Predict the sentiment of Reviews Using LSTM Model

Problem Statement:

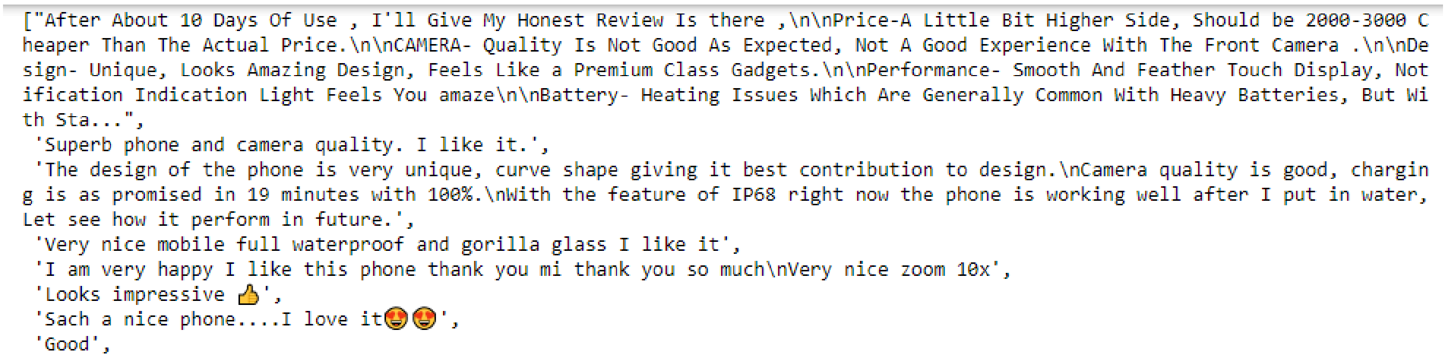
The aim is to predict the Type of the Sentiment By Analyzing the Reviews of the Customers.

Web Scraping:

Initially, the necessary libraries are imported, including webdriver, By, WebDriverWait, and other components from the Selenium package, as well as pandas for data manipulation and storage. The URL of the Flipkart product reviews page is defined and a Chrome WebDriver instance is initiated to open this URL.

A WebDriverWait instance is created to wait for elements to load for a maximum of 10 seconds. An empty list named reviews_list is initialized to store the extracted reviews. The code iterates over a range of indices from 3 to 12, each corresponding to a review element on the page. In each iteration, the XPath of the review element is dynamically constructed and the WebDriverWait instance waits until this element is present in the DOM.

Once the element is located, its text content is extracted and appended to the reviews_list. After collecting all reviews, the length of the reviews_list is printed to verify the number of reviews extracted.



```
["After About 10 Days Of Use , I'll Give My Honest Review Is there ,\n\nPrice-A Little Bit Higher Side, Should be 2000-3000 C
heaper Than The Actual Price.\n\nCAMERA- Quality Is Not Good As Expected, Not A Good Experience With The Front Camera .\n\nDe
sign- Unique, Looks Amazing Design, Feels Like a Premium Class Gadgets.\n\nPerformance- Smooth And Feather Touch Display, Not
ification Indication Light Feels You amaze\n\nBattery- Heating Issues Which Are Generally Common With Heavy Batteries, But Wi
th Sta...",
 'Superb phone and camera quality. I like it.',
 'The design of the phone is very unique, curve shape giving it best contribution to design.\nCamera quality is good, chargin
g is as promised in 19 minutes with 100%.\nWith the feature of IP68 right now the phone is working well after I put in water,
Let see how it perform in future.',
 'Very nice mobile full waterproof and gorilla glass I like it',
 'I am very happy I like this phone thank you mi thank you so much\nVery nice zoom 10x',
 'Looks impressive 👍',
 'Sach a nice phone....I love it😍😍',
 'Good',
```

Finally, a pandas DataFrame is created from the reviews_list and saved as a CSV file named 'reviews.csv'. This allows for easy storage and subsequent analysis of the scraped reviews. This way we scraped 352 reviews from flipkart products.

Reading and Preparing Data:

The next step involves reading a CSV file named 'reviews.csv' into a pandas DataFrame. This CSV file contains product reviews that will be analyzed. The code drops an unnecessary column labeled 'Unnamed: 0', which is likely an index column from the CSV file, and renames the column containing the reviews to 'Reviews'. This step is crucial for cleaning and organizing the data, making it easier to work with.

Text Preprocessing:

The core of the preprocessing is encapsulated in the preprocess function. This function tokenizes the text, converting it into a list of lowercased words. It then removes stop words (common words that do not carry significant meaning, such as 'and', 'the', etc.) and non-alphabetic tokens to focus on meaningful words. Finally, it lemmatizes the tokens, reducing them to their base or root form (e.g., 'running' becomes 'run'). This lemmatization helps in reducing the dimensionality of the text data and ensures that different forms of a word are treated as a single term. After preprocessing, the script creates a Gensim Dictionary from the processed reviews. This dictionary maps each unique word to a unique integer ID. Using this dictionary, the code generates a corpus, which is a list of bag-of-words representations of the reviews. Each review is represented as a list of tuples, with each tuple containing a word ID and its corresponding frequency in the review.

Training the LDA Model:

Latent Dirichlet Allocation (LDA) is a popular topic modeling technique. This technique is particularly useful for discovering hidden themes in large collections of text data.The number of topics for the LDA model is set to 4, indicating that the model will identify four distinct themes within the reviews. The LDA model is then trained using the corpus and dictionary over 15 passes. This iterative process helps the model to converge and produce stable topics.

Extracting and Visualizing Topics:

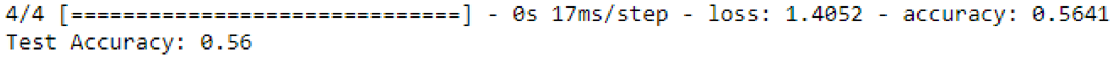
Once the model is trained, the script prints the top 5 words in each of the identified topics, providing a snapshot of the main themes. Although the code for enabling notebook visualization with pyLDAvis and preparing the visualization is included, it is not active in the current form of the script. If enabled, this would provide an interactive way to explore the topics.

Identifying Dominant Topics:

A function named get_dominant_topic is defined to determine the dominant topic for each document in the corpus. This function sorts the topics for each document by their probability and selects the one with the highest probability. The dominant topic for each review is then added as a new column to the DataFrame. This step is crucial for understanding which topics are most prevalent in individual reviews and can help in segmenting the data based on these topics.

Model Building:

It’s begin by the text data is converted into numerical format using one-hot encoding, and sentences are padded to a uniform length. The target variable, 'dominant_topic,' is one-hot encoded to facilitate multi-class classification. A Long Short-Term Memory (LSTM) neural network is built using TensorFlow and Keras. The Sequential model consists of an embedding layer that converts words into dense vectors, an LSTM layer to capture sequential dependencies, and a dense output layer with softmax activation for multi-class classification. The model is compiled with the Adam optimizer and categorical cross-entropy loss function and trained on the dataset for 50 epochs. The script splits the data into training and testing sets and evaluates the model's performance on the test set, printing the test accuracy. But the Accuracy we got from the model is not so good. As we have less no of reviews and we have to do some hyper parameter tuning in topic modeling.



```
4/4 [==============================] - 0s 17ms/step - loss: 1.4052 - accuracy: 0.5641
Test Accuracy: 0.56
```