The code begins by importing necessary Python libraries such as os, pandas, numpy, and seaborn. It then mounts Google Drive using the os library to access a CSV file containing the dataset for analysis.

Next, the code reads in the CSV file using the pandas library's read\_csv() method and assigns it to a variable. It then drops any duplicate rows based on the "question" column using the pandas drop\_duplicates() method.

To prepare the data for analysis, the "is\_correct" column is encoded as 0 or 1 using LabelEncoder from the sklearn.preprocessing library. This is done so that the machine learning model can work with numerical data.

The distribution of the "is\_correct" column is then visualized using a pie chart created with the matplotlib library. This provides a quick overview of the data and allows for insights into the proportion of correct and incorrect answers in the dataset.

Next, the script preprocesses the text in the "question" column. This is done by first tokenizing the text using the nltk library's word\_tokenize() method. The script then removes stop words and punctuation using the nltk library's stopwords and punctuation modules. The text is further preprocessed by applying stemming using the nltk library's PorterStemmer module. The preprocessed text is then stored in a new column named "processed\_question".

The distribution of the length of text in the "question" column is then visualized using histograms and scatter plots created with the seaborn library. This provides insights into the distribution of text length and allows for identifying any patterns or trends.

Finally, I have build a machine learning model using LSTM with TensorFlow and Keras. This model is designed to classify the questions as either correct or incorrect based on the preprocessed text. The model is then displayed on a web application using Streamlit.

Overall, this script demonstrates how to perform data analysis and preprocessing of text data using NLTK, as well as building and training a machine learning model for text classification using TensorFlow and Keras. It also shows how to visualize the data using various Python libraries, such as seaborn and matplotlib, and how to display the machine learning model on a web application using Streamlit.