README

Project Setup:

Extract the zip file you will find `YouTubeSentimentAnalysis` folder which contains all the code.

Find `main.py` file inside project folder which is the starting point of the project.

Development Environment:

Python3, IDE (Visual Studio Code)

Pre-requisites:

Before you may start the execution, you need to setup the YouTube Data API and download the required python libraries.

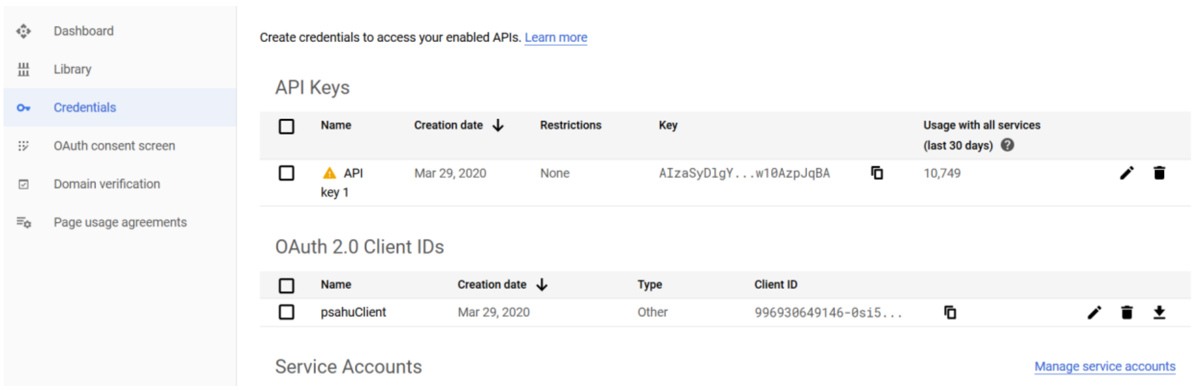
1. Setup YouTube Data API account:

YouTube provides very detailed guide to setup an account to get started using data API. Follow the link to setup your account.

Once you create the project now we have to obtain authentication credentials so that our application can submit API request

NOTE: It is recommended to use Syracuse GSuite id to setup YouTube Data API account as it is the enterprise version and does not require any background checks.

Once you successfully setup YouTube Data API, you will see the credential page as below.



Create credentials to access your enabled APIs. Learn more

API Keys

Name	Creation date ↓	Restrictions	Key	Usage with all services (last 30 days)
API key 1	Mar 29, 2020	None	AIzaSyDlgY...w10AzpJqBA	10,749

OAuth 2.0 Client IDs

Name	Creation date ↓	Type	Client ID
psahuClient	Mar 29, 2020	Other	996930649146-0si5...

Service Accounts

Manage service accounts

Note that you have to follow below two steps to enable our project to successfully authenticate API requests.

1. Copy the API Key 1 and replace APIKey at `<projectHome>/auth/keys.json`.
2. Download OAuth 2.0 Client IDs and replace `<projectHome>/auth/OauthKeys.json`

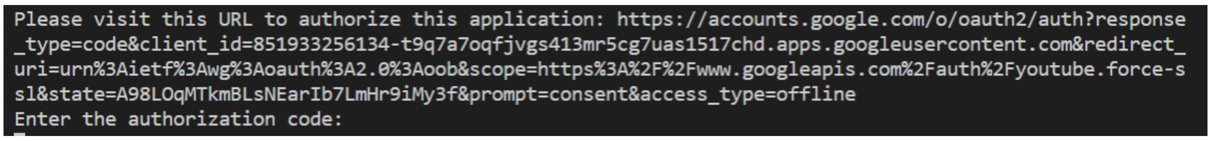
2. Python Libraries

We are using python 3.7 for the development. Download the following python3 libraries.

googleapiclient	google_auth_oauthlib
vaderSentiment	Sklearn
afinn	Json
keras	matplotlib
Itertools	nltk
seaborn	wordcloud

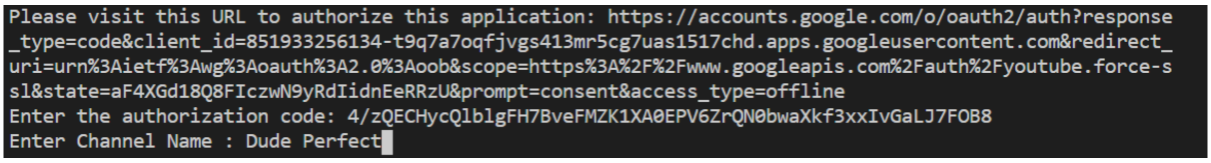
How to Run:

1. Navigate to `main.py` and execute the python file.
2. You will receive a prompt asking for authentication.



```
Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?response
_type=code&client_id=851933256134-t9q7a7oqfjvgs413mr5cg7uas1517chd.apps.googleusercontent.com&redirect_
uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fyoutube.force-s
sl&state=A98LOqMTkmBLsNEarIb7LmHr9iMy3f&prompt=consent&access_type=offline
Enter the authorization code:
```

3. Navigate to above URL and enter the authentication code. Then user would be asked to enter the channel name for which the user wants to run the sentiment analysis and predictions.



```
Please visit this URL to authorize this application: https://accounts.google.com/o/oauth2/auth?response
_type=code&client_id=851933256134-t9q7a7oqfjvgs413mr5cg7uas1517chd.apps.googleusercontent.com&redirect_
uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fyoutube.force-s
sl&state=aF4XGd18Q8FIczwN9yRdIidnEeRRzU&prompt=consent&access_type=offline
Enter the authorization code: 4/zQECHycQlblgFH7BveFMZK1XA0EPV6ZrQN0bwaXkf3xxIvGaLJ7FOB8
Enter Channel Name : Dude Perfect
```

   Note that by default we are scrapping the 2500 comments of each 200 videos. You may configure this in `<projectHome>/constants.json` file.

Once the execution is complete, we are persisting the scraped data for further use, so that once you run out of the API quota for the day, you may continue scraping data from where you left. In addition, once you have the data, you can just choose to run the sentiment analysis and prediction algorithms. You may find `invoker.py` in `<projectHome>`. In below section we will list down where we are persisting `data/results`.

1. When you will successfully complete the execution, you will find below files in `<projectHome>/comments` folder.
   a. `<channelName>_vidlist.json` – contains a list of videos we scraped for the channel
   b. `<channelName>_comment_score.json` – contains list of comments , last update date and their polarity score for Vader, Afinn and NRC.
   c. `<channelName>_stats.json` – saves the sentiment score of each video against 3 sentiment models – Vader, Afinn, NRC Lexicon and other video statistics like view count, comment count, likes and dislikes.

2. We are also saving the sentiment analysis results in human readable form under `<projectHome>/sentimentAnalysis` directory.
3. Various graphs and plots are saved at `<projectHome>/Images`

Baseline analysis:

For baseline analysis execute `<projectHome>/baselinePerformanceAnalysis.py`. You may find the 1200 labelled comments at `<projectHome>/data/labelled_comments.csv`

Briefing on File content:

1. `main.py` – Driver File (Start point)
2. `createTimeSeriesData.py` – Transform data to time series
3. `extractComments.py` – scrap comments for videos
4. `getVideoIds.py` – asks user for channel name, retrieves userId and then scrapes video snippets.
5. `getVideoStatistics.py` – fetch video statistics like view count, comment count, likes and dislikes.
6. `invoker.py` – invokes visualization for existing json files.
7. `mapper.py` – maps video sentiments and stats into single object
8. `predictionModels.py` – used for additional analysis and predicting like/dislike ratio.
9. `predictionTimeSeriesModels.py` – perform prediction on time series data and visualization.
10. `sentiment_afinn.py` – performs sentiment analysis using Afinn
11. `sentiment_NRC.py` – performs sentiment analysis using NRC Lexicon
12. `sentiment_vader.py` – performs sentiment analysis using Vader.
13. `visualisations.py` – all sentiment analysis visualizations and word cloud.