MINOR PROJECT - 2023

SENTIMENT ANALYSIS OF AMAZON PRODUCT REVIEWS

Under the guidance of Mentor:

Ms. Vibha Jain

Naveesha Srivastava
209301372
Section – F
B. Tech CSE - 3rd Year

CONTENTS

1. INTRODUCTION

What is Sentiment Analysis ?

Sentiment analysis is a natural language processing (NLP) method which is utilized to predict whether data is positive, negative, or neutral.

In this project, we particularly look at sentiment analysis of an Amazon product based on the customer reviews. Each review corresponds to a rating from 1 to 5. The goal of sentiment analysis is to determine the overall emotional tone of a review.

According to recent statistics, 77% of consumers read product reviews before buying on Amazon.

2. DATASET DESCRIPTION

Table below contains data about the customer's review details of an electronic product from Amazon. The major focus is on customer review and its corresponding rating for the sentiment analysis. Sample data is obtained from Kaggle.com

```
dataset.head()
```

	Unnamed: 0	reviewerName	overall	reviewText	reviewTime	day_diff	helpful_yes	helpful_no	total_vote	score_pos_neg_diff	score_average_rating
0	0	NaN	4.0	No issues .	2014-07-23	138	0	0	0	0	0.0
1	1	0mie	5.0	Purchased device , worked advertised . never m...	2013-10-25	409	0	0	0	0	0.0
2	2	1K3	4.0	works expected . sprung higher capacity . thin...	2012-12-23	715	0	0	0	0	0.0
3	3	1m2	5.0	think worked great.Had diff . bran 64gb card w...	2013-11-21	382	0	0	0	0	0.0
4	4	2&1/2Men	5.0	Bought Retail Packaging , arrived legit	2013-07-13	513	0	0	0	0	0.0

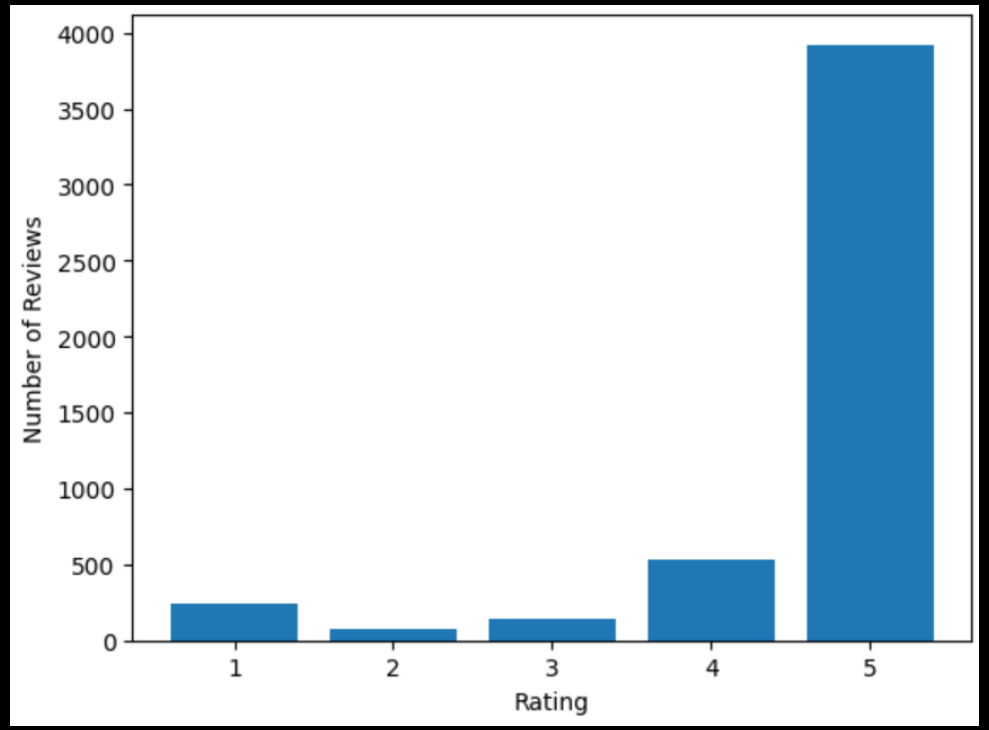
Dataset Information

Number of tuples: 4915
Data columns: 11
Link:
https://www.kaggle.com/datasets/tarkkaanko/amazon

```
dataset.info()
```

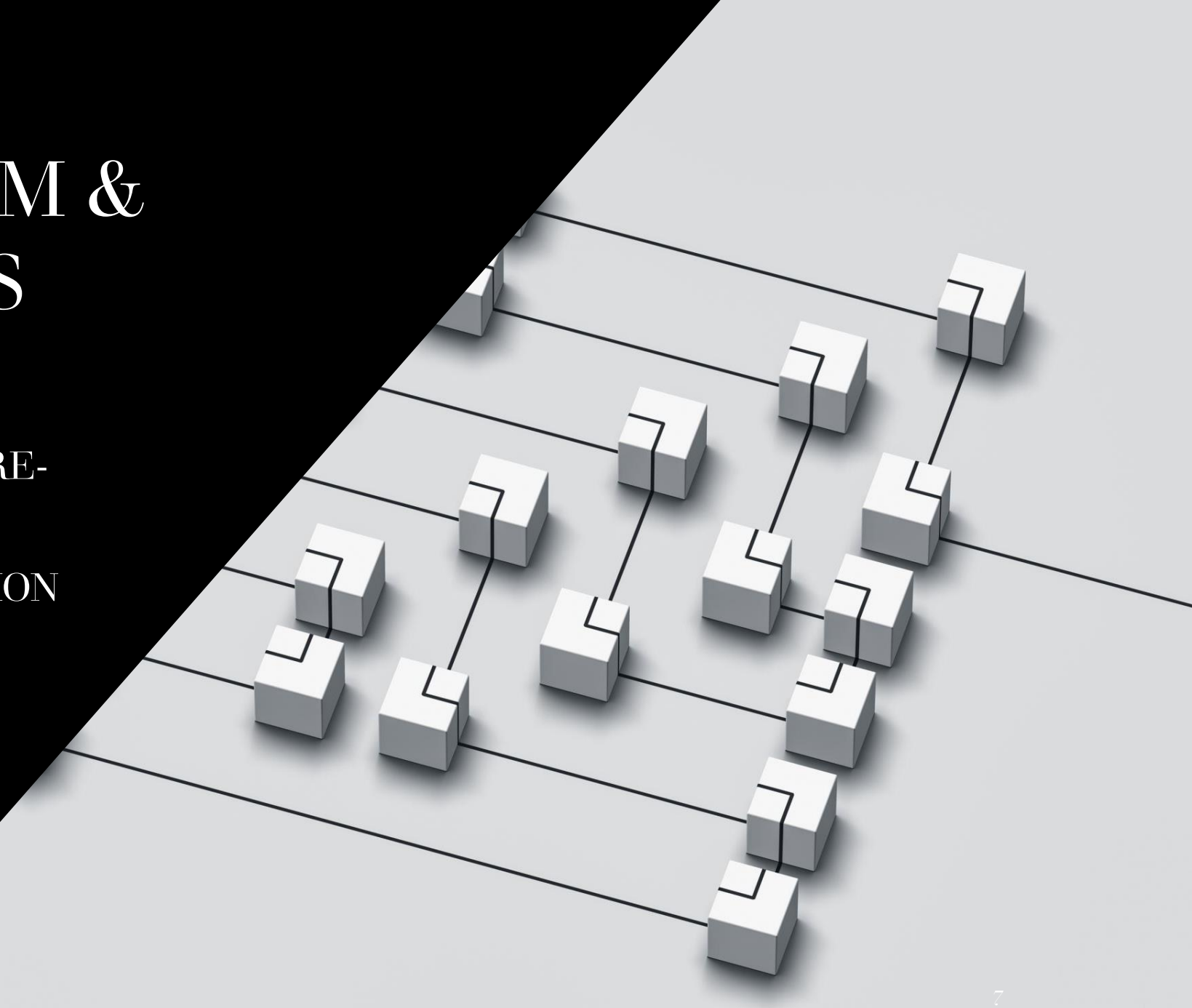
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4915 entries, 0 to 4914
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Unnamed: 0            4915 non-null   int64  
 1   reviewerName          4914 non-null   object 
 2   overall               4915 non-null   float64
 3   reviewText            4914 non-null   object 
 4   reviewTime            4915 non-null   object 
 5   day_diff              4915 non-null   int64  
 6   helpful_yes           4915 non-null   int64  
 7   helpful_no            4915 non-null   int64  
 8   total_vote            4915 non-null   int64  
 9   score_pos_neg_diff    4915 non-null   int64  
 10  score_average_rating  4915 non-null   float64
 11  wilson_lower_bound    4915 non-null   float64
dtypes: float64(3), int64(6), object(3)
memory usage: 460.9+ KB
```

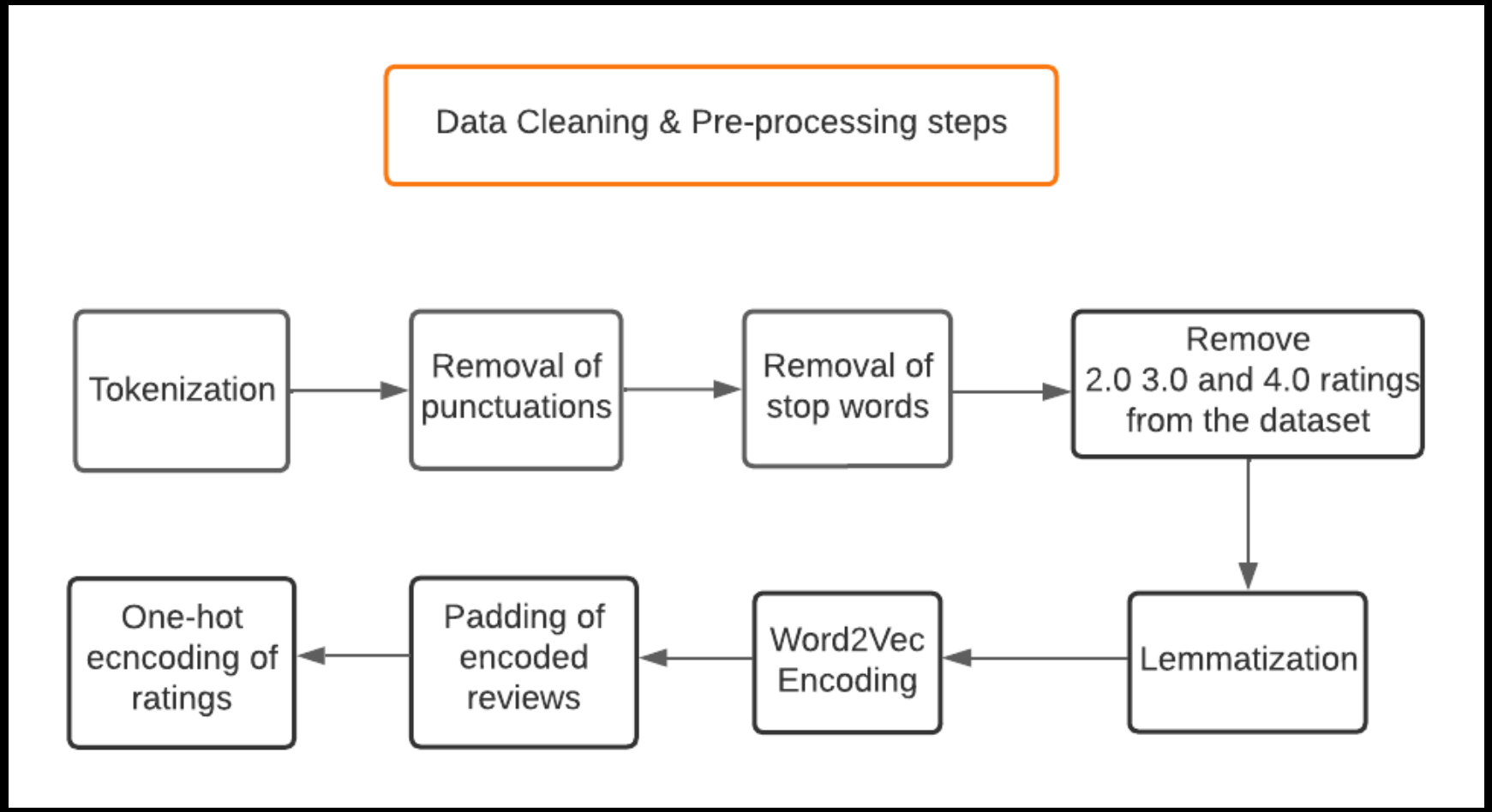
This graph depicts the frequency count of number of reviews per rating from 1 to 5 the dataset offers.



3. ALGORITHM & TECHNIQUES

1. DATA CLEANING AND PRE-PROCESSING
2. MODEL IMPLEMENTATION AND TESTING




Data Cleaning & Pre-processing steps
Tokenization
Removal of punctuations
Removal of stop words
Remove 2.0 3.0 and 4.0 ratings from the dataset
Lemmatization
Word2Vec Encoding
Padding of encoded reviews
One-hot ecncoding of ratings

1. Tokenization

```
reviewText.head()
```

```
0                                        [No, issues, .]
1    [Purchased, this, for, my, device, ,, it, work...
2    [it, works, as, expected, ., I, should, have, ...
3    [This, think, has, worked, out, great.Had, a, ...
4    [Bought, it, with, Retail, Packaging, ,, arriv...
Name: reviewText, dtype: object
```

2. Removal of punctuations

```
reviewText.head()
```

```
0                                             No issues.
1    Purchased this for my device, it worked as adv...
2    it works as expected. I should have sprung for...
3    This think has worked out great.Had a diff. br...
4    Bought it with Retail Packaging, arrived legit...
Name: reviewText, dtype: object
```

3. Removal of stop words

```
reviewText.head()
```

```
0                                           No issues .
1    Purchased device , worked advertised . never m...
2    works expected . sprung higher capacity . thin...
3    think worked great.Had diff . bran 64gb card w...
4    Bought Retail Packaging , arrived legit , oran...
Name: reviewText, dtype: object
```

4. Lemmatization

```
reviewText.head()
```

```
0                                                 no issue .
1    purchase device , work advertise . never much ...
2    work expect . sprung high capacity . think mak...
3    think work great.had diff . bran 64 gb card go...
4    buy retail packaging , arrive legit , orange e...
Name: reviewText, dtype: object
```

5. Word2Vec Encoding

```
embeddings = reviewText[0]
for embedding in embeddings:
    print(embedding)
```

```
tf.Tensor(
[[-2.59923842e-02 -1.00169824e-02  3.52738537e-02 -3.18673439e-03
   2.78390143e-02  2.91083865e-02  2.30574328e-02  3.13552842e-02
  -1.47768389e-02  4.43636701e-02 -6.36036741e-03  2.22535096e-02
   3.64602096e-02 -5.08949831e-02  6.43709004e-02  7.32610375e-02
   4.10345979e-02 -9.39234793e-02  1.07663488e-02 -7.12433681e-02
```

6. One-hot encoding : Technique used to convert categorical data into numerical data.

In this case, the ratings are categorical, either '5.0' or '1.0'. The one-hot encoding will represent these two categories as binary vectors, [1, 0]: for 5 or [0, 1]: for 1

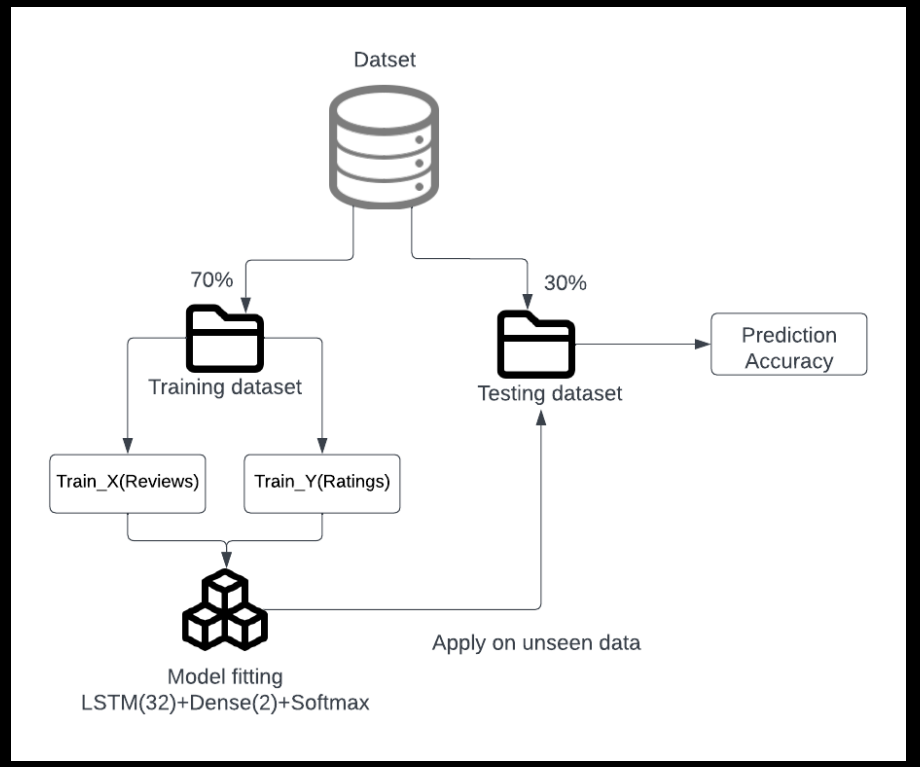
MODEL IMPLEMENTATION

Train-test dataset ratio- 70:30

Deep Learning Model:

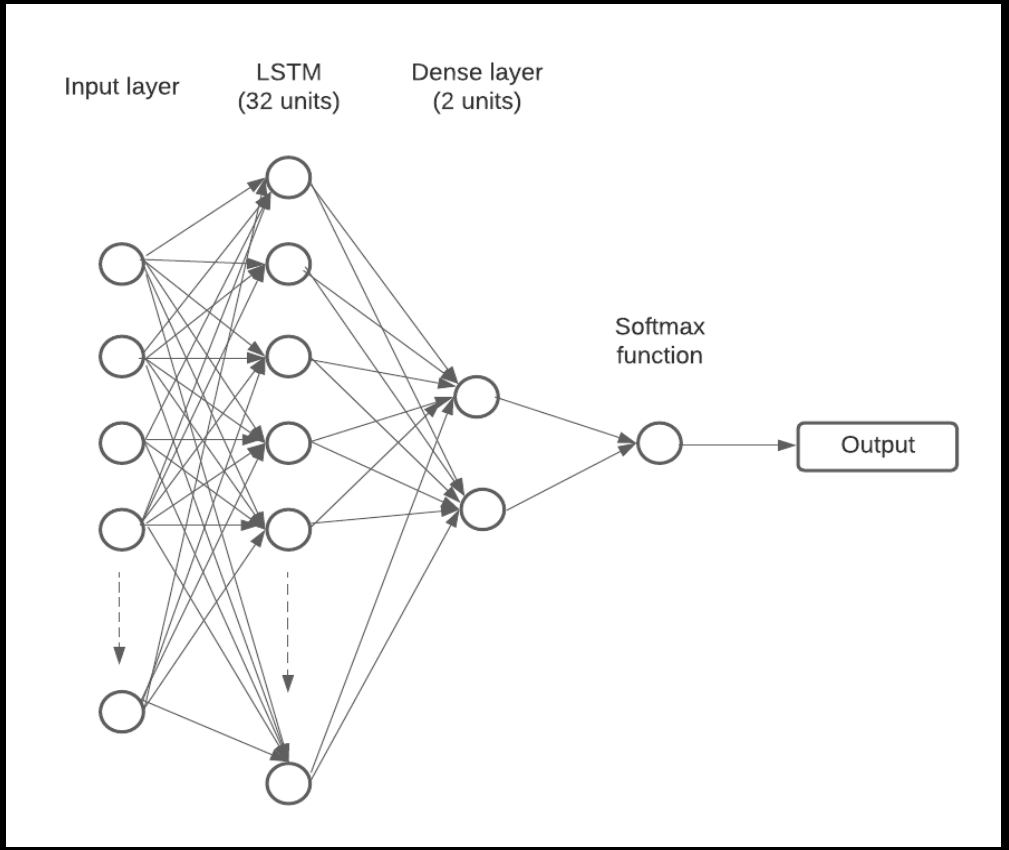
RNN - LSTM

Recurrent Neural Network(RNN) can remember the previous computation of information and can reuse it by applying it to the next element in the sequence of inputs. A special type of RNN is long short-term memory (LSTM), which is capable of using long memory as the input of activation functions in the hidden layer.



The LSTM layer takes as input a sequence of word embeddings, and produces as output a hidden state vector that represents the information learned from the input sequence.

A sequential model object is created and adds an LSTM layer with 32 units. Then, a dense layer with 2 units and a softmax activation function(it converts a vector of value to a probability distribution.)



4. RESULT ANALYSIS

Parameters and results:

1. Total data	4914
2. Split ratio	70:30 train-test ratio
3. Training data	3439
4. Testing data	1474
5. Model used	RNN- LSTM
6. LSTM layer units	32 units
7. Dense layer units	2 units
8. Activation function	Softmax Activation Function
9. Number of Epochs	25
10. Test Loss	0.1114
11. Test Accuracy	0.9653225541114807

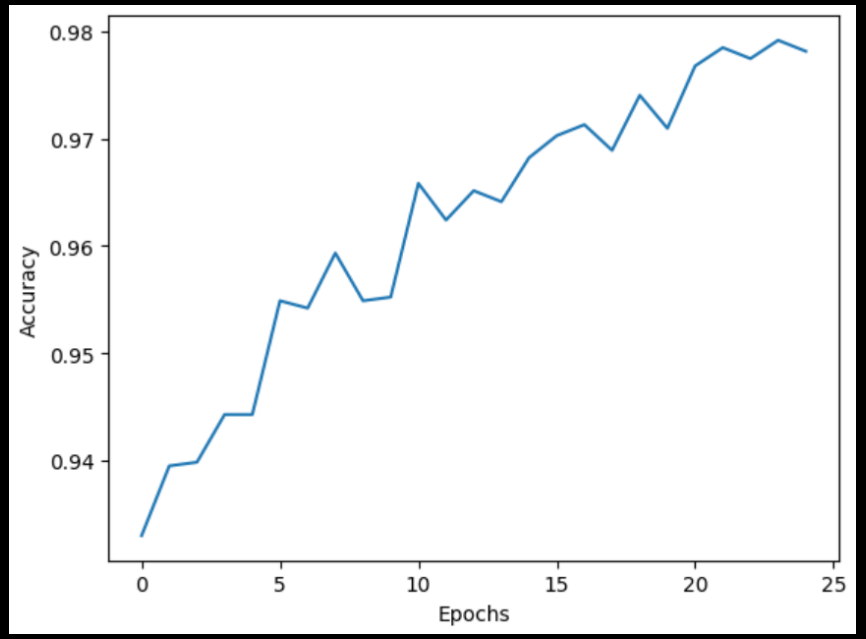


Figure 1 Accuracy score plotted against each epoch. The model reached its highest peak with the accuracy score of 0.9789

Other results:

- Precision: [0.97478992 0.74]
- Recall: [0.98891731 0.55223881]
- F1 Score: [0.98180279 0.63247863]

In the results, the precision, recall, and F1 score values are given in the form of 1-D arrays, where the first element corresponds to the "positive" class (in this case, a rating of 5) and the second element corresponds to the "negative" class (in this case, a rating of 1).

Precision: ratio of true positive predictions to the total number of positive predictions.

Recall: ratio of true positive predictions to the total number of actual positive instances.

F1 score: harmonic mean of precision and recall.

5. CONCLUSION AND FUTURE SCOPE

The proposed system using LSTM, succeeded in obtaining an accuracy of 96.53 %.

Future Scope:

- There is a scope to improve the dataset quality by removing the biasness in the data. Approximately 80% of the reviews are 5-star rated. This will give more accurate results on how efficient the model is.
- Analysis of emoticons is another challenge to be handled, because it has been observed that a lot of reviews contains emoticons, which directly imply the true sentiment of the text. Studying relationship between sequence of emoticons is also a challenging yet necessary task for better analysis.
- Multilingual text analysis is yet another challenge in Sentiment Analysis.

THANK YOU

MINOR PROJECT PRESENTATION

CS3270

JAN-MAY 2023