**Application of NLP and Deep Learning Methods:**

By using Natural Language Processing libraries, we have applied different operations on the pre-processed data with the main purpose of processing the raw text data into a format that can be used as input for machine learning models. Here are the characteristics and purpose of each step in the code:

1. **Prepare the data for training:**

This step involved loading and cleaning the raw data. The raw data is usually in the form of a dataset or a corpus of documents, which needs to be cleaned and preprocessed to remove any noise, such as HTML tags, punctuation, and stop words. The cleaned data is then stored in a panda DataFrame.

Text

Description automatically generated

2. **Split the data into training and testing sets:**

This step involves dividing the data into two separate sets: training data and testing data. The training data is used to train the machine learning models, while the testing data is used to evaluate the performance of the models. This is done to ensure that the models generalize well to new, unseen data.



3. **Tokenize the text data:**

Tokenization is the process of converting text into a sequence of tokens or words. In this step, the text data is tokenized using the Tokenizer class from the Keras API. The tokenizer maps each word in the text data to a unique integer, and creates a word index dictionary that can be used to convert the text data into sequences of integers. The num\_words parameter specifies the maximum number of words to keep in the vocabulary, while the oov\_token parameter represents any out-of-vocabulary words that were not seen during training.

Graphical user interface, text, application, email

Description automatically generated

4. **Convert the text data to sequences:**

This step involves converting the text data into sequences of integers using the word index dictionary created in the previous step. The resulting sequences are then used as input for machine learning models, which typically require numerical data. The sequences are also padded or truncated to ensure that they are of a fixed length, which is required by some models.

Graphical user interface, text

Description automatically generated

Overall, this code is an essential part of the machine learning workflow for text classification tasks, and it helps to ensure that the text data is properly preprocessed and formatted for use in machine learning models.

**Training the Model and Execution:**

We have applied **Long Short-Term Memory (LSTM) neural network architecture** in Keras, for the task of binary text classification to detect fraudulent descriptions. In the Keras code, the LSTM model is defined using the Sequential model API, which allows the user to stack layers on top of each other in a linear fashion. The input layer is followed by an Embedding layer, which maps each word in the input sequence to a dense vector representation.

Finally, the output layer is a dense layer with a single neuron and a sigmoid activation function, which produces a binary output indicating whether the input is fraudulent or not.

Table

Description automatically generated

**Visualize the LSTM Model Training Validations:**

The plot in the left side represents the LSTM Model Accuracy on both test and train results, while the plot on the right represents the Model Loss for both Test and Train results.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

**Model Evaluation:**

The evaluation metrics provided in the output are precision, recall, and F1-score, which are common measures used to evaluate the performance of a binary classification model. Precision measures the proportion of true positive predictions out of all positive predictions, while recall measures the proportion of true positives out of all actual positive cases. The F1-score is the harmonic mean of precision and recall and provides a balanced measure of the model's performance. The macro and weighted averages of these metrics are also provided, which give a summary of the overall performance of the model.

Chart

Description automatically generated

**Confusion Matrix:** Below is the confusion matrix plotted for LSTM model using Keras.

Chart

Description automatically generated