Evaluation Crieteria

There are two methods of calculating text similarity:

Term Frequency-inverse document frequency (TF-idf): This examines terms that exist in both pieces of text and assigns a score depending on how often they appear.

Semantic Similarity: This score words based on how similar they are, even if they are not exact matches.

Data Insights

The dataset provided is textual data. It contains three features, *text, reason,* and *label*. The text column and reason column are same but with different wordings/meanings and the label column has values if given text is more similar to reason provided then flagged to 1 else 0.

Text length distribution:

```
(array([657., 522., 338., 219., 110.,  71.,  45.,  44.,  15.,  15.,   9.,
          2.,   4.,   3.,   0.,   4.,   1.,   1.,   0.,   1.]),
 array([ 4. ,  7.1, 10.2, 13.3, 16.4, 19.5, 22.6, 25.7, 28.8, 31.9, 35. ,
        38.1, 41.2, 44.3, 47.4, 50.5, 53.6, 56.7, 59.8, 62.9, 66. ]),
 <BarContainer object of 20 artists>)
```

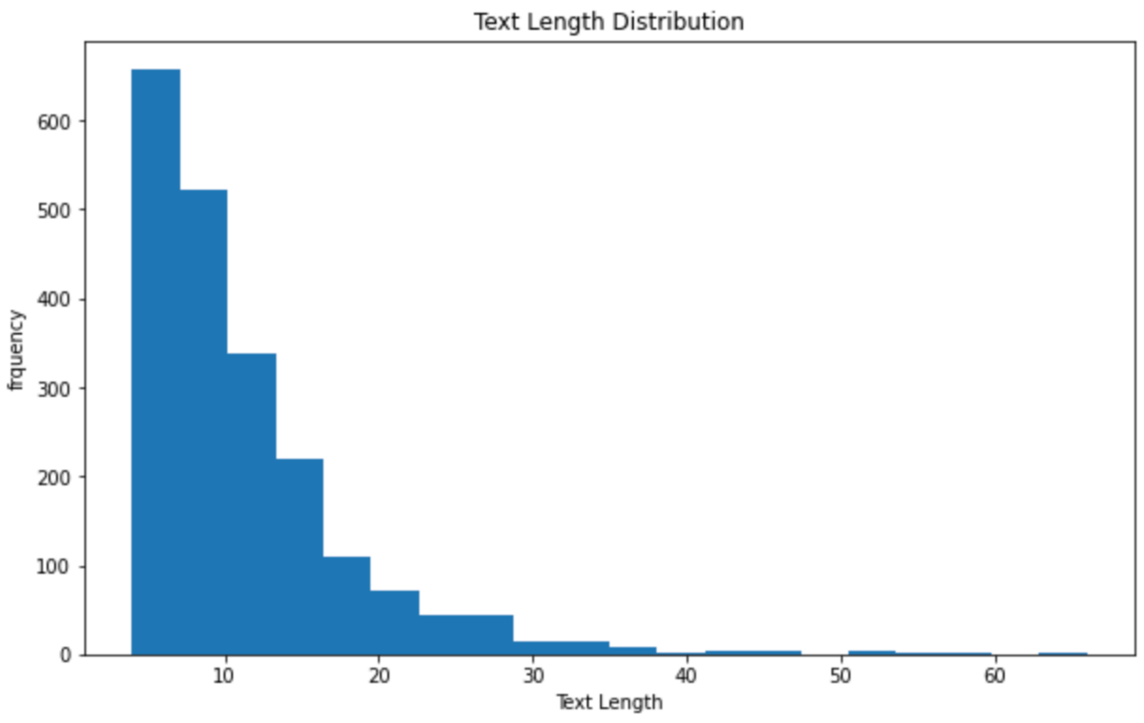
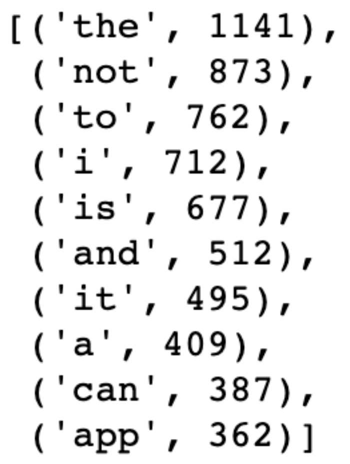


Figure 1 Text distribution

- The above histogram plot shows, there are 657 instances are having text length 4. Similarly, 522 instances are in range 7, 338 instances are in range 10, 219 instances are in range 13, 110 instances are in range 16, 71 instances are in range 19.
- Overall, 1917 instances are in range 20 text length.
- For the given text data has highest text length of 66 and has only one instance, the second highest is 60 and third highest is 57 are also having only one instance each.

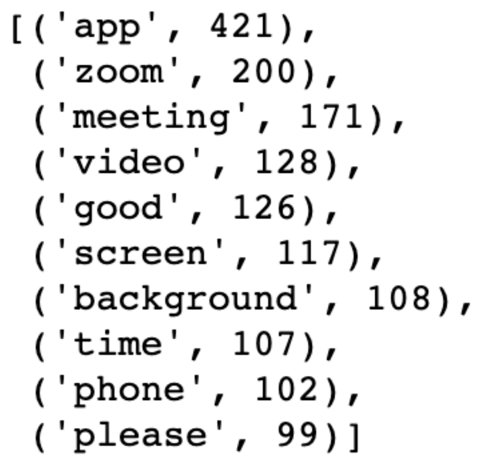
Word Frequency:



```
[('the', 1141),
 ('not', 873),
 ('to', 762),
 ('i', 712),
 ('is', 677),
 ('and', 512),
 ('it', 495),
 ('a', 409),
 ('can', 387),
 ('app', 362)]
```

Figure 2 Word Frequency_1

- It is obvious that most of the words repeated in a text document is shown in Figure 2.
- But, after preprocessing which includes the stopwords (are which does not carry any information) removal and lemmatization (are representing the same word in different wordings), the top 10 words are displayed as shown in Figure 3.



```
[('app', 421),
 ('zoom', 200),
 ('meeting', 171),
 ('video', 128),
 ('good', 126),
 ('screen', 117),
 ('background', 108),
 ('time', 107),
 ('phone', 102),
 ('please', 99)]
```

Figure 3 Word Frequency_2

The visual representation of the most repeated words in wordcloud as shown in Figure 4.

```
<matplotlib.image.AxesImage at 0x7fc17467db50>
```

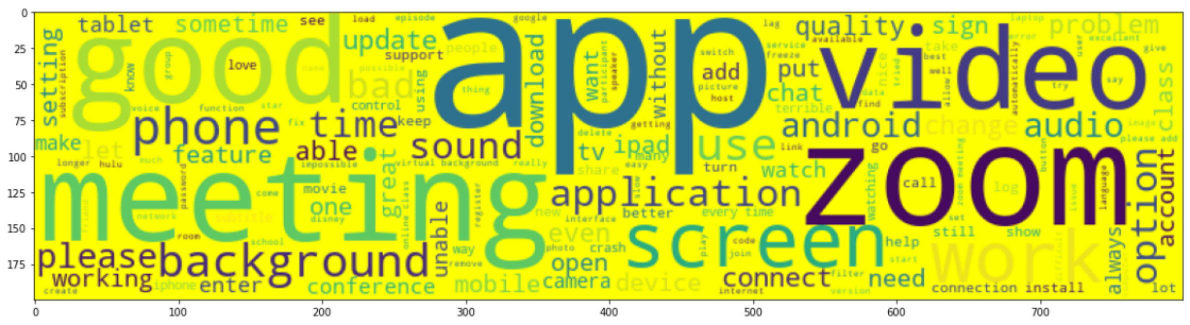


Figure 4 WordCloud

- Also, the least words are repeated in the text document are shown in Figure 5.

```
[('warns', 1),
 ('provided', 1),
 ('homosexuality', 1),
 ('queue', 1),
 ('frame', 1),
 ('childhood', 1),
 ('girl', 1),
 ('automated', 1),
 ('player', 1),
 ('reload', 1)]
```

Figure 5 Word Frequency_3

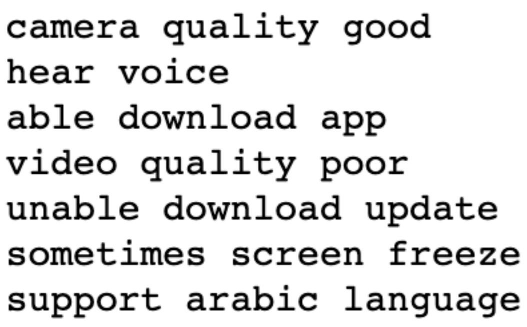
Duplicates:

- For the given original data has no duplicates but after preprocessing there are 7 duplicates generated.

```
camera quality is good, camera quality is not good .but
i can not hear the voice, because of this you can not hear the voice.
i am not able to download this app, i am not able to download the app.
video quality is very poor, video quality is very poor
unable to download updates, unable to download or update
sometimes the screen freezes, sometimes the screen freezes
but it does not support arabic language.., it does not support the arabic
language
```

Figure 6 Duplicates_1

The above Figure 6 has 2 sentences in each line which are separated by “,”. All these sentences look different but after preprocessing the two sentences in each line is formed as one sentence as shown in Figure 7.



```
camera quality good
hear voice
able download app
video quality poor
unable download update
sometimes screen freeze
support arabic language
```

Figure 7 Duplicates_2

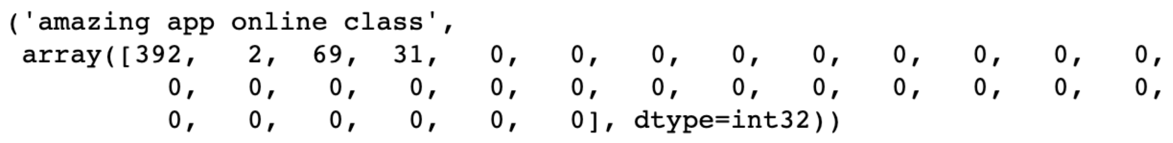
Ablation Table

Table 1Ablation Table

Model	Error percentage (%)
Cosine_similarity	36.05
all-MiniLM-L6-v2	11.27
distilbert-base-nli-mean-tokens	0.0

Training Approach

After all preprocessing steps, the data is free from special characters, stopwords removal, lemmatizing, and all text to lowercase. Now the data is ready to analyze.
Initially, in training approach the text document should be converted into numbers basically weights. There is a technique called Term Frequency-inverse document frequency (TF-idf) which gives weights of each word in a sentence. An example is shown in Figure 8.

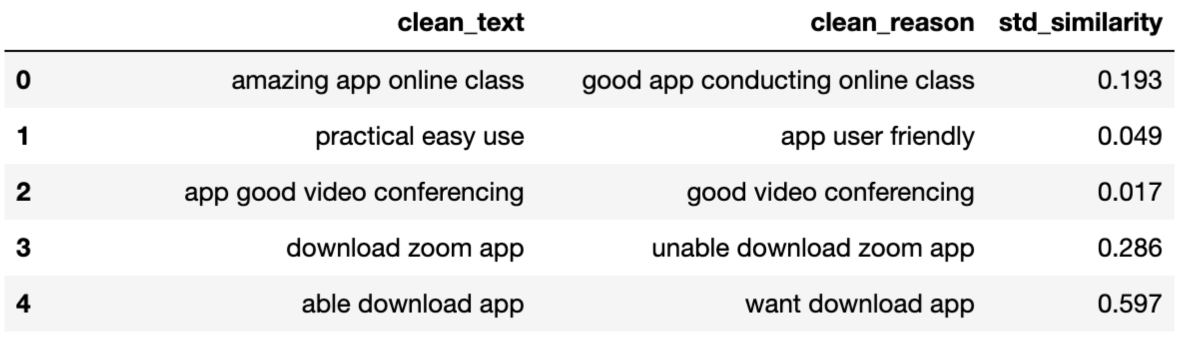


```
('amazing app online class',
 array([392,   2,  69,  31,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
          0,   0,   0,   0,   0,   0], dtype=int32))
```

Figure 8 Word to vector

First model:

From the sci-kit learn library there is a metrics.pairwise module which has a cosine_similarity function. Using cosine_similarity function, the given input text and a reason outputs the similarity value. For the first five examples are shown in Figure 9.



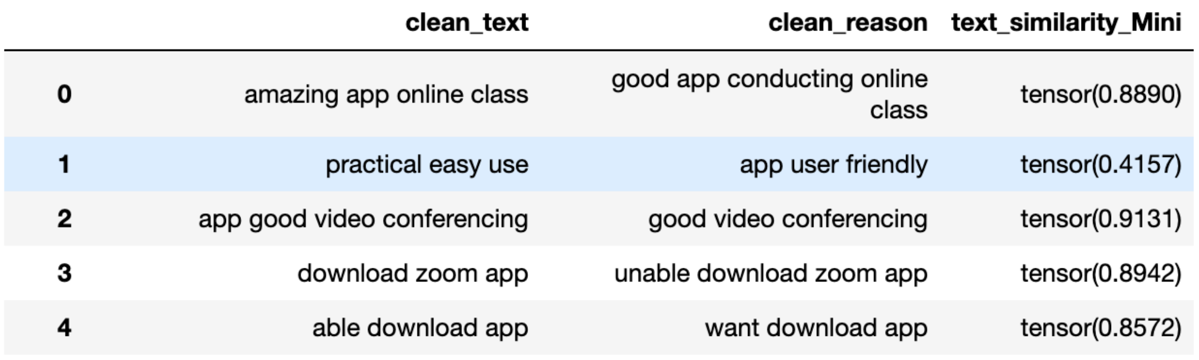
	clean_text	clean_reason	std_similarity
0	amazing app online class	good app conducting online class	0.193
1	practical easy use	app user friendly	0.049
2	app good video conferencing	good video conferencing	0.017
3	download zoom app	unable download zoom app	0.286
4	able download app	want download app	0.597

Figure 9 First model score

The std_similarity column shows the similarity values for each row. With just using cosine_similarity function, the model failed to spot the similarity between the given sentences due to lack of data as shown in Figure 9.

Second model:

The second model is sentence-transformers. The name of the model is 'all-MiniLM-L6-v2' which is pretrained model where it maps sentences and paragraphs to a 384-dimensional dense vector space. For the given input text and reason, the similarity between the sentences outputs a considerable value as shown in Figure 10.

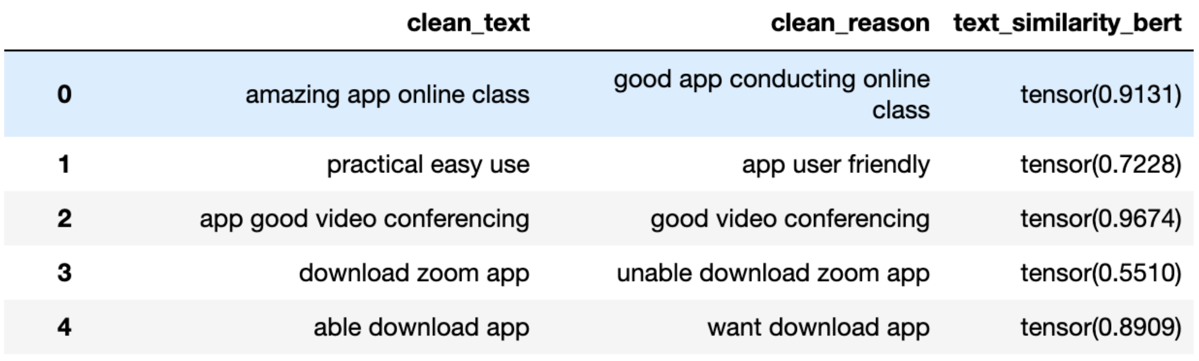


	clean_text	clean_reason	text_similarity_Mini
0	amazing app online class	good app conducting online class	tensor(0.8890)
1	practical easy use	app user friendly	tensor(0.4157)
2	app good video conferencing	good video conferencing	tensor(0.9131)
3	download zoom app	unable download zoom app	tensor(0.8942)
4	able download app	want download app	tensor(0.8572)

Figure 10 Second model score

Third model:

The third model is also a sentence-transformers. The name of the model is 'distilbert-base-nli-mean-tokens' which is pretrained model where it maps sentences and paragraphs to a 768-dimensional dense vector space. For the given input text and reason the similarity between the sentences outputs a even more better value than the second model as shown in Figure 11.



	clean_text	clean_reason	text_similarity_bert
0	amazing app online class	good app conducting online class	tensor(0.9131)
1	practical easy use	app user friendly	tensor(0.7228)
2	app good video conferencing	good video conferencing	tensor(0.9674)
3	download zoom app	unable download zoom app	tensor(0.5510)
4	able download app	want download app	tensor(0.8909)

Figure 11 Third model score

Summary

As the given training data is only 2061 rows and labelled all as 1's, from the training data finding reason for giving all 1's data that is a text similarity value. Based on the above three models used for training, the third model that is 'distilbert-base-nli-mean-tokens' is best in finding the similarity value. Now, finding the threshold value from the generated values by third model.

```
(array([  3.,  11.,   7.,  15.,  20.,  41.,  58.,  88.,  79.,  91., 106.,
        113., 141., 164., 173., 169., 171., 197., 214., 200.]),
 array([0.15262178, 0.1949907 , 0.23735963, 0.27972853, 0.32209748,
        0.3644664 , 0.40683532, 0.44920424, 0.49157315, 0.5339421 ,
        0.576311  , 0.61867994, 0.6610488 , 0.7034178 , 0.7457867 ,
        0.7881556 , 0.83052456, 0.87289345, 0.9152624 , 0.9576313 ,
        1.0000002 ], dtype=float32),
 <BarContainer object of 20 artists>)
```

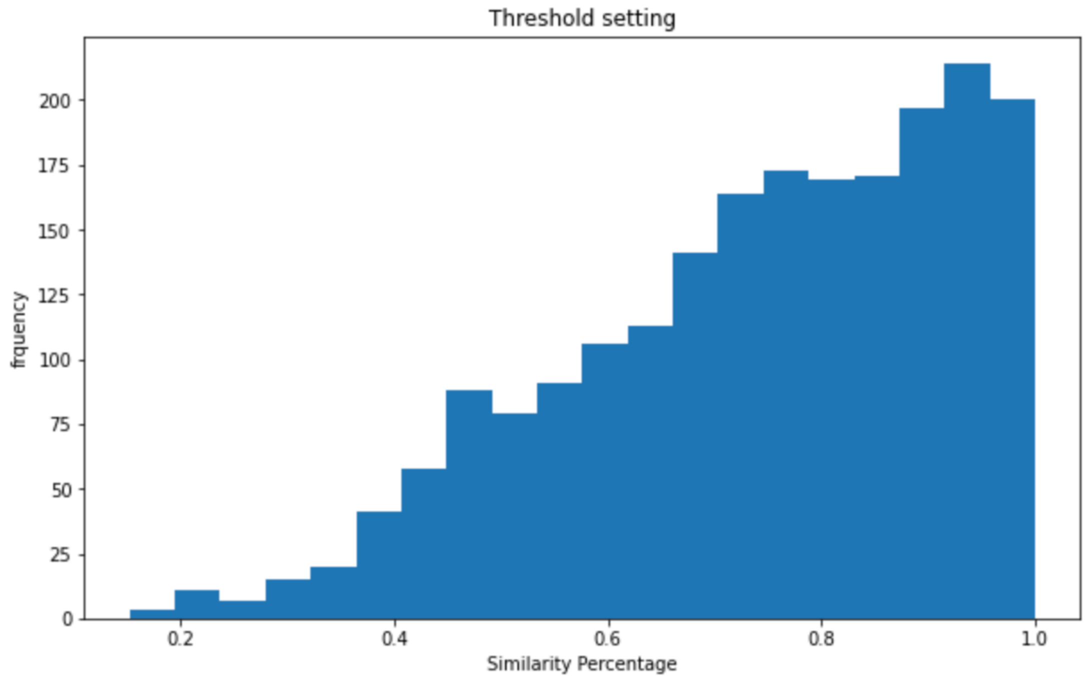
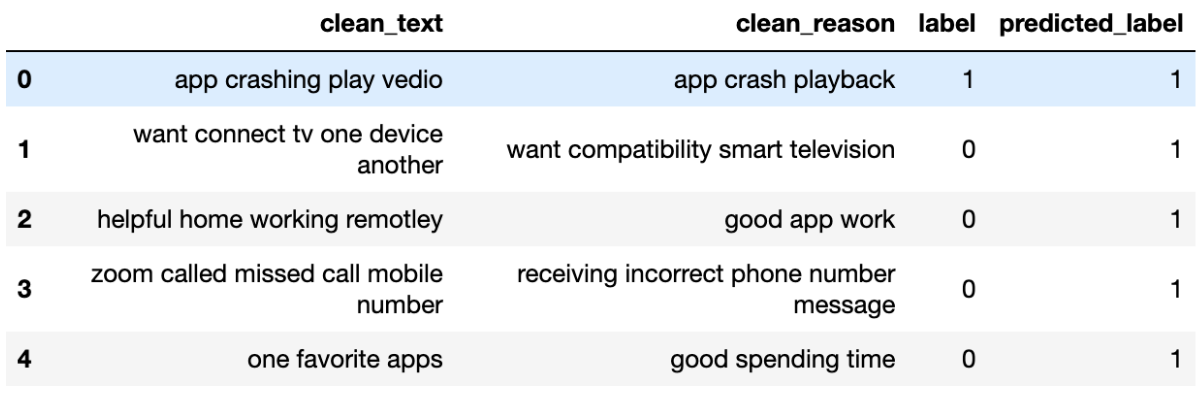


Figure 12 Threshold setting

After calculating all the similarity values for the training data it is found that out of 2061 instances, there are nearly 98% data are falling above 35% similarities and only 2% of data are between 15% to 35%.. Hence Keeping semantic similarity threshold 35%, flagging 0's if similarity value is less than 35%.

Evaluation metric

The evaluation metric used in this project is threshold value. Hence by setting threshold value to 35% similarity to flag 1 if greater else flag 0.
The evaluation dataset is implemented with this metric and the first five instances are shown in Figure 13.



	clean_text	clean_reason	label	predicted_label
0	app crashing play vedio	app crash playback	1	1
1	want connect tv one device another	want compatibility smart television	0	1
2	helpful home working remotley	good app work	0	1
3	zoom called missed call mobile number	receiving incorrect phone number message	0	1
4	one favorite apps	good spending time	0	1

Figure 13 Evaluation data

Error Analysis

Initially, the data is cleaned thoroughly and implemented model to get the similarity value which was acceptable value. Hence keeping this as a baseline model, comparing with different methods.

1) The model is trained without any preprocessing that is no stopwords removal and not performing lemmatization. As the generated result is compared to baseline model, there was an error rate 13.21%.
2) Next, the model is trained with the removal of stopwords, the generated result is again compared to baseline model. Now the error rate is significantly reduced to 1.3%. Hence removal of stopwords is important in text dataset.