Assignment - 1

The given dataset consists of 2358 rows with 6 columns. The ‘id’ column specifies the unique id of each row, but when the dataset is loaded through pandas it automatically assigns a unique id, so the existing id column is dropped. The ‘from’ and ‘to’ columns reveal the position of the aspect term in the sentence, but the position of the aspect term can change during preprocessing, so I dropped the ‘from’ and ‘to’ columns from the dataset.

	Sentence	Aspect Term	polarity
0	I charge it at night and skip taking the cord ...	cord	neutral
1	I charge it at night and skip taking the cord ...	battery life	positive
2	The tech guy then said the service center does...	service center	negative
3	The tech guy then said the service center does...	"sales" team	negative
4	The tech guy then said the service center does...	tech guy	neutral

Fig. 1 First five rows after dropping three columns.

The ‘polarity’ column has four different unique values namely positive, negative, neutral, and conflict.

```
polarity
positive    987
negative    866
neutral     460
conflict     45
Name: count, dtype: int64
```

Figure. 2 Different class values of the ‘polarity’

1. Lowercasing

In the first part of preprocessing the sentences were lowercased along with the aspect term as aspect term is crucial for determining the polarity of the sentence.

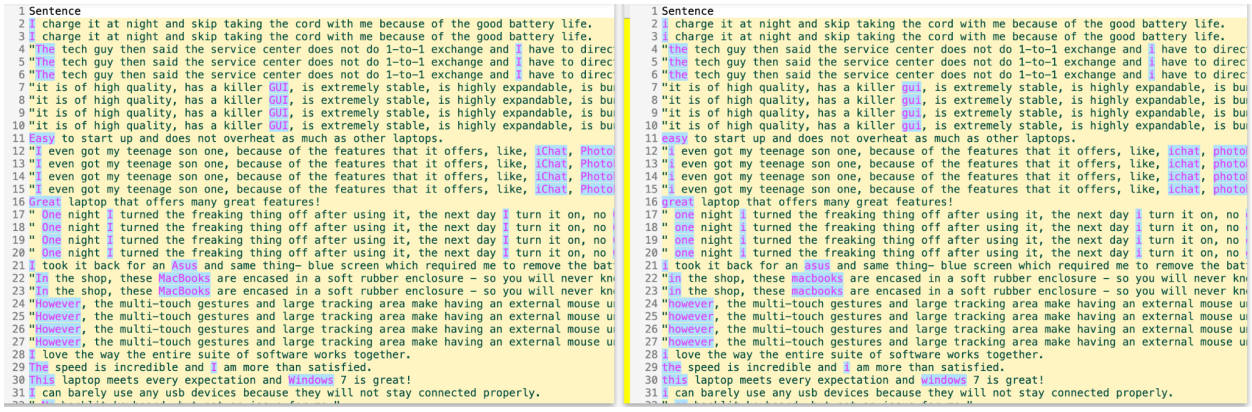

Figure 3. Before and After lowercasing sentences

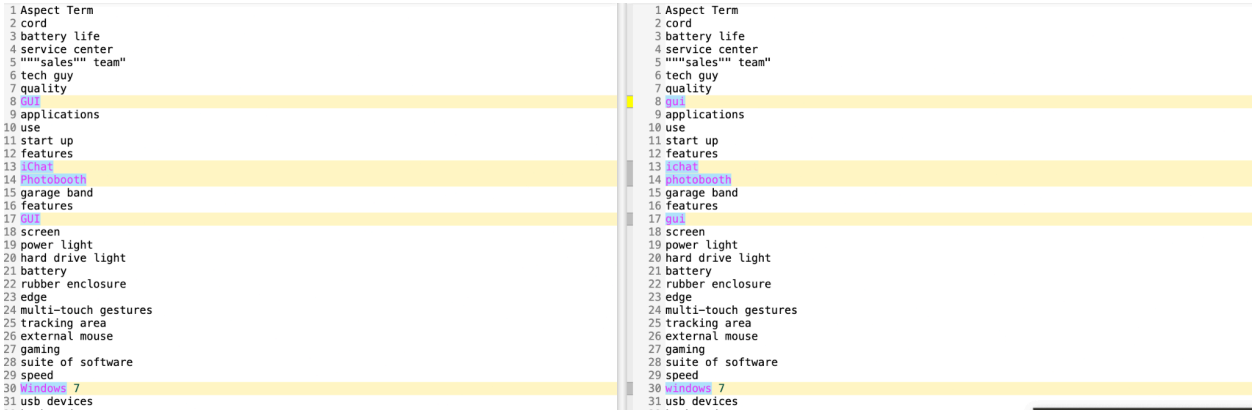

Figure 4. Before and After lowercasing aspect term

2. Removal of URLs, Email Address, and Hashtags

In the second step, sentences were inspected for URLs, Email Address, and Hashtags, but it lacks any of these in it.

3. Removal of white space

This is a common preprocessing task in NLP. In this step we removed extra space and leading/trailing spaces in the sentences and aspect term.

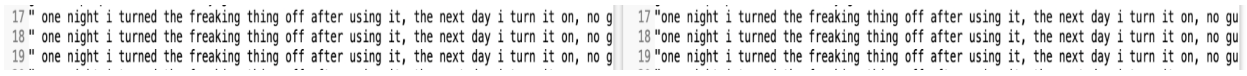

Figure 5. Before and After removal of white space in sentences

Figure 6. Before and After removal of white space in aspect term

4. Handling Contractions

Here we expand the contractions to their full forms. It helps to standardize the text and can improve the performance.

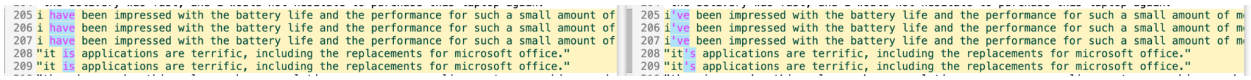

Figure 7. After and Before expanding the contractions.

5. Removal of special characters.

Special characters are removed to reduce noise, and prevent unnecessary tokens from inflating the vocabulary size.

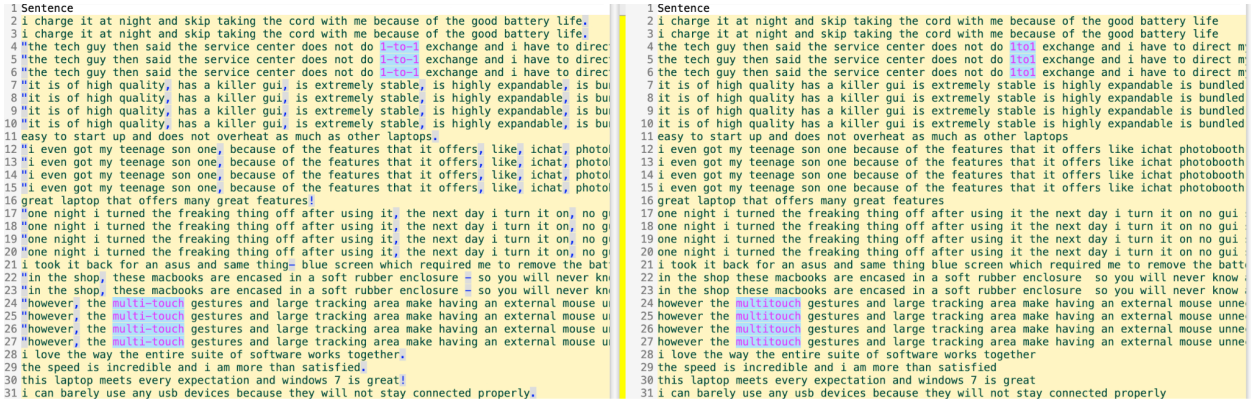

Figure 8. Before and After removal of special characters in sentences.

Figure 9. Before and After removal of special characters in aspect term.

6. Handling numbers.

Numbers are removed to reduce noise, improve generalization, and simplify the text data.

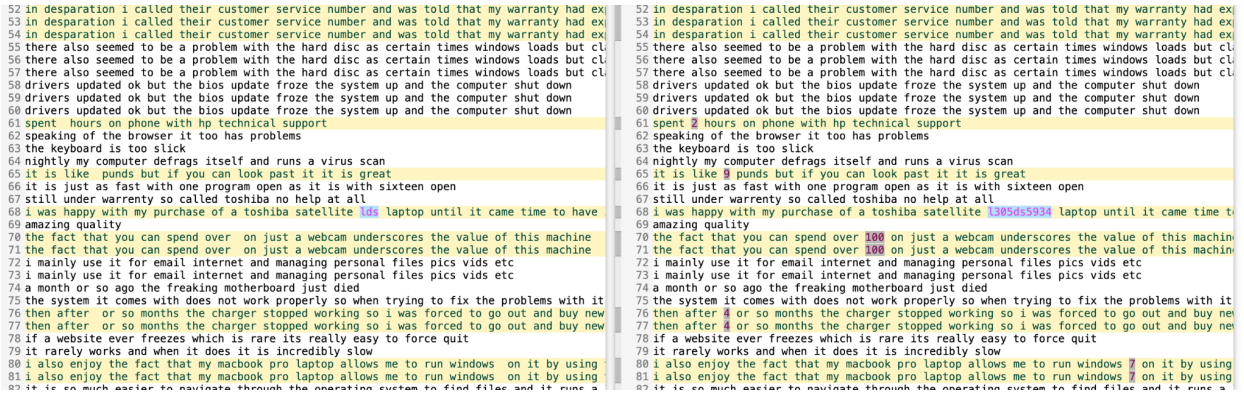

Figure 10. After and Before handling numbers in sentences.

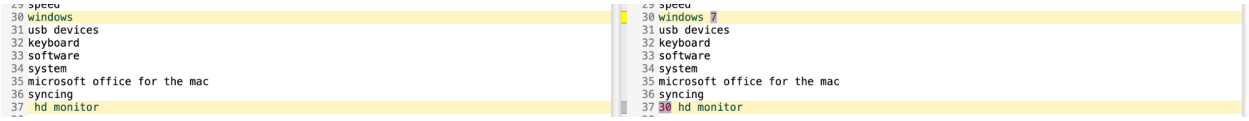

Figure 11. After and Before handling numbers in aspect term.

7. Tokenization

It involves breaking down text into smaller units, known as tokens.

```
'i charge it at night and skip taking the cord with me because of the good battery life'
```

Figure 12. Sentence Before Tokenization

```
['i',
 'charge',
 'it',
 'at',
 'night',
 'and',
 'skip',
 'taking',
 'the',
 'cord',
 'with',
 'me',
 'because',
 'of',
 'the',
 'good',
 'battery',
 'life']
```

Figure 13. Sentence After Tokenization

8. Stop Word removal

Stop words are common words such as "the," "is," "in," "and," and "to," which occur frequently in text but carry little meaning. This step helps to reduce noise and improves model efficiency.

```
['charge', 'night', 'skip', 'taking', 'cord', 'good', 'battery', 'life']
```

Figure 14. Result After Stop Word Removal

9. Lemmatization

The purpose of lemmatization is to group different forms of a word into a single base form so that they can be treated as the same word. This was chosen over stemming as it results in better accuracy.

```
['charge', 'night', 'skip', 'take', 'cord', 'good', 'battery', 'life']
```

Figure 14. Result After Lemmatization

```
0        charge night skip take cord good battery life
1        charge night skip take cord good battery life
2    tech guy say service center exchange direct co...
3    tech guy say service center exchange direct co...
4    tech guy say service center exchange direct co...
```

Figure 15. Dataset after preprocessing.

Bag of Words:

This is the first frequency based method that is used on the dataset. Before applying this method on the dataset, the aspect term column is combined with the sentence column as this approach is more straightforward and can work well when the aspect term naturally fits within the context of the sentence. Even though using separate embeddings and concatenating them gives more flexibility, it results in poor accuracy.

CountVectorizer from sklearn is used for Bag of Words and chi2 is used to select top 100 features. The result from the chi-square is used to split the dataset into train and test data. Before train-test split of the data the polarity column of the dataset is label encoded.
Three classifiers Random Forest, Decision Tree, SVC used on the embedding. The result can be seen in the table below.

Classifier	**Accuracy**
Random Forest Classifier	61.86%
SVC	62.50%
Decision Tree Classifier	61.86%

The following results are obtained during cross validation for this embedding.

Classifier	**Cross Validation Accuracy**
Random Forest Classifier	59.54%
SVC	60.43%
Decision Tree Classifier	58.53%

TF - IDF

This is the second frequency based method that is used on the dataset. Before applying this method on the dataset, the aspect term column is combined with the sentence column as this approach is more straightforward and can work well when the aspect term naturally fits within the context of the sentence. Even though using separate embeddings and concatenating them gives more flexibility, it results in poor accuracy.

TfidfVectorizer from sklearn is used for TF-IDF and chi2 is used to select top 100 features. The result from the chi-square is used to split the dataset into train and test data. Before train-test split of the data the polarity column of the dataset is label encoded.
Three classifiers Random Forest, Decision Tree, SVC used on the embedding. The result can be seen in the table below.

Classifier	**Accuracy**
Random Forest Classifier	60.80%
SVC	61.44%
Decision Tree Classifier	60.16%

The following results are obtained during cross validation for this embedding.

Classifier	**Cross Validation Accuracy**
Random Forest Classifier	56.83%
SVC	59.46%
Decision Tree Classifier	55.64%

For the frequency based approach Bag of Words dominates the TF-IDF method as the accuracy and cross validation accuracy for the former is higher than latter.

GloVe

For this embedding method glove.6B.300d is used from stanford. As a part of this embedding process both sentence and aspect column of the dataset is passed separately to the method to capture embedding for the words in sentence and aspect column. After this the result from the sentence and aspect embedding were combined to form a final embedding. The final embedding is used to obtain train and test data.
Three classifiers Random Forest, Decision Tree, SVC used on the embedding. The result can be seen in the table below.

Classifier	**Accuracy**
Random Forest Classifier	68.64%
SVC	56.99%
Decision Tree Classifier	54.66%

The following results are obtained during cross validation for this embedding.

Classifier	**Cross Validation Accuracy**
Random Forest Classifier	62.09%
SVC	60.05%
Decision Tree Classifier	48.14%

Random Forest's superior performance with GloVe embeddings can be attributed to its ability to handle high-dimensional and complex data. Unlike SVC, which may struggle with non-linearly separable data, and Decision Trees, which tend to overfit, Random Forest reduces overfitting by averaging multiple trees.

Sentence BERT

For this embedding all-distilroberta-v1 is used from sentence-transformers. At first embedding is applied separately on the sentence and aspect column of the dataset. The result is concatenated to form a combined embedding. The final embedding is used to obtain train and test data. Three classifiers Random Forest, Decision Tree, SVC used on the embedding. The result can be seen in the table below.

Classifier	Accuracy
Random Forest Classifier	69.49%
SVC	67.50%
Decision Tree Classifier	57.74%

The following results are obtained during cross validation for this embedding.

Classifier	Cross Validation Accuracy
Random Forest Classifier	64.25%
SVC	66.80%
Decision Tree Classifier	49.37%

The better performance of BERT over GloVe and Bag of Words can be attributed to the differences in how these models capture the meaning of words in context. BERT is a contextualized word embedding model, meaning it understands the meaning of words based on their surrounding context within a sentence. In contrast, GloVe is a static word embedding model, meaning that each word has a fixed vector representation regardless of context, which limits its ability to capture nuanced meanings in different sentences. Bag of Words is even more simplistic, treating words independently without considering their context or relationships, leading to a less expressive representation of the text.

References:

https://nlp.stanford.edu/projects/glove/
https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db
https://www.geeksforgeeks.org/python-lemmatization-with-nltk/
https://medium.com/analytics-vidhya/the-best-feature-selection-technique-for-text-classification-23199b4a4f8d
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html
https://huggingface.co/sentence-transformers/all-mpnet-base-v2