A. Describe the purpose of this data analysis by doing the following:

A. Describe the purpose of this data analysis by doing the following:

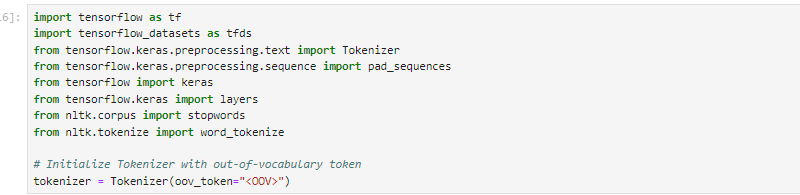
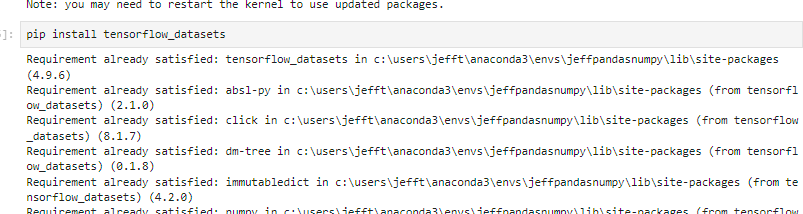
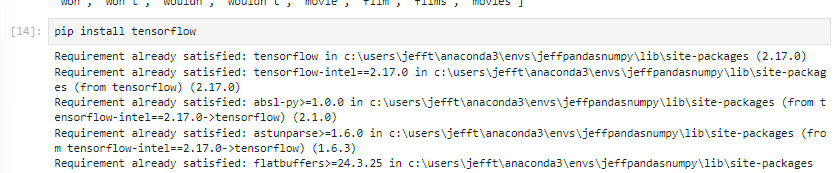
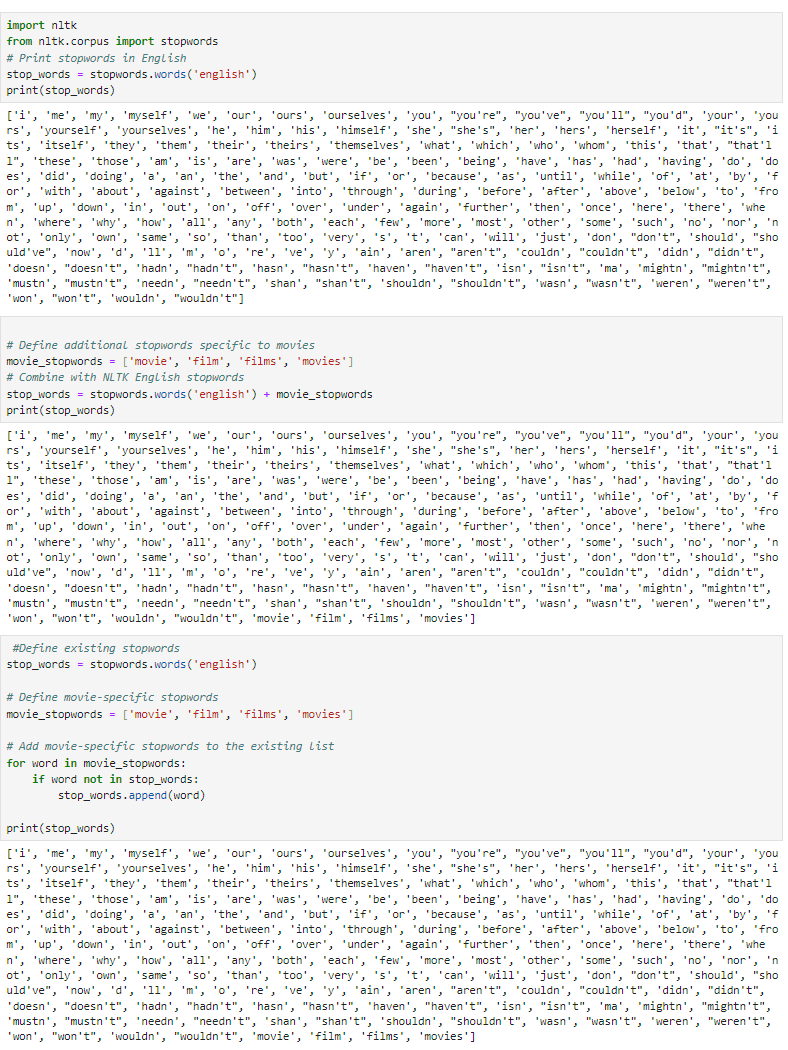
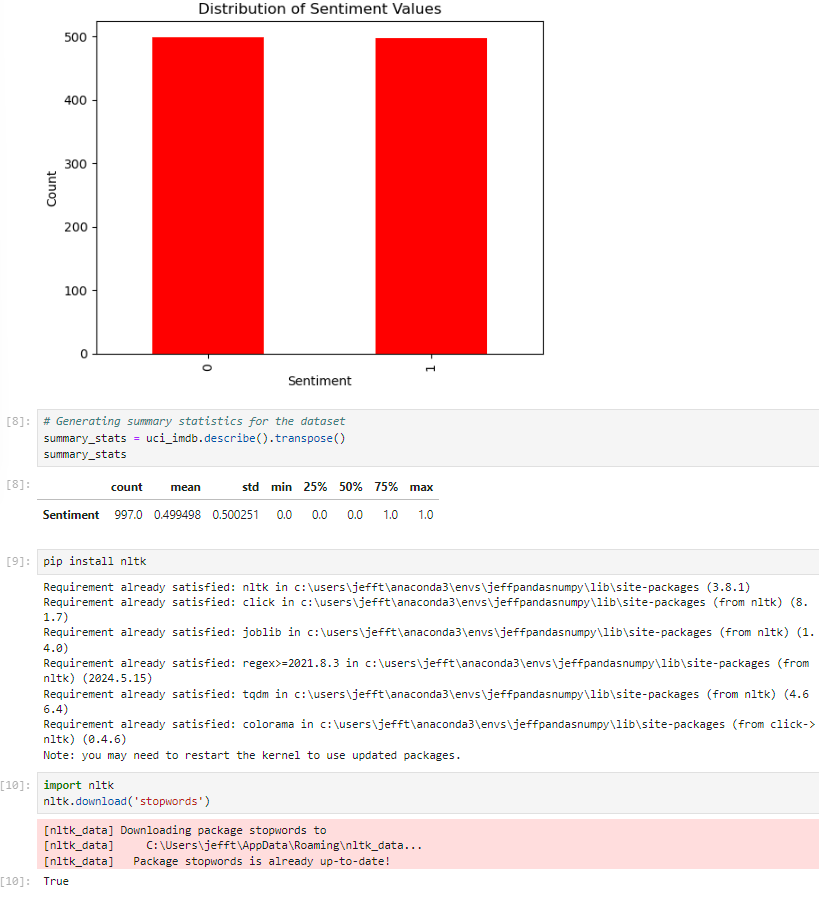
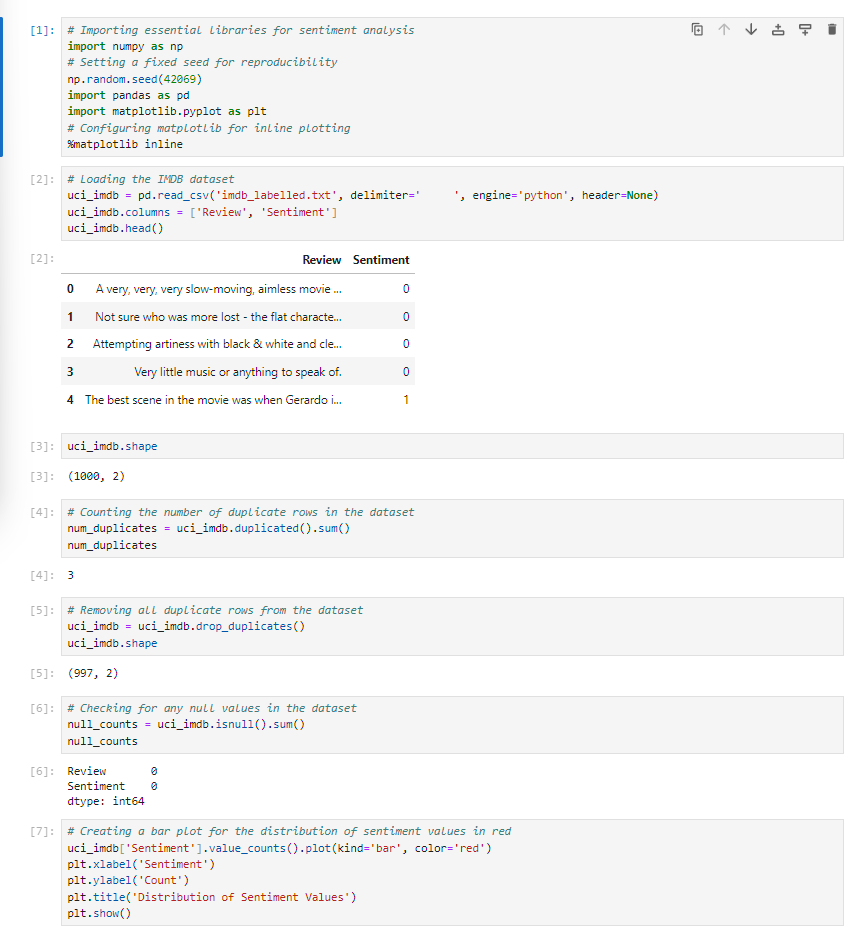
1. Summarize **one** research question that you will answer using neural network models and NLP techniques. Be sure the research question is relevant to a real-world organizational situation and sentiment analysis captured in your chosen data set(s).

*Note: If you choose to use more than one data set, you must concatenate them into one data set for parts II and III.  
Can a part of the UCI-provided IMDB dataset be used to train a neural network model to predict the sentiment of reviews?*

2. Define the objectives or goals of the data analysis. Be sure the objectives or goals are reasonable within the scope of the research question and are represented in the available data.

The goal of this project is to construct a deep learning network and natural language processing pipeline to analyze sentiment on IMDB movie reviews.

3. Identify a type of neural network capable of performing a text classification task that can be trained to produce useful predictions on text sequences on the selected data set.



**Part II: Data Preparation**

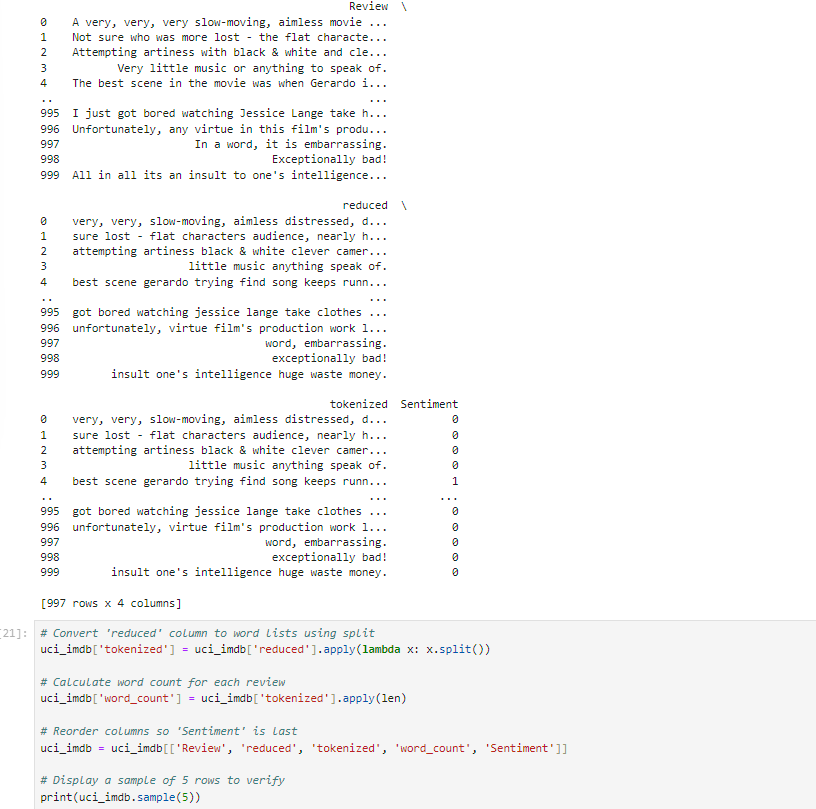
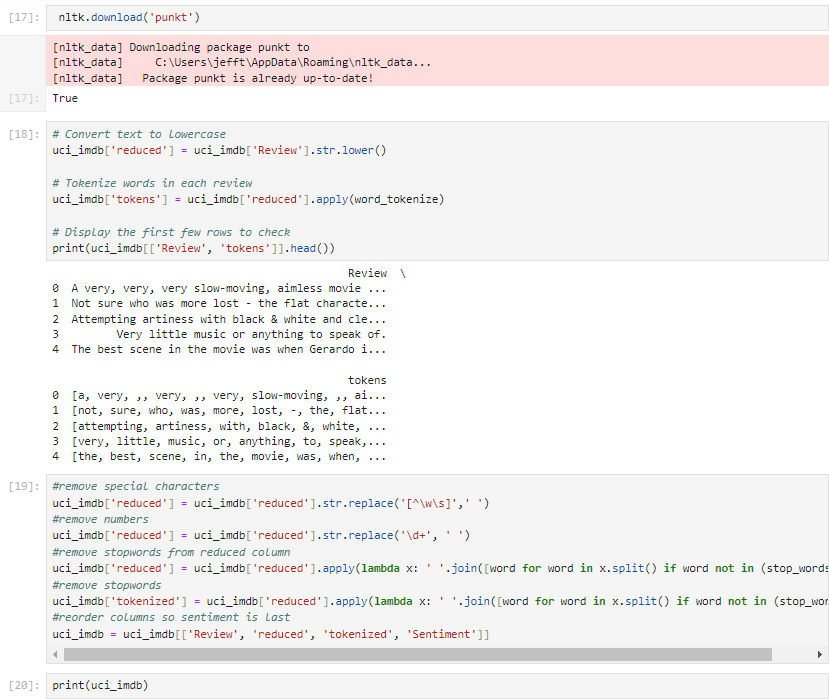
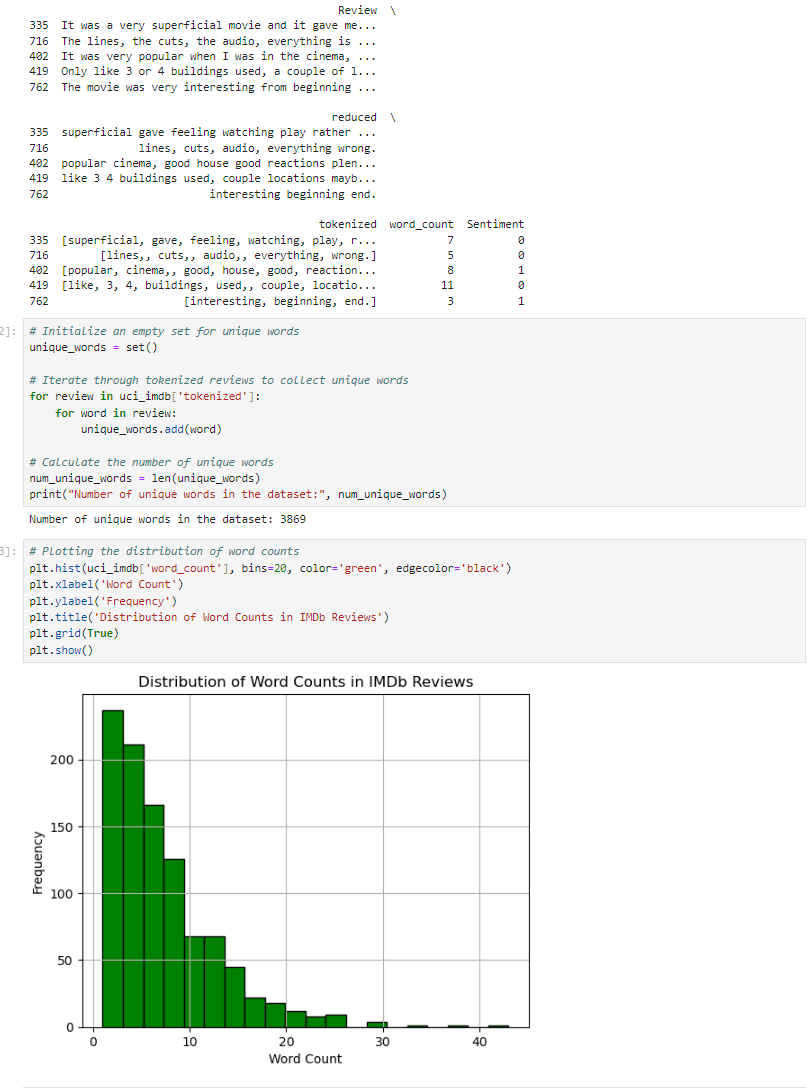
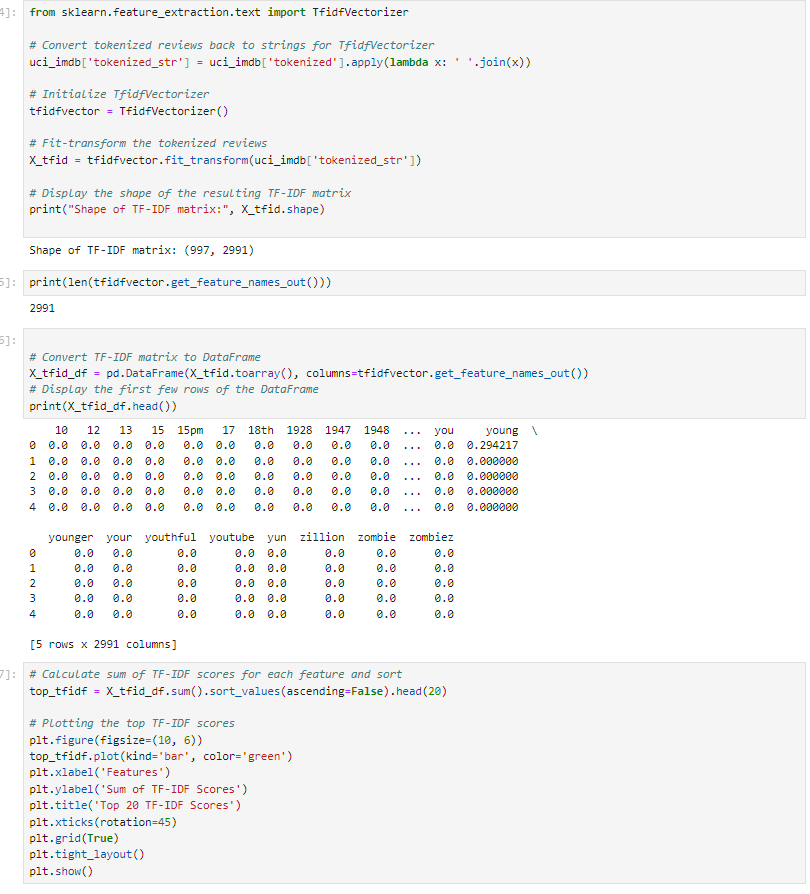
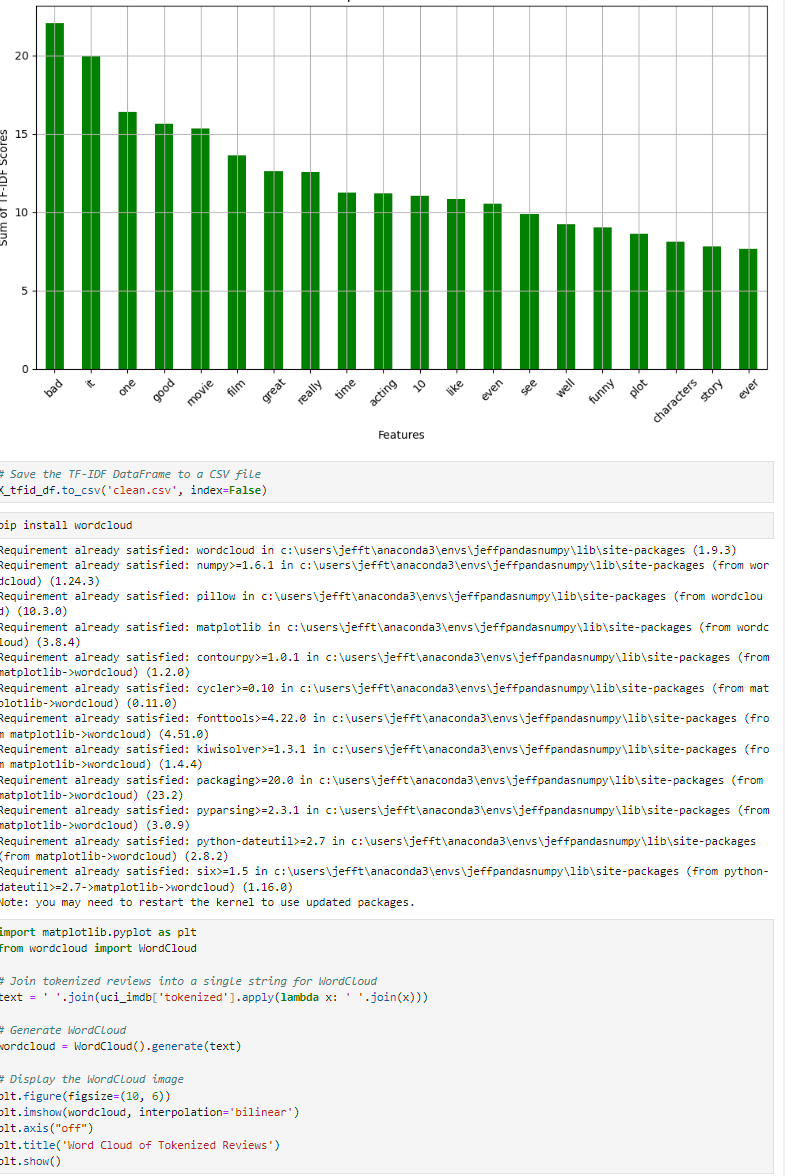
B. Summarize the data cleaning process by doing the following:

1. Perform exploratory data analysis on the chosen data set, and include an explanation of *each* of the following elements:

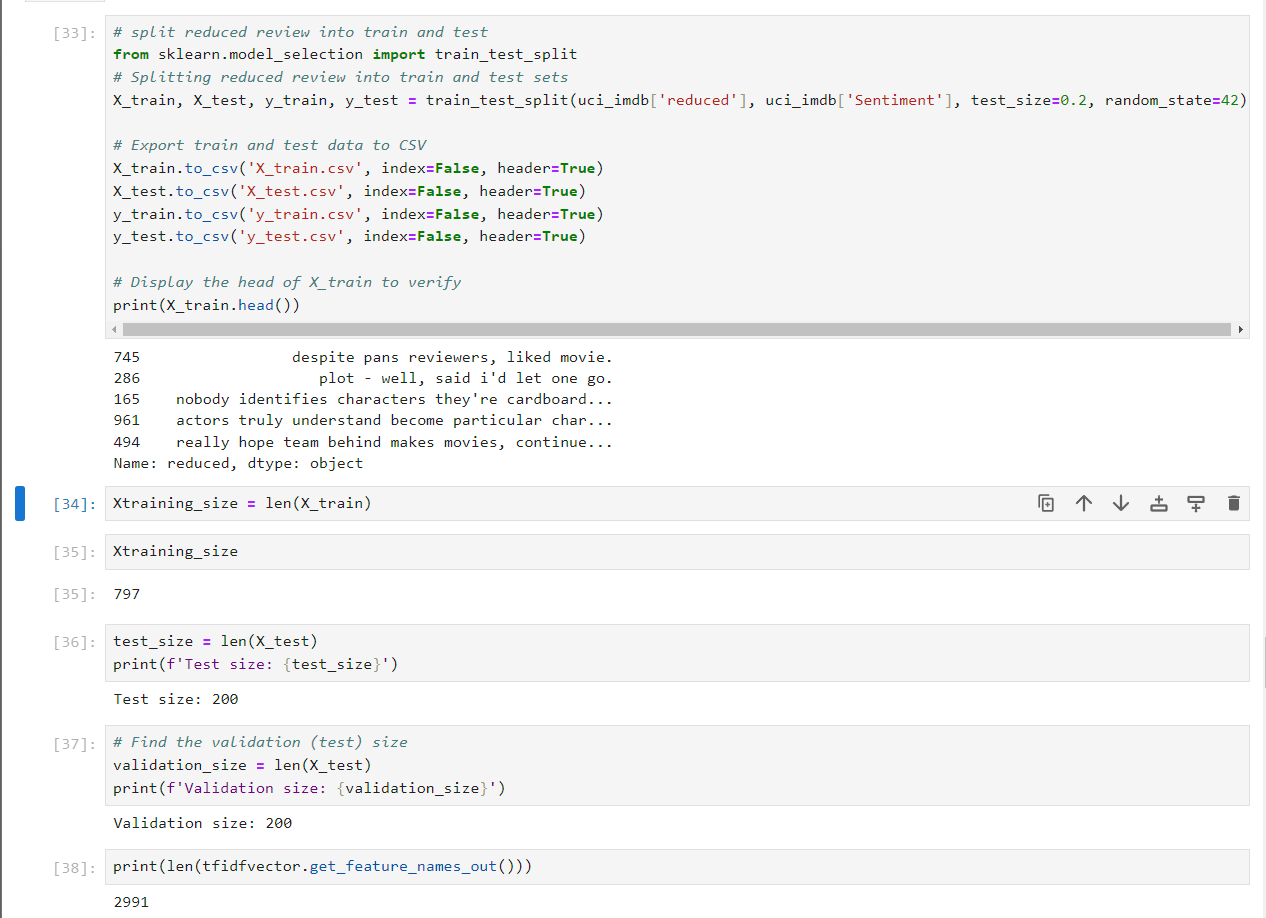
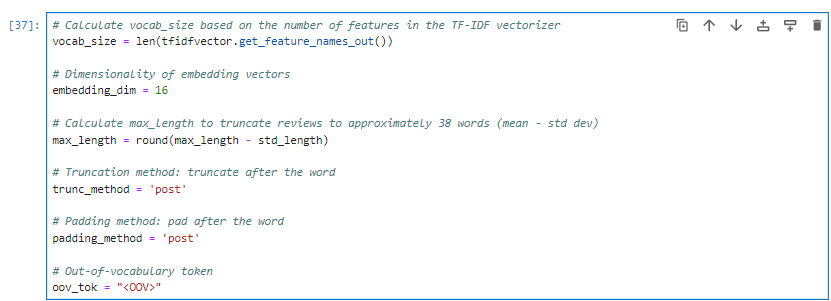
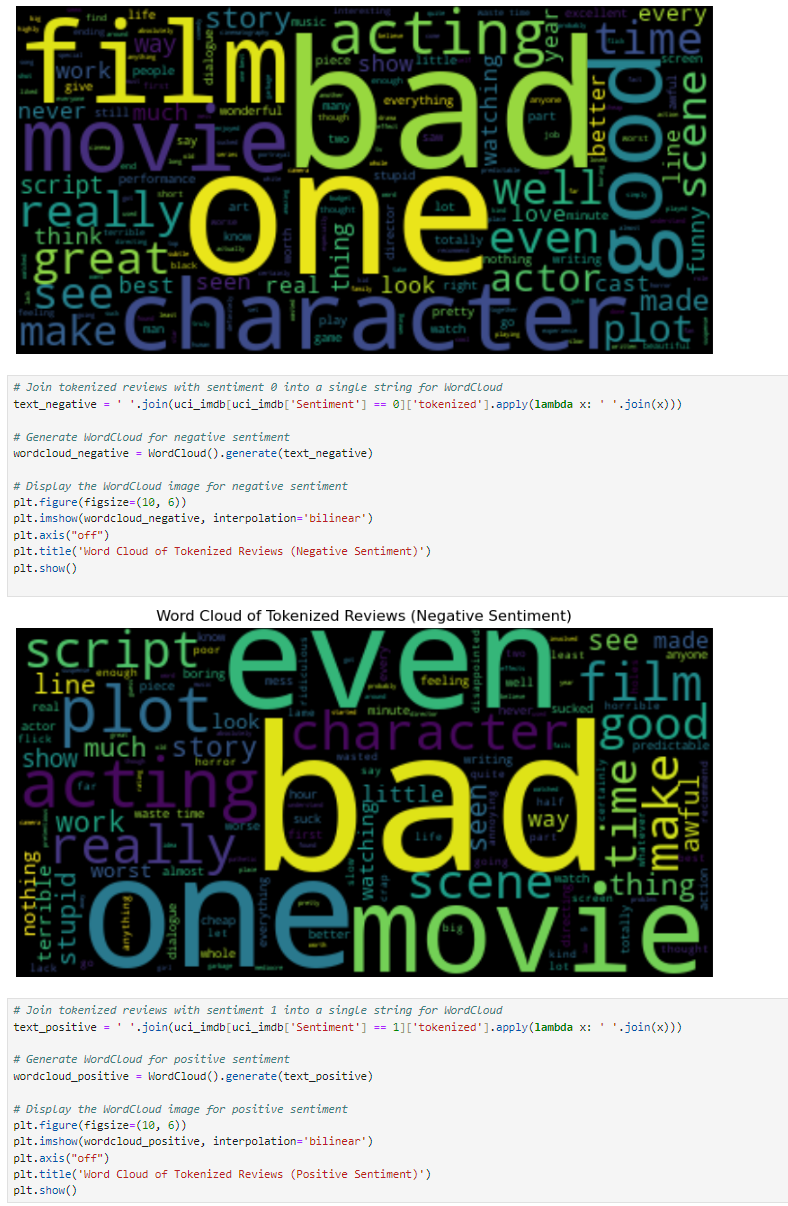
• presence of unusual characters (e.g., emojis, non-English characters)

To look at the presence of unusual characters, I used (uci\_imdb['reduced'] = uci\_imdb['reduced'].str.replace('[^\w\s]', ' ')

This code removes special characters from the text such as punctuation,special symbols,mathematical symbols and any other non Alphanumeric characters.

Above I converted the columns I reduced to word lists using splits. This was done so I can calculate the word count for each review. Then I reordered the columns so sentiment is last and the data is easier to comprehend when printed.Afterwards, I created an empty set so I can collect unique words in the tokenized reviews established earlier. Then I plotted the distributions of word counts for data visualization. Next I begin initializing TFIDF. TF-IDF converts a collection of raw documents into a matrix of TF-IDF features. I Initialize then fit the tokenized reviews so I can see the shape and length of the matrix, then i convert the matrix to a dataframe so I can calculate the sum of features in the TF IDF datafrane and to sort them. 

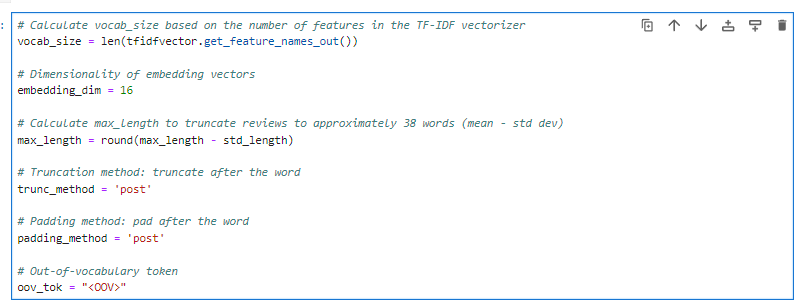
From here I generate my first wordcloud by joining tokenizing reviews into a single string. Then I decide to investigate negative and positive sentiments separately with their own word cloud. I have negative sentiments established as 1 and positive as 0 to generate the separate word clouds. After the word clouds are generate I began splitting my reduced review data into training and testing sets and exporting them to separate CSV files.



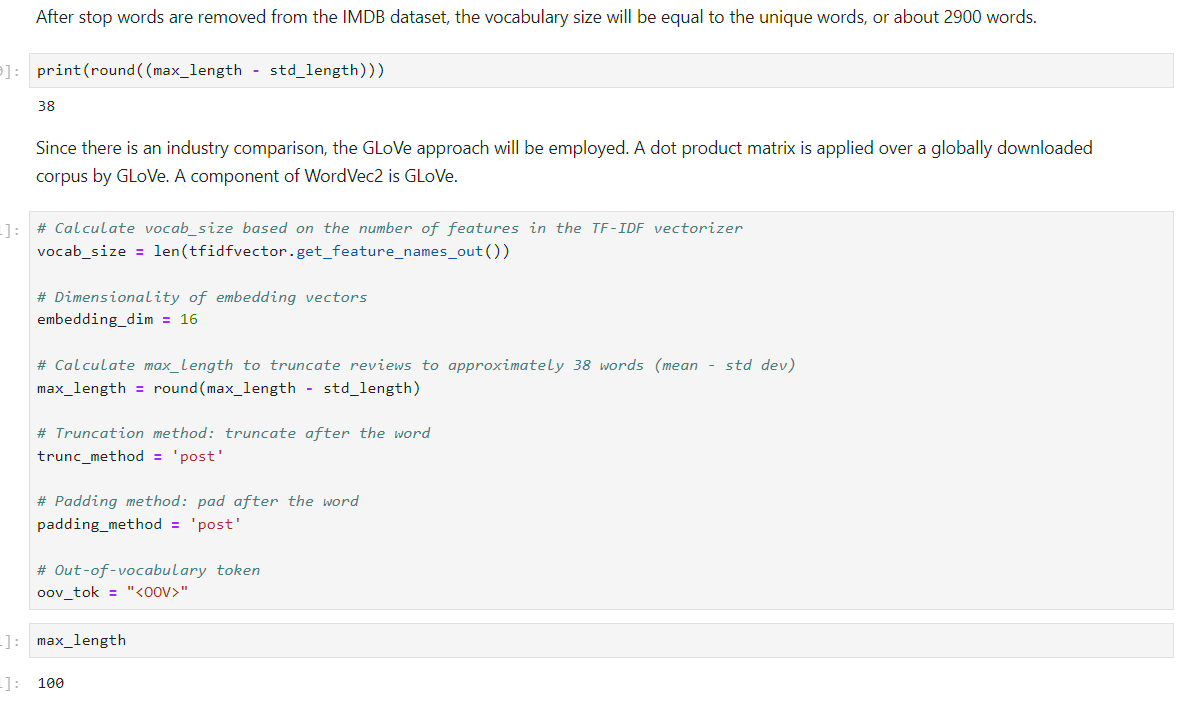
• vocabulary size

After all the stop words are removed from the IMDB dataset, the vocabulary size will then be equal to the unique words, 2900 words.

• proposed word embedding length

industry comparison is available, and the GLoVe method will be used. GLoVe applies a dot product matrix across a globally downloaded corpus. GLoVe is a part of WordVec2.

• statistical justification for the chosen maximum sequence length



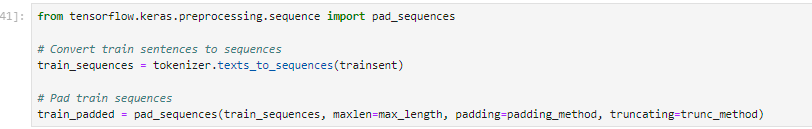
The Maximum sequence length was found by searching for the longest review in the dataset and by number of detected words. From there I Calculated Max length to truncate words to approximately 43 by subtracting the standard deviations of length. Since the sequences don't have to have the same length, there is no maximum sequence length.

2. Describe the goals of the tokenization process, including any code generated and packages that are used to normalize text during the tokenization process.

Tokenization is the process of dissecting language into smaller chunks so that deep learning networks can process them. To construct a tokenized review list, I decided to use the built-in Python string and regex packages in the code above. To create a thorough language model, these tokens can then be examined for uniqueness and connections to other words and tokens.

3. Explain the padding process used to standardize the length of sequences.

The data can be padded out to take on the shape that the model anticipates once it has been divided into training, test, and validation sets. Since that was the length of the longest review following the tokenization of the words, each review in this instance is padded out to 38 words. If padding is required, it happens at the conclusion of the review. The fact that the final "words" in the selections are zeroes indicates that the reviews were not required to be padded out to a maximum of 38 words because they did not include that many words.To do the padding, I converted the train sentences to sequences, then padded using the padding\_method and truncated using trunc\_method



• a screenshot of a single padded sequence



4. Identify how many categories of sentiment will be used and an activation function for the final dense layer of the network.

Sentiment are binary as positive or negative.

5. Explain the steps used to prepare the data for analysis, including the size of the training, validation, and test set split (based on the industry average).

To prepare this data for analysis, first I looked for the presence of unusual characters and removed them. During ths step I removed numbers,stop words, and reordered columns so sentiment appeared last. Next I converted the tokenized column to word lists using splits. From there I was able to calculate the work count for each review. After I found out the word counts, I created an empty data set to then calculate the number of unique words. With that out the way I then proceeded to plot the distributions of word counts. Next I joined the tokenized reviews into a single string so I could generate a WordCloud. I then proceeded to do this again but for reviews with positive and negative sentiments. After I had the word clouds generated, I was able to split the reduced reviews into training and test for the analysis later. After that I then found the max word count,median word count, and mean word count of the reviews. Finally I began the tokenizer process so that the deep learning networks can process the data. I used the built in python string and regex packages to do this. After that, I was able to fit the tokenizer on the train sentences so that it could actually process the data.

Size of

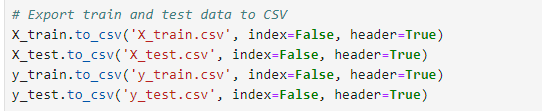
Training:797

Test: 200

Validation:200



6. Provide a copy of the prepared data set.



**Part III: Network Architecture**

C. Describe the type of network used by doing the following:

1. Provide the output of the model summary of the function from TensorFlow.



2. Discuss the number of layers, the type of layers, and the total number of parameters.

Layer 1 embedding

The length of the vocabulary size that was previously kept in max\_length is the same as the vocabulary size. The maximum sentence length is 100 words with 16 embedded dimensions. One hundred sixty thousand parameters in all.

Layer 2, "GlobalAveragePooling1D\_1"

16 embedded dimensions with 0 parameters The vector is also flattened into a single dimension by this layer

Layer 3, Dense\_4 hidden layer: 64 neurons 1088 parameters relu

Layer 4 , dropout\_2 layer: 64 neurons 0 parameters

Layer 5 dense\_5 32 neurons 2080 parameters hidden relu

Layer 6 dropout\_3 32 neurons 0 parameters

Layer 7 dense\_6 6 neurons 198 parameters hidden relu

Layer 8 dense\_7 1 neuron 7 parameters output sigmoid

3. Justify the choice of hyperparameters, including the following elements:

• activation functions

Since this is a binary classification problem and sigmoid is the standard for binary classification problems, relu should be used for the hidden layers and sigmoid for the output layer (Sucky, 2021).

• number of nodes per layer

For the first hidden layer, I used 64 nodes, and for the output layer, I used 1 node. Given that this is a binary classification problem, the output layer just has to provide one result. For a binary classification task, the first hidden layer is a useful place to start because it may be changed as needed and starts close to the square root of the minimum review length.

• loss function

For binary classification problems, binary crossentropy is the accepted measure (Sucky, 2021).

• optimizer

When it comes to binary classification issues, Adam is the norm (Sucky, 2021).

• stopping criteria

I started with 5 epochs and intended to adjust if required.

• evaluation metric

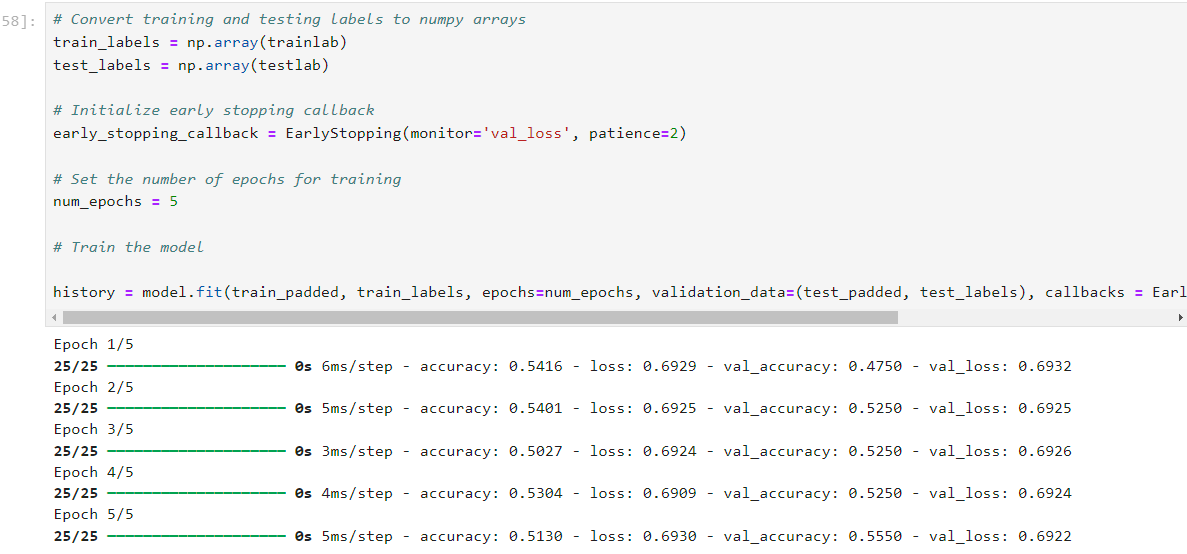
The benchmark for binary classification issues is accuracy.

**Part IV: Model Evaluation**

D. Evaluate the model training process and its relevant outcomes by doing the following:

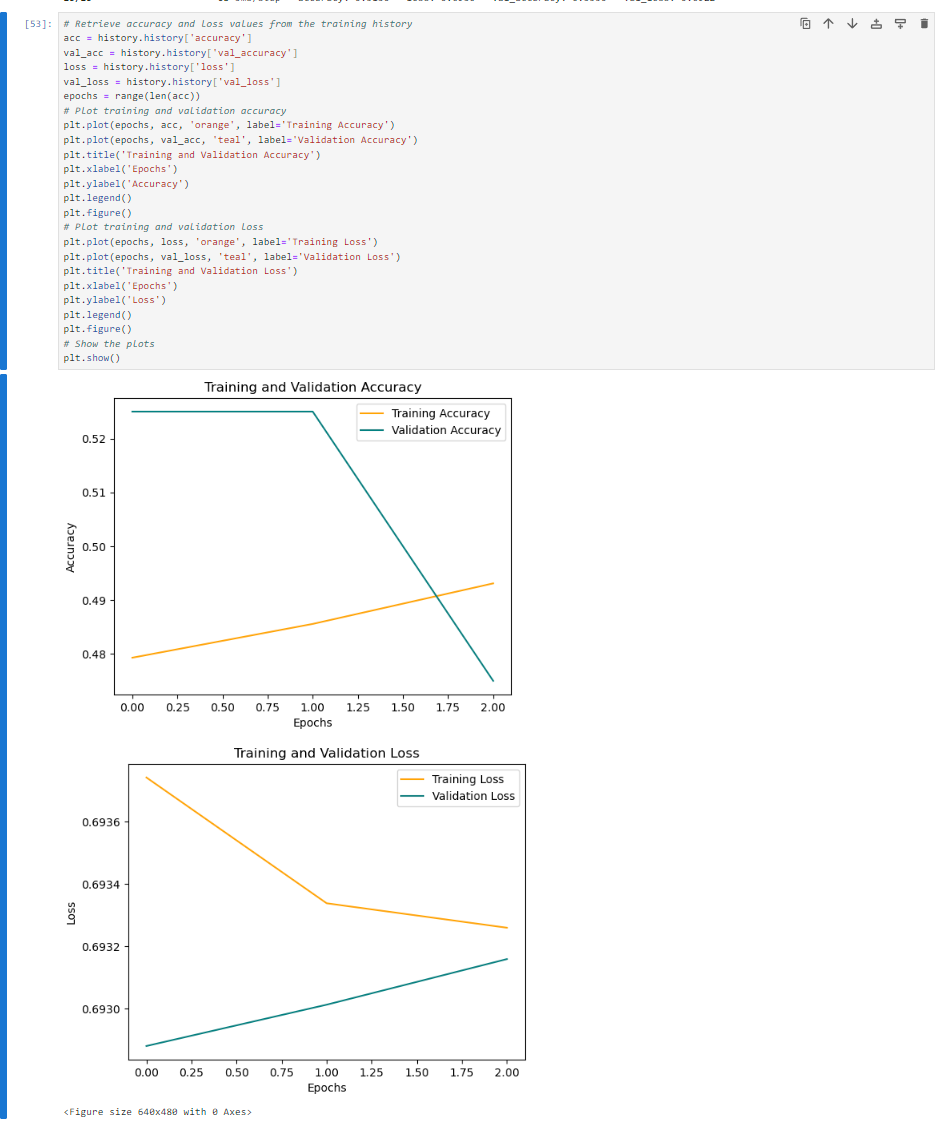
1. Discuss the impact of using stopping criteria to include defining the number of epochs,

including a screenshot showing the final training epoch.

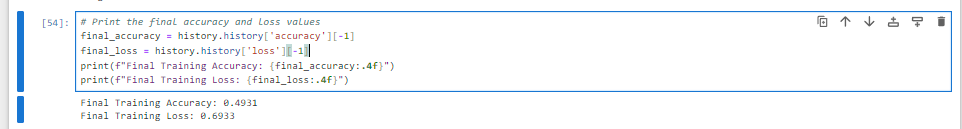
The model can be made to cease training when it is no longer getting better by using stopping criteria. Time and resources can be saved and overfitting can be avoided with this method. A patience parameter can be set to continue looking for a better model even if the validation loss does not improve after a few iterations. This allows the model to stop training when the validation loss stops improving by using a callback earlystop. Although I employed an early stoppage in my model, training continued until the fifth. To determine if the model would have ended training earlier, more epochs would have been required, but the likelihood is that it would have overfit the data additionally.

2. Assess the fitness of the model and *any* actions taken to address overfitting.

For this model I used 5 epochs.I started with 5 epochs and intended to adjust if required to allow easy model adjustment instead of starting high and going lower. To try to prevent overfitting, I used stopping criteria. This stops training when the model is no longer improving. ( use epoch accuracy) Based on the findings, the model did overfit since the validation loss was greater than the training loss. The model could have been trained for a longer period of time, but I had not expected it to overfit to such an extent. The last training metrics show that our TensorFlow model is not performing to its full potential. The model's final training accuracy of 0.4969 indicates that it is only roughly 49.69% accurate in predicting the training data—barely any better than a random guess. The model's final training loss is 0.6931, a value that generally indicates that it performs no better than chance, particularly in tasks involving binary classification. These results highlight significant issues with the model's ability to learn from the training data, suggesting the need for a thorough review of the model architecture, training process, and hyperparameters to improve its performance.

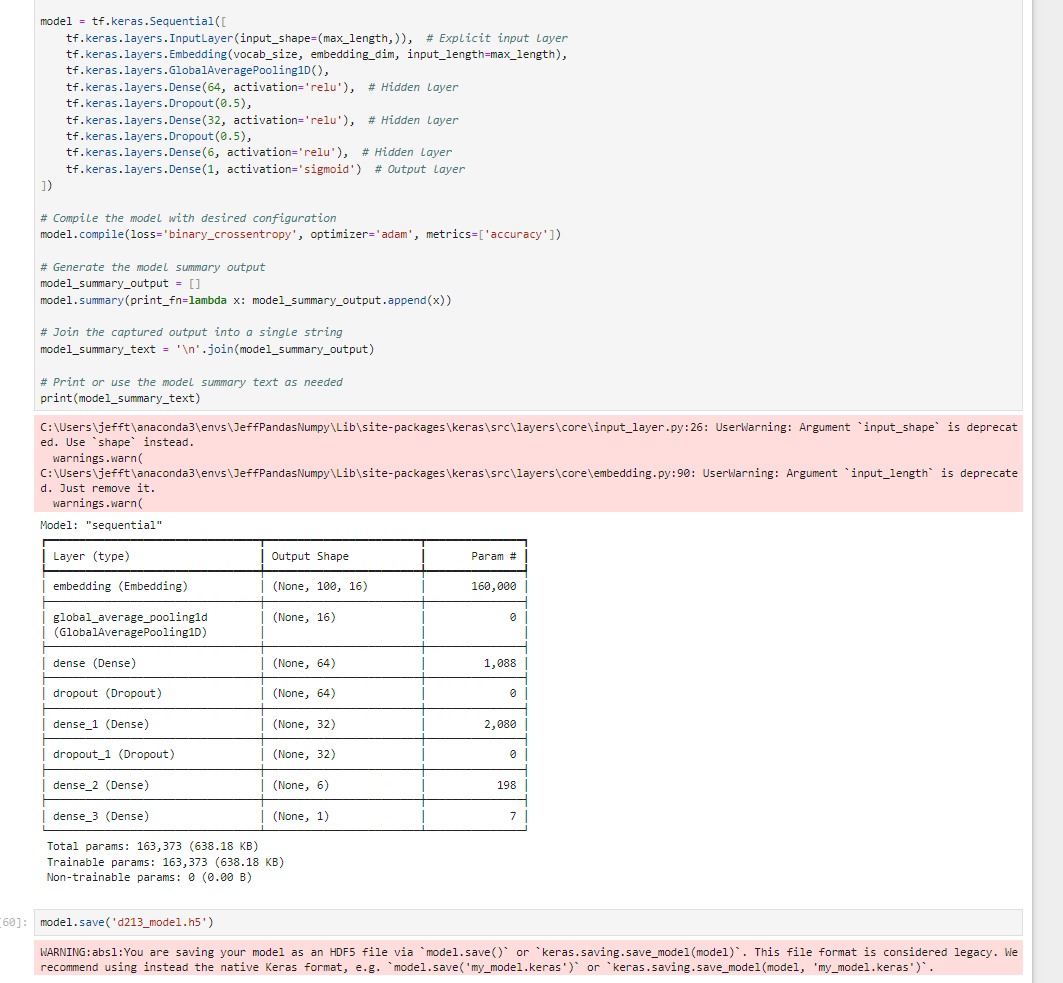
3. Provide visualizations of the model’s training process, including a line graph of the loss and chosen evaluation metric.

4. Discuss the predictive accuracy of the trained network using the chosen evaluation metric from part D3.



The model's final training accuracy of 0.4969 indicates that it is only roughly 49.69% accurate in predicting the training data—barely any better than a random guess. The model's final training loss is 0.6931, a value that generally indicates that it performs no better than chance, particularly in tasks involving binary classification.

**Part V: Summary and Recommendations**

E. Provide the code you used to save the trained network within the neural network.  


F. Discuss the functionality of your neural network, including the impact of the network architecture.

This deep learning model for natural language processing (NLP) was developed to predict movie reviews' likelihood properly, without overfitting. The model was tested on 20% of the data and trained on 80% of the data. The model was trained for five epochs, with no breaks until the fifth epoch. The model was determined to be overfit since the validation loss was greater than the training loss. The model might have been trained for a longer amount of time to see whether it would have stopped training earlier, though it most likely would have just maintained overfitting the data. This model's accuracy is 49%. The significant loss can be attributed to a binary classification issue. There are a total of eight layers: one output layer, two dropout layers, and three hidden layers. Ssigmoid activation was used as the output from the first hidden layer, which used relu. Adam served as the optimizer and binary crossentropy as the loss function in the model itself.

G. Recommend a course of action based on your results.

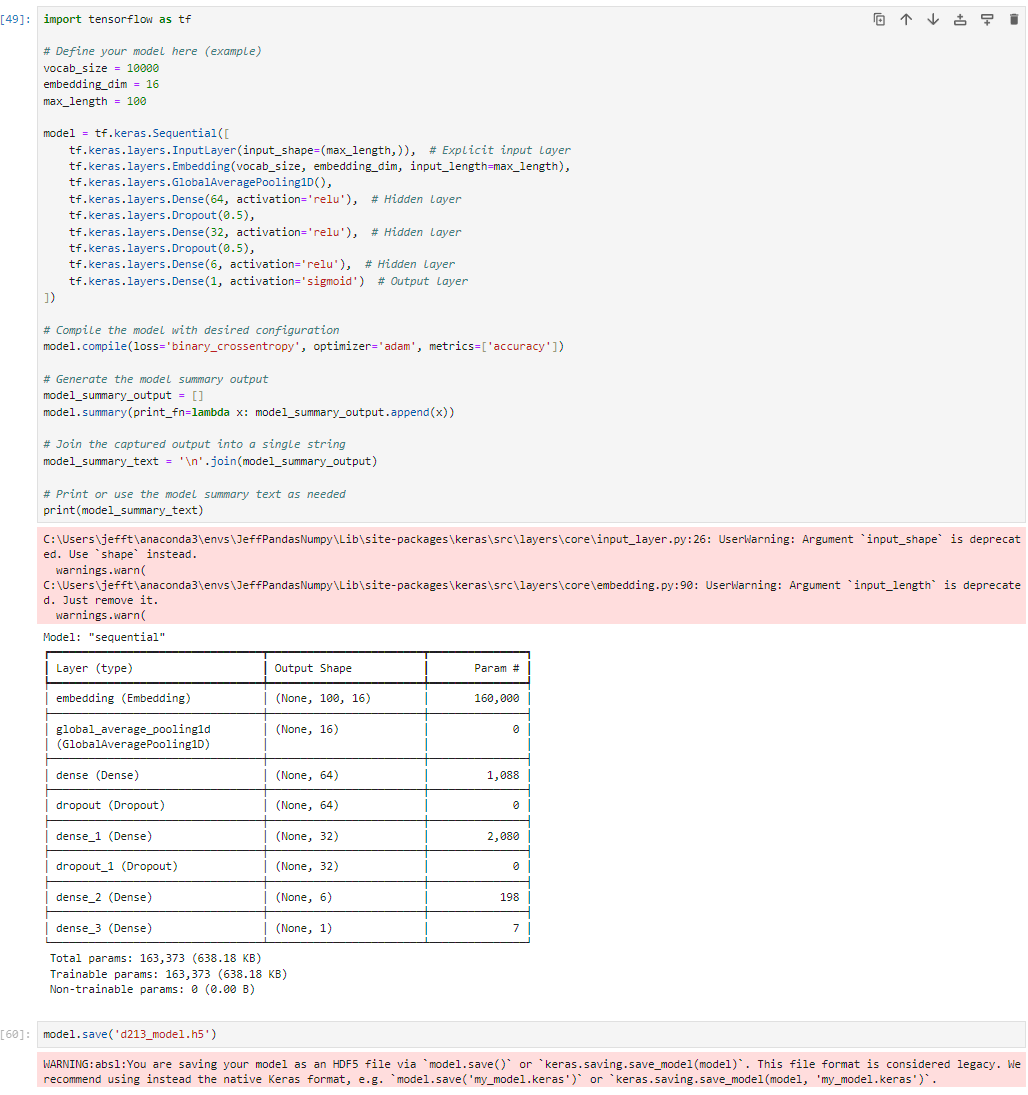
Because of its poor accuracy, this model should not be used to infer sentiment from reviews of movies. The model was trained on movie reviews, not other industries, so it should not be used to forecast sentiment in other sectors of the economy. The model should be tested using a different collection of movie reviews in order to validate it, such as the UCI open source movie review dataset from Amazon.

**Part VI: Reporting**

**Sources**

Sucky, R. N. (2021, July 8). A complete step by step tutorial on sentiment analysis in Keras and tensorflow. Medium. Retrieved July 18, 2022, from <https://towardsdatascience.com/a-complete-step-by-step-tutorial-on-sentiment-analysis-in-keras-and-tensorflow-ea420cc8913f>

Model and model saving



Output saved to included file ‘d213\_model.h5’