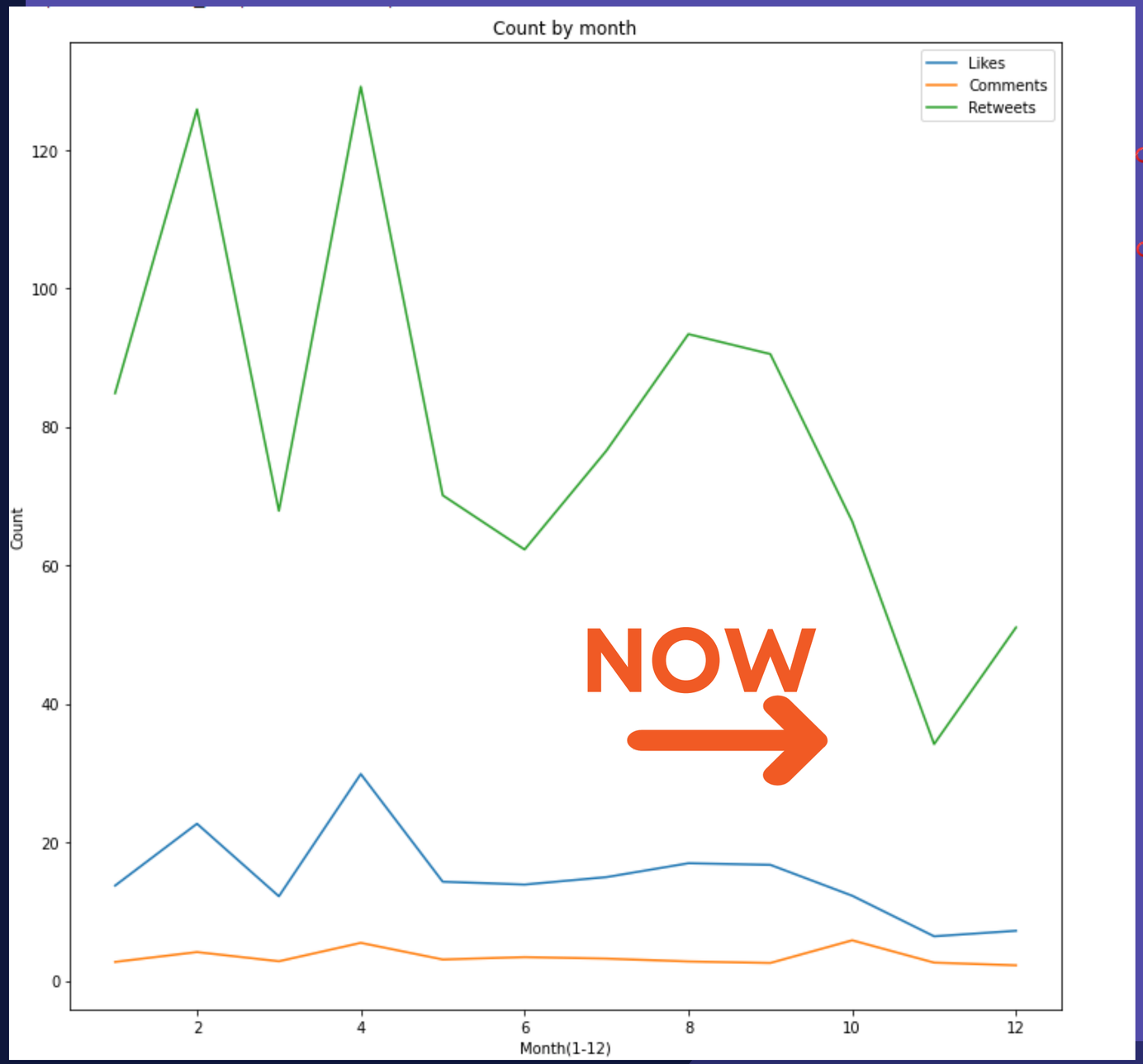
Analysis & Insight

Fake News Released

Increase Brand Volume



Analysis & Insight



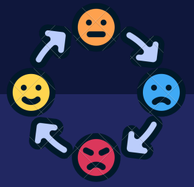
Toyota - Insight

LATEST NEWS

Inactive Than Tesla

While Tesla release new product
the tweet activity about Toyota is dropped

Benefit



Emotion Analysis

Huge Database

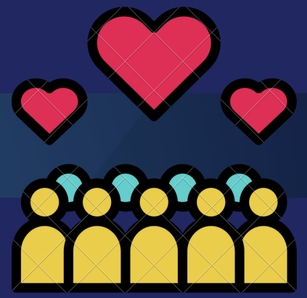
Analyze Any Keyword

Break Official Application Limit

Get Specific User Profile

Understand Tweet Contents Comments 、 Likes

Specify: Location, Period, Language, Hashtag



How News Affect People's impression

Limitation

1. Non official package (compared with Tweepy)
2. Google Chrome is required as scraping method
3. Cleaning datatype def function only suitable in English or Traditional Chinese User interface computer
4. Not fully automation
5. Serveral package required to installl

Reflection

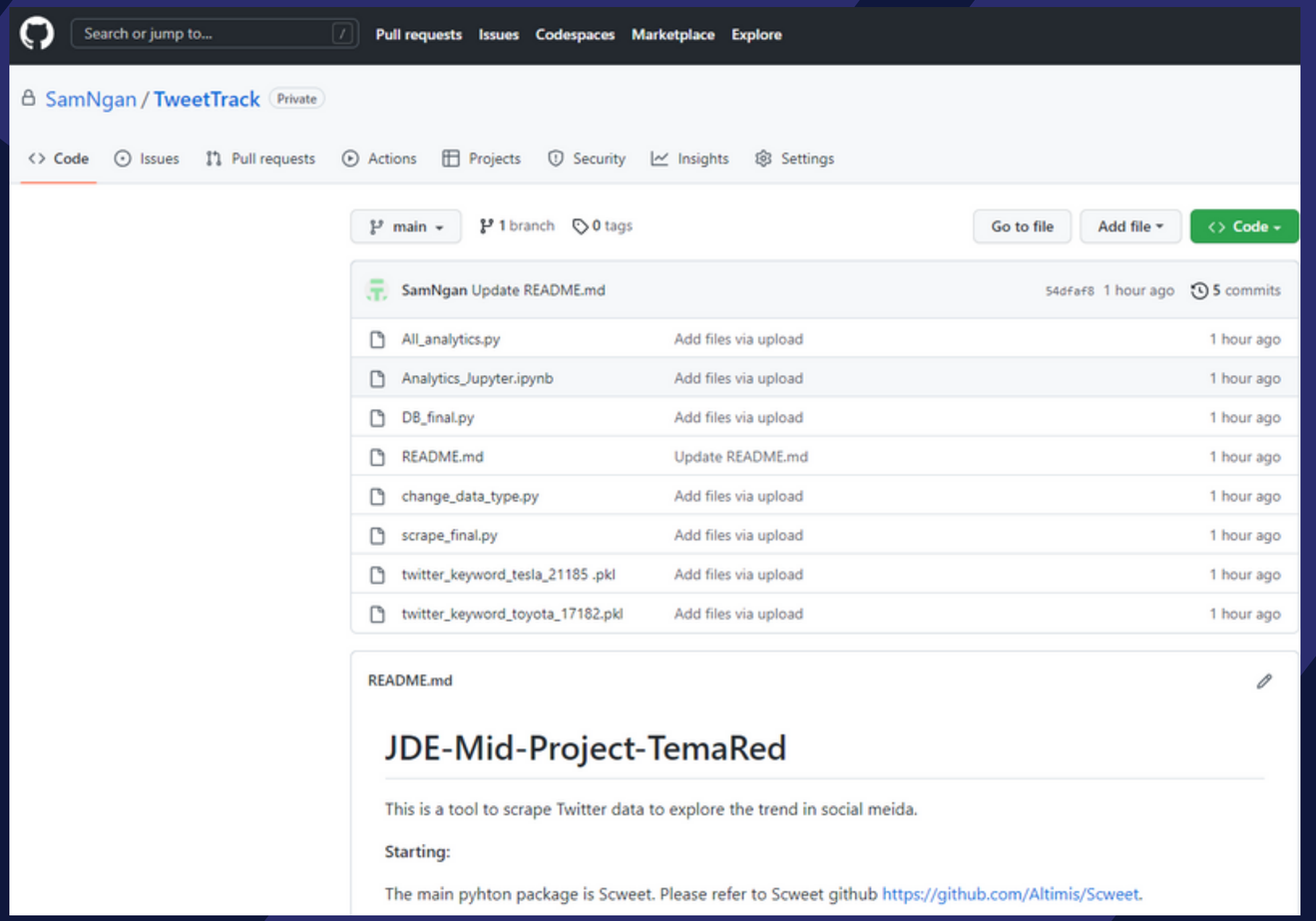
1.Use offical package to get exact data

2.Use more software(such as airflow) to reach fully automation

3.Social Media data shown the actvitiy of the community

Come visit our GitHub

https://github.com/SamNgan/TweetTrack.git



THANKYOU


Q|A