

1. Load the IMDB dataset using pandas
2. For the “Reviews” column, pre-process the reviews by removing html tags, punctuations and numbers, multiple spaces
3. Convert the text to sequences. You may need the following steps

```
tokenizer = Tokenizer(num_words=5000) #Define a Keras tokenizer; from  
keras.preprocessing.text import Tokenizer
```

```
tokenizer.fit_on_texts(X_train) # Fit the tokenizer on the text
```

```
X_train = tokenizer.texts_to_sequences(X_train) # Convert the text to  
sequences
```

```
X_test = tokenizer.texts_to_sequences(X_test)
```

2. Classify each review into positive and negative “sentiment” categories. You may consider positive class as 1 and negative as 0. Use batch-size 128, optimizer - adam, learning rate - anything, validation split - 0.2, test data split - 0.2, epochs - anything, early_stopping - 10
The model should contain following layers

2.1 -> A **trainable** Embedding layer with embedding size 100

2.2 -> A Dense layer on the embedding layer of output size 128

2.3 -> A sigmoid layer for final classification

3. Evaluate on the test data and print accuracy.

4. Print the model summary and model image

5. Use both Functional and Sequential API of Keras.

6. **Optional (No extra credit):** Add attention mechanism before step 2.2. The attention mechanism should follow this. Say, the input matrix is M of shape (batch, timestep, embedding_size), we apply a kernel i.e., weight matrix W of shape (batch, embedding_size, 1) on A and we get $A = WM$ of shape (batch, timestep, 1), we apply a tanh layer on A and get $A2 = \tanh(WM)$, we obtain attention scores by applying a softmax activation in the last axes i.e., $\text{attention_scores} = \text{softmax}(A2, \text{axes} = 2)$ of shape (batch, timestep, 1)

6.1 -> Think why we apply tanh on A to get $A2$

6.2 -> Plot the attention weight for each word. See below for reference.

