**This report will contain three parts: improvement overview, explanation and step through code, and NewsAPI model test**

**Improvement Overview:**

1. Due to efficiency, I need to use Jupyter notebook instead of python because when I do the hyperparameter tuning or modify the code, it is efficient that I do not need to run tokenize (stemming and lemmatization) and encode the dataset again and again. (Simply just rerun the cells below the preprocess cell)
   1. It took around 15 mins to tokenize top 10000 words.



* 1. It took around 25/3/3mins to encode the train/validation/test set.



1. Remove stopword and stemming tokenizer to improve accuracy, keep max\_features = 10000

A picture containing company name

Description automatically generated

1. Add tqdm to check preprocessing progress (progress bar), especially for encoding process:
2. Remove unused library and code (sent\_tokenize, word\_tokenize, FreqDist, and etc.)
3. Use str.isdigit() method to check if each char is a digit
4. Define evaluation function for four models (ANN, CNN, RNN, LSTM) instead of duplicate code

Graphical user interface, text

Description automatically generated

1. Update wordcloud code to make simple visualization

Positive wordcloud

A picture containing text

Description automatically generated

Negative wordcloud

A picture containing logo

Description automatically generated

Some top words for positive: great, love, good, one, show, time

Some top words for negative: even, nothing, bad, still, actually, seem

**Explanation and Step Through Code:**

**Step 1: Import library**

**We need tensorflow to train data, nltk to tokenize data, wordcloud to show data, sklearn to report model performance, CountVectorizer to vectorize text, pandas for dataframe operation.**

**Step 2: Import Dataset (WordCloud)**

**There are three dataset for training, validation, and testing. 40000 samples in training, 5000 in validation and 5000 in testing. Each dataset have two columns: text and label. The text contain movie review in text format and the label column contain pre-labeled for the movie review. The ‘0’ label means the review is negative, the ‘1’ means the review is positive. The wordcloud above have already summarize the top words contribute positive/negative reviews.**

**Step 3: Tokenize and Vectorize for text**

**The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Here, we remove all columns that contain numbers and remove any column with name of 3 or smaller like ‘it’, ‘of’, ‘pre’**

**Step 4: Encode text**

**We encode text into 500 length numeric array to represent original text so the model can recognize them. We use the embedding (the matrix we made in the previous steps) to encode the text into index inside the embedding.**

**The padding method will be used so every encoded sentence will be same length, which is 500, we append the ‘pad’ symbol if the length is less than 500, get first 500 tokens if the length is greater than 500.**

**Step 5: Configure model and train/validation/test model**

**We implement four models (ANN, RNN, LSTM, CNN), we use three dataset(train, validation, test) to evaluate the model. They are all based on tensorflow and keras. And confusion matrix will be generated by sklearn library.**

**Accuracy and Confusion Matrix:**

**Table for four model**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ANN** | **RNN** | **LSTM** | **CNN** |
| **Train Loss** | **0.4882** | **0.6342** | **0.3745** | **0.2843** |
| **Train Accuracy** | **0.7898** | **0.6453** | **0.8386** | **0.8834** |
| **Validation Loss** | **0.5023** | **0.5803** | **0.4618** | **0.5695** |
| **Validation Accuracy** | **0.7682** | **0.7280** | **0.7892** | **0.7766** |
| **Test Loss** | **0.5023** | **0.5803** | **0.4618** | **0.5695** |
| **Test Accuracy** | **0.7682** | **0.7279** | **0.7892** | **0.7766** |

**CM for ANN:**

**[[2163 827]**

**[ 332 1678]]**

**CM for RNN:**

**[[1616 481]**

**[ 879 2024]]**

**CM for LSTM:**

**[[1920 479]**

**[ 575 2026]]**

**CM for CNN:**

**[[2121 743]**

**[ 374 1762]]**

**It seems that LSTM and CNN perform better than ANN and RNN for this dataset, LSTM perform generally better but with more training time and space.**

**NewsAPI (200 samples for two different topic):**

**Two topics: Abortion and Election**

**Get 100 samples for abortion and 100 samples for election.**

Text

Description automatically generated

Text

Description automatically generated

**Top words for abortion: rights, voters, states, midterm**

**Top words for election: polls, state, president, close, results**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **ANN** | **RNN** | **LSTM** | **CNN** |
| **Train Loss** | **0.7347** | **0.5301** | **0.6923** | **0.6798** |
| **Train Accuracy** | **0.5050** | **0.7300** | **0.5400** | **0.5150** |

RNN perform better for the dataset gather from NewsAPI