Question 1

Dataset Loading

*1 Marks for loading each dataset properly*

In this part you are expected to load the dataset using [read\_csv from the Pandas Library](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html).

*1 Marks for concatenating them properly*

In this part you need to use the [concatenate function from pandas](https://pandas.pydata.org/docs/reference/api/pandas.concat.html) to concatenate two dataframes.

Word Embedding

*1 Marks for getting the right vector from w2v*

First create a list of tokens(to simplify we consider each word as a token)

Points to check: If the text is an empty string do not create a (input,output) pair

The template code already loads the w2v dictionary. It is a dictionary, where the word is the key and its corresponding value is the tensor for the word.

Points to check: If the word is present in w2v dictionary. If not, skip that word.

*1 Marks for doing mean pooling correctly*

Add all tensors corresponding to words in a review and divide them by the number of words.

Points to check: If none of the words are present in w2v the length will be zero. If you divide by 0 you will get nan(not a number) in python. So if the length is 0 you just need to take a zero vector as representation of the review.

Model Training

*1 Marks for defining the Neural Network correctly*

You need to define a neural network with multiple linear layers and ReLU connecting them. You can find Linear layer documentation [here](https://pytorch.org/docs/stable/generated/torch.nn.Linear.html). You can find ReLU documentation [here](https://pytorch.org/docs/stable/generated/torch.nn.ReLU.html). For connecting them you can use [nn.sequential](https://pytorch.org/docs/stable/generated/torch.nn.Sequential.html). Keep the model in self.seq object.

*1 Marks for using the model for predicting logits*

Pass your input through the model to predict logits. Logits are basically the probability of the input belonging to the various classes. For example a logit array of (0.1,0.2,0.3,0.35,0.05) means the input belongs to class 0 with 10% probability, class 1 with 20% probability and so on.

*1 Marks for getting class with highest probability*

The final class assigned to the input is the class with the highest probability. For example, if the logit array is (0.1,0.2,0.3,0.35,0.05), the input will be assigned class 3.

Point to note: The code processes (input, output) pairs in batches. So for a batch of 4 you will have (input0,output0), (input1,output1), (input2,output2), (input3,output3). So the logits will be in the form [[0.1,0.2,0.3,0.35,0.05],[0.4,0.5,0.01,0.02,0.07],[0.3,0.5,0.1,0.05,0.05],[0.2,0.2,0.2,0.3,0.1]]. You need to find the highest probability index in 1st dimension to get [3,1,1,3] as output

Evaluation

*1 Marks for the classification report and confusion matrix*

Print classification report using [classification report](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html) from sklearn library and print confusion matrix using [confusion matrix](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html) from sklearn.

*1 Marks for plotting*

Plot the confusion matrix using matplotlib plot and [scikitplot](https://scikit-plot.readthedocs.io/en/stable/metrics.html)

*1 Marks for analysing the result*

Write how you feel about the results and why?

Question 2

Adversarial Assessment

1 Marks for each sub part

Ethics Board Review

2 marks for each part