**SOCIAL MEDIA ANALYSIS – ANALYTIC CODES**

**instagram\_analysis.py**

Main function : instagram\_analysis

Sub functions

**fetch**

Here the instagram scraped data will be fetched from MongoDB collection.

uri = “mongodb://localhost:27017”, mydb = “Manual\_Data\_Push\_For\_Insta”

**transform**

Here data cleaning and the analysis process will happen. This part will read excel lookup files of each insta handle. These lookup files are created to filter only the related posts form the scraped data, since the scraping part will fetch all the posts data from the handles.

(Ex:**Insta\_PS1\_lookup\_madras\_talkies.xlsx,Insta\_PS1\_lookup\_lyca\_productions.xlsx**),

Lookup files are created for each insta handles separately by manually visiting each insta profiles and copying the url of only related the posts.

**insta\_config.ini** file contains the cast insta id and their modified names.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**facebook\_analysis.py**

Main function : facebook\_analysis

Sub functions

**fetch**

Here the facebook scraped data will be fetched from MongoDB collection.

uri = “mongodb://localhost:27017”, mydb = “Manual\_Data\_Push\_For\_FB”

**transform**

Here data cleaning and the analysis process will happen. This part will read excel fb lookup file. These lookup file is created to filter only the related posts form the scraped data, since the scraping part will fetch all the posts data from the FB pages . Lookup file **“FB\_lookup”** is created by manually checking all the unique url’s from the collection and copying the url of only related the posts.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**twitter\_analysis\_daily.py**

* This code will give daily level twitter insights.

Main function : twitter\_analysis\_daily

Sub functions

**fetch**

Here data will be fetched from twitter MongoDB collections with respect to tag name.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Here data cleaning and the analysis process will happen.

**tw\_config.ini** file keyword to differentiate each tag.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**youtube\_analysis\_daily.py**

* This code will gives daily level youtube insights.

Main function : youtube\_analysis\_daily

Sub functions

**fetch**

Here data will be fetched from youtube MongoDB collections with respect to tag name.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Here data cleaning and the analysis process will happen.

Language name for the tag is given manually by checking the each video id from the post.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**overall\_nps\_entity.py**

* In the youtube and twitter tables we will have nps and entity score with respect to each tags differently. This code will gives us Overall nps and Overall Entity score for Youtube and Twitter.And also the Youtube video counts.

This code should executes after twitter\_analysis\_daily.py and yputube\_analysis\_daily.py.

Main function : overall\_nps\_entity

Sub functions

**fetch**

Here the required data will be fetcjed from Database postgre tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**transform**

Here data cleaning and the analysis process will happen.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**wrapper\_overall.py**

overall\_nps\_entity.py is dependent on twitter\_analysis\_daily.py and youtube\_analysis.py.

It should execute only after the completion of these 2 codes. To make sure that overall\_nps\_entity.py will executes only after twitter\_analysis\_daily.py and youtube\_analysis\_daily.py this wrapper\_overall.py file is written.

**twitter\_analysis\_weekly.py**

* This code will give weekly level twitter insights.

Main function : twitter\_analysis\_weekly

Sub functions

**fetch**

Here data will be fetched from twitter MongoDB collections with respect to tag name.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Here data cleaning and the analysis process will happen.

**tw\_config.ini** file keyword to differentiate each tag.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**youtube\_analysis\_week.py**

* This code will gives daily level youtube insights.

Main function : youtube\_analysis\_weekly

Sub functions

**fetch**

Here data will be fetched from youtube MongoDB collections with respect to tag name.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Here data cleaning and the analysis process will happen.

Language name for the tag is given manually by checking the each video id from the post.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**twitter\_competitor\_analysis\_daily.py**

* This code will give daily level twitter competitor insights.

Main function : twitter\_competitor\_analysis\_daily

Sub functions

**fetch**

Here data will be fetched from twitter MongoDB collections with respect to competitor tag name.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Here data cleaning and the analysis process will happen.

**tw\_config.ini** file keyword to differentiate each competitor tag.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**twitter\_competitor\_analysis\_weekly.py**

* This code will give weekly level twitter competitor insights.

Main function : twitter\_competitor\_analysis\_weekly

Sub functions

**fetch**

Here data will be fetched from twitter MongoDB collections with respect to competitor tag name.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Here data cleaning and the analysis process will happen.

**tw\_config.ini** file keyword to differentiate each tag.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**insta\_cast\_analysis.py**

Main function : insta\_cast\_analysis

Sub functions

**fetch**

Here the instagram scraped data will be fetched from MongoDB collection.

uri = “mongodb://localhost:27017”, mydb = “Manual\_Data\_Push\_For\_Insta”

**transform**

Here data cleaning and the analysis process will happen. This part will read excel lookup files of each cast insta handle. These lookup files are created to filter only the related posts form the scraped data, since the scraping part will fetch all the posts data from the handles.

(Ex:**Insta\_PS1\_lookup\_trisha.xlsx,Insta\_PS1\_lookup\_karthi.xlsx**),

Lookup files are created for each cast insta handles separately by manually visiting their insta profiles and copying the url of only related the posts. (These are the same lookup sheets using in instagram\_analysis.py also)

**insta\_config.ini** file contains the cast insta id and their modified names.

The lookup files should get updated every time before the execution of the code.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**twitter\_cast\_analysis.py**

* This code will give cast level twitter insights.

Main function : twitter\_cast\_analysis

Sub functions

**fetch\_and\_transform**

Here data will be fetched from twitter MongoDB collections with respect to competitor tag name data cleaning and the analysis process will happen.

The data will be fetched by reading the excel files. These excel files contains the tweets made by the particular cast. For each cast we have separate csv files(Ex : “Karthi\_All\_Tweets.csv”, “Trisha\_All\_Tweets.csv”). To generate these csv files we have to run “**TwitterCastAnalysis.ipynb**” file separately for each celebrity and handle once and have to save it as csv file then have to push those csv files to the twitter\_cast\_analysis.py file location.

The celebrity tweets csv files should get updated every time before the execution of the code.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**celebrity\_tack.py**

This will give the celebrity level insights by combining insta\_cast\_analysis.py and twitter\_cast\_analysis.py. This code should get executed only after the insta\_cast\_analysis.py and twitter\_cast\_analysis.py

Main function : celebrity\_tracker

Sub functions

**fetch**

Here required data is fetched from Postgre DB tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**transform**

Here data cleaning and the analysis process will happen.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**twitter\_media\_handles\_analysis.py**

* This code will give the twitter level media peoples insights.

This code should get executed only after twitter\_cast\_analysis.py

Main function : twitter\_media\_handles\_analysis

Sub functions

**Fetch**

Here media peoples data will be fetched from twitter MongoDB collections.

uri = “mongodb://localhost:27017”, mydb = “socialmedia”

**transform**

Data cleaning and the analysis process will happen.

**pushdata**

This part will push the required outputs to the Database tables.

database=’psone\_flash\_base’, user=’psone\_flash\_base’, password=’psone’,host=’kalacitra.in’

**wrapper\_celebrity.py**

celebrity\_track.py and twitter\_media\_handles\_analysis.py are dependent on twitter\_cast\_analysis.py and insta\_cast\_analysis.py.

These 2 codes should execute only after the completion twitter\_cast\_analysis.py and insta\_cast\_analysis.py. To make sure that celebrity\_track.py and twitter\_media\_handles\_analysis.py will executes only after twitter\_cast\_analysis.py and insta\_cast\_analysis.py this wrapper\_celebrity.py file is written.