Twitter Data Source:

1. Run twitter streaming API to collect offensive tweets data, clean, save to local disk
   Output File: hate_tweets.txt
   Output format: <screen_name \t user_id \t followers_count \t tweet~|\n>
   Command: python twitter_streaming.py

2. Run twitter REST API to collect, clean, store recent tweets by offensive users recorded in hate_tweets.txt
   Output File: hate_user_tweets.txt
   Output Format: <screen_name \t tweet1>
   <screen_name \t tweet2>...
   <screen_name \t tweetN>
   Command: python twitter_rest_tweets.py

3. Move data from Local Disk to HDFS.
   hdfs dfs -mkdir tweets/
   hdfs dfs -mkdir tweets/followers
   hdfs dfs -put hate_tweets.txt tweets/followers/hate_tweets.txt
   hdfs dfs -put hate_user_tweets.txt tweets/hate_user_tweets.txt

4. Run MR (HateScore) code. Job processes all tweets to find offensive tweets and subsequent “score” for each user.
   Command: hadoop jar HateScoreFinalVersion.jar HateScore tweets/hate_user_tweets.txt tweets/output
   Output File: tweets/output/part-r-00000
   Output Format: <screen_name \t hate_score>\n

5. Run Hive code to create blacklist of offensive twitter users.
   Command: hive -f blacklist.sql
   Output File: tweets/hiveoutput/000000_0
   Output Format: <screen_name \t hate_score>0.3 \t followers_count> (sorted by hatescore and count descending)

6. Move HDFS output to local Disk. Run twitter REST API to obtain twitter profiles blacklist users follow.
   Command: python twitter_rest_friends.py
   Output File: hate_user_friends.txt
   Output Format: <screen_name \t friend1>
   <screen_name \t friend2>...
   <screen_name \t friendN>

7. Run MR (Hate Friends) code to obtain the most popular twitter profiles followed amongst these offensive users.
   Cmd: hadoop jar HateFriendsFinalVersion.jar HateFriends tweets/hate_user_friends.txt tweets/mr2output/
   Output File: tweets/mr2output/part-r-00000
   Output Format: <profile_id \t count>\n

8. Run HIVE to get above list sorted based on count. This is the END RESULT.
   Command: hive -f groups.sql
   Output File: tweets/hiveoutput2/000000_0
   Output Format: <profile_id \t count>\n (sorted)

Reddit Data Source:
(require sqlite3 in linux)

1. Download Kaggle Reddit DataSource into Cloudera VM.
   Link - https://www.kaggle.com/reddit/reddit-comments-may-2015
2. Run following sqlite commands to Extract required data + clean it before storing in HDFS.

   ```
   $ sqlite3 database.sqlite
   sqlite> create view heavy_view as select subreddit, score, body from May2015 where score > 100;
   sqlite> create view heavy_view1 as select subreddit, score, replace(body,'\n',' ') as body from heavy_view;
   sqlite> create view heavy_view2 as select subreddit, score, replace(body,'\t',' ') as body from heavy_view1;
   sqlite> create view heavy_view3 as select subreddit, score, replace(body,',',' ') as body from heavy_view2;
   sqlite> create view heavy_view4 as select subreddit, body from heavy_view3 order by score desc;

   sqlite> .mode tabs
   sqlite> .output reddit_comments.txt
   sqlite> select subreddit, body, '~|' from heavy_view4;
   sqlite> .quit
   ```

3. Move data from Local Disk to HDFS.
   ```
   hdfs dfs -mkdir reddit/
   hdfs dfs -put reddit_comments.txt   reddit/reddit_comments.txt
   ```

4. Run SAME MR (HateScore) code. Job processes all comments to find offensive comments and subsequent “score” for each subreddit.
   Command: hadoop jar HateScoreFinalVersion.jar HateScore reddit/reddit_comments.txt reddit/output/
   Output File: reddit/output/part-r-00000
   Output Format: <subreddit_name \t hate_score>\n

5. Run following HIVE commands to get above list sorted based on count. This is the END RESULT.
   Commands:
   beeline -u jdbc:hive2://quickstart:10000/default -n cloudera -d org.apache.hive.jdbc.HiveDriver
   create external table hatereddits (subreddit_name string, hate_score decimal(5,3)) row format delimited fields terminated by '\t' location '/user/cloudera/reddit/output’;

create table hatereddits_sorted as select subreddit_name, hate_score from hatereddits order by hate_score desc;

insert overwrite directory ‘reddit/hiveoutput/' select concat(subreddit_name,'\t',hate_score) from hatereddits_sorted;

   Output File: reddit/hiveoutput/000000_0
   Output Format: <subreddit_name \t hate_score>\n (sorted)