The project fetch real-time data from Twitter, Reddit and Youtude website corresponding to the 2019 movies from Wiki.

A. Twitter API

1. We harvested the 2019 data from wiki and imported the table to Jupyter to obtain other information and data through Python. The dataset consists of 144 movies and we attached the screen name, searched from Twitter, to the table manually. However, not all movies have official Twitter account. There are 100 screen names while there are 44 movies without screen name.

	Opening date	Title	Studio	Cast and crew	Genre	Country	Screen_name
0	Jan-04	Escape Room	Columbia Pictures	Adam Robitel (director); Bragi F. Schut, Maria...	Drama, Horror, Thriller, Mystery	US	Escape_Room
1	Jan-11	A Dog's Way Home	Columbia Pictures	Charles Martin Smith (director); W. Bruce Came...	Drama, Family	US	ADogsWayHome
2	Jan-11	The Upside	STX Entertainment / Lantern Entertainment	Neil Burger (director); Jon Hartmere (screenpl...	Comedy, Drama	US	TheUpsideFilm
3	Jan-11	Replicas	Entertainment Studios	Jeffrey Nachmanoff (director); Chad St. John (...	Sci-Fi, Thriller	US	replicas_movie
4	Jan-18	Glass	Universal Pictures / Buena Vista International...	M. Night Shyamalan (director/screenplay); Jame...	Superhero, Horror, Thriller	US	GlassMovie
5	Jan-25	The Kid Who Would Be King	20th Century Fox / Working Title Films	Joe Cornish (director/screenplay); Louis Ashbo...	Fantasy, Adventure, Family	UK, US	KidWouldBeKing
6	Jan-25	Serenity	Aviron Pictures / Global Road Entertainment	Steven Knight (director/screenplay); Matthew M...	Drama, Thriller	US	SerenityFilm

2. Assessing to Twitter API through tweepy package:

```
consumer_key = "LaOJGTHUiYOCkJvKrQvcsn5mE"
consumer_secret = "wTbZQKcDsQWyYOuWIrn0kBx71SuzAu2tJOqjYYHvQsvIPBaPnI"
access_token = "1151636618-uVTZH53QJ64V1Yw2sPXgWoFx3i1GwlWnHZntbvT"
access_token_secret = "bYOD512TWVXrLzCWz5pceojQWdi6RuHQOopccqf0TOeQ4"

api = twitter.Api(consumer_key=consumer_key,
                  consumer_secret=consumer_secret,
                  access_token_key=access_token,
                  access_token_secret=access_token_secret)

# Pass OAuth details to tweepy's OAuth handler
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
```

```
# Test whether access is successful
```

3. Define a function to convert the Twitter API time format to the format that SQL can read. We will convert all Twitter time data to the format yyyy-mm-dd hh:mm:ss

```
## the function that convets time to the time format that sql readable
def parse_datetime(value):
    time_tuple = parsedate_tz(value)
    timestamp = mktime_tz(time_tuple)

    return datetime.datetime.fromtimestamp(timestamp)
```

3. Retrieved official movie Twitter accounts' post and related attributes like, the count of favourites, Hashtags.etc., for each screen name through "GetUserTimline" and made the data into a table: 'official_movie_tw_post'. Be careful to skip none-screen-names.

```
for i, movie in enumerate(movies['Screen_name']):

    if movies['Screen_name'].isnull()[i]==True:
        continue
    results = api.GetUserTimeline(screen_name= movie, count=100)


    for each in results:
        status = json.loads(str(each))
        posts_12.append(status['text'])
        post_id_12.append(status['id'])
        favourites_12.append(each.favorite_count)

        retweets_12.append(each.retweet_count)
        time_12.append(parse_datetime(status['created_at']))
        screen_name_12.append(status['user']['screen_name'])

        try:
            hashtags_12.append(status['hashtags'][0]['text'])
        except:
            hashtags_12.append(status['hashtags'])
```

	Screen_name	Post	Post_id	Favourites	Retweets	Hashtags	Posted_time
0	Escape_Room	Get to know the amazing cast and check out a n...	1121524570408624128	12	3	[]	2019-04-25 17:21:09
1	Escape_Room	New phone, who dis? #NationalTelephoneDay 📞 ht...	1121486031671291904	8	3	NationalTelephoneDay	2019-04-25 14:48:00
2	Escape_Room	Centered on six strangers who must overcome li...	1121138475527315456	28	4	[]	2019-04-24 15:46:56
3	Escape_Room	Can I take a message? #AdminProfessionalsDay ☎...	1121095688706646017	5	1	AdminProfessionalsDay	2019-04-24 12:56:55
4	Escape_Room	Welcome to the Escape Room. Find the clues	1120849977494786048	6	3	EscapeRoomMovie	2019-04-23

4. Retrieved official movie Twitter accounts' information and related attributes like movie_id, user name, the count of followers, for each screen name through "GetUserTimline" and made the data into a table: 'movie_tw_info'. Be careful to skip none-screen-names.

```
for i, movie in enumerate(movies['Screen_name']):

    if movies['Screen_name'].isnull()[i]==True:
        continue
    results_1 = api.GetUser(screen_name= movie)


    movie_id_1.append(results_1.id)
    name_1.append(results_1.name)
    screen_name_1.append(results_1.screen_name)

    followers_1.append(results_1.followers_count)
    friends_count_1.append(results_1.friends_count)
    no_tweet_1.append(results_1.statuses_count)
    time_1.append(parse_datetime(results_1.created_at))
```
;

	Movie_id	Screen_name	User_name	Followers	Friends_count	tweets_count	Joined_time
0	1042668670617440256	Escape_Room	Escape Room	1730	5	101	2018-09-20 02:55:57
1	918927445029302272	ADogsWayHome	A Dog's Way Home	2401	8	163	2017-10-13 15:52:30
2	788766161852854273	TheUpsideFilm	The Upside	2434	20	957	2016-10-19 11:38:02
3	1074869646594068481	replicas_movie	ReplicasMovie	1633	303	341	2018-12-17 22:31:08
4	920922266503434241	GlassMovie	Glass	23849	7	135	2017-10-16 21:00:56
5	1001973055747338240	KidWouldBeKing	The Kid Who Would Be King	1466	19	278	2018-05-30 19:46:07

5. Made Cast and Crew into another table:

The “Cast and crew” column corresponding movie titles is the important information but it is not atomic Therefore, we need to split the names into separate rows corresponding to each movie, attached the jobs they did, like actors, directors, screenplay and set the cast and crew id to each row.

	Opening date	Title	Studio	Cast and crew	Genre	Country	Screen_name
0	Jan-04	Escape Room	Columbia Pictures	Adam Robitel (director); Bragi F. Schut, Maria...	Drama, Horror, Thriller, Mystery	US	Escape_Room
1	Jan-11	A Dog's Way Home	Columbia Pictures	Charles Martin Smith (director); W. Bruce Came...	Drama, Family	US	ADogsWayHome
2	Jan-11	The Upside	STX Entertainment / Lantern Entertainment	Neil Burger (director); Jon Hartmere (screenpl...	Comedy, Drama	US	TheUpsideFilm
3	Jan-11	Replicas	Entertainment Studios	Jeffrey Nachmanoff (director); Chad St. John (...	Sci-Fi, Thriller	US	replicas_movie
4	Jan-18	Glass	Universal Pictures / Buena Vista International...	M. Night Shyamalan (director/screenplay); Jame...	Superhero, Horror, Thriller	US	GlassMovie
5	Jan-25	The Kid Who Would Be King	20th Century Fox / Working Title Films	Joe Cornish (director/screenplay); Louis Ashbo...	Fantasy, Adventure, Family	UK, US	KidWouldBeKing
6	Jan-25	Serenity	Aviron Pictures / Global Road Entertainment	Steven Knight (director/screenplay); Matthew M...	Drama, Thriller	US	SerenityFilm

	cast_id	Name	Job	Movie
0	00101	Adam Robitel	director	Escape Room
1	00102	Bragi F. Schut	actor	Escape Room
2	00103	Maria Melnik	screenplay	Escape Room
3	00104	Logan Miller	actor	Escape Room
4	00105	Deborah Ann Woll	actor	Escape Room

6. Made Studios into a new table: M_to_S table:

In the same way as above, the “Studio” column corresponding movie titles is the important information but not atomic. Therefore, we need to split the studios into separate rows corresponding to each movie, and set the cast and crew id to each row.

```
# Make a new company table
studio = []
movie_3= []
M_to_S_id=[]
company= movies['Studio']
# Split studios corresponding to movies into different rows
for i, stu in enumerate(company):

    studio_list = stu.split(" / ")  ## studio list for each movie

    for j, s in enumerate(studio_list):
        studio.append(s)
        movie_3.append(movies['Title'][i])

        temp = chr(ord('a')+j)  ##creat id
        start_num= i+1
        M_to_S_id.append(str(start_num)+temp)
```

	M_to_S_id	Movie	Studio
0	1a	Escape Room	Columbia Pictures
1	2a	A Dog's Way Home	Columbia Pictures
2	3a	The Upside	STX Entertainment
3	3b	The Upside	Lantern Entertainment
4	4a	Replicas	Entertainment Studios

7. We searched for the screen name of each studio on Twitter. However, not all studios have official Twitter account, especially the studios outside of US. There are totally distinct 103 studios on list while there are 23 studios without screen name. (The table below does not need to import to database, just for attaching screen

name manually)

	Studio	Screen_name
0	Columbia Pictures	ColumbiaStudios
1	STX Entertainment	STXEnt
2	Lantern Entertainment	lantern_ent
3	Entertainment Studios	ESGlobalMedia

8. Retrieved studio Twitter accounts' information and related attributes like user name, the count of followers, corresponding to each studio screen name through "GetUserTimline" and made the data into a table: 'tw_studio_table'. Be careful to skip none-screen-names.

```
for i, company in enumerate(s_studio['Screen_name']):

    if s_studio['Screen_name'].isnull()[i]==True:
        continue
    results_3 = api.GetUser(screen_name= company)


    company_id.append(results_3.id)
    screen_name_3.append(results_3.screen_name)

    followers_3.append(results_3.followers_count)
    friends_count_3.append(results_3.friends_count)
    no_tweet_3.append(results_3.statuses_count)
    time_3.append(parse_datetime(results_3.created_at))
    ## Add  studio title correspoding the info to the new table
    company_name_3.append(s_studio['Studio'][i])
```

	Studio	Screen_name	Studio_id	Followers	Friends_count	tweets_count	Joined_time
0	Columbia Pictures	ColumbiaStudios	1465344716	1865	6	12	2013-05-28 14:15:24
1	STX Entertainment	STXEnt	2998868947	40281	206	1084	2015-01-27 20:15:53
2	Lantern Entertainment	lantern_ent	993917042343403520	61	2	7	2018-05-08 14:14:23
3	Entertainment Studios	ESGlobalMedia	34728131	2669	246	2231	2009-04-23 17:01:54
4	Universal Pictures	UniversalPics	21904217	3695593	704	13312	2009-02-25 14:14:14

9. Retrieved studio Twitter accounts' post and related attributes like, the count of favourites, Hashtags.etc., for each screen name through "GetUserTimline" and made the data into a table: 'studio_tw_post'. Be careful to skip none-screen-names.

```
for i, stu in enumerate(df_3['Screen_name']):

    results = api.GetUserTimeline(screen_name= stu, count=100)


    for each in results:
        status = json.loads(str(each))
        posts_32.append(status['text'])
        post_id_32.append(status['id'])
        favourites_32.append(each.favorite_count)

        retweets_32.append(each.retweet_count)
        time_32.append(parse_datetime(status['created_at']))
        screen_name_32.append(status['user']['screen_name'])

        try:
            hashtags_32.append(status['hashtags'][0]['text'])
        except:
            hashtags_32.append(status['hashtags'])
```

	Screen_name	Post	Post_id	Favourites	Retweets	Hashtags	Posted_time
0	ColumbiaStudios	Jeri Ryan joins the cast of Helix as Ilaria's ...	434954542367514625	12	12	[]	2014-02-16 02:36:58
1	ColumbiaStudios	With stunning performances that have wowed cri...	426329337600569344	9	6	[]	2014-01-23 07:23:29
2	ColumbiaStudios	She's always hot on the trail of danger. #Gwen...	426329060306722817	8	6	Gwensday	2014-01-23 07:22:23
3	ColumbiaStudios	Captain Phillips just landed on Blu-ray and Di...	426018623895789569	10	4	[]	2014-01-22 10:48:49
4	ColumbiaStudios	Who are your real-life heroes? #MonumentsMen h...	426018153626210304	8	2	MonumentsMen	2014-01-22 10:46:57

10. To see how users comment on movies, we retrieved Twiiter users' post and related attributes like, the count of favourites, Hashtags.etc., corresponding to each movie as the query-term through "GetSearch" and made the data into a table: 'tw_user_comment'. However, some movie terms, for example, 'Us' and 'After' are not suitable for searching comments because both are very common words in our daily life. You may obtain something not relating to movies. Therefore, we replace 'Us' and 'After' with 'Us%20movie' and 'After%20movie' as the terms respectively.

```
for i, title in enumerate(movies['Title']):
    if title is 'Replicas':
        results_4 = api.GetSearch(term='Replicas%20movie', count= 100)

    elif title is 'Glass':
        results_4 = api.GetSearch(term='Glass%20movie', count= 100)

    elif title is 'Serenity':
        results_4 = api.GetSearch(term='Serenity%20movie', count= 100)
```

	User	Movie	Followers	Post	Post_id	Hashtags	Posted_time	Favourites
0	Heather Wixson	Escape Room	19294	Hey y'all! #EscapeRoom is out on Blu/DVD today...	1120748112945901568	EscapeRoom	2019-04-23 13:55:47	59
1	CLL at UNC-CH SON	Escape Room	377	#FlashBackFriday to when we went to @bullcitye...	1121840033718263808	FlashBackFriday	2019-04-26 14:14:41	0
2	[illegible]	Escape	[illegible]	RT @BaptieGretchen: Today we	[illegible]	[]	2019-04-26	[illegible]

B. Reddit API

	Opening date	Title	Studio	Cast and crew	Genre	Country	Screen_name
0	Jan-04	Escape Room	Columbia Pictures	Adam Robitel (director); Bragi F. Schut, Maria...	Drama, Horror, Thriller, Mystery	US	Escape_Room
1	Jan-11	A Dog's Way Home	Columbia Pictures	Charles Martin Smith (director); W. Bruce Came...	Drama, Family	US	ADogsWayHome
2	Jan-11	The Upside	STX Entertainment / Lantern Entertainment	Neil Burger (director); Jon Hartmere (screenpl...	Comedy, Drama	US	TheUpsideFilm
3	Jan-11	Replicas	Entertainment Studios	Jeffrey Nachmanoff (director); Chad St. John (...	Sci-Fi, Thriller	US	replicas_movie
4	Jan-18	Glass	Universal Pictures / Buena Vista International...	M. Night Shyamalan (director/screenplay); Jame...	Superhero, Horror, Thriller	US	GlassMovie
5	Jan-25	The Kid Who Would Be King	20th Century Fox / Working Title Films	Joe Cornish (director/screenplay); Louis Ashbo...	Fantasy, Adventure, Family	UK, US	KidWouldBeKing
6	Jan-25	Serenity	Aviron Pictures / Global Road Entertainment	Steven Knight (director/screenplay); Matthew M...	Drama, Thriller	US	SerenityFilm

1. Based on the movie titles, we retrieved the data from Reddit under the subreddit—movies (under the category: movies.) Then, we made the data into a table.

	id	title	body	movie	score	comms_num	url	created
0	adjst5	Box Office Week: Aquaman is #1 again for third...	|*Rank*|*Title*|*Domestic Gross (Weekend) *|*Wo...	Escape Room	1322	695	https://www.reddit.com/r/movies/comments/adjst...	1.546881e+09
1	bae8r1	Movies similar to Escape Room	I watched the movie Escape Room the other day ...	Escape Room	12	25	https://www.reddit.com/r/movies/comments/bae8r...	1.554625e+09
2	ac84qv	Official Discussion: Escape Room [SPOILERS]	# Poll\n\n**If you've seen the film, please ra...	Escape Room	203	949	https://www.reddit.com/r/movies/comments/ac84q...	1.546571e+09
3	bf0bzk	What Did You Think Of: Escape	I got around to watching it last night, and it...	Escape Room	1	13	https://www.reddit.com/r/movies/comments/bf0bz...	1.555687e+09

2. Notice that the “created” column is the unix format of time. Therefore, we need to convert the form of time to that form that is readable.

```
# Fixing the date column("created")
def get_date(created):
    return dt.datetime.fromtimestamp(created)

_timestamp = df_1["created"].apply(get_date)
df_1 = df_1.assign(timestamp = _timestamp)

df_1= df_1.drop(columns='created')
```

	id	title	body	movie	score	comms_num	url	timestamp
0	adjst5	Box Office Week: Aquaman is #1 again for third...	|*Rank*|*Title*|*Domestic Gross (Weekend) *|*Wo...	Escape Room	1322	695	https://www.reddit.com/r/movies/comments/adjst...	2019-01-07 12:16:42
1	bae8r1	Movies similar to Escape Room	I watched the movie Escape Room the other day ...	Escape Room	12	25	https://www.reddit.com/r/movies/comments/bae8r...	2019-04-07 04:19:31
2	ac84qv	Official Discussion: Escape Room [SPOILERS]	# Poll\n\n**If you've seen the film, please ra...	Escape Room	203	949	https://www.reddit.com/r/movies/comments/ac84q...	2019-01-03 22:00:24
3	bf0bzk	What Did You Think Of: Escape Room?	I got around to watching it last night, and it...	Escape Room	1	13	https://www.reddit.com/r/movies/comments/bf0bz...	2019-04-19 11:21:23

C. YouTube API

In [1]:

```
# -*- coding: utf-8 -*-

import os

import google.oauth2.credentials

import google_auth_oauthlib.flow
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError
from google_auth_oauthlib.flow import InstalledAppFlow

# The CLIENT_SECRETS_FILE variable specifies the name of a file that contains
# the OAuth 2.0 information for this application, including its client_id and
# client_secret.
CLIENT_SECRETS_FILE = "client_secret.json"

# This OAuth 2.0 access scope allows for full read/write access to the
# authenticated user's account and requires requests to use an SSL connection.
SCOPES = ['https://www.googleapis.com/auth/youtube.force-ssl']
API_SERVICE_NAME = 'youtube'
API_VERSION = 'v3'
```

1. Use YouTube API V3 to get data through search a specific topic about our domain - movies. My google account and password was stored as json file so that I can use this API with authorized execute.

```
def get_authenticated_service():
  flow = InstalledAppFlow.from_client_secrets_file(CLIENT_SECRETS_FILE, SCOPES)
  credentials = flow.run_console()
  return build(API_SERVICE_NAME, API_VERSION, credentials = credentials)

def print_response(response):
  print(response)

# Build a resource based on a list of properties given as key-value pairs.
# Leave properties with empty values out of the inserted resource.
def build_resource(properties):
  resource = {}
  for p in properties:
    # Given a key like "snippet.title", split into "snippet" and "title", where
    # "snippet" will be an object and "title" will be a property in that object.
    prop_array = p.split('.')
    ref = resource
    for pa in range(0, len(prop_array)):
      is_array = False
      key = prop_array[pa]

      # For properties that have array values, convert a name like
      # "snippet.tags[]" to snippet.tags, and set a flag to handle
      # the value as an array.
      if key[-2:] == '[]':
        key = key[0:len(key)-2:]
        is_array = True

      if pa == (len(prop_array) - 1):
        # Leave properties without values out of inserted resource.
        if properties[p]:
          if is_array:
            ref[key] = properties[p].split(',')
          else:
            ref[key] = properties[p]
      elif key not in ref:
        # For example, the property is "snippet.title", but the resource does
        # not yet have a "snippet" object. Create the snippet object here.
        # Setting "ref = ref[key]" means that in the next time through the
        # "for pa in range ..." loop, we will be setting a property in the
        # resource's "snippet" object.
        ref[key] = {}
        ref = ref[key]
      else:
        # For example, the property is "snippet.description", and the resource
        # already has a "snippet" object.
        ref = ref[key]
  return resource

# Remove keyword arguments that are not set
```

```
# Remove keyword arguments that are not set
def remove_empty_kwargs(**kwargs):
  good_kwargs = {}
  if kwargs is not None:
    for key, value in kwargs.iteritems():
      if value:
        good_kwargs[key] = value
  return good_kwargs

def videos_list_most_popular(client, **kwargs):
  # See full sample for function
  kwargs = remove_empty_kwargs(**kwargs)

  response = client.videos().list(
    **kwargs
  ).execute()

  return print_response(response)


if __name__ == '__main__':
  # When running locally, disable OAuthlib's HTTPs verification. When
  # running in production *do not* leave this option enabled.
  os.environ['OAUTHLIB_INSECURE_TRANSPORT'] = '1'
  client = get_authenticated_service()

  videos_list_most_popular(client,
    part='snippet,contentDetails,statistics',
    chart='mostPopular',
    regionCode='US',
    videoCategoryId='20')
```

```
Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?respo
nse_type=code&client_id=784889702052-u7nmuacm8069moa361pvn2cqrd6u2n1d.apps.googleusercontent.com&red
irect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fyoutub
e.force-ssl&state=3QL39uxHfT5FhwwI5tUnIfOHFw4la1&prompt=consent&access_type=offline
Enter the authorization code: 4/FwHOScXjPuLtylcfAXAvxLDQhlcOlFqwS5bONJGx67-AOJz0b8mtCo0
{u'nextPageToken': u'CAUQAA', u'items': [{u'kind': u'youtube#video', u'statistics': {u'commentCoun
t': u'19714', u'viewCount': u'6503190', u'favoriteCount': u'0', u'dislikeCount': u'2601', u'likeCoun
t': u'147260'}, u'contentDetails': {u'definition': u'hd', u'projection': u'rectangular', u'caption':
 u'false', u'duration': u'PT1OM2S', u'licensedContent': True, u'dimension': u'2d'}, u'snippet': {u't
humbnails': {u'default': {u'url': u'https://i.ytimg.com/vi/b6yDfg1y-fA/default.jpg', u'width': 120,
 u'height': 90}, u'high': {u'url': u'https://i.ytimg.com/vi/b6yDfg1y-fA/hqdefault.jpg', u'width': 48
0, u'height': 360}, u'medium': {u'url': u'https://i.ytimg.com/vi/b6yDfg1y-fA/mqdefault.jpg', u'widt
h': 320, u'height': 180}, u'maxres': {u'url': u'https://i.ytimg.com/vi/b6yDfg1y-fA/maxresdefault.jp
g', u'width': 1280, u'height': 720}, u'standard': {u'url': u'https://i.ytimg.com/vi/b6yDfg1y-fA/sdde
```

2. After we authorized, python file will get the data which contains detailed information about movies we entered in the search area.