1. **Preprocessing:**

   Written a script that will preprocess the file.
   - Merge 3 lines of user's info (T, U & W) to 1 line and separate them by ",".
   - Remove empty tweets
   - Remove tweets with No Post Title
   - Remove special chars from the tweets other than alphanumeric and @# but this happening in mapper

   After preprocessing added the preprocessed file to hdfs

```
T 2009-06-07 02:07:42,U zaibatsu,W rt @agungp Twitter to roll out Verified Accounts this summer
T 2009-06-08 21:49:34,U alphaexe,W @lessallan I just played give it away in response to your tweet
T 2009-06-08 21:49:36,U digitalcrimeinv, W Apple Readies Snow Leopard for September, Will Charge 29
T 2009-06-08 21:49:36,U gabrielphb,W @LeOzInHoxP Ai cara vamo faze uma parceria entre nossos blogs Qualquer coisa me add no ms
T 2009-06-08 21:49:36,U gingerkid416,W @msali_sobb how work was today miss me
T 2009-06-08 21:49:36,U itscarol26,W de tanto meu irmão cantar smelly cat, eu fiquei com essa música na cabeça hahahaha meu ir
T 2009-06-08 21:49:36,U momfluential,W Just in case you missed my Celebrity Retail Therapy Du Jour. Go shop yourselves sane
T 2009-06-08 21:49:36,U ps_led_zeppelin,W Me duele la panza
```

2. **Part1 -** ***Find the users that tweet the most.***

   - For this task, I have written a custom writable class TwitterUser which will info of the user like a list of tweets, list of timestamps of that tweets and username and the total count of the tweet
   - I have added a mapper and reducer which keep track of the total tweet count for the user and also will give the top 5 users with the most tweets

For running this MapReduce job you need to give the command:
yarn jar **mr-1.0.jar** edu.usfca.cs.mr.part_1_user_withmost_tweets.UserWithMostTweetsJob **/50_06_pre.txt /test01**

where {mr-1.0.jar → jar name,s_pre6.txt → input file already on hdfs, test01→ output_dir of hdfs}

{input → 50k lines from a tweets2009-06.txt file and preprocessed} The output of the top 5 Twitter users with a list of their tweets and timestamp.

3. **Part2 -** ***Find the top 5 hashtags for each week.***

- For this task, I have written two custom writable class HashTag and HashTagByWeek which will have info about hashtags for each week
- I have added a mapper and reducer which keep track of the total hashtag count for the week and also will sort and give the top 5 hashtags for each week. I have added the key of week_year to group them together.

For running this MapReduce job you need to give the command:
yarn jar **mr-1.0.jar** edu.usfca.cs.mr.part_2_top_hashtag.TopHashtagJob /**50_06_pre.txt** /**test02**

where {mr-1.0.jar → jar name,s_pre6.txt → input file already on hdfs, test02→ output_dir of hdfs}

{input → 50k lines from a tweets2009-06.txt file and preprocessed} The output of the top 5 hashTags for each week.

4. **Part3 a) -** ***Sentiment analysis to calculate the overall sentiment associated with the users***

    - For this task, I have used a dataset AFINN.txt which has a list of positive and negative words with sentiment counts attached to each of them
    - I have added a mapper and reducer which keep track of the total sentiment count for each tweet.

For running this MapReduce job you need to give the command:
yarn jar **mr-1.0.jar**
edu.usfca.cs.mr.part_3_sentiment_analysis.TwitterUserSentimentAnalysisJob **/AFINN-111.txt /s_pre6.txt /test03**

where {mr-1.0.jar → jar name, AFINN-111.txt → dataset which contains a list of words with their sentiments, s_pre6.txt → input file already on hdfs, test03→ output_dir of hdfs}

{input → 30k lines from a tweets2009-06.txt file and preprocessed} The output contains a list of sentiment counts for each tweet by the user.

Part3 b) -***Sentiment analysis to calculate the overall sentiment of hashtags.***

- For this task, I have used a dataset AFINN.txt which has a list of positive and negative words with sentiment counts attached to each of them
- I have added a mapper and reducer which keep track of the total sentiment count for each hashTag.

For running this MapReduce job you need to give the command:
yarn jar **mr-1.0.jar** edu.usfca.cs.mr.part_3_sentiment_analysis.HashTagSentimentAnalysisJob **/AFINN-111.txt /50_06_pre.txt /test03**

where {mr-1.0.jar → jar name, AFINN-111.txt → dataset which contains a list of words with their sentiments, s_pre6.txt → input file already on hdfs, test03→ output_dir of hdfs}

```
#ass     -4
#comedy  1
#crazy   -2
#fail    -6
#fun     4
#funny   4
#lol     18
#love    6
#odd     -4
#rofl    4
#safety  1
#sexy    3
#sunshine        2
#wtf     -4
```

{input → 50k lines from a tweets2009-06.txt file and preprocessed} The output contains a list of sentiment counts for each hashtag.

5. **Part4 -** ***Implemented top 5 mentioned user each year***
    - For this task, I am getting the list of top 5 users who were most mentioned like @userName.
    - I have added a mapper and reducer which keep track of the total count of Top mentioned user for each year.

For running this MapReduce job you need to give the command:

yarn jar **mr-1.0.jar** edu.usfca.cs.mr.part_4_top_mentioned_user.TopMentionedUserJob /**50_06_pre.txt /test04**

where {mr-1.0.jar → jar name,s_pre6.txt → input file already on hdfs, test04→ output_dir of hdfs}

```
2009    UserName: @debruyndesign ->  Total number of Mentions :2
2009    UserName: @ituneiphone ->  Total number of Mentions :2
2009    UserName: @pedrojimenez ->  Total number of Mentions :2
2009    UserName: @Advocates4Youth ->  Total number of Mentions :2
2009    UserName: @RuthZ ->  Total number of Mentions :2
```

{input → 50k lines from a tweets2009-06.txt file and preprocessed} The output contains a list of the top 5 mentioned users counts for each year.

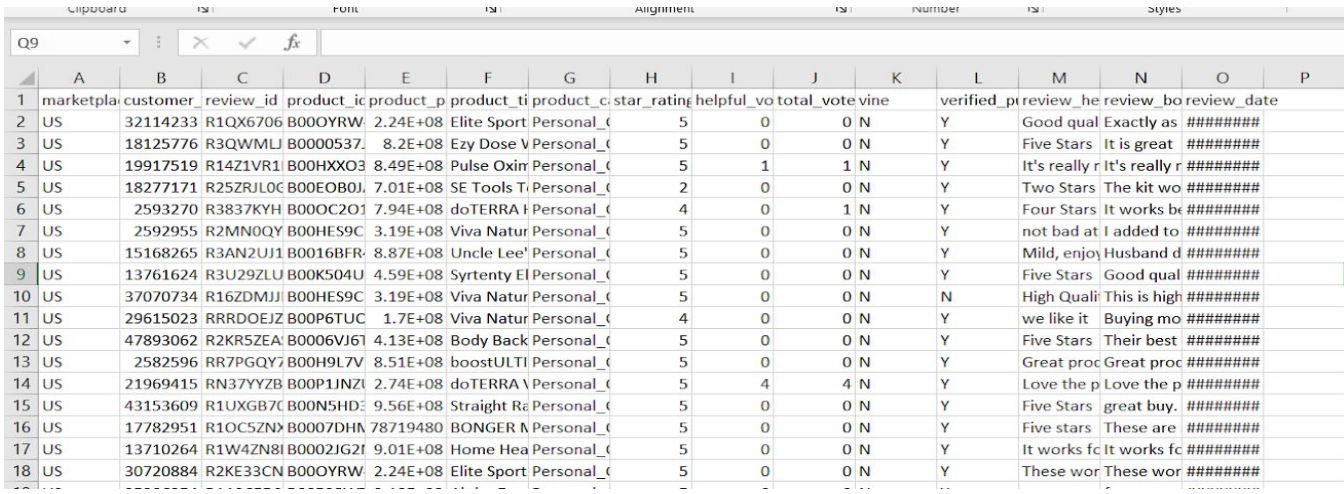
6. **Part5 -** ***Final project (Amazon users reviews dataset) 12GB***

Dataset:

- Dataset: Amazon all product review dataset is around 12 GB (https://snap.stanford.edu/data/web-Amazon.html)

- But for this assignment, I am working on a subset of the complete dataset, "**amazon_reviews_us_Personal_Care_Appliances_v1_00"** which is 50 MB. I am analyzing reviews related to the product that belongs to the personal care appliances category.

- The dataset contains the following important info.
  Customer_Id: Customer who has given the review
  Review_id: Review Id for each review
  Product_id: product id of that product
  Product title: Short title of the product
  Star_rating: Rating of that product



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	marketpla	customer_	review_id	product_ic	product_p	product_ti	product_c	star_rating	helpful_vo	total_vote	vine	verified_p	review_he	review_bo	review_date	
2	US	32114233	R1QX6706	B00OYRW	2.24E+08	Elite Sport	Personal_C	5	0	0	N	Y	Good qual	Exactly as	########	
3	US	18125776	R3QWMLJ	B0000537	8.2E+08	Ezy Dose V	Personal_C	5	0	0	N	Y	Five Stars	It is great	########	
4	US	19917519	R14Z1VR1	B00HXXO3	8.49E+08	Pulse Oxin	Personal_C	5	1	1	N	Y	It's really r	It's really r	########	
5	US	18277171	R25ZRJL0C	B00EOB0J	7.01E+08	SE Tools T	Personal_C	2	0	0	N	Y	Two Stars	The kit wo	########	
6	US	2593270	R3837KYH	B00OC2O1	7.94E+08	doTERRA I	Personal_C	4	0	1	N	Y	Four Stars	It works be	########	
7	US	2592955	R2MN0QY	B00HES9C	3.19E+08	Viva Natur	Personal_C	5	0	0	N	Y	not bad at	I added to	########	
8	US	15168265	R3AN2UJ1	B0016BFR	8.87E+08	Uncle Lee'	Personal_C	5	0	0	N	Y	Mild, enjoy	Husband d	########	
9	US	13761624	R3U29ZLU	B00K504U	4.59E+08	Syrtenty El	Personal_C	5	0	0	N	Y	Five Stars	Good qual	########	
10	US	37070734	R16ZDMJJ	B00HES9C	3.19E+08	Viva Natur	Personal_C	5	0	0	N	N	High Quali	This is high	########	
11	US	29615023	RRRDOEJZ	B00P6TUC	1.7E+08	Viva Natur	Personal_C	4	0	0	N	Y	we like it	Buying mo	########	
12	US	47893062	R2KR5ZEA	B0006VJ6T	4.13E+08	Body Back	Personal_C	5	0	0	N	Y	Five Stars	Their best	########	
13	US	2582596	RR7PGQY7	B00H9L7V	8.51E+08	boostULTI	Personal_C	5	0	0	N	Y	Great proc	Great proc	########	
14	US	21969415	RN37YYZB	B00P1JNZU	2.74E+08	doTERRA \	Personal_C	5	4	4	N	Y	Love the p	Love the p	########	
15	US	43153609	R1UXGB7C	B00N5HD3	9.56E+08	Straight Ra	Personal_C	5	0	0	N	Y	Five Stars	great buy.	########	
16	US	17782951	R1OC5ZNX	B0007DHM	78719480	BONGER M	Personal_C	5	0	0	N	Y	Five stars	These are	########	
17	US	13710264	R1W4ZN8I	B0002JG2I	9.01E+08	Home Hea	Personal_C	5	0	0	N	Y	It works fc	It works fc	########	
18	US	30720884	R2KE33CN	B00OYRW	2.24E+08	Elite Sport	Personal_C	5	0	0	N	Y	These wor	These wor	########	

Preprocessing:

- Dataset was provided in the TSV format so the first thing I did convert it to CSV comma-separated file for easy processing.
- Also, I tried removing reviews with empty product title
- Most of the dataset was clean so not much needed in the preprocessing step

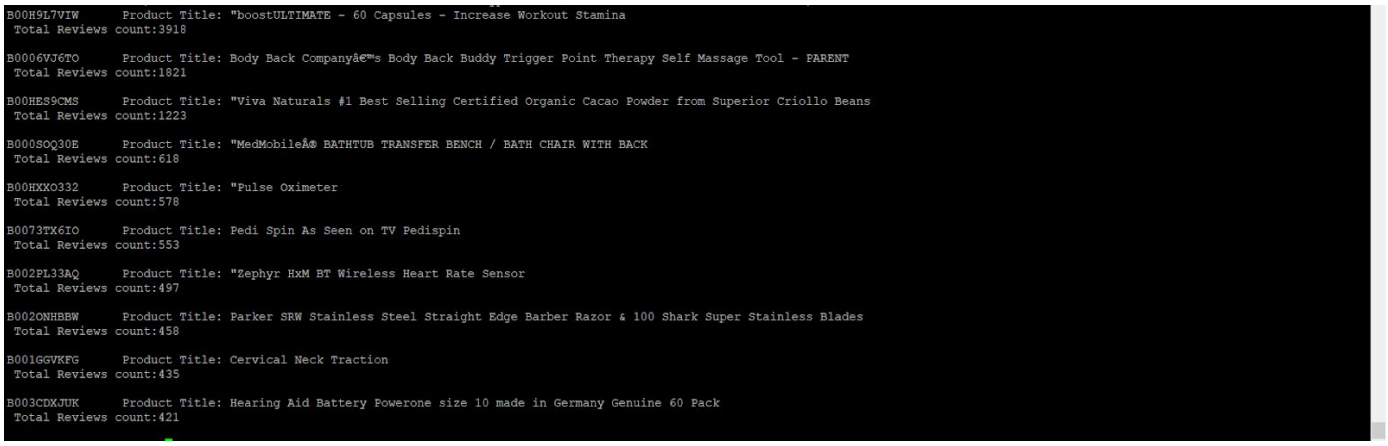
Map Reduce 1: ***Amazon Most Reviewed Product [personal_appliances category]***

- This MapReduce job will display the top 5 most reviewed product that belongs to personal appliances category from this dataset

For running this MapReduce job you need to give the command:

yarn jar **mr-1.0.jar**
edu.usfca.cs.mr.part_5_final_project_amazon_reviews.AmazonMostReviewedProductJob
/**amazon_reviews_us_Personal_Care_Appliances_v1_00.csv /test05**

where {mr-1.0.jar → jar name,amazon_reviews_us_Personal_Care_Appliances_v1_00.csv → input file already on hdfs, test05→ output_dir of hdfs}



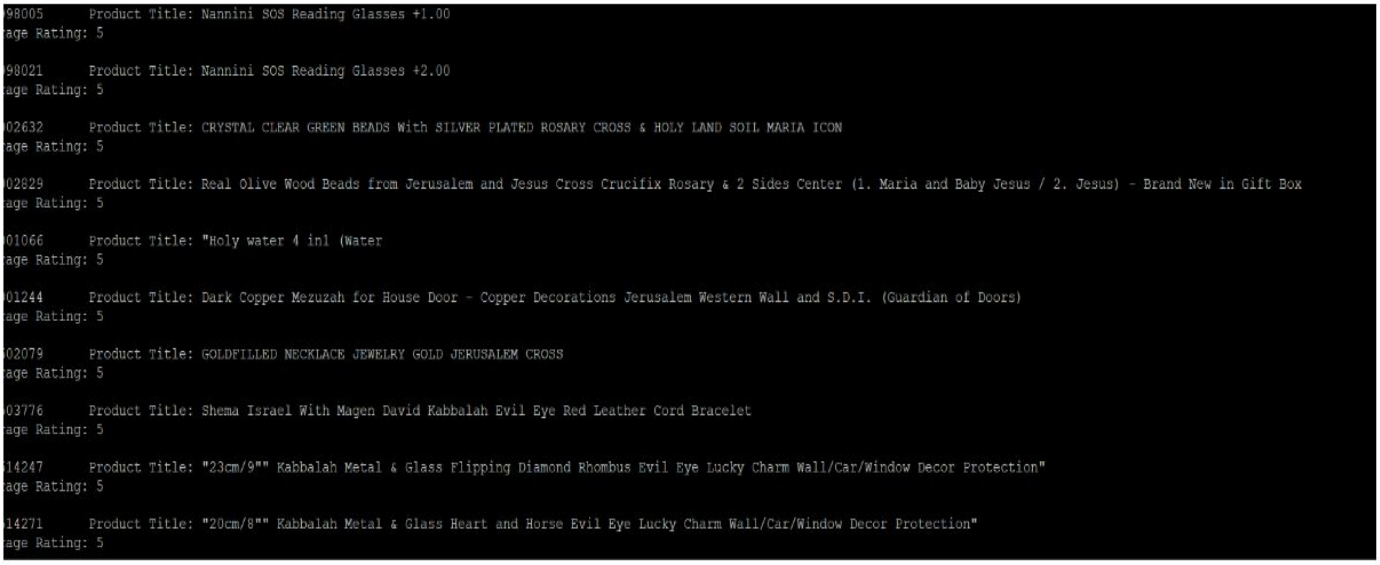
{input →Amazon reviews CSV file and preprocessed} The output contains a list of the top 5 amazon products reviewed by the customer.

Map Reduce 2: ***Amazon Top rated Product [personal_appliances category]***

- This MapReduce job will display the top 5 rated product that belongs to the personal care appliances category from this dataset.

For running this MapReduce job you need to give the command:

yarn jar **mr-1.0.jar**
edu.usfca.cs.mr.part_5_final_project_amazon_reviews.TopRatedAmazonProductJob
/**amazon_reviews_us_Personal_Care_Appliances_v1_00.csv /test05rated**



where {mr-1.0.jar → jar name,amazon_reviews_us_Personal_Care_Appliances_v1_00.csv → input file already on hdfs, test05rated→ output_dir of hdfs}