Sentiment Analysis POC Report

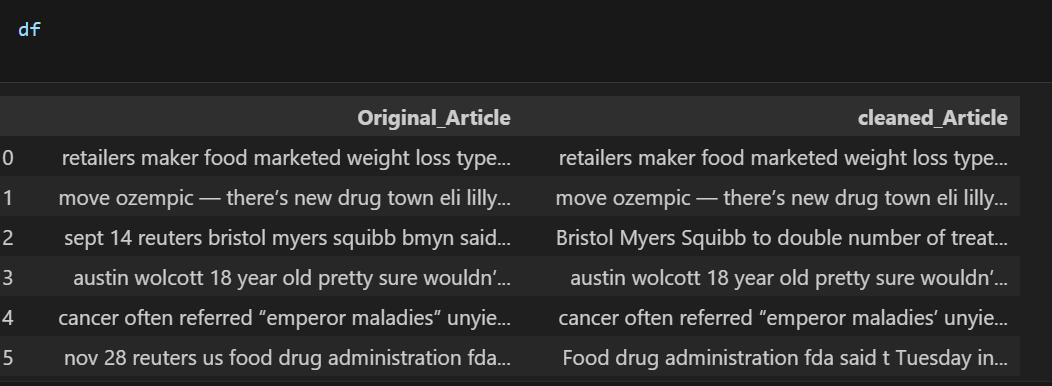
**Step1 →data cleaning**

We use Bart model to summarize original\_Article columns from the dataset and create cleaned\_Article columns and put the summarized text data of all original\_Article .

BART model, short for Bidirectional and Auto-Regressive Transformers, introduced by Facebook AI for text summarization.

I used to Transformers library of deep learning which include features (like BartTokenizer, BartForConditionalGeneration)

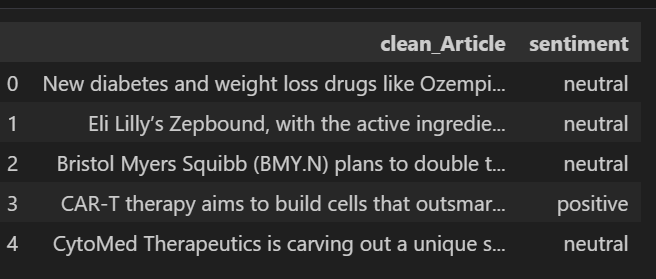
**Here is cleaned data :**



**Step 2 —> apply sentiment Analysis for all cleand\_Article columns of dataset:**

I used Bert model - to apply Sentiment analysis on all dataset of cleaned\_article and check. And just check the mood (**like positive ,negative and Neutra**l), this is used to analyze the text and classify the Mood.

**Here is sentiment analysis data sample:**

****

**Step3 —> model Training on specific text dataset**

**Pandas (pd) Library:**

Pandas is a powerful Python library for data manipulation and analysis. Here, it's used to read and handle the dataset from a CSV file, making it easier to work with structured data.

**Numpy (np) Library :**

Numpy is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. In this script, it's used for numerical operations and array handling.

**Pickles Library :**

Pickling is a way to serialize Python objects into a byte stream, allowing them to be saved to a file and later reconstructed. Here, it's used to save the tokenizer object, which is crucial for converting text data into a format suitable for model training and inference.

**Tokenizer:**

The Tokenizer class from Keras preprocessing is used to convert text data into numerical sequences. It tokenizes the text by splitting it into individual words (or tokens) and assigns a unique integer to each word based on its frequency in the dataset. This preprocessing step is essential for training machine learning models on text data.

**Sequential Model:**

Sequential is the simplest type of Keras model, allowing you to build a neural network layer by layer in a linear fashion. Here, it's used to define the architecture of the sentiment analysis model.

**Embedding Layer:**

Embedding layers are used to convert integer-encoded words into dense vectors of fixed size. These vectors are trainable and can capture semantic relationships between words. In this script, the embedding layer is the first layer in the neural network model, responsible for transforming the input text into a format suitable for LSTM processing.

**LSTM (Long Short-Term Memory):**

LSTM is a type of recurrent neural network (RNN) architecture, specifically designed to model sequence data and capture long-term dependencies. It's well-suited for tasks like sentiment analysis, where the order of words in a sentence matters. Here, it's used as the core processing unit within the neural network model.

**Dense Layer:**

Dense layers, also known as fully connected layers, are used to perform classification based on the features learned by the preceding layers. In this script, dense layers with activation functions are added to the model to map the learned features to the output sentiment classes.

**Dropout Layer:**

Dropout is a regularization technique used to prevent overfitting in neural networks by randomly setting a fraction of input units to zero during training. Here, it's employed to improve the generalization ability of the model by reducing the likelihood of overfitting to the training data.

**Model Compilation:**

The compile method configures the model for training by specifying the optimizer, loss function, and evaluation metrics. In this script, the Adam optimizer is used along with sparse categorical cross-entropy loss, which is suitable for multi-class classification tasks.

**Model Training:**

The fit method trains the model on the provided dataset. Here, it's used to train the sentiment analysis model on the input text data and corresponding sentiment labels.

**Model Saving:**

The save method is used to save the trained model to a file in the Hierarchical Data Format (HDF5). This allows the model to be reused or deployed for inference tasks without needing to retrain it from scratch.

**Tokenizer Saving:**

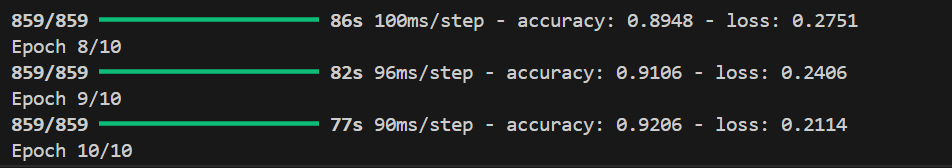
Lastly, the tokenizer object used for preprocessing the text data is saved to a file using pickling. This ensures that the same preprocessing steps can be applied consistently during model deployment or inference.

**Tensorflow library which include features (likeTokenizer,Sequential,Model,Embedding Layer,LSTM,dense layer,Dropout Layer,Model Compilation and Model training)**,

As we gone through the sample data only 26 record but if we train the model then not give good accuracy when a new data test we use dataset from kaggle during the training model.

Url-link:-<https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset?select=train.csv>

**Model Accuracy Report:**

****

**Setup script file Setup**

**Step1→**  create python environment python==3.12

Cmd: conda create -n sentiment\_analysis python=3.12

**Step2→** After creating environment we use activate environment name in terminal

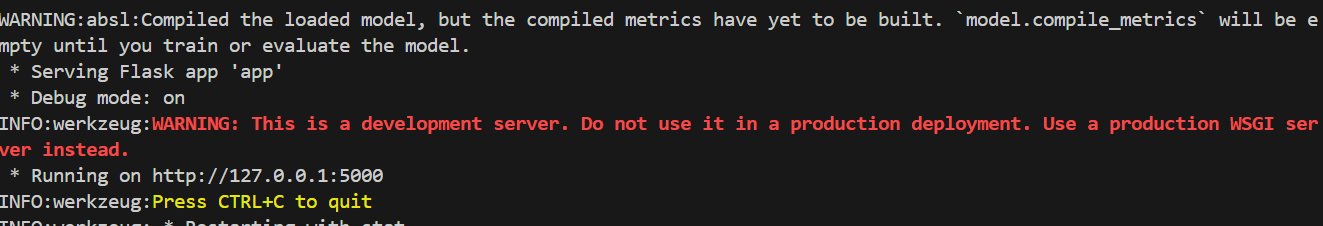
Cmd: conda activate sentiment\_analysis

**Step3→** run command pip install -r requirements.txt which will install all dependency library

Cmd: pip install -r requirements.txt

**Step4→** run app.py file for testing our model. After the ruined file we will receive the URL in the terminal and copy the urls and paste in the browser. Like

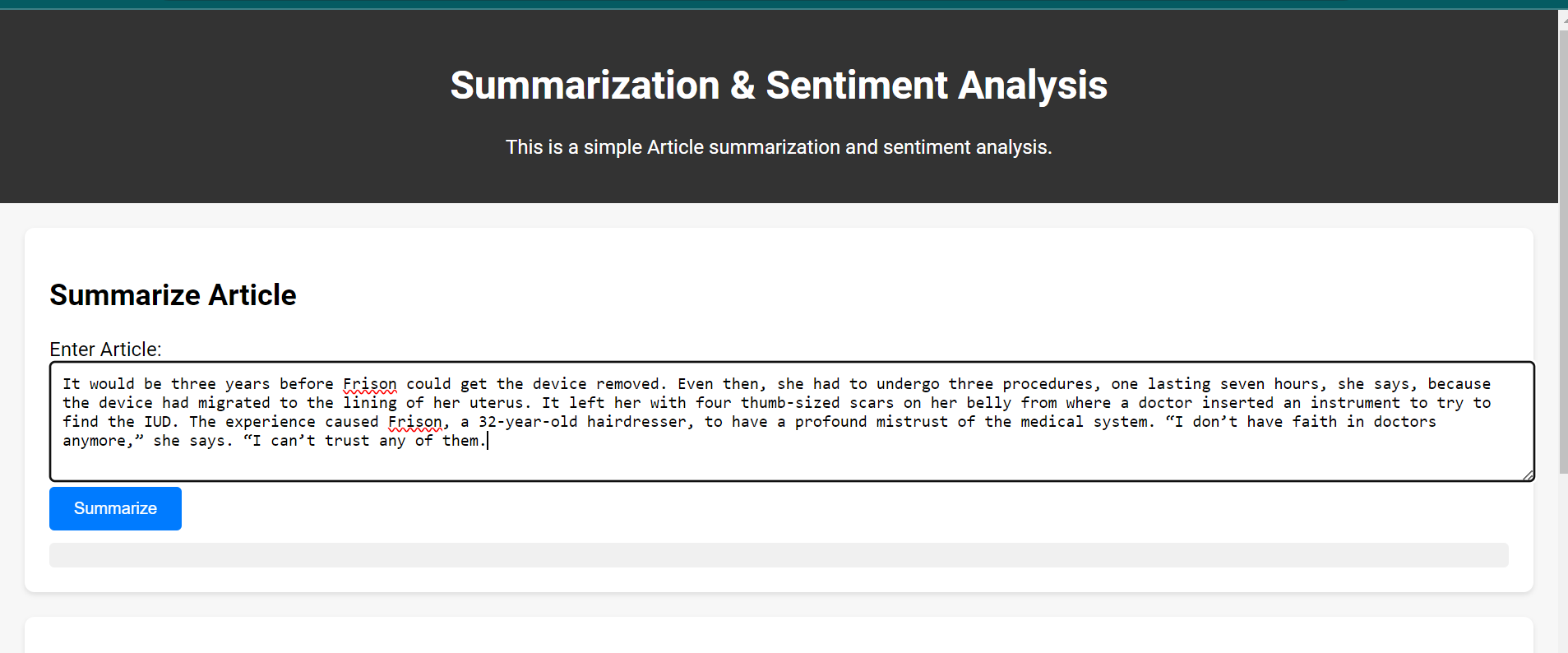
Cmd: python app.py



I skipped the training part because i have already trained the model and keep model folder

**Test the application:**

**Step1→** copy text of a Article from your source and paste the text in Summary Article Blok and click Summarize button

****

After clicking the summarize button then you get summarized text, copy the summarized text and paste it in the Analyze Sentiment Block and click on the Analyze Sentiment button. After that you will get the result.

