For this project, we have used 3 Hadoop technologies developing analytics for our project:

1. Hive
2. MapReduce
3. Impala

Below are the commands that we used as part of our analyses through the Hadoop ecosystem-

Table Creation and Merging through Hive: (yelpTableCreateMergeHive.hql)

```
create external table Yelp_data(id string, date_stamp string, reviews string, business_id string,stars float) row format delimited fields terminated by ',' LINES TERMINATED BY '\n' location '/user/ss13449/project/cleaned_data/data_yelp/';

create external table Yelp_business_data(business_id string, business_name string, city string, state string) row format delimited fields terminated by ',' LINES TERMINATED BY '\n' location '/user/ss13449/project/cleaned_data/data_yelp_business/';

create external table Yelp_review(id string, date_stamp date,reviews string,business_id string,business_name string,stars float,city string,state string)
STORED AS TEXTFILE
LOCATION 'hdfs://dumbo/user/ss13449/project/cleaned_data/tabledata/yelp';

INSERT OVERWRITE TABLE Yelp_review select id, cast(to_date(from_unixtime(unix_timestamp(date_stamp, 'dd-MM-yyyy'))) as date),reviews,yelp_data.business_id,yelp_business_data.business_name,stars,city,state from yelp_data JOIN yelp_business_data ON yelp_data.business_id=yelp_business_data.business_id;
```

Taking Data in Text File from Hive: (yelpHiveToTextFile.hql)

```
INSERT OVERWRITE DIRECTORY '/user/ss13449/project/cleaned_data/yelp' ROW FORMAT DELIMITED FIELDS TERMINATED BY ':' SELECT * FROM yelp_review;
```

Sentiment analysis is done through MapReduce Code:

There are two code for this – analytics.py and python_wrapper.py (attached with this file)

Analytics code for Yelp:

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics.py" -mapper "python_wrapper.sh analytics.py" -input /user/ss13449/project/cleaned_data/yelp -output /user/ss13449/project/cleaned_data/analysed_data
```

Analytics code for Reddit:

Starbucks:

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics.py" -mapper "python_wrapper.sh analytics.py" -input /user/ss13449/project/cleaned_data/data_reddit_starbucks/ -output /user/ss13449/project/cleaned_data/analysed_data/analysed_data_reddit_starbucks
```

McDonalds:

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics.py" -mapper "python_wrapper.sh analytics.py" -input /user/ss13449/project/cleaned_data/data_reddit_mcdonalds/ -output /user/ss13449/project/cleaned_data/analysed_data/analysed_data_reddit_mcdonalds
```

Analytics code for Twitter:

Starbucks:

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics.py" -mapper "python_wrapper.sh analytics.py" -input /user/ss13449/project/cleaned_data/data_twitter_starbucks -output /user/ss13449/project/cleaned_data/analysed_data/analysed_data_twitter_starbucks/
```

McDonald’s:

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics.py" -mapper "python_wrapper.sh analytics.py" -input /user/ss13449/project/cleaned_data/data_twitter_mcdonalds -output /user/ss13449/project/cleaned_data/analysed_data/analysed_data_twitter_mcdonalds/
```

Putting Data back in Impala: (putDataInImpala.iql)

```
create external table data_for_analyses(id string, date_stamp string, business_id string, business_name string, user_rating float, city string, state string, polarity float, subjectivity float) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/analysed_data/';
```

```
create external table data_yelp(id string, date_stamp string, business_id string, business_name string, user_rating float, city string, state string, polarity float, subjectivity float) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/analysed_data/';
```

```
create external table data_reddit_starbucks(id string, date_stamp string, business_id string, business_name string, user_rating string, city string, state string, polarity float, subjectivity float) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/analysed_data/analysed_data_reddit_starbucks';
```

```
create external table data_reddit_mcdonalds(id string, date_stamp string, business_id string, business_name string, user_rating string, city string, state string, polarity float, subjectivity float) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/analysed_data/analysed_data_reddit_mcdonalds';
```

```
create external table data_twitter_starbucks(id string, date_stamp string, business_id string, business_name string, user_rating string, city string, state string, polarity float, subjectivity float) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/analysed_data/analysed_data_twitter_starbucks';
```

```
create external table data_twitter_mcdonalds(id string, date_stamp string, business_id string, business_name string, user_rating string, city string, state string, polarity float, subjectivity float) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/analysed_data/analysed_data_twitter_mcdonalds/';
```

Merging Data in Impala: (dataMergeInImapal.iql)

```
Insert into Table data_for_analyses select id, cast(to_timestamp(trim(date_stamp), 'yyyy-MM-dd') as string), business_id, business_name, cast(user_rating as float), city, state, polarity, subjectivity from data_reddit_starbucks;
```

```
Insert into Table data_for_analyses select id, cast(to_timestamp(trim(date_stamp), 'yyyy-MM-dd') as string), business_id, business_name, cast(user_rating as float), city, state, polarity, subjectivity from data_reddit_mcdonalds;
```

```
Insert into Table data_for_analyses select id, cast(to_timestamp(trim(date_stamp), 'yyyy-MM-dd') as string), business_id, business_name, cast(user_rating as float), city, state, polarity, subjectivity from data_twitter_starbucks;
```

```
Insert into Table data_for_analyses select id, cast(to_timestamp(trim(date_stamp), 'yyyy-MM-dd') as string), business_id, business_name, cast(user_rating as float), city, state, polarity, subjectivity from data_twitter_mcdonalds;
```

Taking Data from analyses table to hadoop: (dataForAnalyses.iql)

```
INSERT OVERWRITE DIRECTORY '/user/ss13449/project/cleaned_data/data_for_analyses' ROW FORMAT DELIMITED FIELDS TERMINATED BY ':' SELECT * FROM data_for_analyses;
```

Combining Files of all data sources:

```
hdfs dfs -cp /user/ss13449/project/cleaned_data/yelp/* /user/ss13449/project/cleaned_data/all_data
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_reddit_mcdonalds/* /user/ss13449/project/cleaned_data/all_data
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_reddit_starbucks/part-00000 /user/ss13449/project/cleaned_data/all_data/part-00002
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_reddit_starbucks/part-00001 /user/ss13449/project/cleaned_data/all_data/part-00003
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_twitter_mcdonalds/part-00000 /user/ss13449/project/cleaned_data/all_data/part-00004
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_twitter_mcdonalds/part-00001 /user/ss13449/project/cleaned_data/all_data/part-00005
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_twitter_starbucks/part-00000 /user/ss13449/project/cleaned_data/all_data/part-00006
hdfs dfs -cp /user/ss13449/project/cleaned_data/data_twitter_starbucks/part-00001 /user/ss13449/project/cleaned_data/all_data/part-00007
```

Final Result through Impala for faster processing: (resultsImpala.iql)

Checking average rating by sentiment analysis and user star rating:

```
Select avg(user_rating) as average_user_rating, avg((polarity*5)+5)/2 as average_polarity from data_for_analyses where business_name like "%Starbucks%" and user_rating is not null;
Select avg(user_rating) as average_user_rating, avg((polarity*5)+5)/2 as average_polarity from data_for_analyses where business_name like "%McDonalds%" and user_rating is not null;
```

Frequent Topics from Reviews:

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics_starbucks.py" -mapper "python_wrapper.sh analytics_starbucks.py" -input /user/ss13449/project/cleaned_data/all_data/ -output /user/ss13449/project/cleaned_data/word_count_starbucks
```

```
hadoop jar /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/hadoop-streaming.jar -D mapreduce.job.reduces=0 -files "python_wrapper.sh,analytics_mcdonalds.py" -mapper "python_wrapper.sh analytics_mcdonalds.py" -input /user/ss13449/project/cleaned_data/all_data/ -output /user/ss13449/project/cleaned_data/word_count_mcdonalds
```

(frequentTopics.iql) -

```
create external table fc_starbucks(feature string, count int) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/word_count_starbucks/';
```

```
create external table fc_mcdonalds(feature string, count int) row format delimited fields terminated by ':' location '/user/ss13449/project/cleaned_data/word_count_mcdonalds/';
```

```
select trim(feature) as features,sum(count) as frequency from fc_starbucks where count > 1 group by features order by sum(count) desc limit 10;
```

```
select trim(feature) as features,sum(count) as frequency from fc_mcdonalds where count > 1 group by features order by sum(count) desc limit 10;
```

Ratings by State: (ratingsByState.iql)

```
Select state, avg(user_rating) as average_user_rating, avg((polarity*5)+5)/2 as average_polarity from data_for_analyses where business_name like "%Starbucks%" and user_rating is not null group by state;
```

```
Select state, avg(user_rating) as average_user_rating, avg((polarity*5)+5)/2 as average_polarity from data_for_analyses where business_name like "%McDonalds%" and user_rating is not null group by state;
```

Rating by Period: (ratingsByPeriod.iql)

```
Select year(to_timestamp(trim(date_stamp), 'yyyy-MM-dd')) as period, avg(user_rating) as average_user_rating, avg((polarity*5)+5)/2 as average_polarity from data_for_analyses where business_name like "%Starbucks%" and user_rating is not null group by period order by period;
```

```
Select year(to_timestamp(trim(date_stamp), 'yyyy-MM-dd')) as period, avg(user_rating) as average_user_rating, avg((polarity*5)+5)/2 as average_polarity from data_for_analyses where business_name like "%McDonalds%" and user_rating is not null group by period order by period;
```