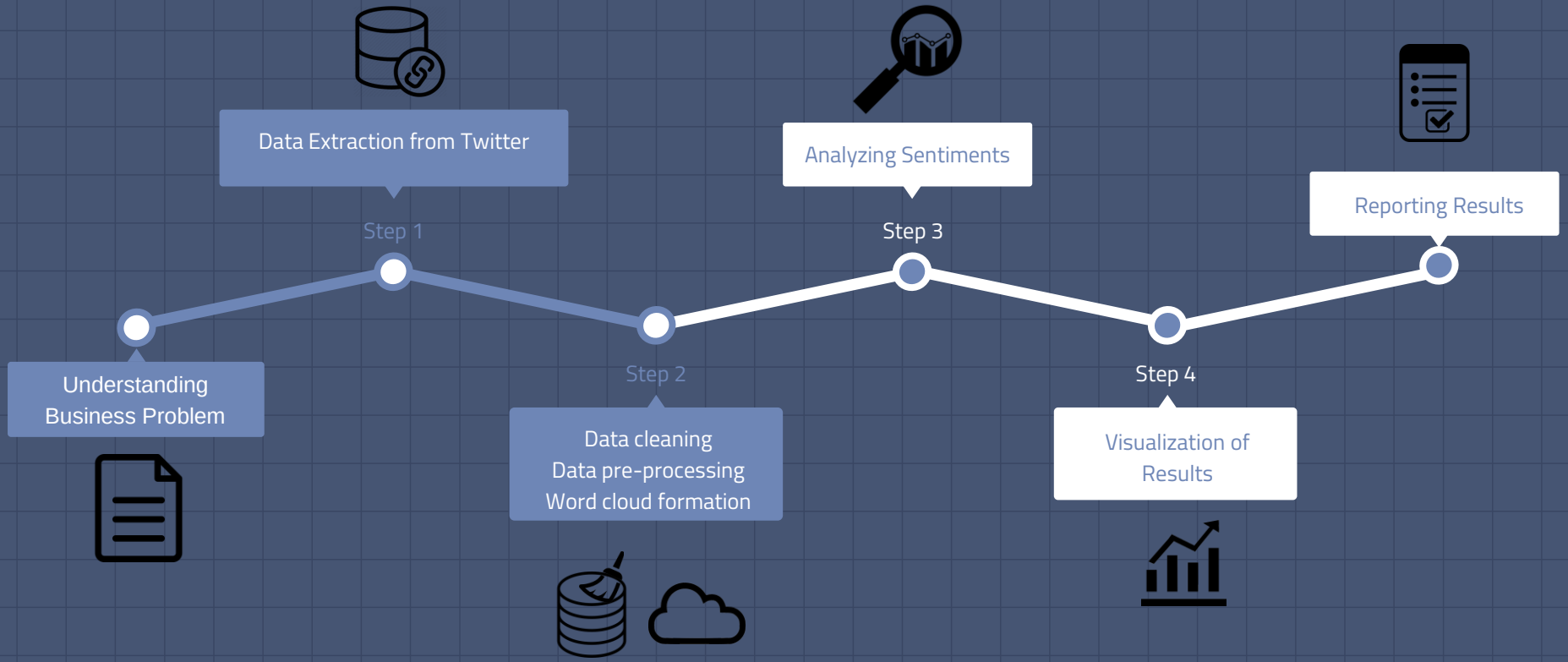
Sentiment analysis from Twitter data using python

Project workflow



Understanding Business Problem

- Problem statement and its scenario

Understanding business problem

Problem statement: Sentiment analysis for the below scenario.

An e-commerce company wants to analyze the sentiment of their competitors in India, as the company is planning to commence their operation in India. Their competitors include Amazon India, Flipkart and Snapdeal. Also to form a word cloud for each competitors.

The solution is divided into 2 parts as follows:

Data extraction:

- ❑ Extracting data from twitter
- ❑ Cleaning/preparation of data

Analysis:

- ❑ Analyzing sentiments
- ❑ Results

DATA EXTRACTION

- Extracting data from Twitter
  - Setting up a twitter developer account
  - Twitter data scraping using python
- Cleaning and data preparation of data
- Word cloud formation

2

Extracting data from twitter:

Setting up a twitter developer account :

- Accessing twitter data can be done using the twitter's API.
- After signing in to your twitter account, go to Developer.twitter.com

- Go to Apps -> Create an app and fill up forms that twitter asks.
- After confirming the email, Developer account is created successfully.
- Create an app. Apps -> Create an app
- Generate the required authentication keys
  - Consumer API key
  - Consumer API secret key
  - Access token
  - Access token secret

- These keys are important for accessing the twitter data. This provides authentication to any other apps to access the twitter data.

Twitter data scrapping:

python

```
#import required packages
import tweepy as tw
#twitter authentication credentials
consumer_key = 'xxxxxxxxxx-Your key here-xxxxxxxxxx'
consumer_secret = 'xxxxxxxxx-Your secret here-xxxxxxxxxx'
access_token= 'xxxxxxxxxx-your token here-xxxxxxxxxx'
access_token_secret = 'xxxxxxxxxx-Your tocken secret here-xxxxxxxxxx'
auth = tw.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tw.API(auth, wait_on_rate_limit=True)
notwt=3000 #No. of tweets to scrape
#Amazon Data Scraping, Keyward:AmazonIN
new_search = "AmazonIN" + " -filter:retweets" #neglect Retweets
date_since = "2020-02-01" #Tweets form date
tweets = tw.Cursor(api.search,q=new_search,lang="en", since=date_since,
        tweet_mode='extended').items(notwt)  #scrapping tweets in English
```

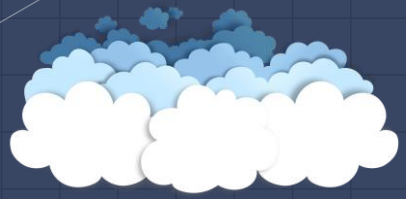

```
#Saving scraped tweets in a list
twlst=[]
for tweet in tweets:
    twlst.append((tweet.full_text))

#Data for flipkart and snapdeal is also scraped in the same way
```

Cleaning / preparation of data:

Data cleaning:

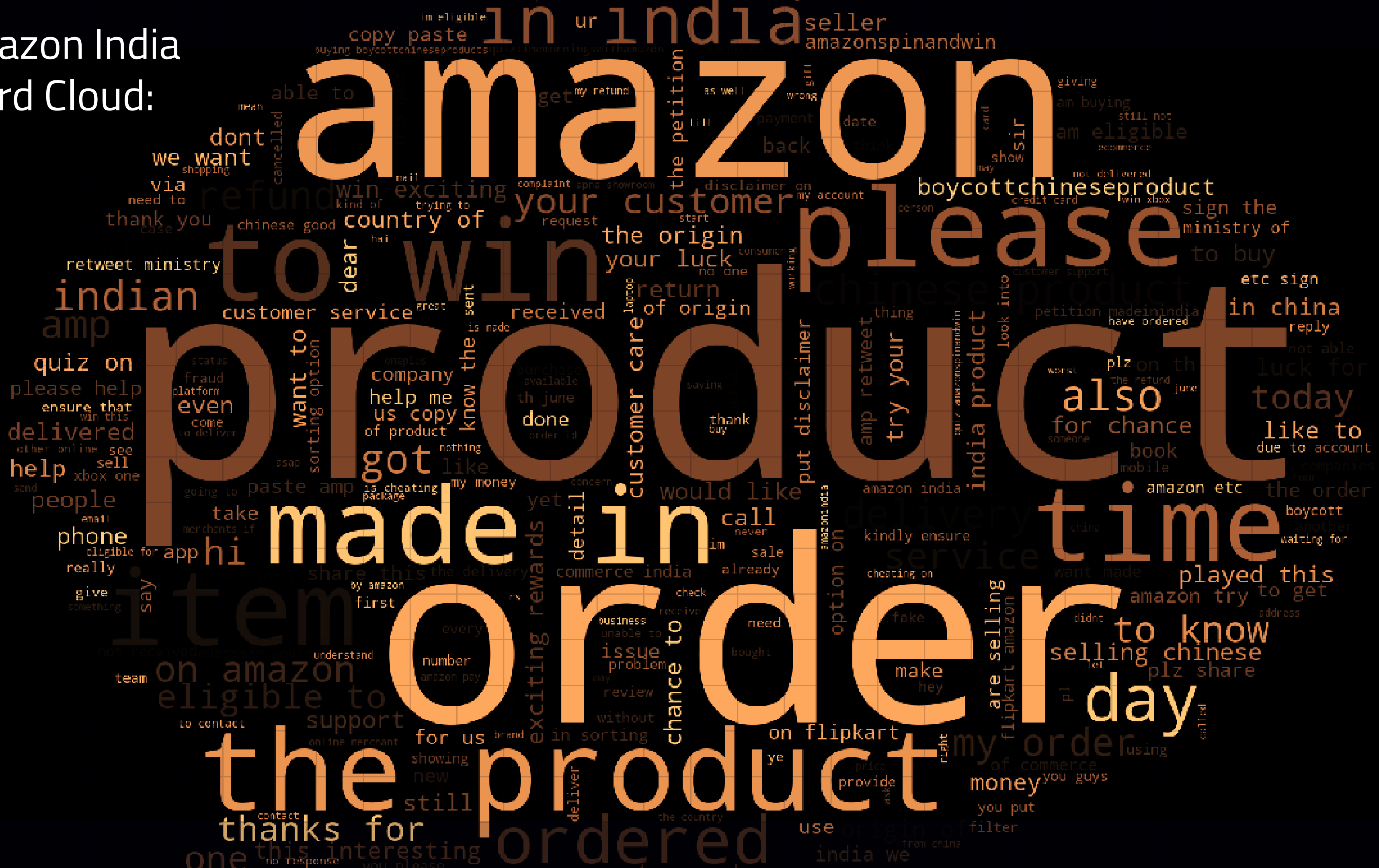
```
#import required packages
import pandas as pd
import preprocessor as p #pip install tweet-preprocessor
import re
#Combine all lists and convert into a DataFrame
data=pd.DataFrame({'Amazon_tweets':twlst, 'Flipkart_tweets':Ftwlst,
'Snapdeal_tweets':Stwlst })
#Conditions set to remove URLs, Emoji/Smiley, Mentions in tweets
p.set_options(p.OPT.URL,p.OPT.EMOJI,p.OPT.SMILEY,p.OPT.MENTION)

clnamz=[]
for twt in data.Amazon_tweets:
    twt=p.clean(twt) #Removes URLs, Emoji/Smiley, Mentions
    twt=re.sub("\d+","", twt) #Removes Numbers
    twt=re.sub(r'[^\w\s]',"",twt) #Removes Punctuations
    clnamz.append(re.sub(" +"," ",twt).strip().lower()) #Removes extra spaces
                                                        and Converts to lowercase
```

Word cloud Formation:



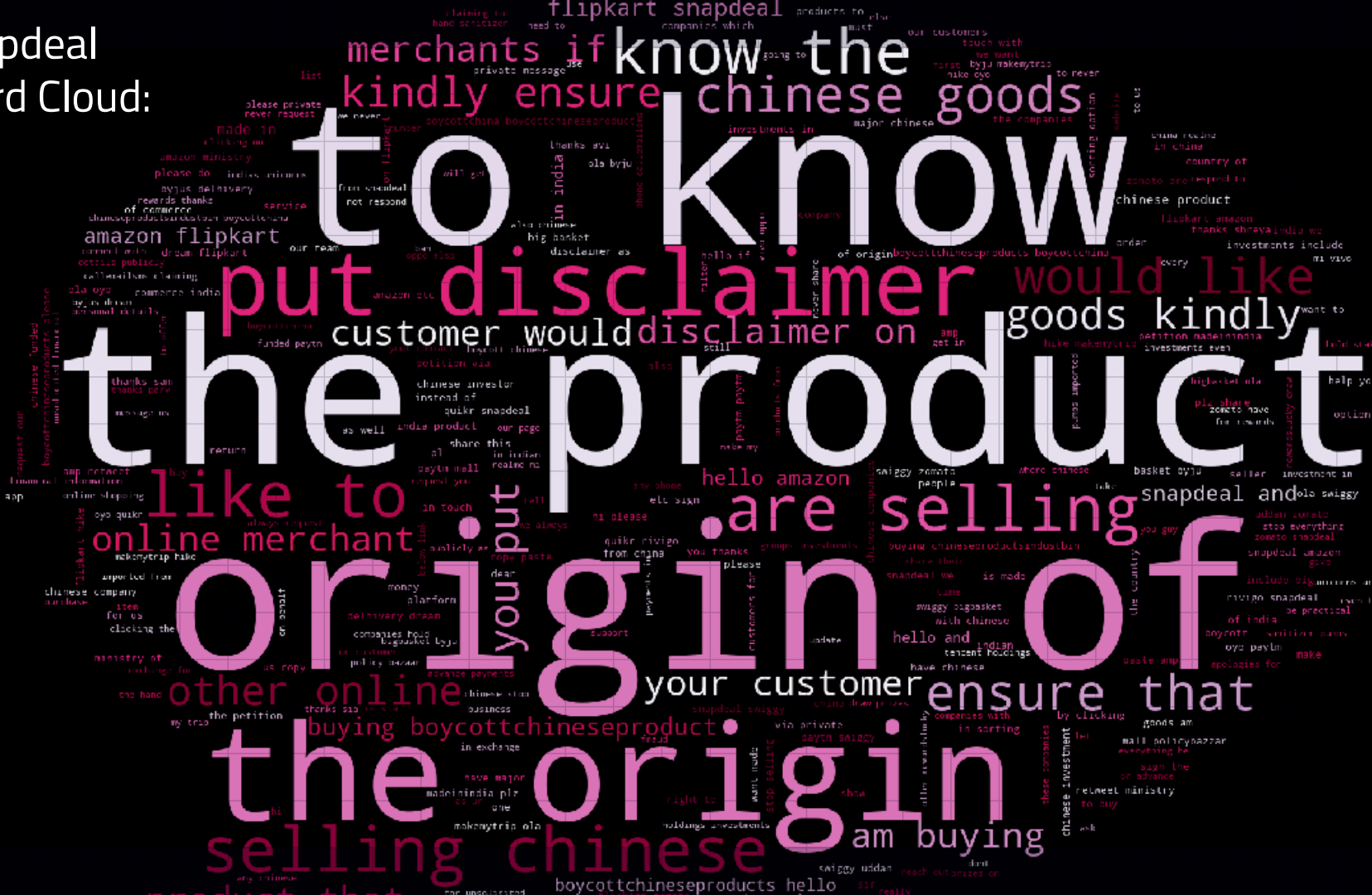
```
#import required packages
import matplotlib.pyplot as plt
from wordcloud import WordCloud
from nltk.corpus import stopwords
import numpy as ny
from PIL import Image
#set mask image
mask=ny.array(Image.open('Path/File name'))
Stopwords = stopwords.words('english') #stop words dictionary from nltk
#word cloud function
wordcloud= WordCloud(background_color = "Black", width=800, height=400,
max_words = 300, mask=mask,colormap='copper',stopwords =
Stopwords).generate(data_cleaned.str.cat(sep='\t'))
#image output
plt.figure( figsize=(30,15))
plt.title('Wordcloud')
plt.imshow(wordcloud)
plt.axis("off")
plt.show()
#Repeat the same for flipkart data and snapdeal data also.
```

Amazon India Word Cloud:



Flipkart Word Cloud:

Snapdeal
Word Cloud:



Sentiment Analysis

Analysing sentiments
Results
Visualization of Results

3

Analysis

Sentiment analysis:

```
#import required packages
import matplotlib.pyplot as plt
from textblob import TextBlob
for col in data_cleaned.columns:
    df=data_cleaned[col]
    #initialize required variables
    polarity = 0, positive = 0, wpositive = 0, spositive = 0
    negative = 0, wnegative = 0,snegative = 0, neutral = 0
    for tweet in df:
        analysis = TextBlob(tweet)
        # adding reaction of how people are reacting to find average later
        polarity += analysis.sentiment.polarity
        if (analysis.sentiment.polarity == 0):
            neutral += 1
        elif (analysis.sentiment.polarity > 0 and analysis.sentiment.polarity <= 0.3):
            wpositive += 1
        elif (analysis.sentiment.polarity > 0.3 and analysis.sentiment.polarity <= 0.6):
            positive += 1
        elif (analysis.sentiment.polarity > 0.6 and analysis.sentiment.polarity <= 1):
            spositive += 1
        elif (analysis.sentiment.polarity > -0.3 and analysis.sentiment.polarity <= 0):
            wnegative += 1
        elif (analysis.sentiment.polarity > -0.6 and analysis.sentiment.polarity <= -0.3):
            negative += 1
        elif (analysis.sentiment.polarity > -1 and analysis.sentiment.polarity <= -0.6):
            snegative += 1#End of 2nd loop
```

```
#Percentage function
def percentage(part, whole):
    Perc = 100 * float(part) / float(whole)
    return format(Perc, '.2f')
```

```
#1st loop continues
positive = percentage(positive, notwt)
wpositive = percentage(wpositive, notwt)
spositive = percentage(spositive, notwt)
negative = percentage(negative, notwt)
wnegative = percentage(wnegative, notwt)
snegative = percentage(snegative, notwt)
neutral = percentage(neutral, notwt)
# finding average reaction
polarity = polarity / notwt
# printing General Report of analysis
print("Feedback/Reaction for "+ str(col) +
"by analyzing " + str(notwt) + " tweets.")
print("General Report: ")
if (polarity == 0):
    print("Neutral")
elif (polarity > 0 and polarity <= 0.3):
    print("Weakly Positive")
elif (polarity > 0.3 and polarity <= 0.6):
    print("Positive")
elif (polarity > 0.6 and polarity <= 1):
    print("Strongly Positive")
elif (polarity > -0.3 and polarity <= 0):
    print("Weakly Negative")
elif (polarity > -0.6 and polarity <= -0.3):
    print("Negative")
elif (polarity > -1 and polarity <= -0.6):
    print("Strongly Negative")
print()
```

```
#1st loop continues
# printing General Report of analysis
    print("Detailed Report: ")
    print(str(spositive) + "% people thought it
was strongly positive")
    print(str(positive) + "% people thought it
was positive")
    print(str(wpositive) + "% people thought it
was weakly positive")
    print(str(neutral) + "% people thought it
was neutral")
    print(str(wnegative) + "% people thought it
was weakly negative")
    print(str(negative) + "% people thought it
was negative")
    print(str(snegative) + "% people thought it
was strongly negative")
    print()
```

Results:

Feedback/Reaction for Amazon_tweets by analyzing 3000 tweets.

General Report:
Weakly Positive

Detailed Report:
5.27% people thought it was strongly positive
12.57% people thought it was positive
22.50% people thought it was weakly positive
37.83% people thought it was neutral
14.23% people thought it was weakly negative
5.67% people thought it was negative
1.30% people thought it was strongly negative

Feedback/Reaction for Flipkart_tweets by analyzing 3000 tweets.

General Report:
Weakly Positive

Detailed Report:
2.10% people thought it was strongly positive
9.57% people thought it was positive
20.40% people thought it was weakly positive
44.03% people thought it was neutral
14.53% people thought it was weakly negative
6.50% people thought it was negative
1.57% people thought it was strongly negative

Feedback/Reaction for Snapdeal_tweets by analyzing 3000 tweets.

General Report:
Weakly Positive

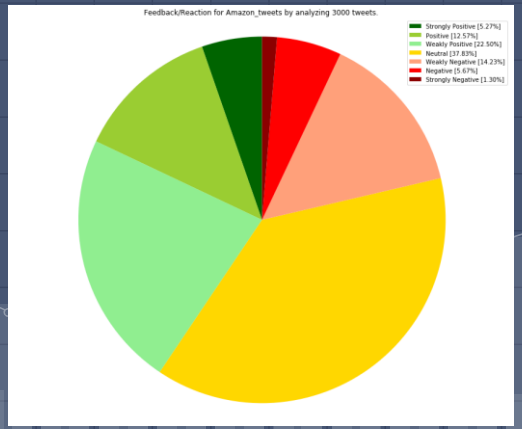
Detailed Report:
0.97% people thought it was strongly positive
5.83% people thought it was positive
51.77% people thought it was weakly positive
31.80% people thought it was neutral
7.63% people thought it was weakly negative
1.60% people thought it was negative
0.30% people thought it was strongly negative

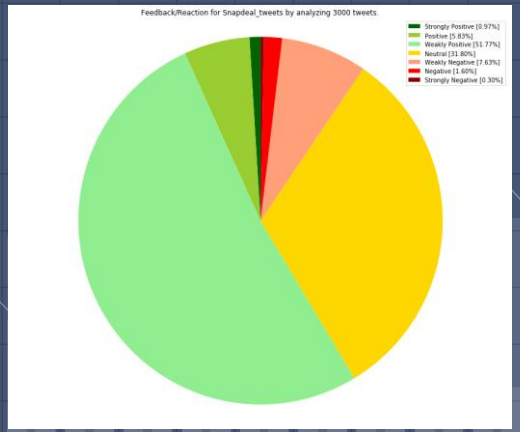
Visualization of results:

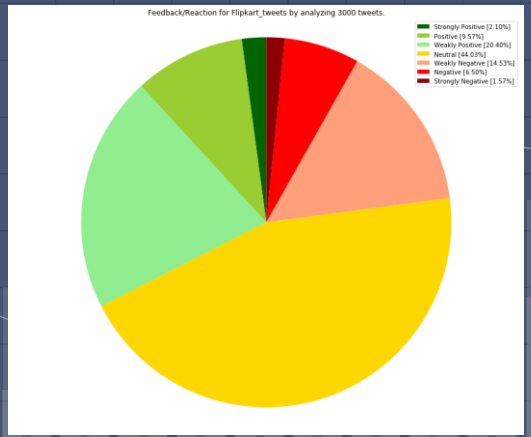
#import required packages
Import matplotlib.pyplot ad plt
#Plotting the analysed values #1st loop continues

```
labels = ['Strongly Positive [' + str(spositive) + '%]', 'Positive [' + str(positive) + '%]', 'Weakly Positive [' + str(wpositive) + '%]','Neutral [' + str(neutral) + '%]',
          'Weakly Negative [' + str(wnegative) + '%]', 'Negative [' + str(negative) + '%]', 'Strongly Negative [' + str(snegative) + '%]']
sizes = [spositive, positive, wpositive, neutral, wnegative, negative, snegative]
colors = ['darkgreen','yellowgreen','lightgreen','gold','lightsalmon', 'red', 'darkred']
plt.figure( figsize=(12,10))
patches, texts = plt.pie(sizes, colors=colors, startangle=90)
plt.legend(patches, labels, loc="best")
plt.title("Feedback/Reaction for "+ str(col) + " by analyzing " + str(notwt) + " tweets.")
plt.axis('equal')
plt.tight_layout()
plt.show() #End of 1st  loop
```

output:





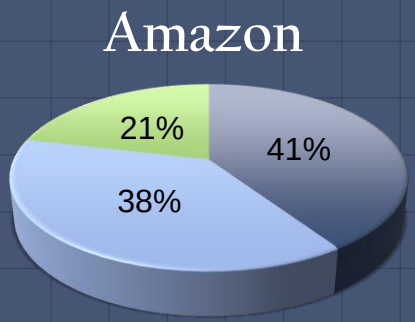


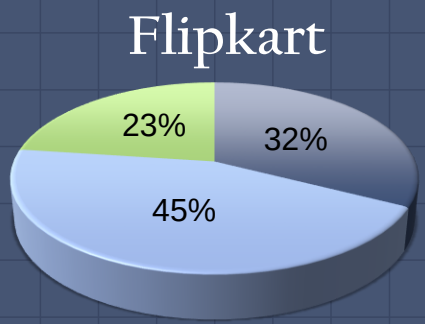
Reporting Results

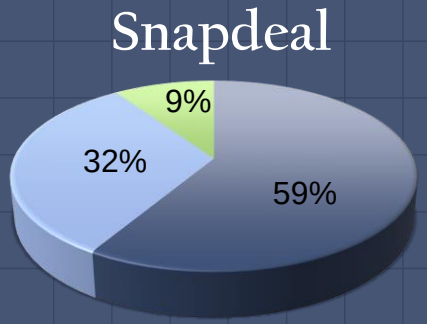
- Reporting results in understandable format

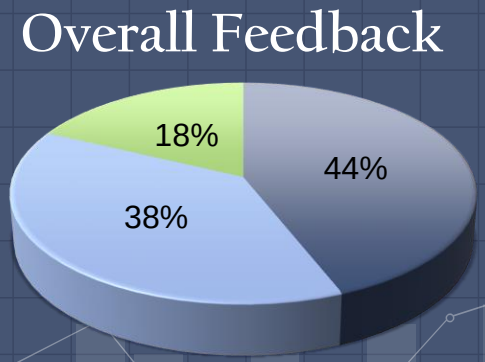
Reporting Results

Feedback	Positive	Neutral	Negative
Amazon	40.3%	37.8%	21.2%
Flipkart	32.1%	44.0%	22.6%
Snapdeal	58.6%	31.8%	9.5%
Overall Feedback	43.7%	37.9%	17.8%





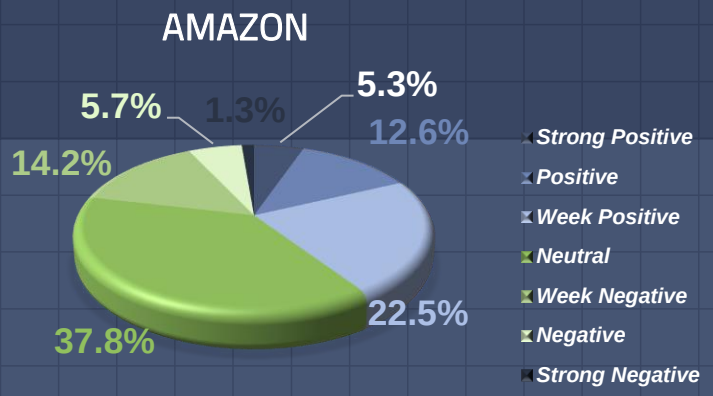


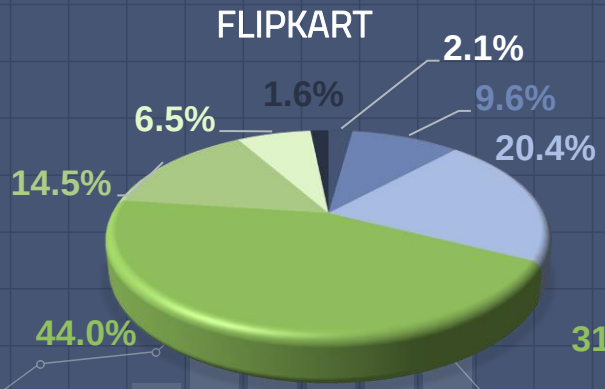


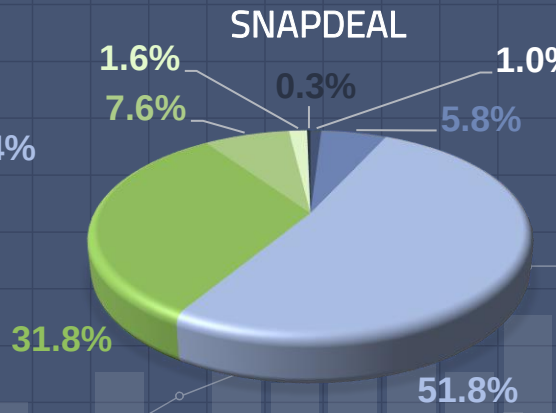
Positive Neutral Negative

Detailed Report

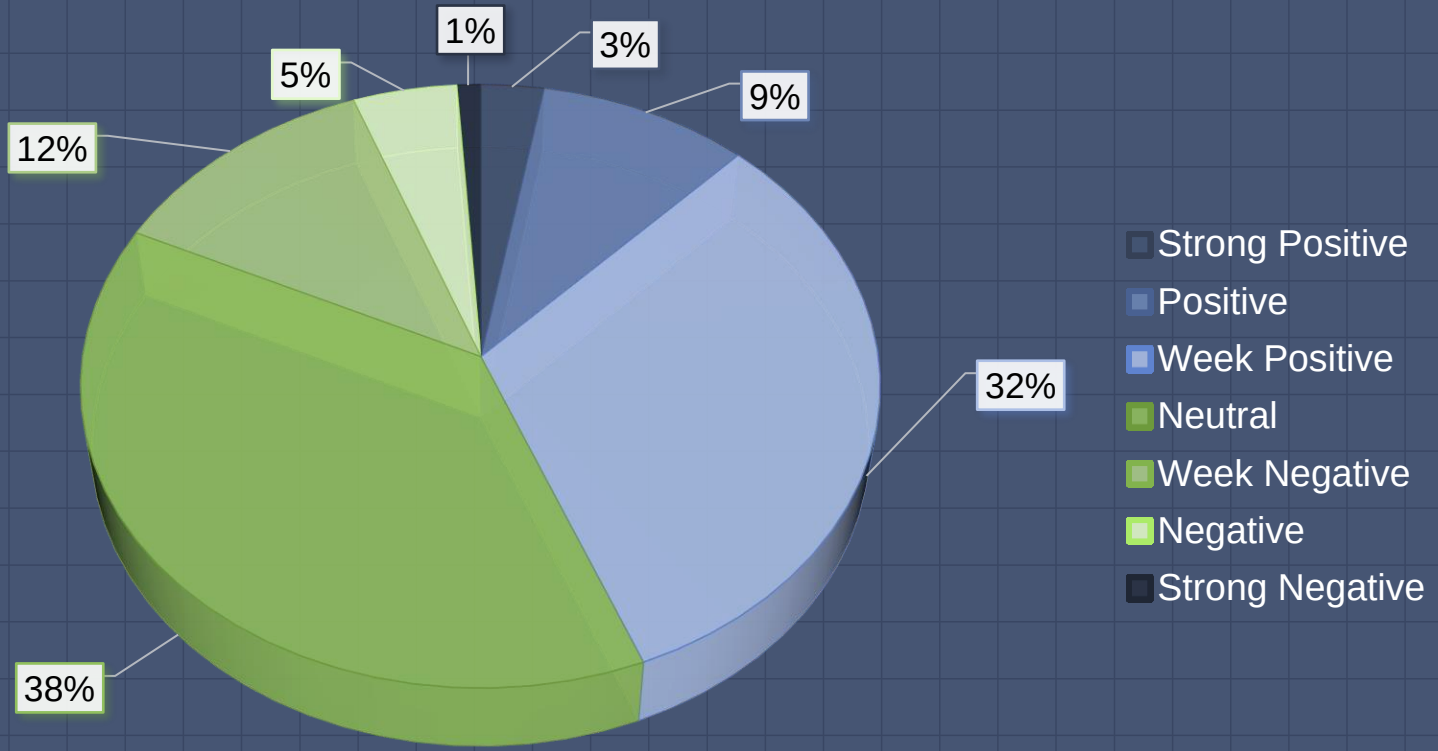
Feedback	Amazon	Flipkart	Snapdeal	Overall feedback
Strong Positive	5.3%	2.1%	1.0%	2.8%
Positive	12.6%	9.6%	5.8%	9.3%
Week Positive	22.5%	20.4%	51.8%	31.6%
Neutral	37.8%	44.0%	31.8%	37.9%
Week Negative	14.2%	14.5%	7.6%	12.1%
Negative	5.7%	6.5%	1.6%	4.6%
Strong Negative	1.3%	1.6%	0.3%	1.1%







OVERALL FEEDBACK



Verdict

Amazon vs Flipkart:

- Amazon has 9% more positive feedbacks and 5.4% is strongly positive.
- Amazon has 7% less neutral feedbacks.
- Amazon has 2% less negative feedback.

Amazon vs Snapdeal:

- Amazon has 18% less positive feedbacks, but strong positives are 4.3% more.
- Amazon has 6% more neutral feedbacks.
- Amazon has got a 12% more negative feedback, and has1.3% strong negatives.

Flipkart vs Amazon:

- Flipkart has 9% less positive feedbacks.
- Flipkart has 7% more neutral feedbacks.
- Flipkart has 2% more negative feedback out of which 1.6% strongly negative.

Flipkart vs Snapdeal:

- Flipkart has 27% less positive feedbacks, but has 2.1% strong positives.
- Flipkart has 13% more neutral feedbacks.
- Flipkart has got a 14% more negative feedback, and has 1.6% strong negatives.

Snapdeal vs Flipkart:

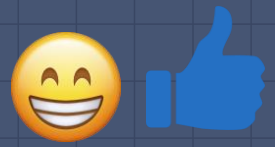
- Snapdeal has 27% more positive feedbacks. But, has only 1% is strong positives.
- Snapdeal has 7% more neutral feedbacks.
- Snapdeal has 2% more negative feedback.

Snapdeal vs Amazon:

- Snapdeal has 18% more positive feedbacks.
- Snapdeal has 6% less neutral feedbacks.
- Snapdeal has 12% less negative feedback. Also, has only 0.3% strong negatives.

Over all there is a weekly positive sentiment towards the e-commerce competitors, in which there are 38% neutral and 32% week positive feedbacks. But, there are 18% Negative feedback in which only 1% is strongly negative. Operations in India is possible to reach success, with best success metrics.

#THANKS!



THANK YOU!