ML Intern - Assignment

By Mahak Gupta

NOTE:- I have performed this assignment in very limited resources and constraints, which may affect performance of the model. I have tried to build it without using pretrained models but face many issues with my system, it's crashing many times.

Approach:
(Category Prediction)

The approach involved several key steps:

Data Preprocessing: The dataset was preprocessed using techniques such as text cleaning, sentiment analysis, and feature extraction. This included removing stop words, punctuation, and non-alphanumeric characters, as well as lemmatization to reduce words to their base form.
Feature Extraction: Features such as polarity, subjectivity, and length of the cleaned text were extracted to provide additional information for the classification model.
Model Training: A Naive Bayes classifier was trained on the preprocessed data using a pipeline that included CountVectorizer, TfidfTransformer, and MultinomialNB. This pipeline helped transform text data into numerical vectors and applied term frequency-inverse document frequency (TF-IDF) weighting before training the classifier.

Model Evaluation: The trained model was evaluated using the testing set to measure its performance in accurately predicting the category of news articles.

Metrics Chosen:

The performance and accuracy of the model were evaluated using the following metric:

Accuracy: Accuracy represents the ratio of correctly predicted instances to the total number of instances in the dataset. It was chosen as the primary metric to measure the overall performance of the classification model.

Results:

The developed text classification model achieved an accuracy of approximately 71% on the testing set. This indicates that the model correctly predicted the category of news articles with an accuracy of 71%.

Approach:

(Title Generation)

The approach involved several key steps:

Data Preprocessing: The dataset was preprocessed to remove duplicates, handle missing values, and clean the text data. This included tokenization, removing stop words and non-alphanumeric characters, and converting words to their base forms.

Model Architecture: The Seq2Seq model architecture consisted of an encoder and a decoder. The encoder processed the input sequence (news article content) and generated a context vector, which was then fed into the decoder. The decoder generated the summary sequence based on the context vector.

Training: The model was trained using the preprocessed dataset. The input sequences (news article content) were fed into the encoder, while the target sequences (news headlines) were fed into the decoder. The model was trained to

minimize the loss function and maximize the accuracy of generating the correct summaries.

Evolution: The trained model was evaluated using the testing set to measure its performance in accurately predicting the category of news articles.

Metrics Chosen:

The accuracy metric chosen for evaluating the Seq2Seq model for text summarization is word-level accuracy, which measures the percentage of correctly predicted words in the generated summaries compared to the ground truth summaries. This metric provides insight into how accurately the model captures the essence of the input text and generates relevant summaries.

Results:

The developed Seq2Seq model achieved an accuracy of 80.10% in generating summaries for news articles. This indicates that the model performed reasonably well in generating concise and accurate summaries based on the input content.