Steps for the email classification task:

Step 1: Data Analysis

Counts the number of all emails on each topic and visualizes this information in a pie chart.

Step 2: Data Cleaning

I did the following steps for data cleaning:

1. Eliminate URLs

2. Remove punctuations

3. Remove English stop words like "and, is, a, on, etc." (define in data_fields())

4. Convert all the words to lower case (define in data_fields())

5. Tokenize the datapoints' subjects and body

6. Convert every word to its stem

7. Remove data points with empty subjects and empty body

Step 3: Feature Extraction

In this step, we build vocabulary and initialize the words with the pre-trained embeddings, i.e., Glove_100d. We then convert the vocabularies into integer sequences (indexes).

Step 4: Training and Validating

In the training phase, we set the model for the training phase. We calculate the loss, backpropagate it and compute the gradients. Then, we update the weights.

In the evaluation phase, we set the model on the evaluation phase and save the model if it performed better than the previous time.

Step 5: Testing

In this step, we retrieve the saved classification model, and then for all test samples, we do the prediction and calculate the performance metrics including accuracy, precision, and F1_Score on the test data.

Points:

1. I used Adam for the optimizer because of its state-of-the-art performance

2. CrossEntropyLoss() is used for the loss function because we are dealing with a multi-class classification problem, also I used Softmax for the LSTM activation function.

3. Bidirectional LSTM is used as the deep learning model

4. I did a hyper-parameter fine-tuning as much as possible regarding the time constraint. More advanced methods for parameter fine-tuning could be applied, like the cyclic learning rate approach, etc. The best parameter setting I found is as follows:

```
embedding_dim = 100

num_hidden_nodes = 25

num_classes = 4

n_layers = 2

bidirection = True

N_EPOCHS = 10

batch_size = 25

l_rate = 0.001
```

for which I got around 94% of accuracy, 95% of f1-score, and 94% of precision.

For the learning rate, I fine-tuned over [0.001, 0.005, 0.008, 0.01, 0.1] values.

5. For the stopping condition of the DL training, because of time constraints, I considered a fixed number of epochs (N_EPOCHS). Another option is to train until the maximum number of epochs is reached or the validation loss of two consecutive epochs is less than the loss threshold.

6. We use the email body for the email classification task unless the body is empty. In that case, we use the email subject for the text classification purpose.