TEXT MINING GROUP PROJECT

Twitter Data Analysis

Group 4 - Project presentation

G Likith Chandra 21020845001
Pritam 21020845008
Sachin 21020845014
Naik Dhanasri Shivdas 21020845015

Case - Motivation

Covid19 has changed many things around eCommerce space. One of these is the change in customer behavior and perception towards their shopping preferences. The scope of this project is to understand this change in customer behaviour specifically in the eCommerce space.

Text, such as social media posts and customer reviews, is a gold mine waiting to be discovered. With the help of suitable text mining techniques, this unstructured data can be turned into useful insights, which can help companies better understand how customers perceive their products or services, and how can they make required business improvements to tackle the ever changing customer dynamics.

Project Objective

The focus area of our project is 'Flipkart'- the ecommerce giant of India. We collected more than a million tweets of data on Flipkart for our analysis.

Text mining on the user's tweets regarding their shopping experiences and reviews of the products and services on Flipkart can help undestand consumer behaviour and find answers to below mentioned research questions from a business context:

- Which are the items that are primarily purchased or being talked about?
- Is there a difference in customer sentiment/behavior during covid vs. post covid?
- Which specific issues or pain points of customers, should the company address on priority?
- What should be the focus area of improvements for the company (products or delivery or website experience, etc.) that could help in building better brand sentiment among users.?
- Understanding keyword burstiness along with timeline.
- What is the topic focus of the tweets for Flipkart and how it has evolved through time?

Project flow -

1. Downloading Data from Twitter API (v2 bearer token) and loading text into a pandas data frame.
2. Dropping irrelevant columns
3. Taking out hours, day and year number
4. Creating context windows based on dates
5. Data Preprocessing (Including Lemmatization)
    a. Emoji
    b. URL, HTML,
    c. StopWords, Tokenization,
    d. Length of words with characters less than 3 removals, Removing whitespace
6. **Dump 1 - Preprocessing pickle file**
7. Gensim preprocessing and lemmatization
8. **Dump 2 - Lemmatization**
9. id2Word implementation - Word Embedding
10. **Dump 3 - Corpus pickle file.**
11. **Dump 4 - id2Word**
12. LDA model implementation

13. Getting a **perplexity score.**
14. **Coherence score** calculation using the 'u_mass' argument
15. Checking the best number of topics using a coherence score
16. Obtaining 20 as our ideal topic number
17. Dumping optimal topic into a text file
18. Getting **dominant topics** - for each of the tweets

19. **Text Summarization** - Reverse tracking of Tweets under each of the Topics obtained
20. Assigning Text and Weight for each of the Tweets & Setting a **threshold value** to filter **summarized tweets**.
21. Checking burst keywords using word cloud within the summarized texts for each of the tweets
22. Visualization using **pyLDAvis** library
23. **Sentiment analysis** for each topic within our 2 Context window
24. Implementing **Word2Vec** - **CBOW** for each of our contextes

Libraries used

- nltk -> ntl.tokenize, nltk.corpus, nltk.tem,
- emoji
- sklearn
- spacy
- wordnet
- gensim

NLP Concepts used

- Stemming
- Lemmatization
- Stopwords removal techniques
- POS Tagging
- Topic modeling using LDA
- Text summarization
- Sentiment Analysis
- Word2Vec
- Wordcloud, pyLDAvis for visualization

Data Dictionary -

- Original dataframe size : 1050000 X 12
- Post Splitting based on year

    Context 1 for the year 2020-2021: (574480,6)

    Context 2 for the year 2022 : (478074, 6)

	text	created_at	datetime_val	day	hour	year	clean_text
0	#Flipkart ordered nothin wireless tws, didnot ...	2021-12-31T23:58:03.000Z	2021-12-31 23:58:03+00:00	31	23	2021	#Flipkart ordered nothin wireless tws didnot w...
1	@Flipkart @FlipkartStories @flipkartsupport @_...	2021-12-31T23:49:47.000Z	2021-12-31 23:49:47+00:00	31	23	2021	@Flipkart @FlipkartStories @flipkartsupport @_...
2	@Flipkart @FlipkartStories @flipkartsupport @_...	2021-12-31T23:46:32.000Z	2021-12-31 23:46:32+00:00	31	23	2021	@Flipkart @FlipkartStories @flipkartsupport @_...
3	@Flipkart @flipkartsupport #marq\nYou guys are...	2021-12-31T23:30:52.000Z	2021-12-31 23:30:52+00:00	31	23	2021	@Flipkart @flipkartsupport #marq You guy compl...
4	@Flipkart I've Ground floor - retail space ava...	2021-12-31T22:52:00.000Z	2021-12-31 22:52:00+00:00	31	22	2021	@Flipkart 've Ground floor retail space availa...

LDA score analysis & graph

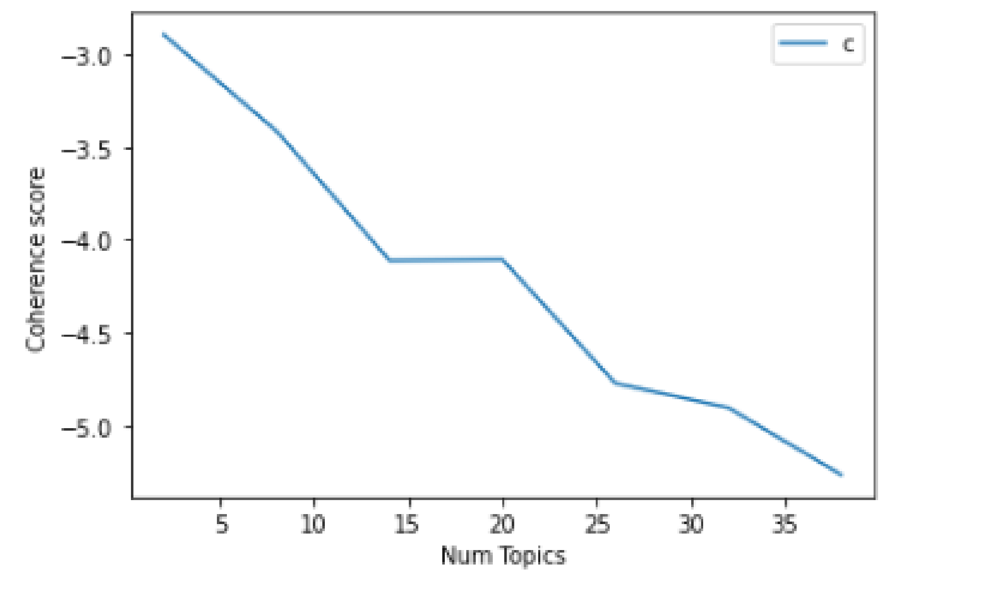
For context window 1 i.e. year 2021

```
In [50]:  lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                                     id2word=id2word,
                                                     num_topics=10,
                                                     random_state=100,
                                                     update_every=1,
                                                     chunksize=100,
                                                     passes=10,
                                                     alpha='auto',
                                                     per_word_topics=True)

In [51]:  print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.

          Perplexity:  -7.413599748700645

In [52]:  # Compute Coherence Score
          coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='u_mass')
          coherence_lda = coherence_model_lda.get_coherence()
          print('\nCoherence Score: ', coherence_lda)

          Coherence Score:  -4.289524903796476
```

- We see that our model Perplexity score is -7.41, and Coherence score is -4.28

- From the coherence graph, it could be seen that the inflection points are at 15, 20, and 25. From our topic_list generated via the coherence computation - we have taken 20 as our ideal number of topics as the maximum flattening of the curve shows a steep incline after that point.

```
limit=40; start=2; step=6;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()
```



For context window 2 i.e. year 2022

```
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                           id2word=id2word,
                                           num_topics=10,
                                           random_state=100,
                                           update_every=1,
                                           chunksize=100,
                                           passes=10,
                                           alpha='auto',
                                           per_word_topics=True)
```

```
print('\nPerplexity: ', lda_model.log_perplexity(corpus))  # a measure of how good the model is. lower the better.
```

```
Perplexity:  -7.350538119251896
```

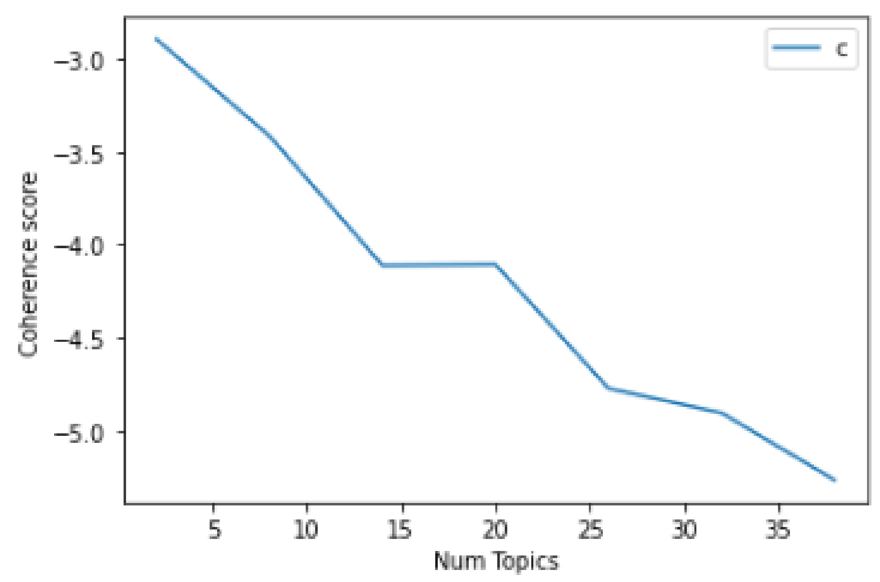
```
# Compute Coherence Score
coherence_model_lda = CoherenceModel(model=lda_model, texts=data_lemmatized, dictionary=id2word, coherence='u_mass')
coherence_lda = coherence_model_lda.get_coherence()
print('\nCoherence Score: ', coherence_lda)
```

```
Coherence Score:  -4.498526126801104
```

- We see that our model Perplexity score is -7.35, and Coherence score is -4.49

- From the coherence graph, it could be seen that the inflection points are at 15, 20, and 25. From our topic_list generated via the coherence computation - we have taken 20 as our ideal number of topics as the maximum flattening of the curve shows a steep incline after that point.

```
limit=40; start=2; step=6;
x = range(start, limit, step)
plt.plot(x, coherence_values)
plt.xlabel("Num Topics")
plt.ylabel("Coherence score")
plt.legend(("coherence_values"), loc='best')
plt.show()
```



Topics - 2021

((0, '0.069*"number" + 0.068*"contact" + 0.061*"email" + 0.055*"com" + 0.047*"reply" + 0.044*"check" + 0.042*"understand" + 0.042*"concern" + 0.040*"share" + 0.033*"detail"')

- <u>This topic depicts unresolved queries.</u>

(1, '0.110*"send" + 0.076*"item" + 0.054*"really" + 0.044*"deal" + 0.043*"wrong" + 0.035*"very" + 0.031*"exchange" + 0.028*"message" + 0.025*"way" + 0.017*"member"')

- <u>This topic depicts that the customer received wrong product and they want it exchanged.</u>

(4, '0.114*"product" + 0.070*"get" + 0.049*"buy" + 0.047*"return" + 0.041*"request" + 0.038*"money" + 0.031*"replacement" + 0.028*"month" + 0.024*"wait" + 0.023*"laptop"')

- <u>This topic depicts customer's frustrations towards a requested for exchange of product.</u>

(3, '0.087*"fraud" + 0.079*"take" + 0.065*"amazon" + 0.058*"company" + 0.035*"never" + 0.034*"pathetic" + 0.033*"well" + 0.026*"cheater" + 0.024*"sell" + 0.023*"do"')
<u>This topic depicts that the customer is comparing the services offered by amazon and flipkart.</u>

(7, '0.154*"flipkart" + 0.083*"order" + 0.048*"customer" + 0.036*"day" + 0.033*"service" + 0.030*"call" + 0.029*"time" + 0.029*"deliver" + 0.022*"delivery" + 0.020*"still"')

- <u>This topic depicts customer's frustrations regarding delay in order delivery.</u>

Topics - 2022

(0, '0.137*"phone" + 0.083*"mobile" + 0.049*"redmi" + 0.042*"new" + 0.041*"offer" + 0.035*"reply" + 0.032*"app" + 0.023*"flipkart" + 0.023*"soon" + 0.023*"change"')

- This topic depicts phones, maybe of redmi brand, could be the launch of new phone.

(1, '0.085*"pay" + 0.084*"take" + 0.058*"flipkart" + 0.047*"book" + 0.042*"consumer" + 0.038*"action" + 0.035*"flight" + 0.033*"ticket" + 0.028*"amount" + 0.026*"respond"')

- This topic depicts flight ticket options available on Flipkart.

(3, '0.115*"product" + 0.085*"order" + 0.083*"return" + 0.064*"od" + 0.039*"damage" + 0.034*"flipkart" + 0.033*"request" + 0.026*"refund" + 0.023*"replacement" + 0.022*"policy"')

- This topic depicts customers' frustrations towards possible damaged products and the refund and replacement policy.

Under the broader category of customer experience, we found 2 topics that depict frustrations faced by the customer, so via the tweets, we can capture the multiple themes ie the different kinds of issues dealt by the customers.

(5, '0.163*"customer" + 0.096*"follow" + 0.061*"support" + 0.050*"care" + 0.039*"flipkart" + 0.025*"cheat" + 0.024*"make" + 0.022*"guy" + 0.020*"call" + 0.020*"fool"')

- This topic depicts customers' frustration with customer support care.

(15, '0.196*"service" + 0.122*"bad" + 0.074*"flipkart" + 0.059*"customer" + 0.045*"experience" + 0.039*"never" + 0.030*"pathetic" + 0.029*"poor" + 0.024*"ever" + 0.021*"shopping"')

- This topic also depicts the poor service experience faced by the customer.

Topics - 2022

(16, '0.071*"flipkart" + 0.041*"well" + 0.040*"think" + 0.033*"amazon" + 0.030*"thing" + 0.030*"purchase" + 0.028*"buy" + 0.026*"year" + 0.022*"invoice" + 0.022*"go"')

- This topic depicts that customers might go to amazon from Flipkart.
- This can be because of 2 reasons:
    1. The customers are frustrated with Flipkart and that's why they are going to amazon OR
    2. The customers found a product on Flipkart, but they also want to compare if buying the product from amazon is a better option.

(19, '0.140*"company" + 0.136*"fraud" + 0.117*"problem" + 0.067*"want" + 0.066*"solution" + 0.062*"solve" + 0.053*"do" + 0.026*"post" + 0.021*"member" + 0.020*"asap"')

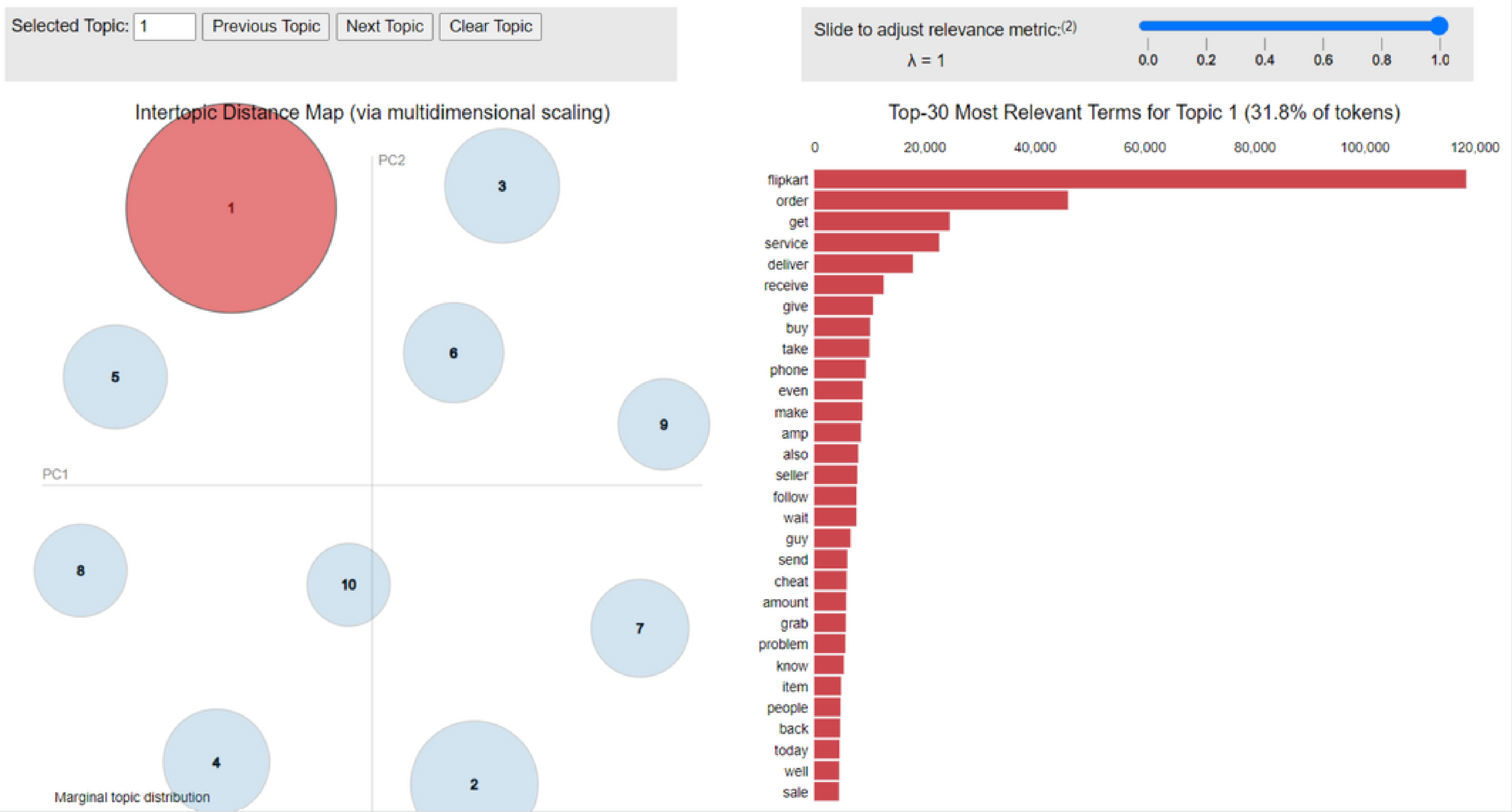
- This topic depicts a possible fraud or fake purchase that the customer wants to be resolved quickly.

(6, '0.118*"number" + 0.063*"flipkart" + 0.054*"check" + 0.047*"email" + 0.047*"request" + 0.044*"contact" + 0.042*"share" + 0.041*"understand" + 0.040*"concern" + 0.037*"detail"')

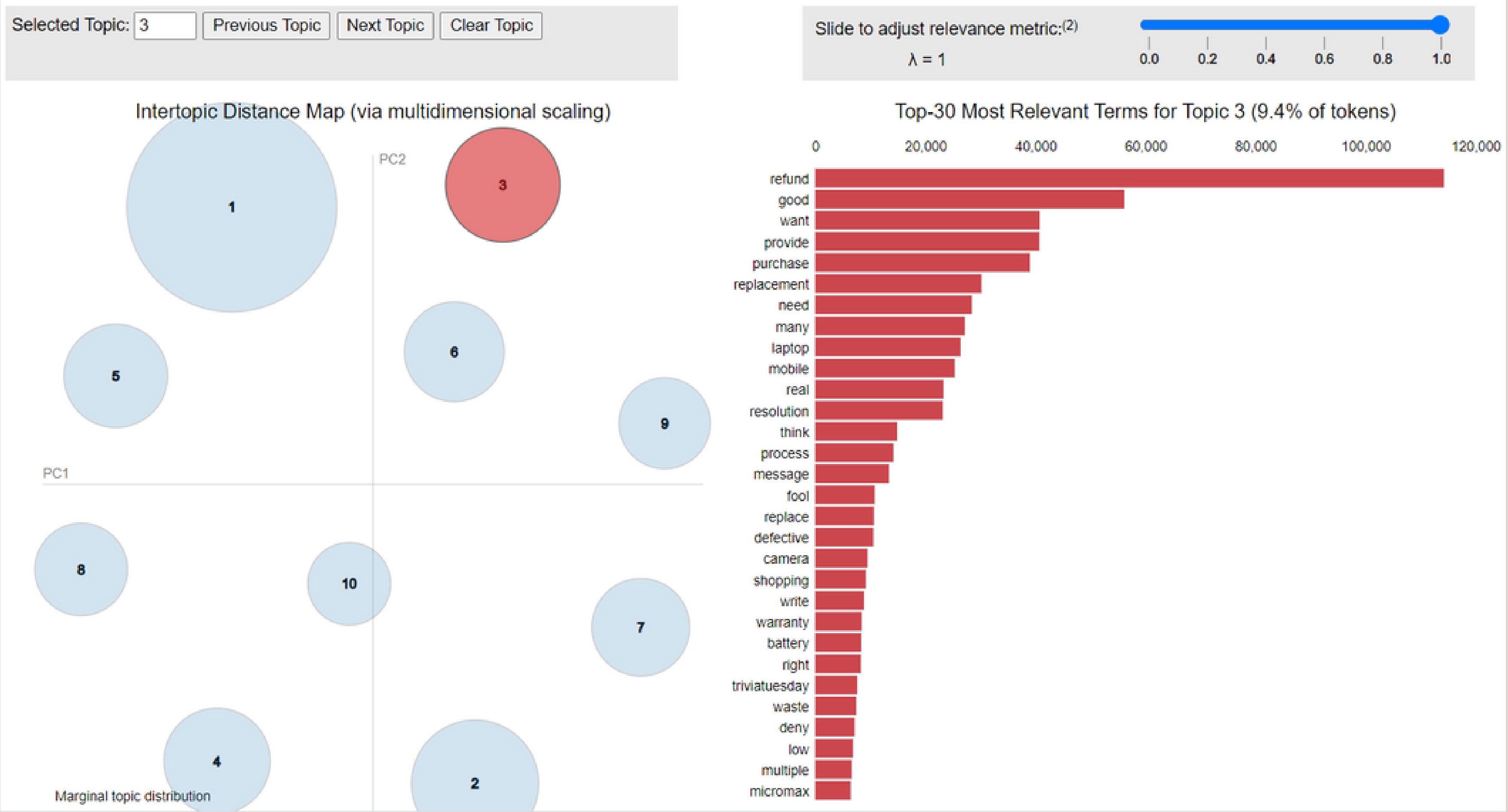
- This topic depicts unresolved queries.

Topics & PyLDAvis plot interpretation

Topic 1 -



Topic 3 -



We are using pyLDAvis to visualize the topics that were generated by the algorithm. The library creates an HTML page where we can see the keywords associated with the topics that are generated from the LDA model.

Applying Text Similarity on top of LDA topics after categorizing them under dominant topic category

```python
def summarization(text):  #calculating score over here by statement by statement
    #print(text, "------")
    number_tokens=len(word_tokenize(text))
    tagged=nltk.pos_tag(word_tokenize(text))
    number_nouns=len([word for word, pos in tagged if pos in ["NN","NNP","VBG"]])   # Trying to find number of nouns from the
    #use NER to tag the named entities
    ners=nltk.ne_chunk(nltk.pos_tag(word_tokenize(text)),binary=False)
    number_ners=len([chunk for chunk in ners if hasattr(chunk,'label')]) # Through chunking we are able to find out NER -> NE
    score=(number_ners+number_nouns)/float(number_tokens) # Out of total tokens, how many no of nouns and no of ners are comi
    return (score,text)
```

Summarising text based on POS Tagging i.e. filtering out Noun singular (NN), Proper Noun singular (NNS) etc.

```python
In [ ]: final_topic1_df.head(10)
        final_topic1_df[final_topic1_df.sort_values('weights', ascending=False)['weights']>0.5].sort_values('weights', as
```

1	Weights	Text
2	0.944	fraud cheater scamer meri thik karo kitni bar bolne ba date date dia rhe kuch solve rha hate
3	0.933	order nonsense going consumer court film complain damage jacket come called going consumer court commerce
4	0.923	voo sabi phone mee hai sab phone mee google dailer hai except samsung
5	0.923	answer mah battery mp main camera mediatek helio g processor allroundsuperstar redmiprime redmi
6	0.909	bhad javo .... murdabaad haa service center going consumer court ...
7	0.909	processor qualcomm snapdragon rear camera mp front camera mp battery mah
8	0.909	woh bhi ..but camera battery wise dekhe toh realme sahi
9	0.909	promise collect refund till date collect refund consumer forum complain flipkart
10	0.909	hope win fullonblockbusterchallenge galaxyf fullonblockbusterchallenge fullonblockbusterchallenge galaxyf fullonblockbusterchallenge fullonblockbusterchallenge galaxyf fullonblockbustercha
11	0.900	file complaint national consumer helpline consumer grievance call register grievance
13	0.900	frustrating unboxing video recording guy solve issue hour consumer forum
14	0.900	fullonblockbusterchallenge fullonblockbuster galaxyf fullonblockbusterstar anser true mp quad camera win
15	0.889	bear cost case give pre aprroval delivery boy entry building security register cctv footage call record ball court
16	0.889	mp quad camera mah battery hz refresh rate fullonblockbusterchallenge
17	0.889	customer satisfaction guy spoil happiness ... tomorrow brother birthday
18	0.889	bas waiting powerplaywithchampions season jaldi lao sir ipl babaswag
19	0.889	final resort calling consumer forum number filing complaint today
20	0.889	commerce mafia mail higher authority mail .... grievance .officer.com
21	0.889	thug company hai sir inka xeo maha chor hai
22	0.889	claiming via appliance protection pain moving consumer forum complaining
23	0.875	answer true mp quad camera fullonblockbusterchallenge join friend
24	0.875	samsung brand flipkart series series amazon series chinese
25	0.875	fullonblockbusterchallenge fullonblockbuster galaxyf fullonblockbusterstar anser denim black win
26	0.875	misleading customer get benifit seek addressal consumer court
27	0.875	bro file complaint consumer court aginst flipkart seller

We have filtered the summarized texts based on their relative weights and our **threshold value > 0.5**

Area of Improvement

- We could create n-grams and then build a network graph on top of it.
- Sliding window concept could be implemented via epoch concept.
- Skipgram implementation could be done
- Levenshtein distance could be used to get the similarity of the tokens within the tweets. This way, we could build a network graph of the vocabulary and understand the co-occurrence contexts of the word.
- We could try to improve the LDA score better to predict the latent topic representation of the corpus.
- Due to resoruce constraints, we couldn't implement the model for both the context windows but we would like to run it for both the windows and then compare what are the visible differences.
- We could try different values of alpha and beta and accordingly check for the coherence score. Kullback Leibler Divergence Score could also be used to figure out better number of topics.

THANK YOU