DEMOKRITOS
NATIONAL CENTER FOR SCIENTIFIC RESEARCH

UNIVERSITY of the PELOPONNESE

Deep Learning

Text-Based Sentiment Analysis

based on IMDb movie reviews

Sykallou Sofia - 2022202204009

Pouli Stavroula - 2022202204008

What is text-based sentiment analysis?

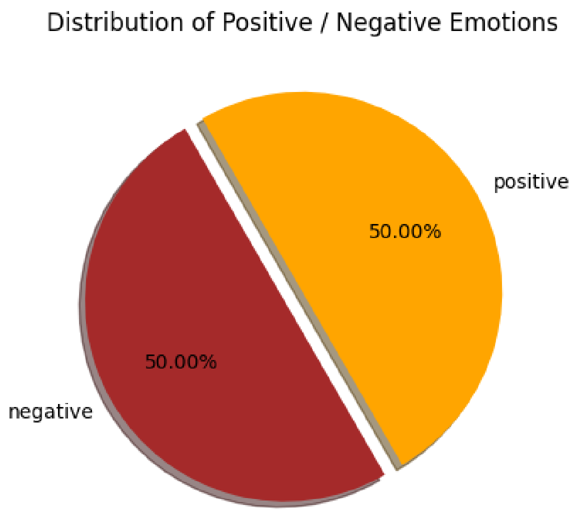
Branch of natural language processing (NLP)

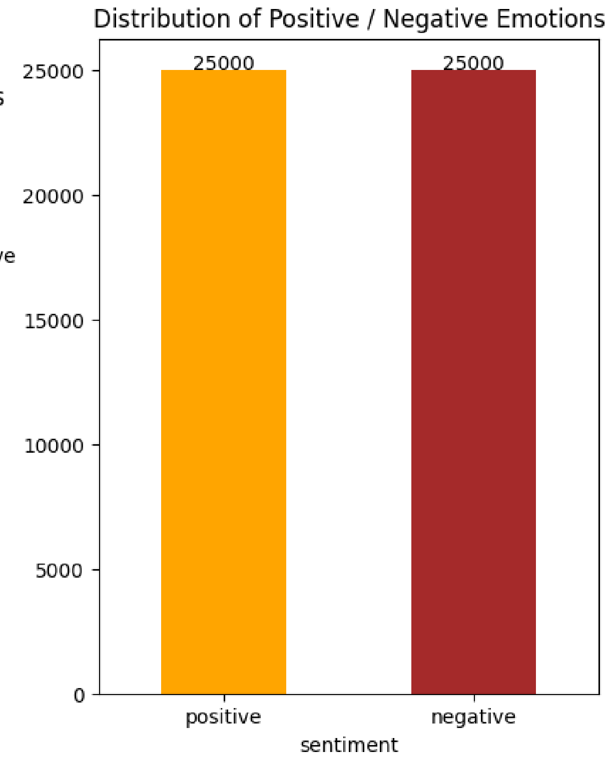
Identifies the emotional tone behind a body of text

- ✓ positive sentiment
- ✓ negative sentiment
- ✓ (or neutral)

The IMDb Dataset

- Data source: [Kaggle]
- Contains 50K Movie Reviews
  - 25.000 positive labeled reviews
  - 25.000 negative labeled reviews





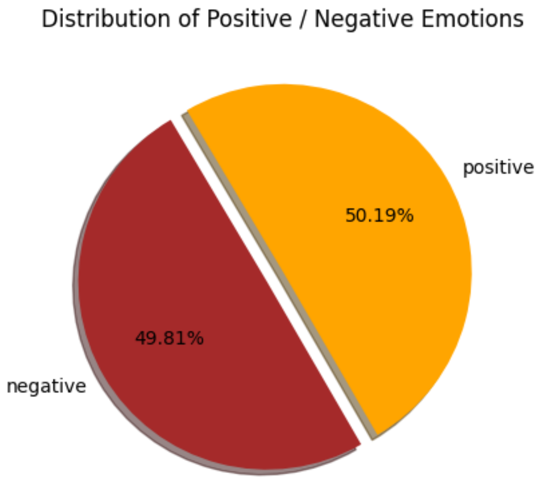
Features

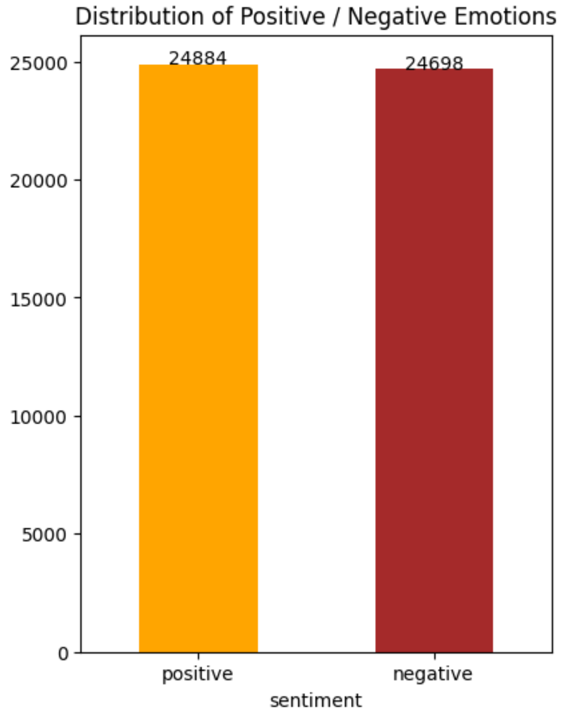
Review	The text content of the movie reviews.
Sentiment	Indicates whether the review is positive or negative.

Data cleaning and preprocessing

Duplicate Values

- There are 418 duplicate values in our dataset.
- After deleting them, the dataset consisted of 49.582 instances.





Data cleaning and preprocessing
Label transformation

Sentiment

- positive
- negative

Sentiment

- 1
- 0

Data cleaning and preprocessing
Label transformation

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

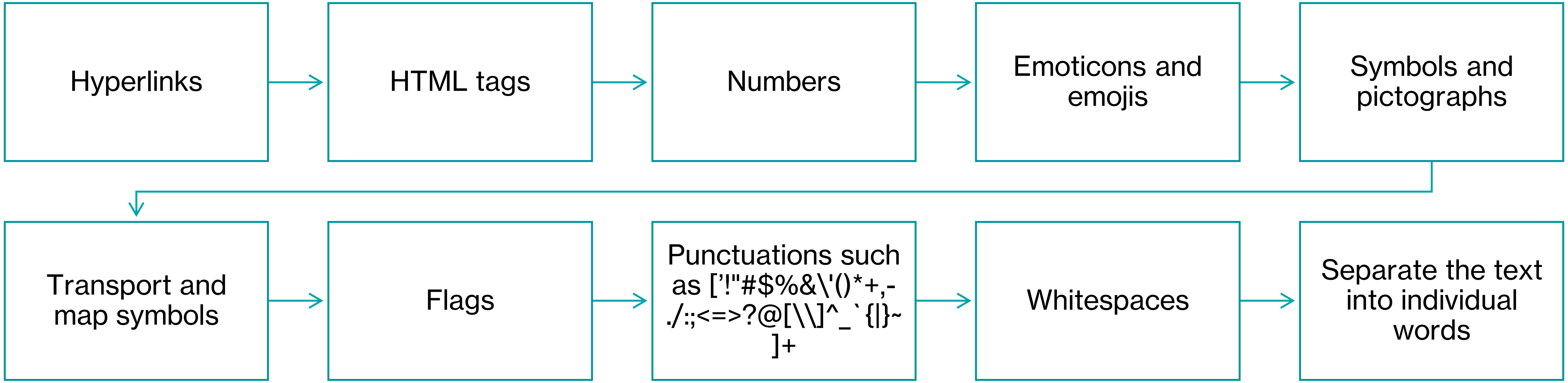
	review	sentiment
0	One of the other reviewers has mentioned that ...	1
1	A wonderful little production. The...	1
2	I thought this was a wonderful way to spend ti...	1
3	Basically there's a family where a little boy ...	0
4	Petter Mattei's "Love in the Time of Money" is...	1

Data cleaning and preprocessing
Stemming

- Reduces words to their base or root form, known as a stem.
- Removes suffixes or affixes from words to normalize them and group similar words together.
- Stemming algorithms
  - **Porter**
  - Snowball
  - Lancaster

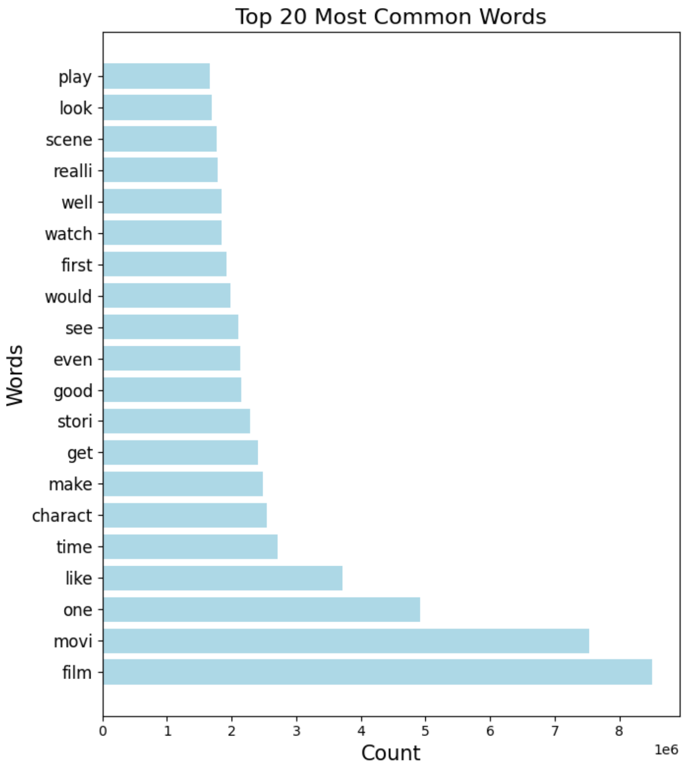
Data cleaning and preprocessing
Transformations and removals

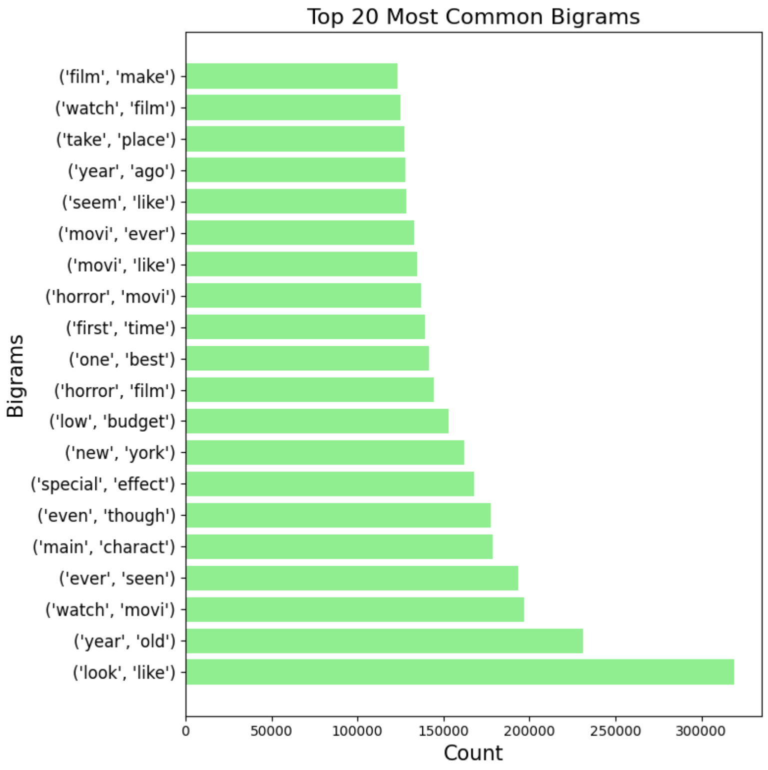
Before performing stemming, we defined a word mapping dictionary:
informal contractions or abbreviations are transformed to their corresponding expanded form.



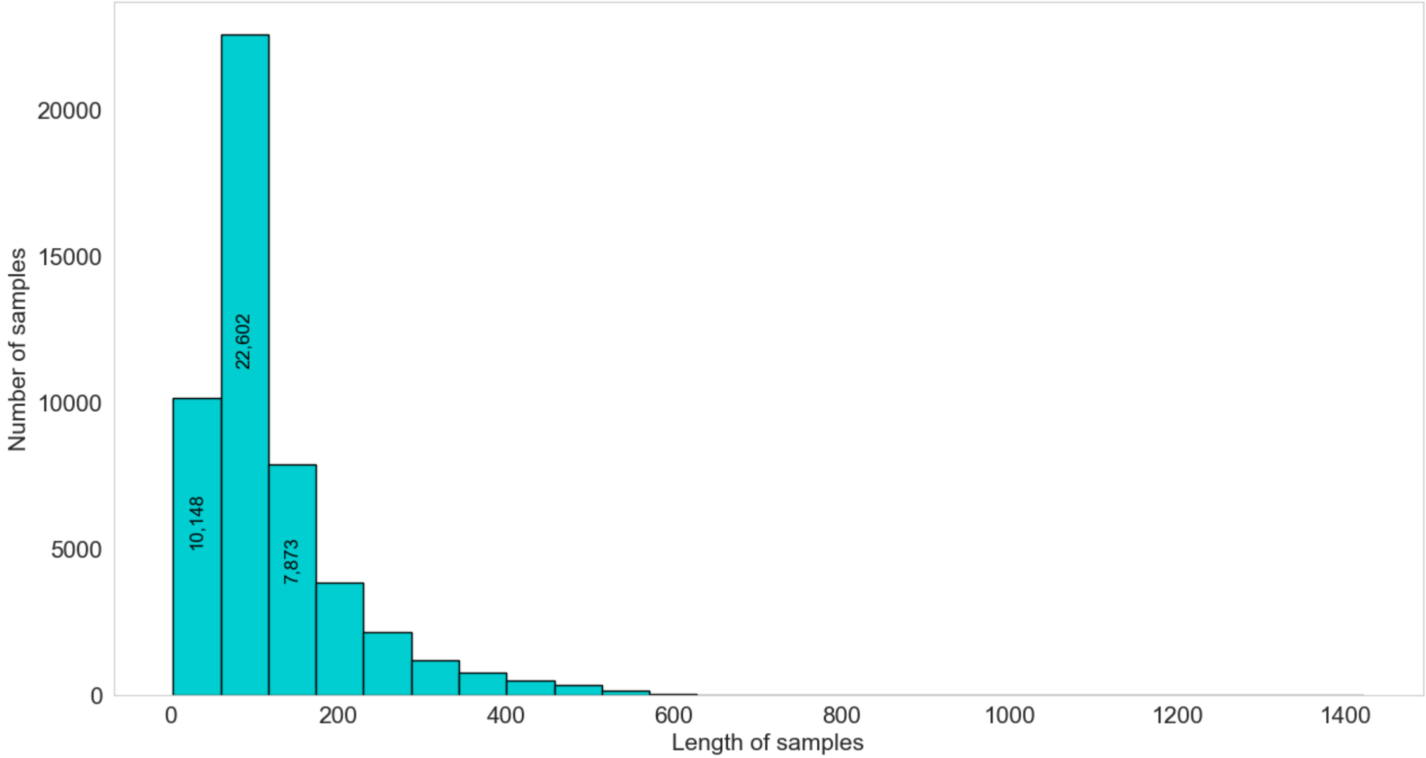
Exploratory analysis

Top 20 most common words and bigrams



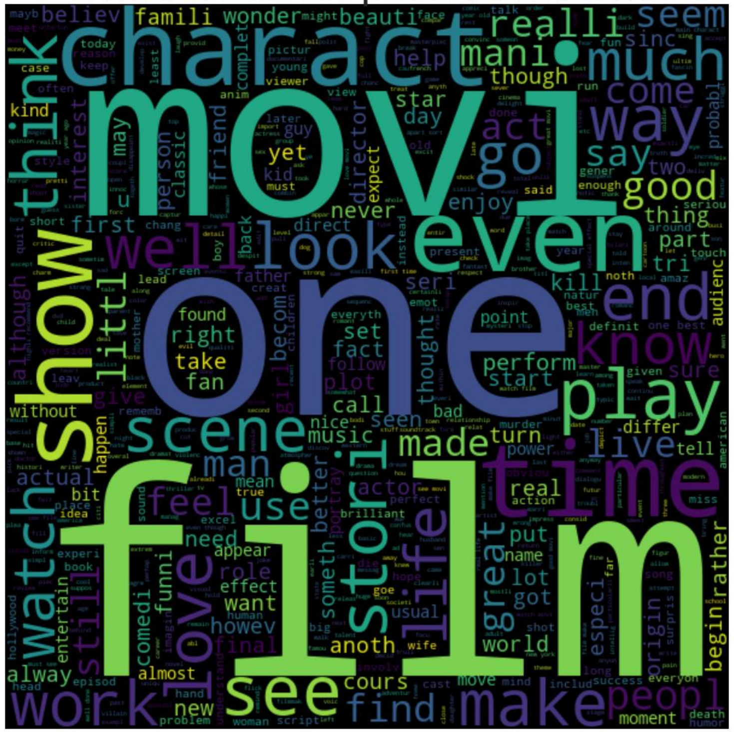


Exploratory analysis
Distribution of the lengths of the reviews

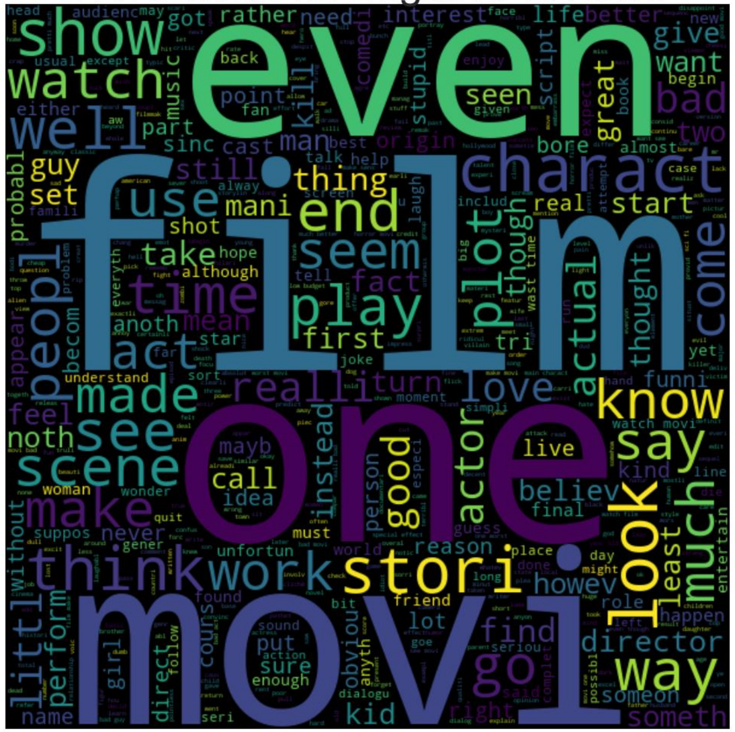


Exploratory analysis
Word clouds

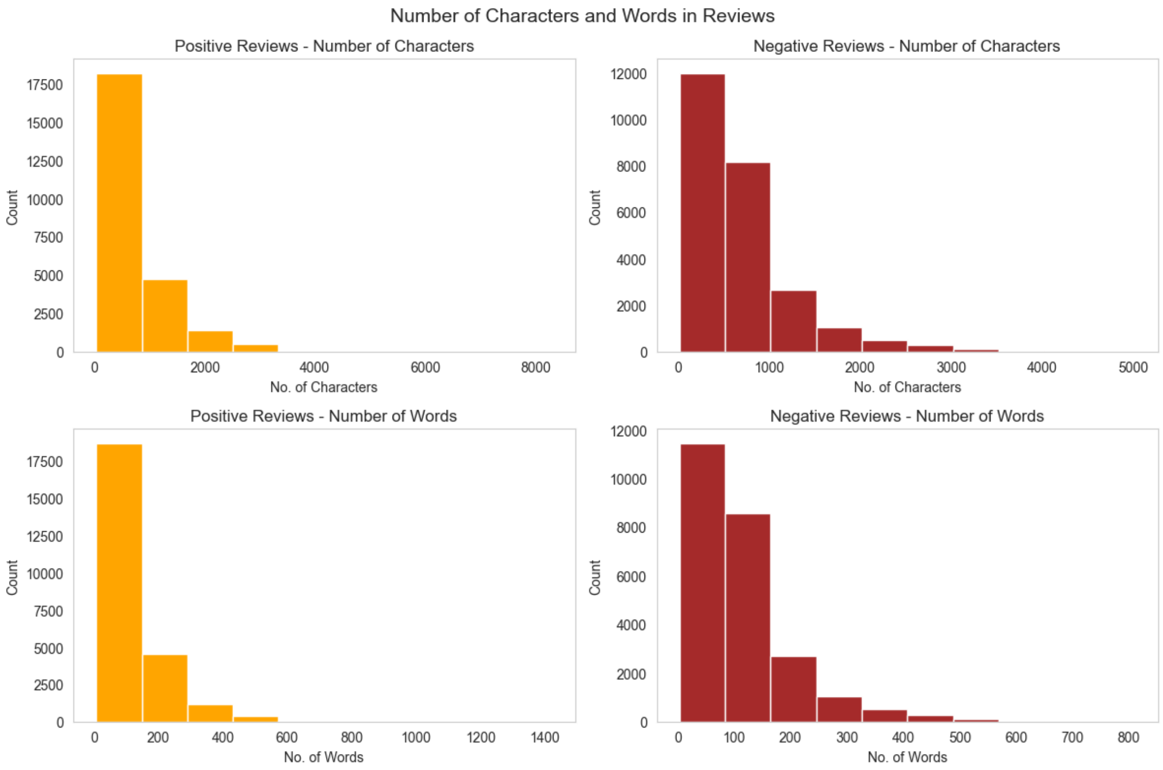
Word cloud for positive reviews



Word cloud for negative reviews

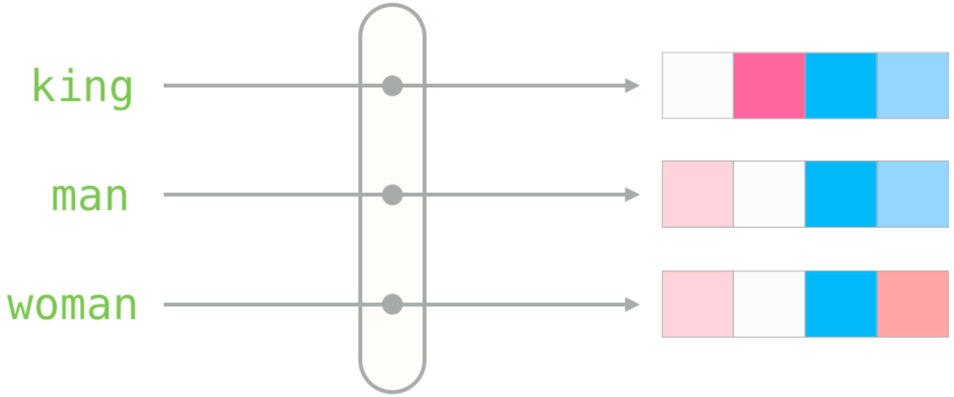


Exploratory analysis
Characters and words in the reviews



Represent words as dense vectors
Word2Vec model

- To discover the semantic relationships between words
- To analyze the context of each word and generate word embeddings
- The model will learn the relationships between words based on their positions
- After executing the above process, the vocabulary's length is equal to 24221 words.



Tokenization and padding

Tokenization	Padding
Breaking down a sequence of text into smaller units called tokens	Ensure that all sequences have the same length

Embedding matrix

Store the embedding vectors in a matrix.
If the model contains the word, then retrieve the embedding vector

```
Embedding Matrix Shape: (35000, 100)

array([[ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.25320712, -0.32531893, -2.18189287, ...,  0.63631606,
         0.05103426, -0.31895107],
       ...,
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ],
       [ 0.        ,  0.        ,  0.        , ...,  0.        ,
         0.        ,  0.        ]])
```

Model selection

Split the dataset

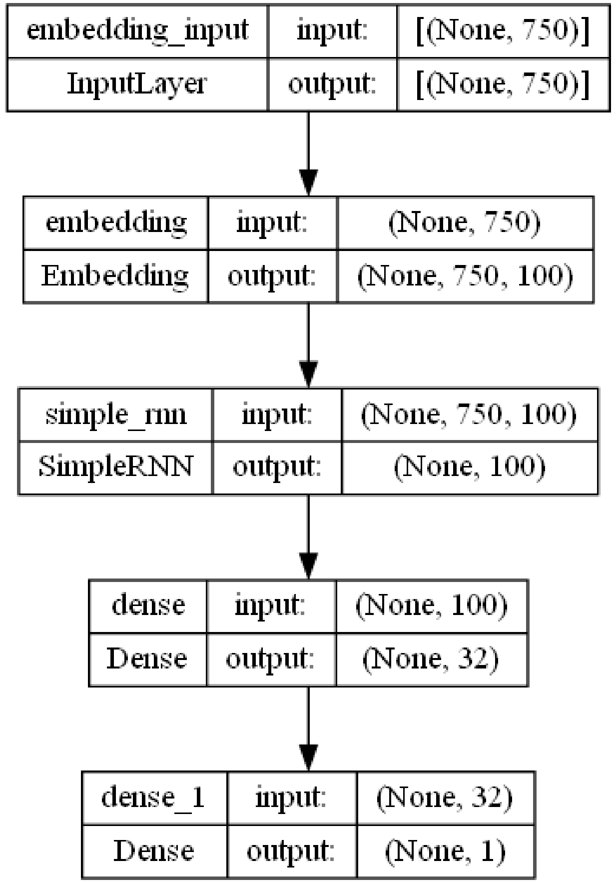
Training set - 80% of the data (39.665 records)

Test set - 20% of the data (9.917 records)

Model selection
RNN (Recurrent Neural Network)

- RNN Loss: 35%
- RNN Accuracy: 84%

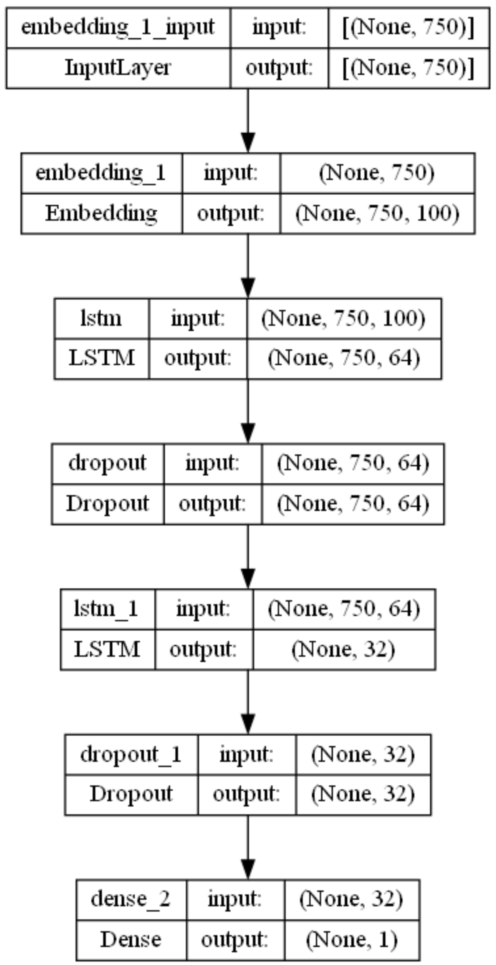
```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 750, 100)          3500000

 simple_rnn (SimpleRNN)      (None, 100)               20100

 dense (Dense)               (None, 32)                3232

 dense_1 (Dense)             (None, 1)                 33

=================================================================
Total params: 3,523,365
Trainable params: 23,365
Non-trainable params: 3,500,000
_________________________________________________________________
```



Model selection
LSTM (Long Short-Term memory)

- LSTM Loss: 30%
- LSTM Accuracy: 87%

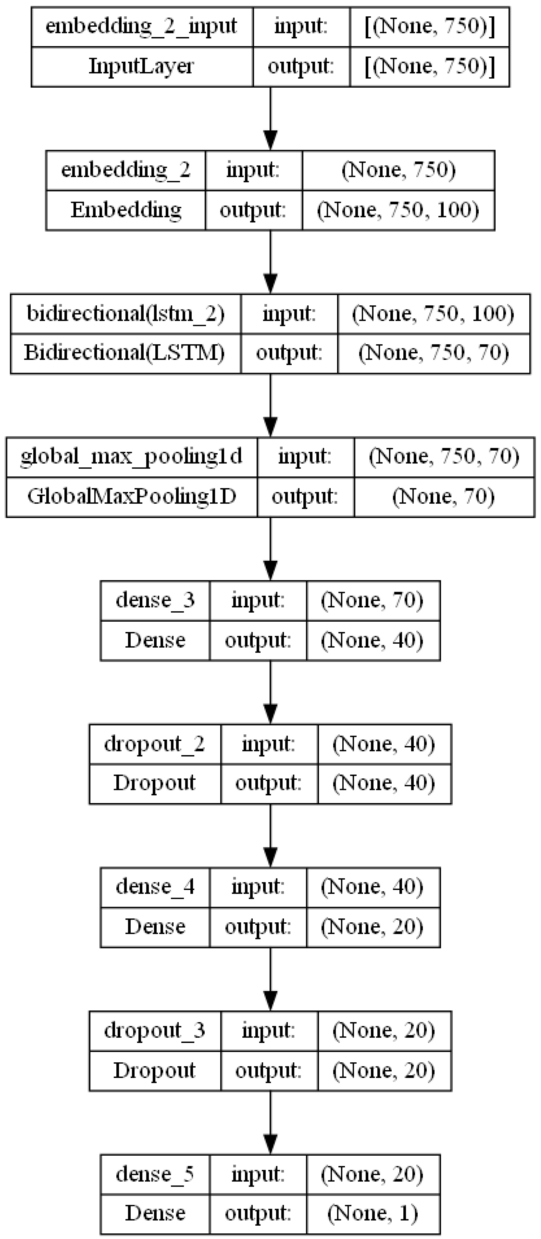
```
Model: "Sentiment_Model_LSTM"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 750, 100)          3500000

 lstm (LSTM)                 (None, 750, 64)           42240

 dropout (Dropout)           (None, 750, 64)           0

 lstm_1 (LSTM)               (None, 32)                12416

 dropout_1 (Dropout)         (None, 32)                0

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 3,554,689
Trainable params: 54,689
Non-trainable params: 3,500,000
_________________________________________________________________
```



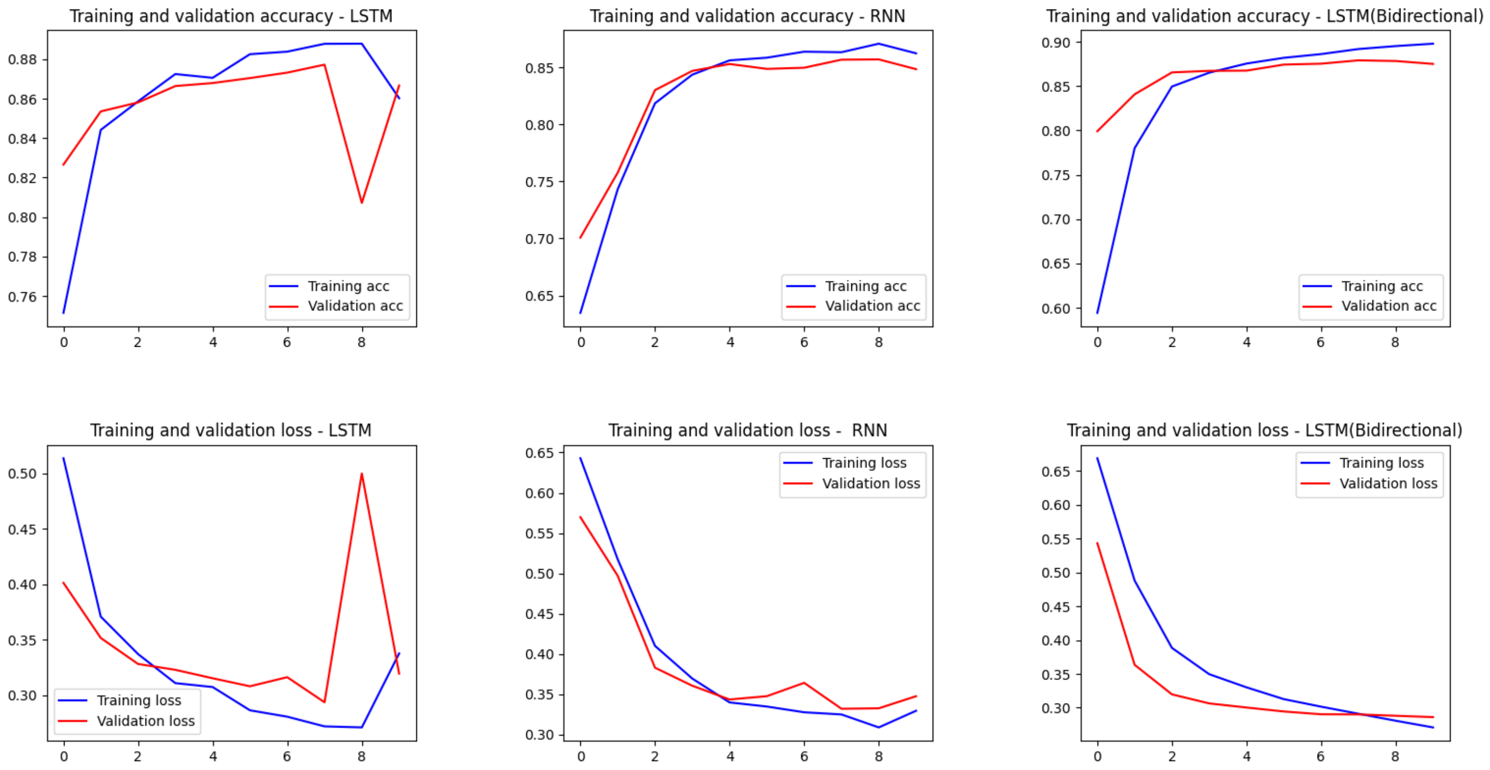
Model selection
BiLSTM (Bidirectional LSTM)

- BiLSTM Loss: 28%
- BiLSTM Accuracy: 88%

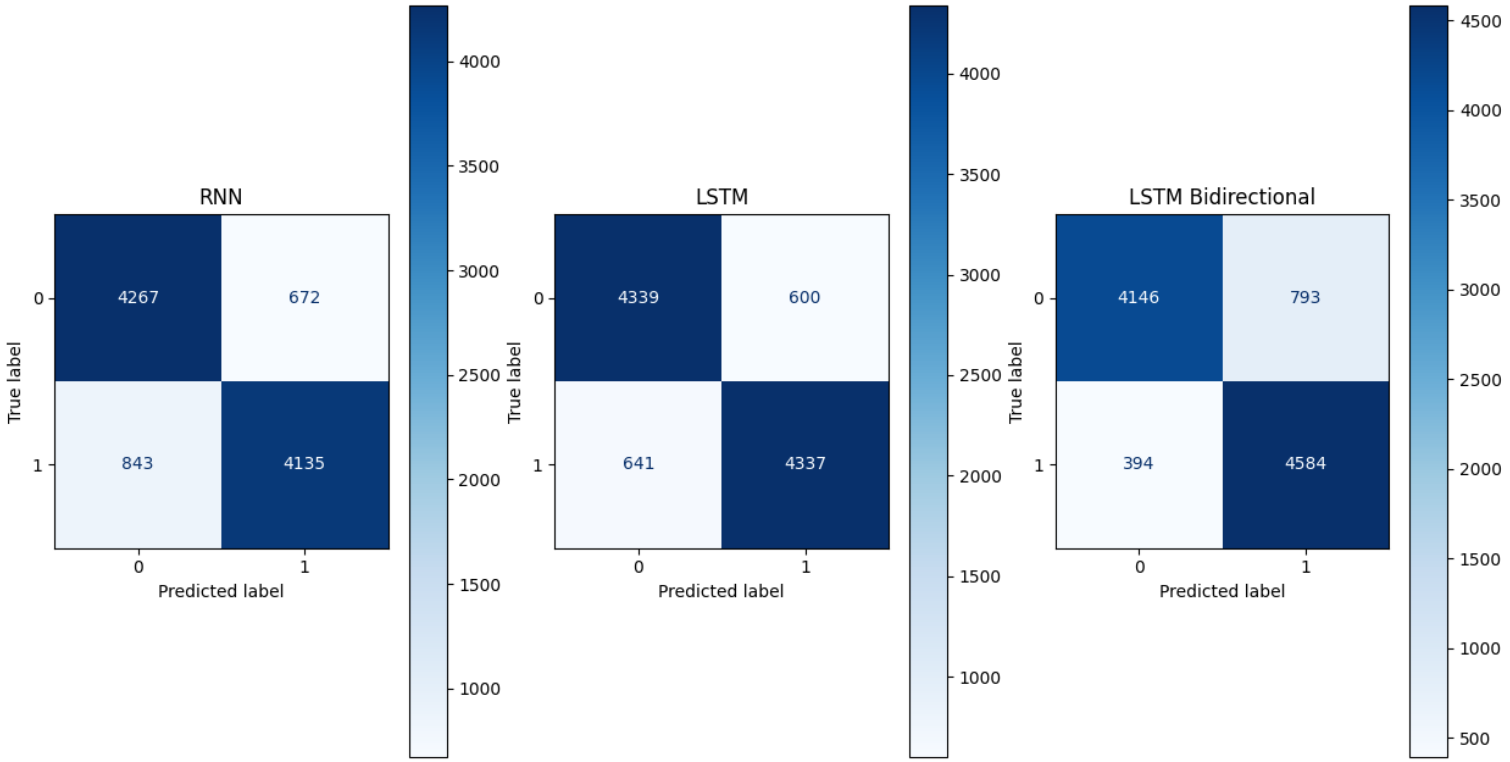
```
Model: "Sentiment_Model_LSTM_Bidirectional"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 750, 100)          3500000

 bidirectional (Bidirectiona  (None, 750, 70)          38080
 l)

 global_max_pooling1d (Globa  (None, 70)               0
 lMaxPooling1D)

 dense_3 (Dense)             (None, 40)                2840

 dropout_2 (Dropout)         (None, 40)                0

 dense_4 (Dense)             (None, 20)                820

 dropout_3 (Dropout)         (None, 20)                0

 dense_5 (Dense)             (None, 1)                 21

=================================================================
Total params: 3,541,761
Trainable params: 41,761
Non-trainable params: 3,500,000
_________________________________________________________________
```



Model evaluation



Confusion matrix



Classification report

All the evaluation metrics for our (classification) problem.

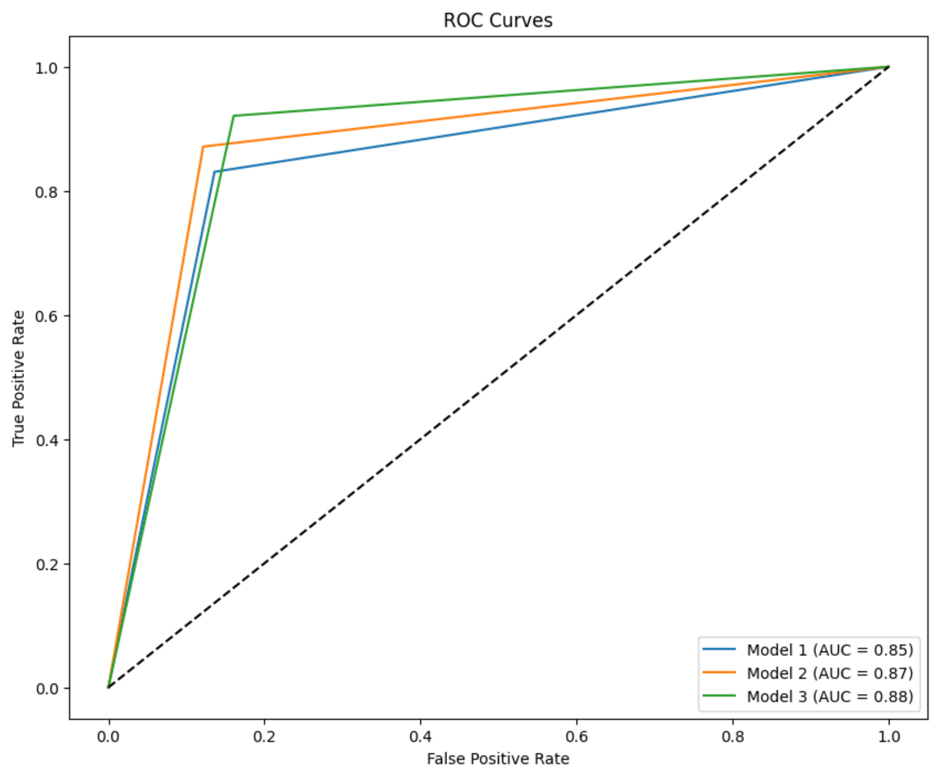
- uses the true labels (y_test) and the predicted labels (y_pred)
- creates a report with the
  - precision
  - recall
  - f1-score
  - support

```
              precision    recall  f1-score   support

           0       0.84      0.86      0.85      4939
           1       0.86      0.83      0.85      4978

    accuracy                           0.85      9917
   macro avg       0.85      0.85      0.85      9917
weighted avg       0.85      0.85      0.85      9917
              precision    recall  f1-score   support

           0       0.87      0.88      0.87      4939
           1       0.88      0.87      0.87      4978

    accuracy                           0.87      9917
   macro avg       0.87      0.87      0.87      9917
weighted avg       0.87      0.87      0.87      9917
              precision    recall  f1-score   support

           0       0.91      0.84      0.87      4939
           1       0.85      0.92      0.89      4978

    accuracy                           0.88      9917
   macro avg       0.88      0.88      0.88      9917
weighted avg       0.88      0.88      0.88      9917
```

ROC AUC Curves

ROC curve: represents the performance of a binary classification model.

AUC: represents the overall performance of the classification model.



Winner model
BiLSTM

The BiLSTM model process the input sequence both forward and backwards.
As a result, the model can better interpret the overall sentiment of the text by capturing contextual information from both past and future situations.

Loss
28%

Accuracy
88%

Thank you

any questions?