**PART 1**

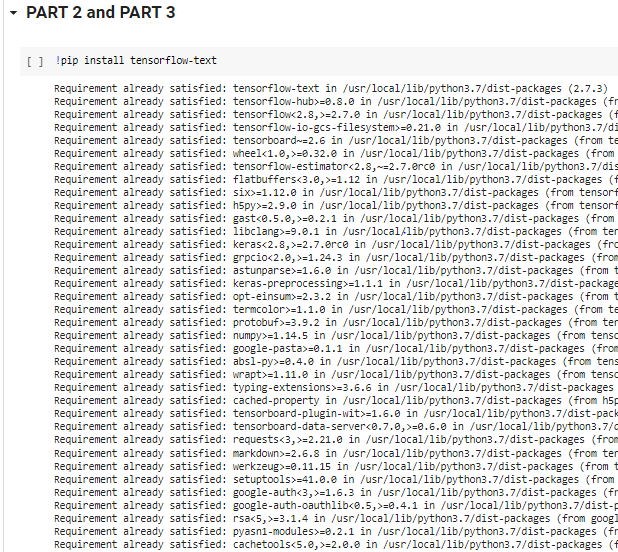
**Choose a Text Classification dataset which is easily downloaded into the Jupyter Notebook. One example is the imdb reviews data set used in the Tensor Flow RNN tutorial.**

We will use the following text classification dataset and is available on Kaggle and the Link is given below as well

**https://www.kaggle.com/c/nlp-getting-started/data?select=test.csv**

**PART 2 & 3**

**In the first step, I will install the tensorflow-text library**

****

**Now I will import the required libraries**

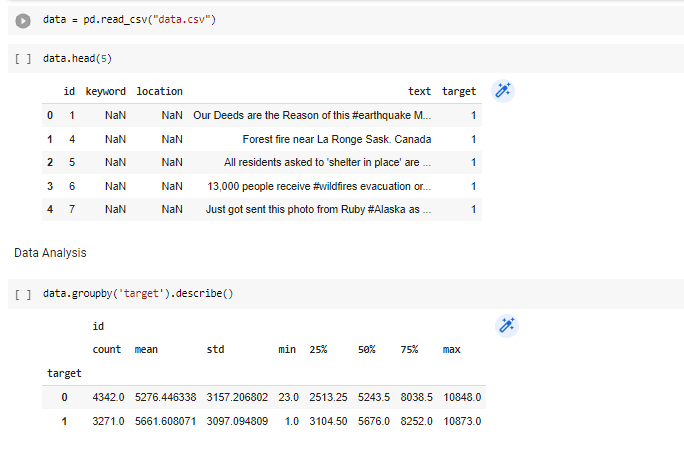
**import tensorflow as tf**

**import tensorflow\_hub as hub**

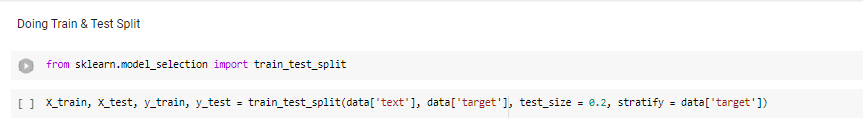
**import tensorflow\_text as text**

**import pandas as pd**

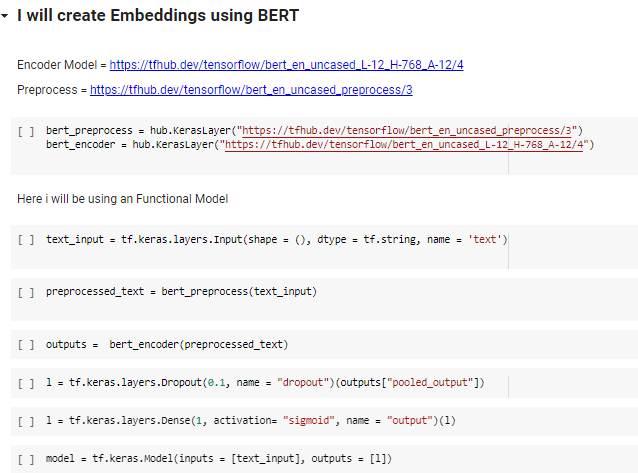
**Load the Dataset**

****

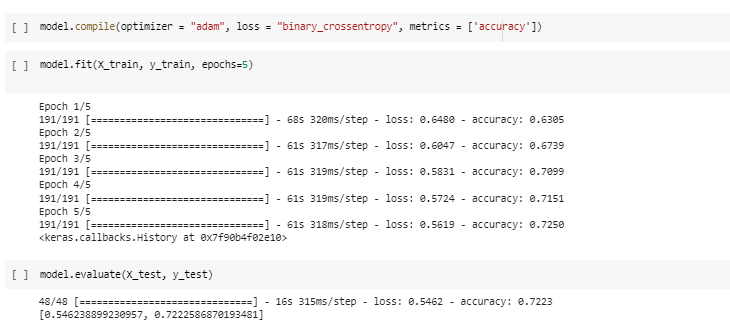
**Now I will be doing Train, Test Split**

****

**I will create embedding using BERT and will create a classification model**

****

**Now we will compile the model**

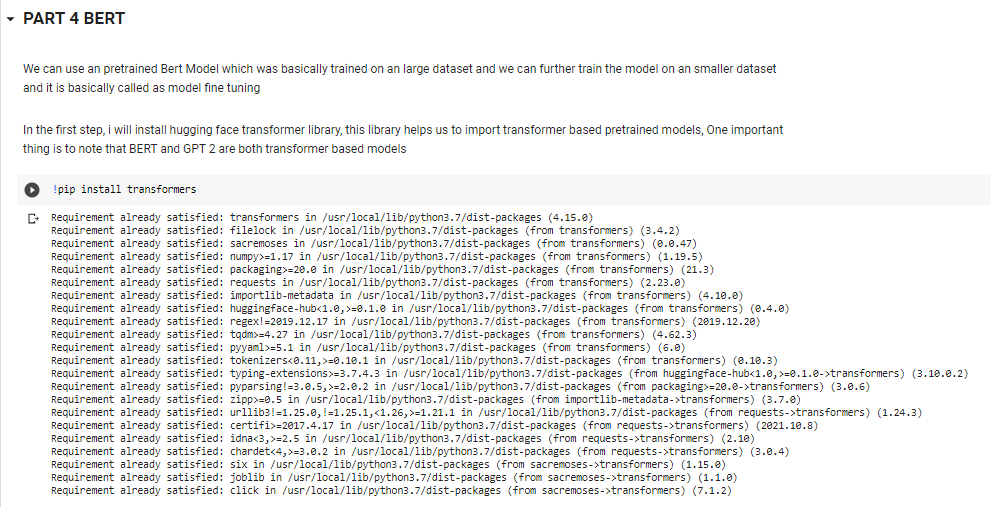
****

**PART 4**

**Fine Tune a pretrained BERT Model on the same dataset by extracting the embeddings from the output of the pre trained model and add your own layer at the top of the classification model**

**We can use an pretrained Bert Model which was basically trained on an large dataset and we can further train the model on an smaller dataset and it is basically called as model fine tuning**

**In the first step, i will install hugging face transformer library, this library helps us to import transformer based pretrained models, One important thing is to note that BERT and GPT 2 are both transformer based models**



Now, in the next step, I will install all the required libraries.

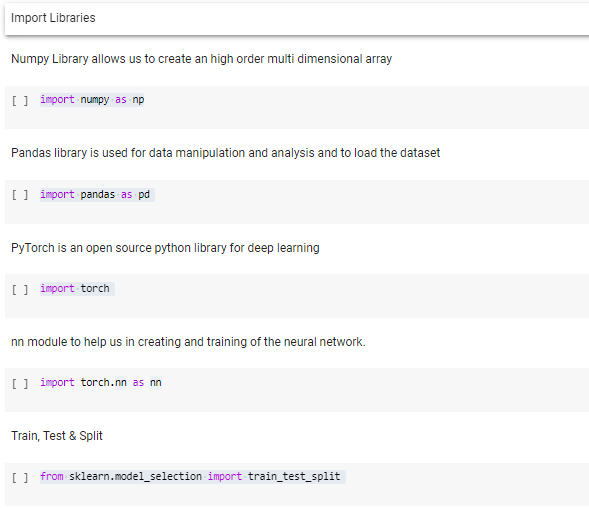
import numpy as np

import pandas as pd

import torch

import torch.nn as nn

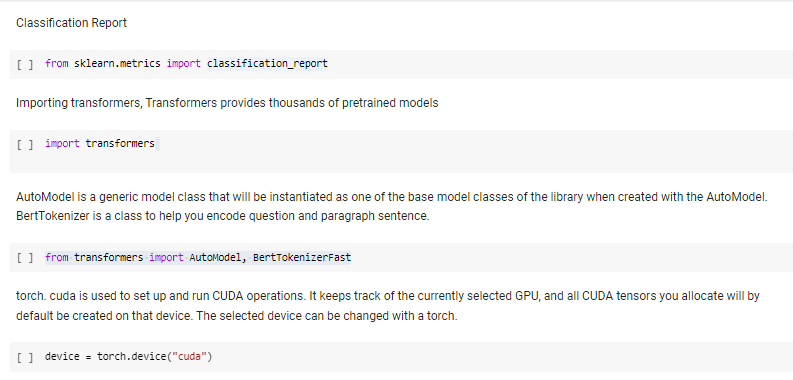
from sklearn.model\_selection import train\_test\_split



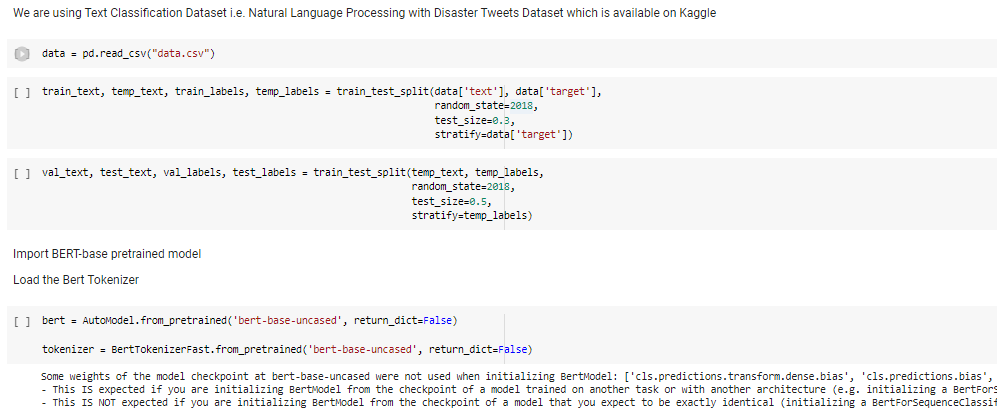
from sklearn.metrics import classification\_report

import transformers

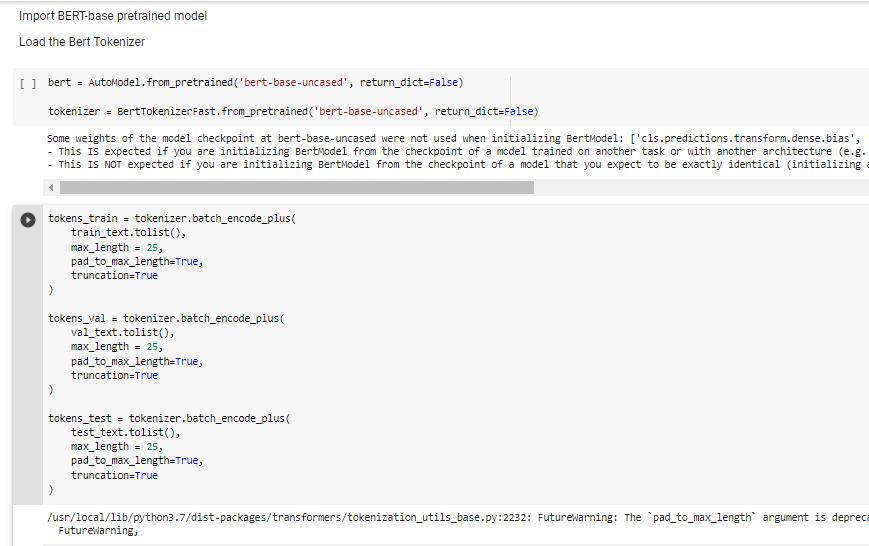
from transformers import AutoModel, BertTokenizerFast

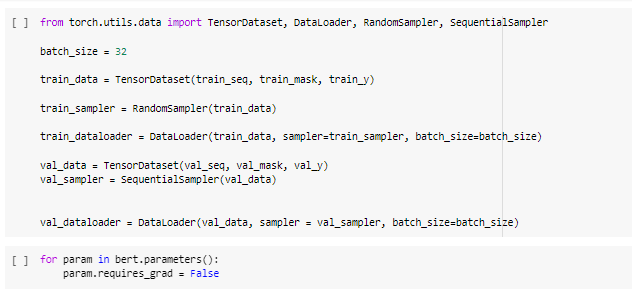


We will import the BERT-base model that has 110 million parameters. There is an even bigger BERT model called BERT-large that has 345 million parameters.



We can clearly see that most of the messages have a length of 25 words or less. Whereas the maximum length is 175. So, if we select 175 as the padding length then all the input sequences will have length 175 and most of the tokens in those sequences will be padding tokens which are not going to help the model learn anything useful and on top of that, it will make the training slower. Therefore, we will set 25 as the padding length





So, we have now converted the messages in train, validation, and test set to integer sequences of length 25 tokens each.

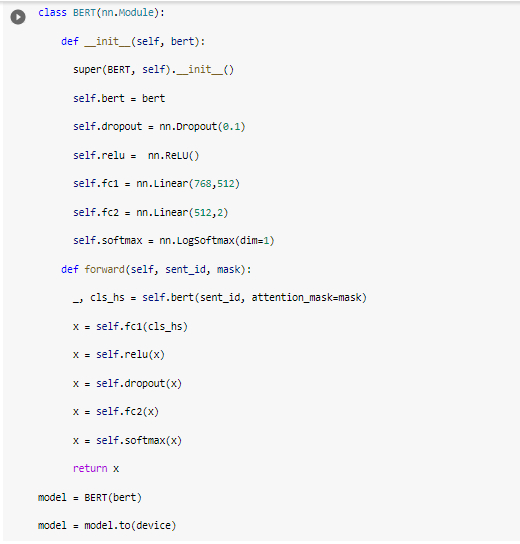
Next, we will convert the integer sequences to tensors.



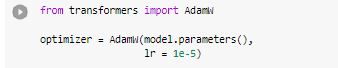
Now I  would freeze all the layers of the model before fine-tuning it.



Defining our model architecture.



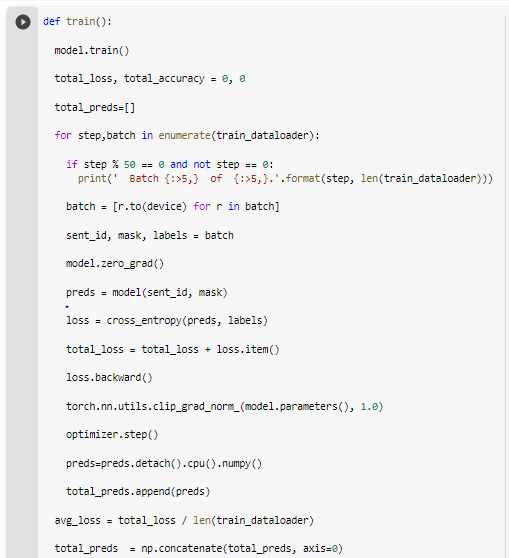
Now I will be using an AdamW optimizer which is basically an improved version of Adam optimizer

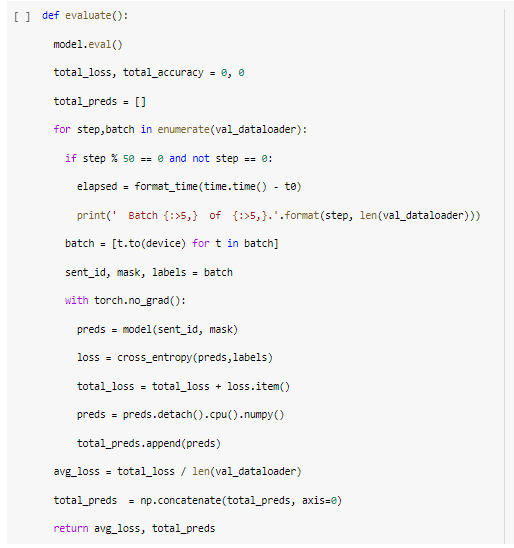


So, we will first class weights for the labels in the train set



In the next step, I will be creating an training and evaluation function.





Now we will be computing training loss and the validation loss

