We first mounts the Google Drive using the drive.mount() function provided by Colab. The IMDB dataset is located in the aclImdb folder in Google Drive. The script then imports necessary libraries and initializes some variables.

The dataset is read and stored in two lists: texts and labels. The texts list contains the movie reviews, and the labels list contains the sentiment labels (0 for negative, 1 for positive). The data is split into training and validation sets using the train\_test\_split() function from the scikit-learn library.

To convert the text data into numerical data, we use tokenization. This involves the utilization of the Tokenizer class provided by the Keras library. Firstly, we fit the Tokenizer to our text data creating a dictionary that maps words to integers which is stored as a vocabulary. Then, using this vocabulary and texts\_to\_sequences() method of the Tokenizer class, we can obtain sequences of integers to represent each sentence in our textual dataset.

Lastly, Keras's pad\_sequences() method is utilized to uniformly set all sentence sequences to equal length. Pre-trained GloVe embeddings are then loaded from Google Drive for further processing. The first step involves loading embeddings into a dictionary format where word tokens serve as keys and corresponding embedded feature vectors are their respective values. Subsequently, an embedding matrix makes use of these embedded feature vectors to create word representations specific to content found within the IMDB dataset.

These newly crafted word representations initiate weight calculation for subsequent neural network processing realized through implementation of Tensorflow's Keras Sequential Model. Specifically, hyperparameter configurations for this model entail an embeding unit followed by two fully connected layers assigned ReLu and sigmoidal activation functions respectively. Having constructed the model, we apply the binary cross-entropy loss function as well as the RMSprop optimizer.

We subsequently train and evaluate the network on respective training and validation sets using Keras' fit() method. The trained model is saved to disk under the name pre\_trained\_glove\_model.h5.

Through evaluation of our model's performance during validation, we analyzed the results pertaining to two types of word embeddings- pretrained word embeddings and embedding layers. The overall validation accuracy achieved for our model was recorded at 54%. Upon further examination conducting this study, it was observed that if pretrained word embedding is utilized over the traditional embedding layer, there exists relatively higher precision rate scores available as evident through its measurement at around 56%, whereas use of a regular embedding layer showed slightly lesser success records thus measuring around only approximately 54%.Overall, the script preprocesses the IMDB dataset and trains a neural network model for sentiment analysis using pre-trained GloVe embeddings.