Fake News Detection from YouTube

Submitted By,

Manthiramoorthy Cheranthian

Objective:

- In this current Information age there is a huge increase in online sharing of medical information. Studies have shown that half of the health-related videos in YouTube contain misleading and ingenuine information.
- Primary objective of this project is to detect and deter the ingenuine medical information from videos and provide genuine and quality medical information to the med-info seeker.

Research model and variables:

- Every individual videos were segregated as Animated/Non- Animated using Image recognition algorithms like OpenCV.
- Transcripts extraction for all the videos were automated using python.
- Topic Modelling to identify the topics most prevalent in a video using Unsupervised and semi-supervised algorithms.
- Sentiment analysis in video comments using predefined packages like “syuzhet” are employed.
- Quality of the medical information in the videos are assessed using DISCERN scale scoring based on a 15 key Questions by discern.org.uk.
- A final logistic regression model is built over a training data consisting of transcripts, comments, likes/dislikes, views, Topics, DISCERN scores from a sample of 100 videos.

Variables:

Appendix A1: variable description

Variable Names	Description
DISCERN Score	Discern score to assess the quality of medical information arrived manually from a 15 key questions by Discern.org.uk
Upvotes	Likes from a video taken by connecting YouTube API.
Downvotes	Dislikes from a video taken by connecting YouTube API
Views	Number of viewers for a video.
Transcript	Transcript from a video taken by connecting YouTube Api.
Comment Sentiment	Comment sentiments
Topics	Topics identified after perfoming Topic modelling over transcripts
Target	Informative/Non-Informative

Dataset:

- The following keywords were used to collect the videos from YouTube

  1.)”Breast Cancer and treatments”.

  2.)”Miracle cures for breast cancer”.

  3.)”Natural remedies for breast cancer”.
- Conducted our search using similar terms but with slight modifications to trigger misinformative content.
- Initially to develop a prototype model we create a dataset with 40 sample videos.
- Final dataset involves 100 videos with equal proportion from the search results obtained using above three keywords.
- Attributes of a video like likes/dislikes, views, transcripts and comments are extracted to form key variables of the dataset.

Filtering videos based on Discern Scale Recommendations:

1) Primary Content should be about Breast Cancer
2) The videos should not contain animations/artificial voices
3) Language should be English
4) Duration should be less than 30mins
5) Video should not have more than 2 speakers.

Comment Collection:

Extracting comments from the videos including all the reply threads by connecting with YouTube API using developer key.

Data scraping:

- Framing all the comments and reply as a single comment thread and summing up the the sentiment scores for all words from all comments.

Transcriptions:

- Video transcriptions has been obtained using YouTube automatic captioning service.
- Speech to Text service by Google Cloud has been used to transcribe audio to text for video without automatic transcripts.
- The following table shows the difference between a genuine and misinformation transcript.

Sample transcript for Informative/Misinformative videos:

Misinformative	Trustworthy
How I healed this prostate cancer, naturally. I was diag-nosed back in 2012. My urologist wanted to have me cut it, cut the prostate out. I was just freaked me out. I chose the alternative route. There's lots of research with a guy named Don Tolman? Do a Google search on him? He is amazing: lots of raw fruit and vegetables. Lots of water, fasting sunshine, fresh air, exercise, lots of turmeric and curcumin. No doubt about that, and the miracle cure, as I mentioned before, you can get into the description. Click on the link to the miracle cure video and have a look. It really is amazing.	in the United States there are approximately 280,000 women diagnosed with breast cancer annually and about 50 - 60 percent of them will receive radiation as part of their care and when women go through it they have a understandably a lot of anxiety going through treatment and having x-ray therapy for the treatment of their breast cancer but some of the most common side effects are behavioral one such as fatigue depression stress and anxiety and when we look at our own studies we find that in the long run up to 30% of women in the long run several months two years after their treatment will continue to suffer from fatigue.

Appendix A2: Transcript sample.

Viewer Engagement Features:

The following are the viewer engagement features we include in our final dataset:

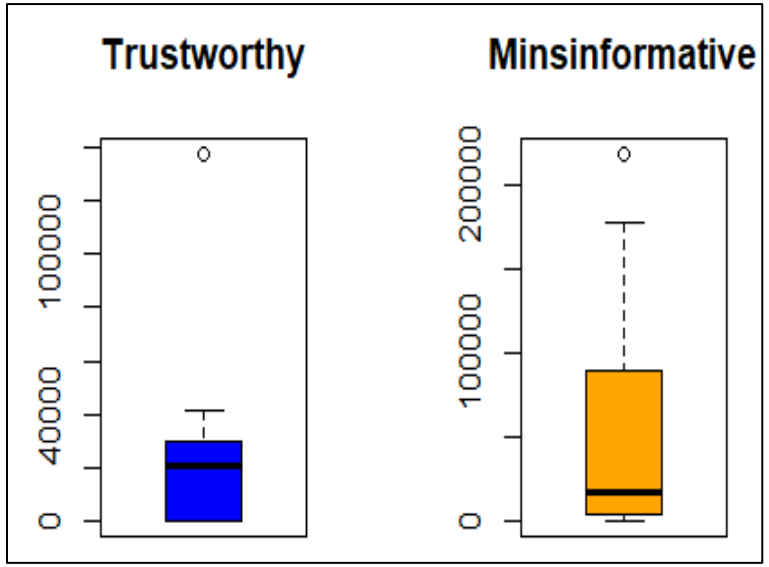
- Number of views for the video.
- Number of thumbs up.
- Number of thumbs down.
- Number of Comments in a video.

All the viewer engagement features were extracted using official YouTube API.

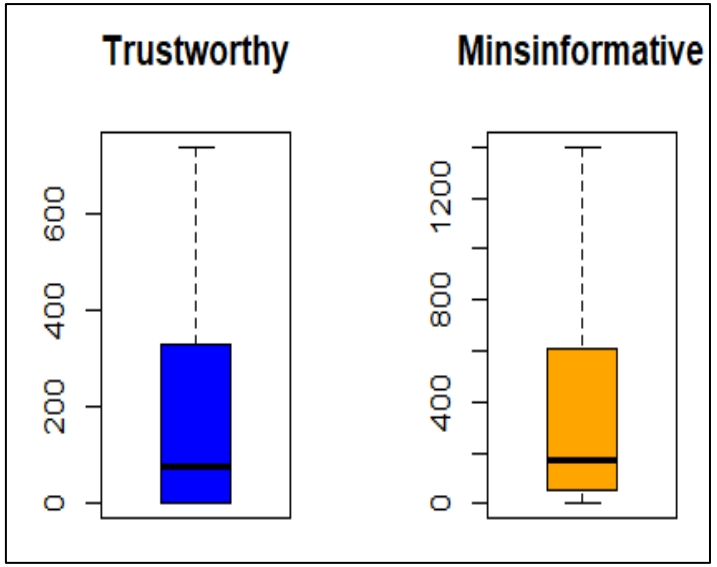
Below Figure shows the distribution of viewer engagement features for both Informative and Misinformative videos.

Distribution of Viewer Engagement Features:

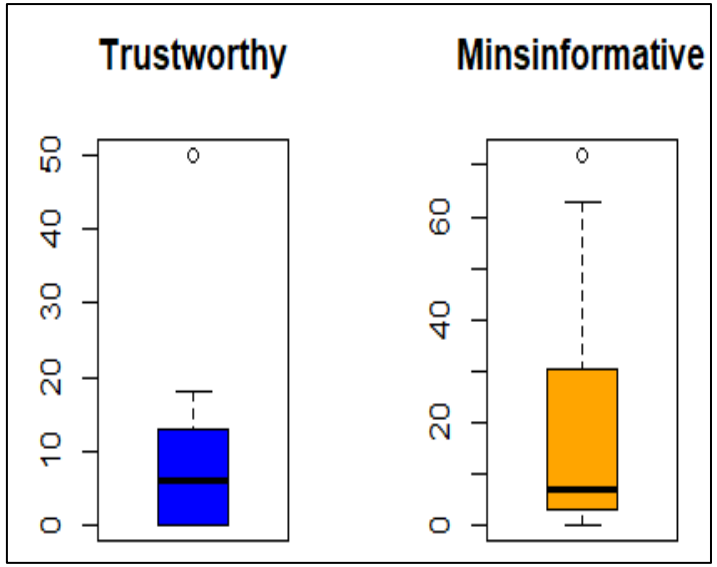
Avg No of views



Thumbs up



Thumbs Down



Appendix A3: Descriptive statistics for viewer engagement features.

A Snapshot of the dataset:

ID	URL	Discern Sc	Upvotes	Downvote	Views	Transcript	Comment	Topics	Informative
1	https://w	24	7	0	71	learning	NA	0	Non-Informative
2	https://w	26	11	1	421	doctors	1	6	Non-Informative
3	https://w	25	2	0	126	hi I'm dr.	NA	4	Informative
4	https://w	21	1300	63	218536	darling	1	6	Non-Informative
5	https://w	26	3	0	257	we're	1	1	Non-Informative
6	https://w	25	439	7	27827	Wayne	1	1	Non-Informative
7	https://w	23	228	2	7785	I went to	1	1	Non-Informative
8	https://w	22	31	4	2625	mm a	1	6	Non-Informative
9	https://w	46	749	21	44152	learning n	2	0	Non-Informative
10	https://w	48	464	35	119848	Hi, I'm	0	0	Non-Informative
11	https://w	52	388	37	123726	breast	0	0	Non-Informative
12	https://w	51	1400	72	177439	want to	2	6	Non-Informative
13	https://w	44	94	7	21213	Most	1	1	Non-Informative
14	https://w	48	464	35	119848	Hi, I'm	NA	0	Non-Informative
15	https://w	46	50	7	11611	Breast	NA	0	Non-Informative
16	https://w	47	737	50	137546	my name	0	2	Informative
17	https://w	50	762	10	39961	the	1	1	Non-Informative
18	https://w	53	230	12	20978	have you	0	5	Informative
19	https://w	52	75	6	26710	evaluatin	NA	2	Informative
20	https://w	49	330	18	41259	how to	NA	4	Informative
21	https://w	30	3	0	273	dr. V welc	0	2	Informative

Appendix A4: Initial dataset.

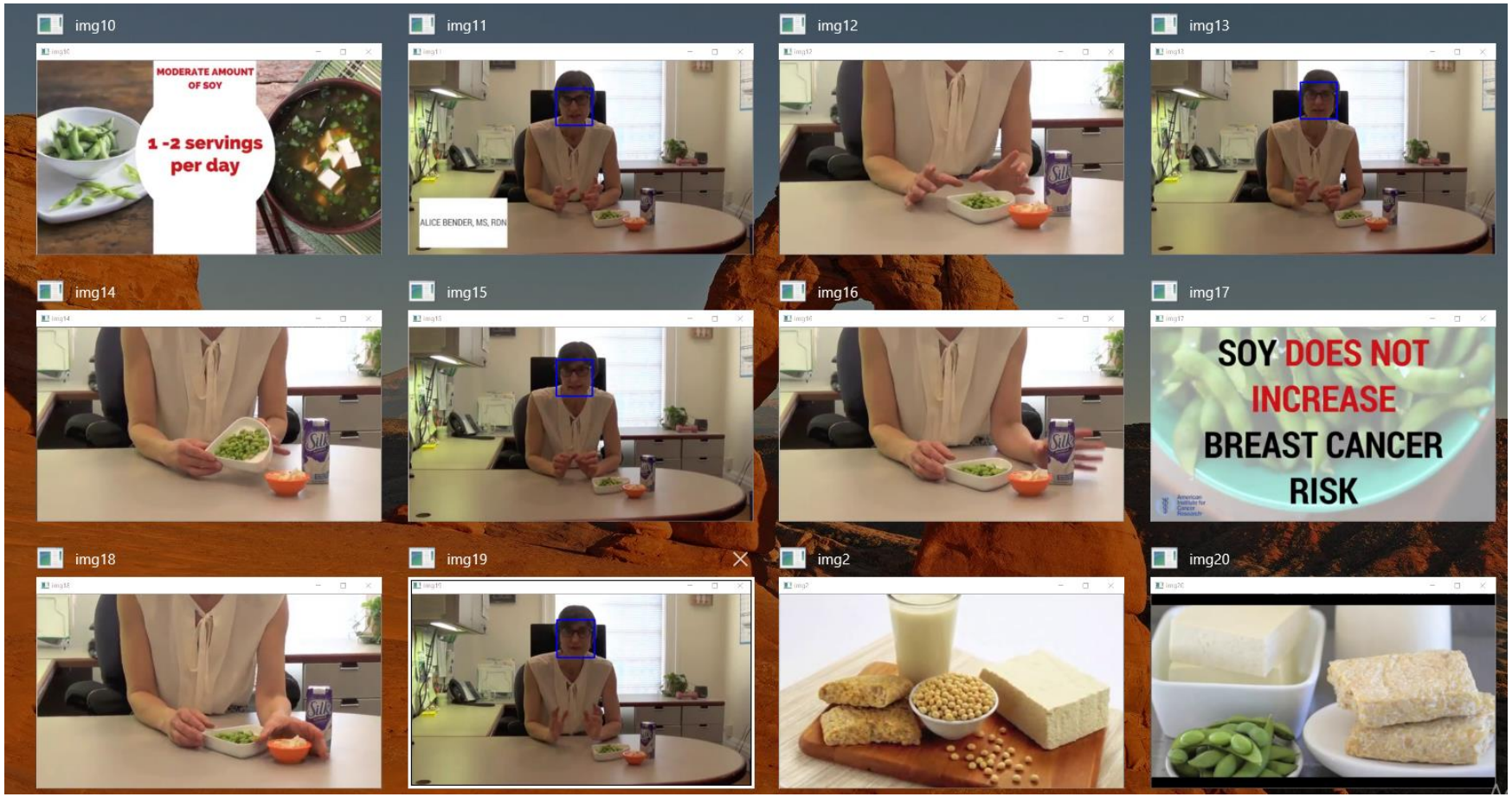
Image Analysis:

- We perform image analysis techniques to classify whether a video has animated speaker or a human speaker.

Iteration:

- All the videos are passed through a loop: in the loop 2000 frames of each videos are extracted.
- Each second in the video contains multiple frames. 21 frames of the video, covering frames at the beginning of the video, at the middle of it and the end are checked.
- OpenCV’s HAAR cascade front face classifier is an XML file containing facial features of a person. Different frames are imported and each of them are passed through OpenCV’s detect Multiscale function to detect if a human Face is available.

Frames of the YouTube video with an expert speaking – Non animated video:



Appendix A5: Sample Non animated video.

- In this example, 7 of the 21 frames contain a human face. So at-least a third of the frames contain a human face (a medical expert or a patient) speaking about breast cancer. Hence the video is declared to be non-animated.

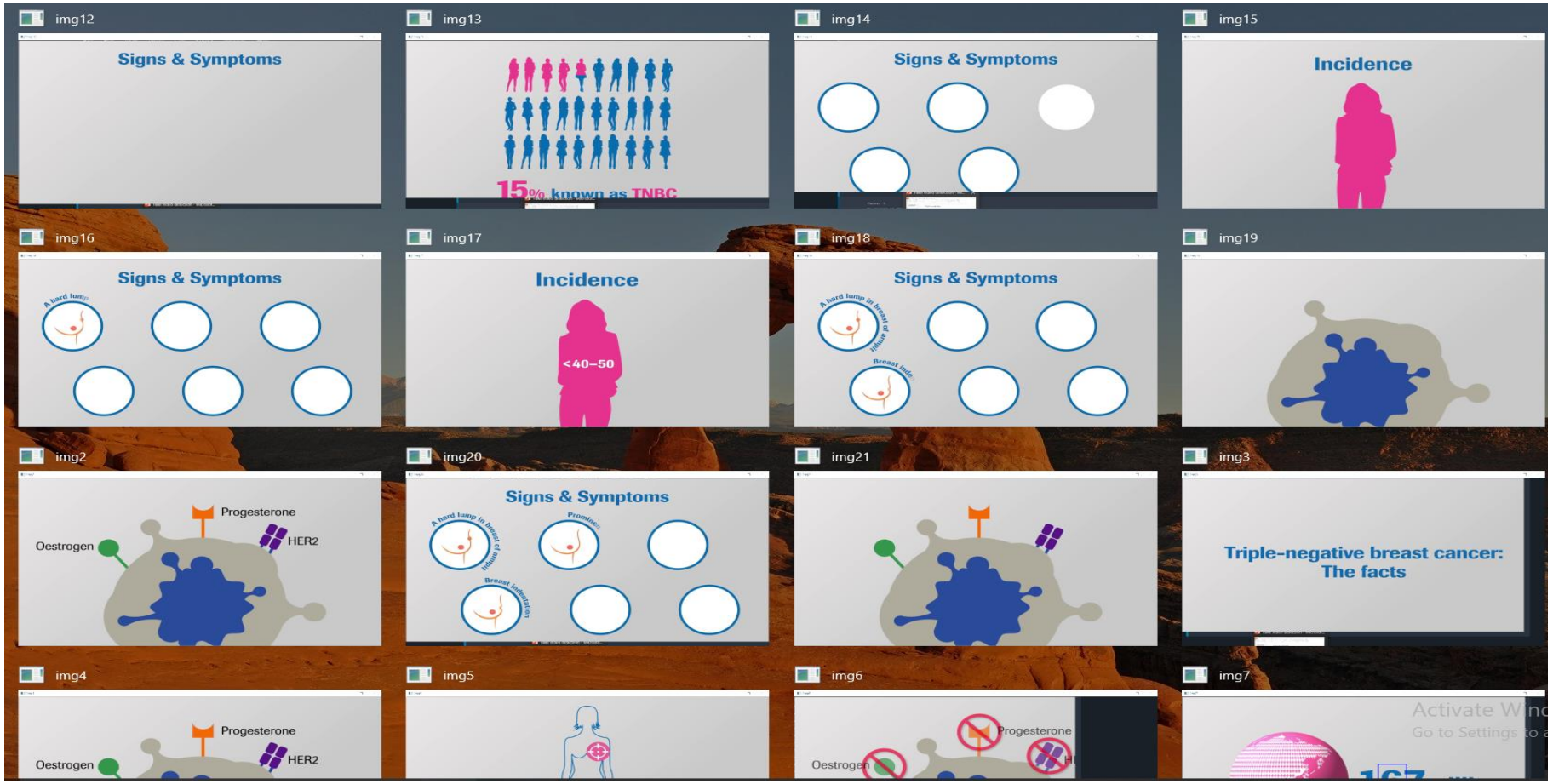
```
cv2.imshow('img1', img1)
cv2.imshow('img2', img2)
cv2.imshow('img3', img3)
cv2.imshow('img4', img4)
cv2.imshow('img5', img5)
cv2.imshow('img6', img6)
cv2.imshow('img7', img7)
cv2.imshow('img8', img8)
cv2.imshow('img9', img9)
cv2.imshow('img10', img10)
cv2.imshow('img11', img11)
cv2.imshow('img12', img12)
cv2.imshow('img13', img13)
cv2.imshow('img14', img14)
cv2.imshow('img15', img15)
cv2.imshow('img16', img16)
cv2.imshow('img17', img17)
cv2.imshow('img18', img18)
cv2.imshow('img19', img19)
cv2.imshow('img20', img20)
cv2.imshow('img21', img21)

cv2.waitKey()
```

```
7
Video is not animated
```

Appendix A6: Image frames for Non animated video.

Different frames of an animated YouTube video (with No human expert speaking):



Appendix A7: Image frames for Animated video.

Instances of animated videos

- As seen, the animated video contains no human face
- There is no expert/patient speaking about cure or treatment
- The model detects no face and deems it animated

- Considering the 100 videos it is observed that if there are 3 or lesser instances of human face then the video is most likely to be an animated video
- Around 80% of the listed videos were found to be non-animated and hence used further for text analysis.

```
cv2.imshow('img2', img2)
cv2.imshow('img3', img3)
cv2.imshow('img4', img4)
cv2.imshow('img5', img5)
cv2.imshow('img6', img6)
cv2.imshow('img7', img7)
cv2.imshow('img8', img8)
cv2.imshow('img9', img9)
cv2.imshow('img10', img10)
cv2.imshow('img11', img11)
cv2.imshow('img12', img12)
cv2.imshow('img13', img13)
cv2.imshow('img14', img14)
cv2.imshow('img15', img15)
cv2.imshow('img16', img16)
cv2.imshow('img17', img17)
cv2.imshow('img18', img18)
cv2.imshow('img19', img19)
cv2.imshow('img20', img20)
cv2.imshow('img21', img21)

cv2.waitKey()

Faces: 0
No person in the video. Mostly animated
```

Appendix A8: Classifying an image with no human faces.

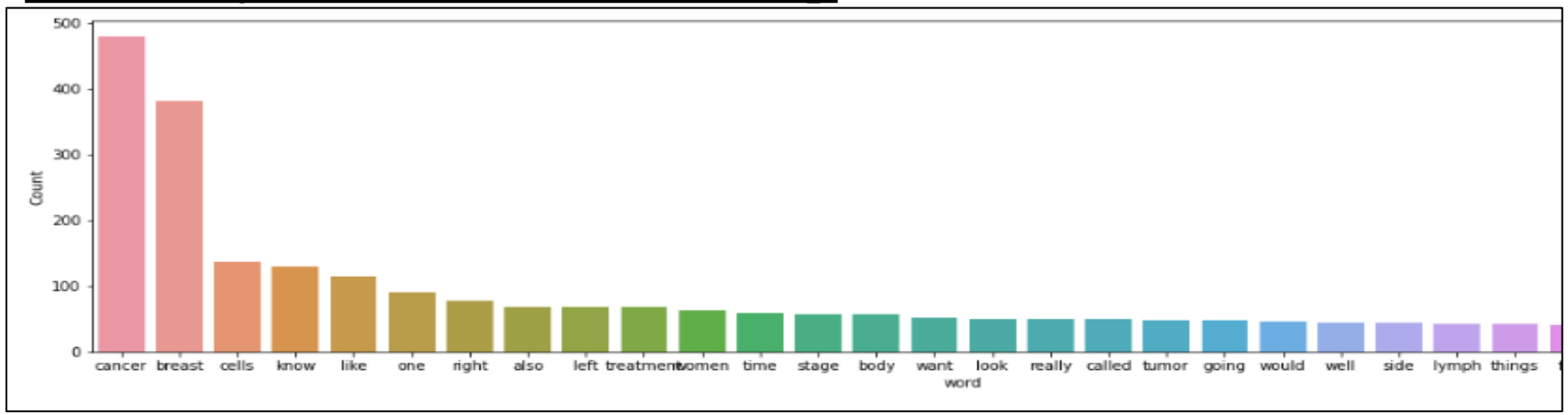
Analysis:

Data Pre-Processing:

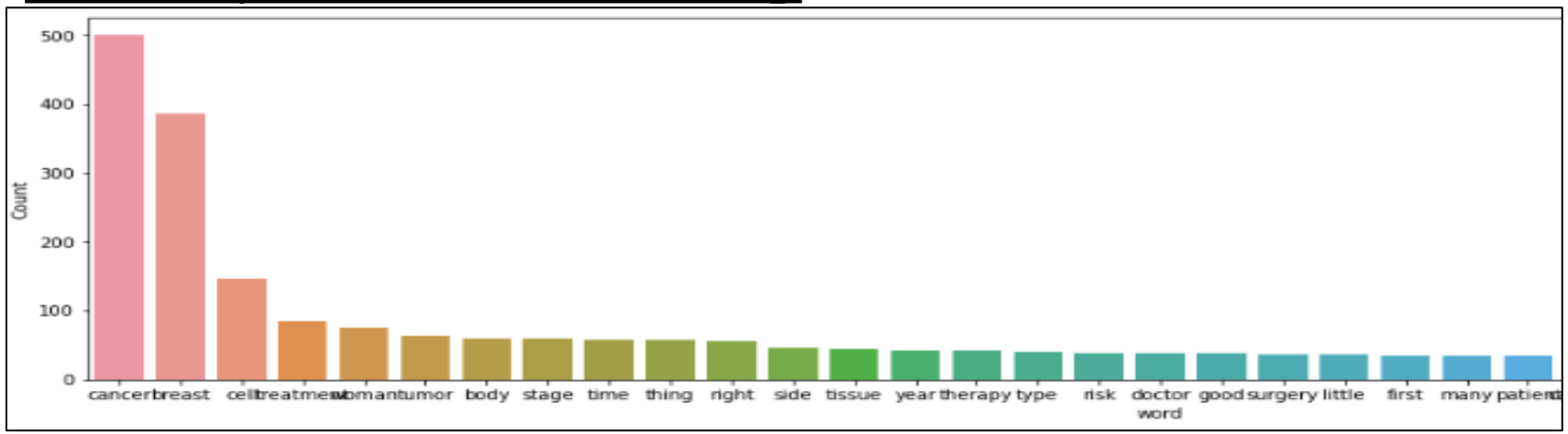
- Steps involved in Data Processing:
- Step 1: Removing the Stop words, Unwanted characters like ("[^a-zA-Z#]", @," "), converting the case of the transcript texts to Lower.
- Step 2: Tokenizing the scripts into word tokens.
- Step 3: We use Lemmatization using spaCY library to reduce a given word to its base i-e reducing the multiple forms of a word to a single word.
- Step 4: Creating the dictionary and Document term matrix for as specified input formats for LDA Gensim Model.

Visualizing the frequent terms:

Term Frequencies Before Processing:



Term Frequencies After Processing:



Appendix A9: Distribution of words before and after processing

Video: Transcript Analysis

- We perform Topic Modelling to identify the different topics available in the videos.
- We perform both Unsupervised and semi supervised topic modelling to identify the topics.
- We have used unsupervised LDA modelling to identify the topics discussed frequently on transcriptions from all the videos.
- We used gensim library to perform LDA unsupervised topic modelling.
- Identified four distinct topics and created a bag of words for these topics identified through various research papers regarding breast cancer and cancer.
- Compared these bag of words with all the transcriptions using corex algorithm.

*Bag of words file has been attached along with the code file.

Unsupervised Topic Modelling:

- We build our LDA model over the Document term matrix prepared from the dictionary.
- Running our model with number of topics as 7 and number of words per topic as 10 to reduce the overlap of words between topics to get a clear distinction between topics.
- Once we print the results of LDA model we name the topics identified as follows based on the words available under the topic.
- *Treatment and therapy (Topic 0), Breast cancer after becoming Mother (Topic 1), Cancer Lumps (Topic 2), Natural Medicines (Topic 3), Immune System (Topic 4), Breast cancer Diagnosis (Topic 5), Oil Usage cures (Topic 5).*

Unsupervised Topic Modelling:

Identified Topics (Number of Words=7, Number of Words =7):

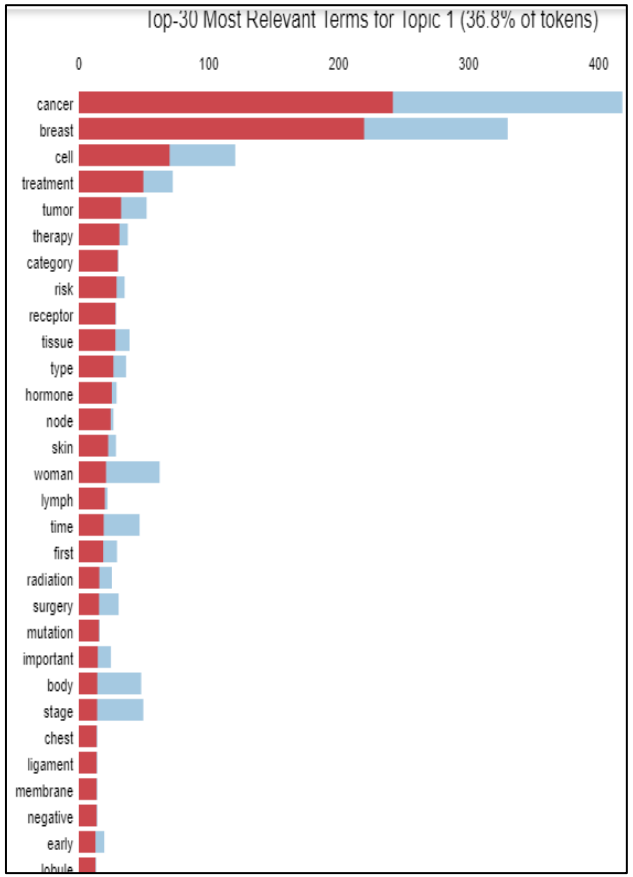
- **Treatment and therapy:**(0, '0.079*"cancer" + 0.072*"breast" + 0.023*"cell" + 0.016*"treatment" + 0.011*"tumor" + 0.010*"therapy" + 0.010*"category" + 0.009*"risk" + 0.009*"tissue" + 0.009*"receptor"')
- **Breast cancer after becoming Mother:**(1, '0.034*"cancer" + 0.021*"breast" + 0.020*"doctor" + 0.018*"mom" + 0.010*"life" + 0.010*"surgery" + 0.010*"good" + 0.008*"year" + 0.008*"treatment" + 0.008*"woman"')
- **Cancer Lumps:**(2, '0.028*"breast" + 0.028*"woman" + 0.015*"cancer" + 0.014*"lump" + 0.012*"thing" + 0.010*"people" + 0.009*"time" + 0.007*"number" + 0.007*"young" + 0.006*"little"')
- **Natural Medicines:**(3, '0.012*"cancer" + 0.010*"growth" + 0.008*"help" + 0.008*"spice" + 0.005*"inflammatory" + 0.005*"inhibit" + 0.005*"turmeric" + 0.005*"herb" + 0.005*"tumor" + 0.003*"black"')
- **Immune System:**(4, '0.040*"cancer" + 0.022*"cell" + 0.018*"breast" + 0.011*"body" + 0.009*"image" + 0.008*"view" + 0.008*"system" + 0.008*"immune" + 0.007*"third" + 0.007*"next"')
- **Breast cancer Diagnosis:**(5, '0.031*"right" + 0.025*"cancer" + 0.024*"side" + 0.024*"cell" + 0.017*"breast" + 0.017*"difference" + 0.015*"mouse" + 0.014*"tumor" + 0.013*"mammary" + 0.012*"patient"')
- **Oil Usage cures:**(6, '0.049*"cancer" + 0.021*"stage" + 0.021*"breast" + 0.012*"oil" + 0.009*"good" + 0.008*"many" + 0.007*"day" + 0.007*"cell" + 0.007*"use" + 0.007*"patient"')]

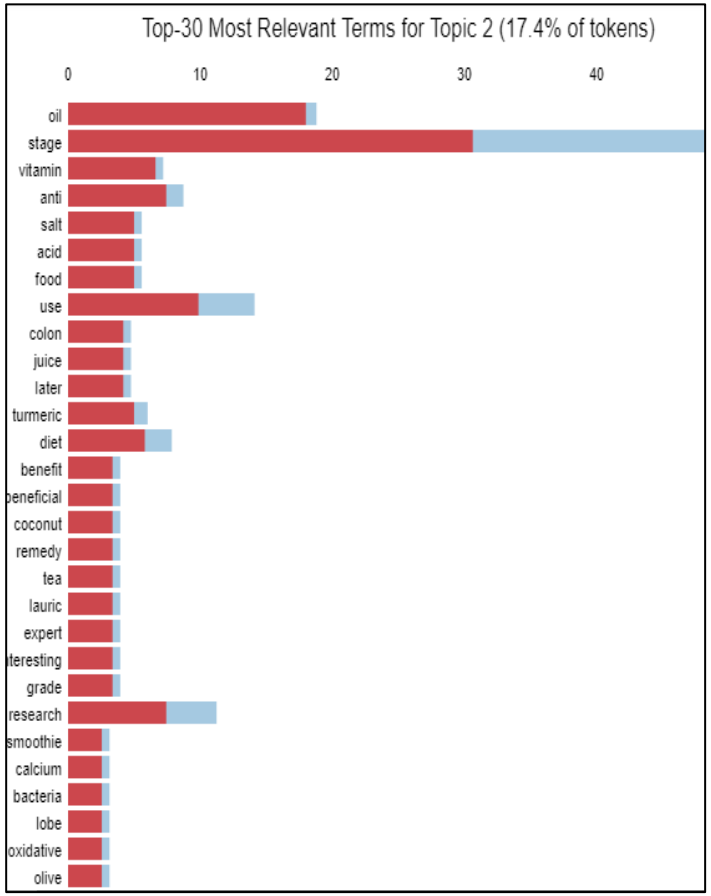
Appendix A10: % Distribution of words in topics LDA.

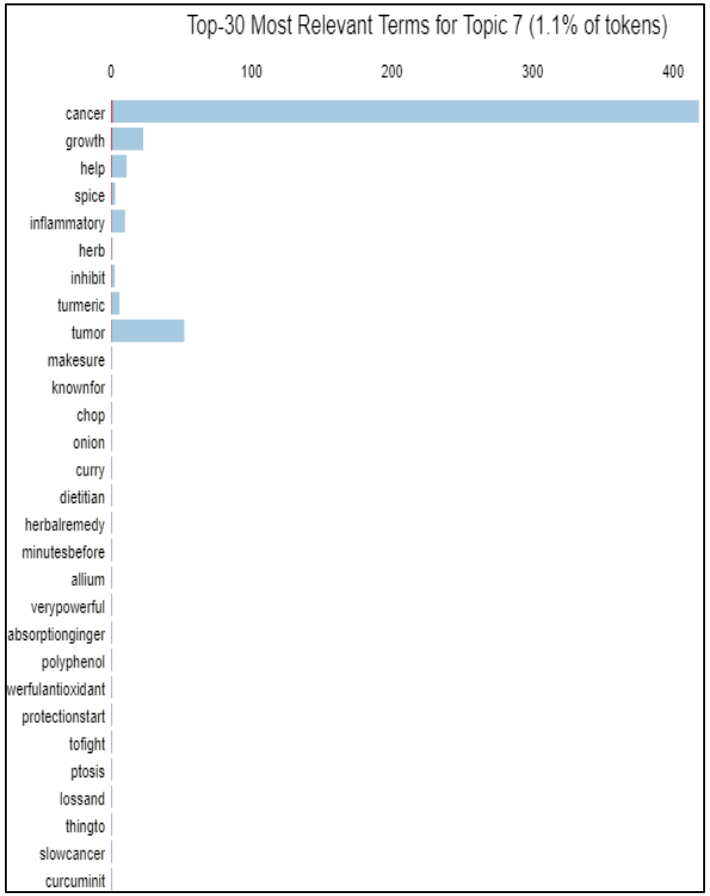
Topic Visualization:

- We use pyLDAvis library to visualize our topics and the word distribution within the topics.
- Significant words like “Breast”, “Cancer” are scarcely distributed in misinformative topics like “Natural cures” and “Oil usage Cures” whereas these words are concentrated in other topics.
- LDA displayed only the top 10 words whereas pyLDAvis chart helped to drill down from the very frequent to less frequent words.
- Comparing transcriptions of every videos with the identified topics and labelled them as topics 0 through 1.
- Combining the videos with dominant Topics 3 (Natural Medicines),Topic 1 and 6 (Oil Usage Cures)as **Non-Informative**.
- Combining the videos with dominant Topics 0(Treatment and therapy), 2 (Cancer Lumps), 4(Immune System) and 5(Breast cancer Diagnosis)as **Informative** videos
- Clear distribution of words from the transcripts are clearly displayed in the below chart.

Word Distribution between Topics:







Appendix A11: Distribution of words between topics.

Identifying Dominant Topics in all Videos:

	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	0.0	0.9986	cancer, breast, cell, treatment, tumor, therapy, category, risk, tissue, receptor	learning medicine hard work osmosis makes easy takes lectures notes create personalized study plan exclusive videos practice questions flashcards much try free today breast cancer breast carcinoma...
1	1	6.0	0.9952	cancer, stage, breast, oil, good, many, day, cell, use, patient	doctors shock kills colon cancer two days unfortunately colon cancer one common cancer types men women terrible disease takes millions lives around world wait good news group experts found powerfu...
2	2	4.0	0.9879	cancer, cell, breast, body, image, view, system, immune, third, next	miller tasty way fight cancer next time waiter armed giant pepper mill asked whether like fresh ground salad say yes please recent lab study ingredient responsible peppers pungent zing called pip ...
3	3	6.0	0.9208	cancer, stage, breast, oil, good, many, day, cell, use, patient	darling vertical garden logo shop found heart yeah god herbs lemongrass use also cure cancer bail good mosquitoes good antibiotic sweet basil oregano another antibiotic got call good brain help li...
4	4	1.0	0.9931	cancer, breast, doctor, mom, life, surgery, good, year, treatment, woman	taking action health next several weeks going take look ways improve well new procedures may give better quality life week looking breakthroughs breast cancer drug cocktail called miracle women ag...
5	5	1.0	0.9948	cancer, breast, doctor, mom, life, surgery, good, year, treatment, woman	wayne higgins girlfriend heather dated four years finally popped question set date started planning big day one thing failed consider wayne worked postal worker lost weight developed persistent co...

Appendix A12: Dominant topics in each videos.

Semi Supervised Topic Modelling:

- Semi supervised algorithm is flexible and allows user to use words as anchors to promote a clear separability and classification of topics from transcriptions.
- Creating a bag of words for 4 different topics “Symptoms”, ”Prevention”, ”Treatment” and “Remedies identified from research papers related to breast cancer.
- We use corex or Correlation explanation a semi supervised topic model allowed to provide the model with “anchor words” using which our model compares and identifies the potential topic in each transcripts.
- Each transcript is then subjected to semi supervised modelling where percentage of the transcript covering the 4 topics is shown by the model
- We also specify in the model how much weights it should give for the anchor words.

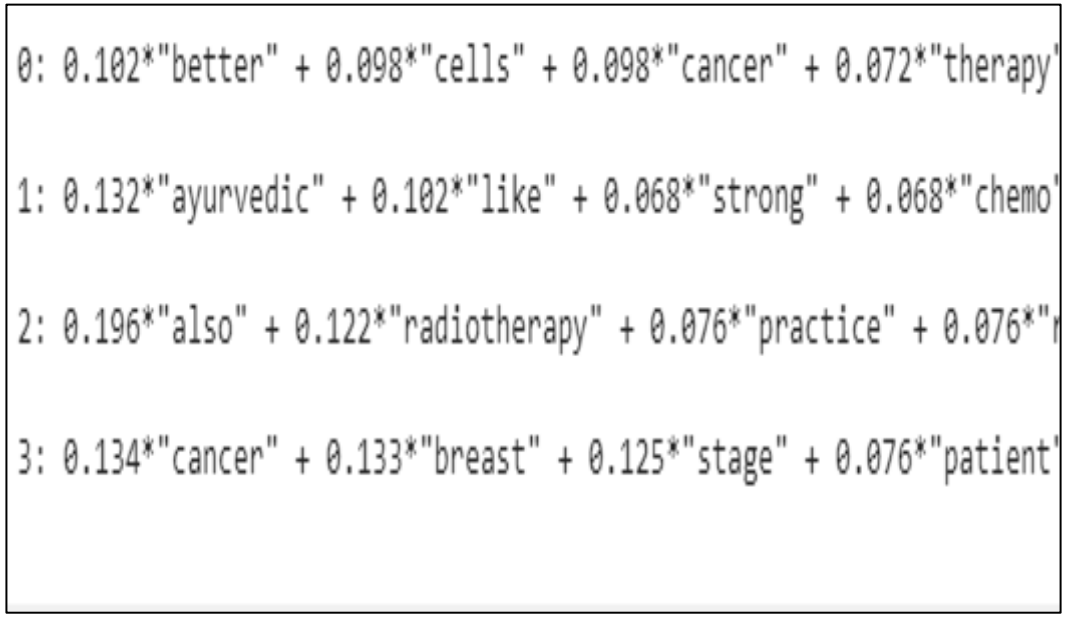
Distribution of words:

Bag of Words

Symptoms	Prevention	Treatment	NaturalRemedies
invasive	concerned	Tests	centuries
ductal	developing	procedures	herbs
carcinoma	wondering	diagnose	plants
IDC	steps	exam	medicinal
milk	help	doctor	purposes
ducts	prevent	check	food
distinct	risk	retain	variety
breast	factors	immune	active
lump	family	stimulating	phytochemicals
feel	history	anti-tumor	carotenoids
lobular	lifestyle	properties	flavonoids
ILC	changes	Mammogram	ligands
forms	make	X-ray	polyphenolics
milk-producing	lower	Mammograms	terpenoids

Appendix A13: Bag of words for breast cancer

word distribution in topics



Appendix A14: Word distribution in semi supervised topics

Corpus dictionary being created for the 4 topics:

```
In [52]:
dictionary = gensim.corpora.Dictionary(gen_docs)
print(dictionary.token2id)

corpus = [dictionary.doc2bow(gen_doc) for gen_doc in gen_docs]
corpus
```

```
{',': 0, '.': 1, 'abdominal': 2, 'abnormal': 3, 'aching': 4, 'affected': 5, 'alertness': 6, 'antibiot': 7, 'antibiotics': 8, 'appeara
nce': 9, 'arm': 10, 'basal': 11, 'beneath': 12, 'biopsy': 13, 'bloody': 14, 'blurred': 15, 'bones': 16, 'brain': 17, 'breast': 18, 'b
reast-feeding': 19, 'breasts': 20, 'breath': 21, 'breathing': 22, 'burning': 23, 'calcium': 24, 'carcinoma': 25, 'carcinomas': 26, 'c
ells': 27, 'change': 28, 'changes': 29, 'characteristics': 30, 'chest': 31, 'collarb': 32, 'collection': 33, 'color': 34, 'concern':
35, 'constipation': 36, 'cord': 37, 'coughing': 38, 'cyst': 39, 'd': 40, 'dcis': 41, 'detected': 42, 'development': 43, 'differences
': 44, 'difficulty': 45, 'dimpling': 46, 'discharge': 47, 'distinct': 48, 'division': 49, 'ductal': 50, 'ducts': 51, 'eczema': 52, 'e
strogen': 53, 'evaluation': 54, 'even': 55, 'exam': 56, 'extreme': 57, 'facing': 58, 'fatigue': 59, 'feel': 60, 'feet': 61, 'felt': 6
2, 'flaking': 63, 'fluid': 64, 'forms': 65, 'fractures': 66, 'generating': 67, 'genetic': 68, 'girth': 69, 'glands': 70, 'hands': 71,
'hard': 72, 'headache': 73, 'heaviness': 74, 'history': 75, 'hormone': 76, 'ibc': 77, 'idc': 78, 'ilc': 79, 'illness': 80, 'increase
': 81, 'infection': 82, 'inflammatory': 83, 'invasive': 84, 'inverted': 85, 'investigate': 86, 'inward': 87, 'irritated': 88, 'itchin
ess': 89, 'itchy': 90, 'larger': 91, 'lcis': 92, 'lining': 93, 'liver': 94, 'lobular': 95, 'loss': 96, 'lump': 97, 'lumps': 98, 'lung
s': 99, 'lymph': 100, 'mass': 101, 'mastitis': 102, 'memory': 103, 'metastatic': 104, 'microscope': 105, 'milk': 106, 'milk-producing
': 107, 'movement': 108, 'mucinous': 109, 'mucus': 110, 'nausea': 111, 'nipple': 112, 'nipples': 113, 'nodes': 114, 'notice': 115, 'n
ursing': 116, 'occur': 117, 'oran': 118, 'orange': 119, 'pain': 120, 'painless': 121, 'papillary': 122, 'peau': 123, 'peel': 124, 'pe
eling': 125, 'pitted': 126, 'pitting': 127, 'pregnant': 128, 'progesterone': 129, 'puckering': 130, 'rarely': 131, 'receptor': 132, '
receptors': 133, 'red': 134, 'redness': 135, 'retraction': 136, 'ridged': 137, 'risk': 138, 'routine': 139, 'scaly': 140, 'seeing': 1
41, 'seizure': 142, 'self-exam': 143, 'shape': 144, 'shortness': 145, 'similar': 146, 'situ': 147, 'size': 148, 'skin': 149, 'sorenes
s': 150, 'speech': 151, 'spinal': 152, 'spreads': 153, 'stage': 154, 'surface': 155, 'swelling': 156, 'swollen': 157, 'symptoms': 15
8, 'tender': 159, 'texture': 160, 'that': 161, 'therapy': 162, 'thickened': 163, 'thickening': 164, 'those': 165, 'touch': 166, 'trip
le-negative': 167, 'tube-like': 168, 'tumors': 169, 'turning': 170, 'uncontrolled': 171, 'under': 172, 'underarm': 173, 'unique': 17
4, 'unrelated': 175, 'unresolved': 176, 'visible': 177, 'vision': 178, 'wall': 179, 'warm': 180, 'yellowing': 181, '’': 182, "'s": 18
3, 'active': 184, 'advanced': 185, 'aggressive': 186, 'aggressiveness': 187, 'aim': 188, 'alongside': 189, 'analysis': 190, 'analyzed
': 191, 'anti-tumor': 192, 'appropriate': 193, 'area': 194, 'areas': 195, 'areola': 196, 'aromatase': 197, 'attack': 198, 'axillary':
```

Appendix A15: word corpus for 4 topics

Sample output of topic modelling:

```
In [49]: # for i,topic in lda_model.show_topics(formatted=True, num_topics=num_topics, num_words=10):
         #     print(str(i)+": "+ topic)
         #     print()
```

```
0: 0.102*"better" + 0.098*"cells" + 0.098*"cancer" + 0.072*"therapy" + 0.068*"enhances" + 0.068*"research" + 0.068*"outcomes" + 0.064*"migration" + 0.059*"growth" + 0.034*"system"

1: 0.132*"ayurvedic" + 0.102*"like" + 0.068*"strong" + 0.068*"chemo" + 0.068*"using" + 0.068*"music" + 0.068*"immune" + 0.068*"iowa" + 0.068*"system" + 0.068*"expert"

2: 0.196*"also" + 0.122*"radiotherapy" + 0.076*"practice" + 0.076*"recorded" + 0.057*"stage" + 0.041*"iowa" + 0.041*"outcomes" + 0.041*"use" + 0.041*"physician" + 0.041*"chemo"

3: 0.134*"cancer" + 0.133*"breast" + 0.125*"stage" + 0.076*"patient" + 0.074*"used" + 0.059*"treatment" + 0.057*"three" + 0.046*"every" + 0.039*"tissues" + 0.039*"help"
```

% of topics from transcripts being shown:

```
In [63]: for i in tf:
             print('Comparing Result:', sims[i])
```

```
Comparing Result: [0.         0.         0.         0.07358667]
Comparing Result: [0.         0.         0.         0.07358667]
Comparing Result: [0.         0.         0.04270986 0.03679334]
Comparing Result: [0.         0.         0.08541973 0.        ]
Comparing Result: [0.         0.         0.04270986 0.03679334]
Comparing Result: [0.         0.         0.09550215 0.01645448]
Comparing Result: [0. 0. 0. 0.]
Comparing Result: [0.05691027 0.         0.         0.05203363]
Comparing Result: [0. 0. 0. 0.]
Comparing Result: [0.         0.         0.06040087 0.05203363]
Comparing Result: [0.         0.         0.04270986 0.03679334]
Comparing Result: [0.05365552 0.01023726 0.01423662 0.04905778]
Comparing Result: [0.         0.         0.         0.07358667]
Comparing Result: [0. 0. 0. 0.]
Comparing Result: [0.08048327 0.         0.         0.        ]
Comparing Result: [0.         0.06142355 0.04270986 0.03679334]
Comparing Result: [0.04024164 0.03071178 0.         0.        ]
Comparing Result: [0.04464409 0.00851791 0.05922792 0.04081854]
Comparing Result: [0. 0. 0. 0.]
Comparing Result: [0.         0.         0.         0.07358667]
Comparing Result: [0.         0.         0.08541973 0.        ]
Comparing Result: [0.         0.         0.         0.07358667]
Comparing Result: [0.08048327 0.         0.         0.        ]
Comparing Result: [0.         0.0354629  0.09863421 0.        ]
```

- For example result of:

[0.04464409 0.00851791 0.05922792 0.04081854]

Tells us that transcript consist of 4.4%, 8%, 5% and 8% of topics covered in the corpus of dictionary created for of “Symptoms”, ”Prevention”, ”Treatment” and “Remedies”.

Appendix A16: %share of topics in each transcriptions

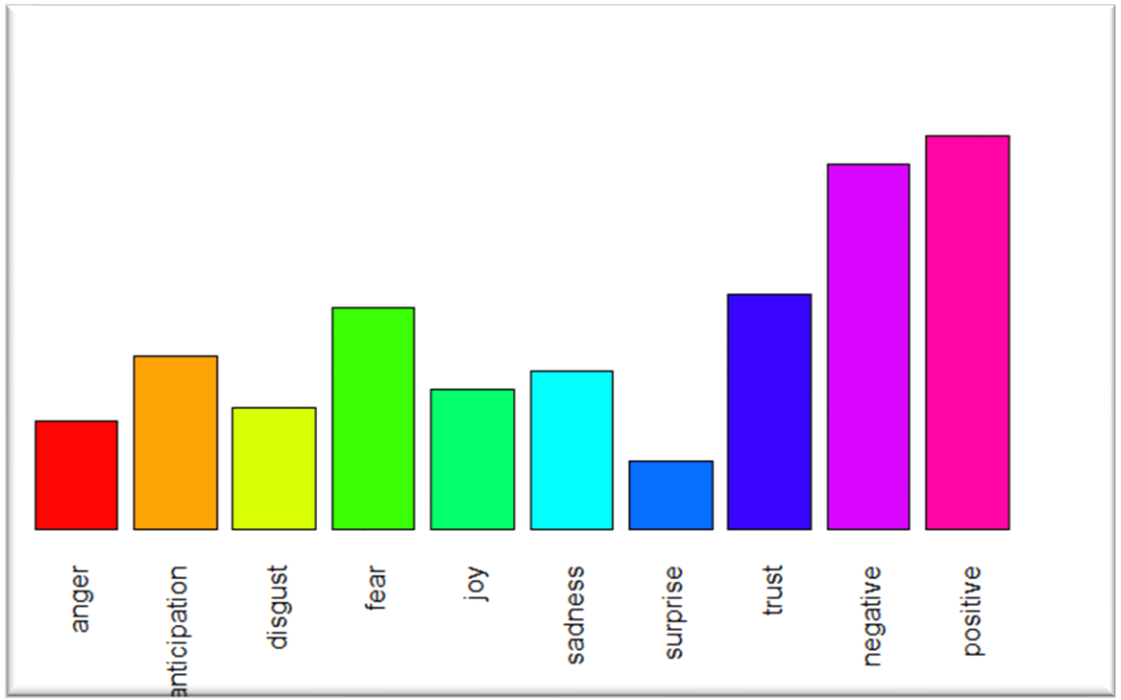
Sentiment Analysis in Comments:

- We use a coreNLP sentiment analysis package named “syuzhet” to get the sentiment scores for every comments.
- Aggregate all the sentiment scores of individual comments and produce a final sentiment value for the video.
- Following are the values assigned for sentiment values.

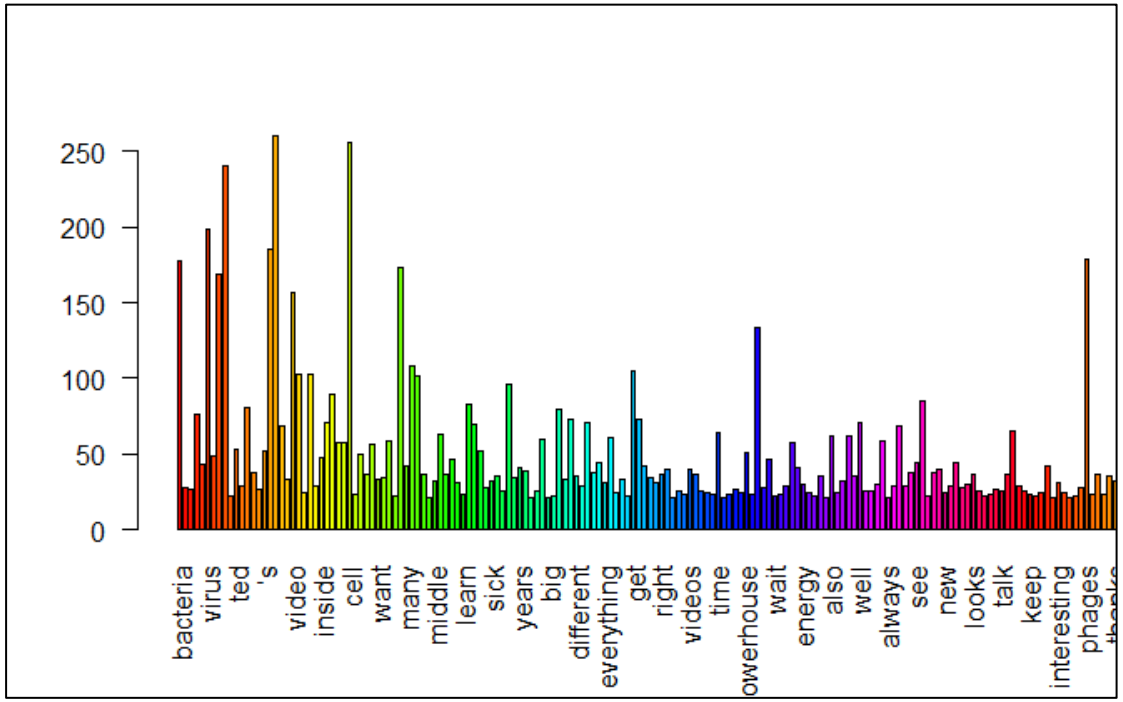
Sentiment Value	Sentiment Description
0	Negative Sentiment
1	Positive Sentiment
2	Neutral Sentiment

Appendix A17: Comment sentiment labels

Sentiment and Word Distribution in Comments:



Appendix A18: Aggregated Comment sentiment scores



Appendix A19: word frequency distribution in comments

Modelling:
BAG OF WORDS & TF-IDF

One tool we can use for doing this is called **Bag of Words**. BoW converts text into the matrix of occurrence of words within a given document. It focuses on whether given words occurred or not in the document, and it generates a matrix that we might see referred to as a BoW matrix or a document term matrix.

We'll also want to look at the **TF-IDF (Term Frequency-Inverse Document Frequency)** for our terms. This sounds complicated, but it's simply a way of normalizing our Bag of Words(BoW) by looking at each word's frequency in comparison to the document frequency. In other words, it's a way of representing how important a particular term is in the context of a given document, based on how many times the term appears and how many other documents that same term appears in. The higher the TF-IDF, the more important that term is to that document.

TRAIN TEST DATA SPLIT IN 70% AND 30% PROPORTIONS

```
# Split the data into train and test set

In [69]: from sklearn.model_selection import train_test_split

         X = data['Transcript'] # the features we want to analyze
         ylabels = data['Informative'] # the labels, or answers, we want to test against

         X_train, X_test, y_train, y_test = train_test_split(X, ylabels, test_size=0.3)
```

Appendix A20: Data split for Modelling

MODEL IS TRAINED AND TESTED ON A LOGISTIC REGRESSION CLASSIFIER

Cleaning, Vectorizer and Logistic regression classifier

In [70]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()

# Create pipeline using Bag of Words
pipe = Pipeline([("cleaner", predictors()),
                 ('vectorizer', bow_vector),
                 ('classifier', classifier)])

# model generation
pipe.fit(X_train,y_train)
```

```
Pipeline(memory=None,
     steps=[('cleaner', <__main__.predictors object at 0x000001E2B1B0AA58>), ('vectorizer', CountVectorizer(analyzer='word', binary=Fals
e, decode_error='strict',
        dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
        lowercase=True, max_df=1.0, max_features=None, min_df=1,
       ...penalty='l2', random_state=None, solver='warn',
          tol=0.0001, verbose=0, warm_start=False))])
```

Appendix A21: Transforming the data

Logistic Regression model and it’s accuracy

Running the pipeline and predicting using Logistic regression

In [71]:

```
from sklearn import metrics
# Predicting with a test dataset
predicted = pipe.predict(X_test)

# Model Accuracy
print("Logistic Regression Accuracy:",metrics.accuracy_score(y_test, predicted))
```

```
Logistic Regression Accuracy: 0.6666666666666666
```

Appendix A22: Prediction and classification accuracy

Conclusions:

- Logistic regression to classify the videos by integrating features like viewer engagement features, topics etc trained over a dataset of 100 YouTube videos.
- We achieved a final accuracy percentage of 66.66% for logistic regression.
- Unsupervised topic modelling to classify the transcripts as Informative/Non-Informative based on the topic dominance explained by the transcriptions of the videos.
- Semi-supervised topic modelling was able to give accurate classification guided in the right direction using anchor words collected in bag of words.