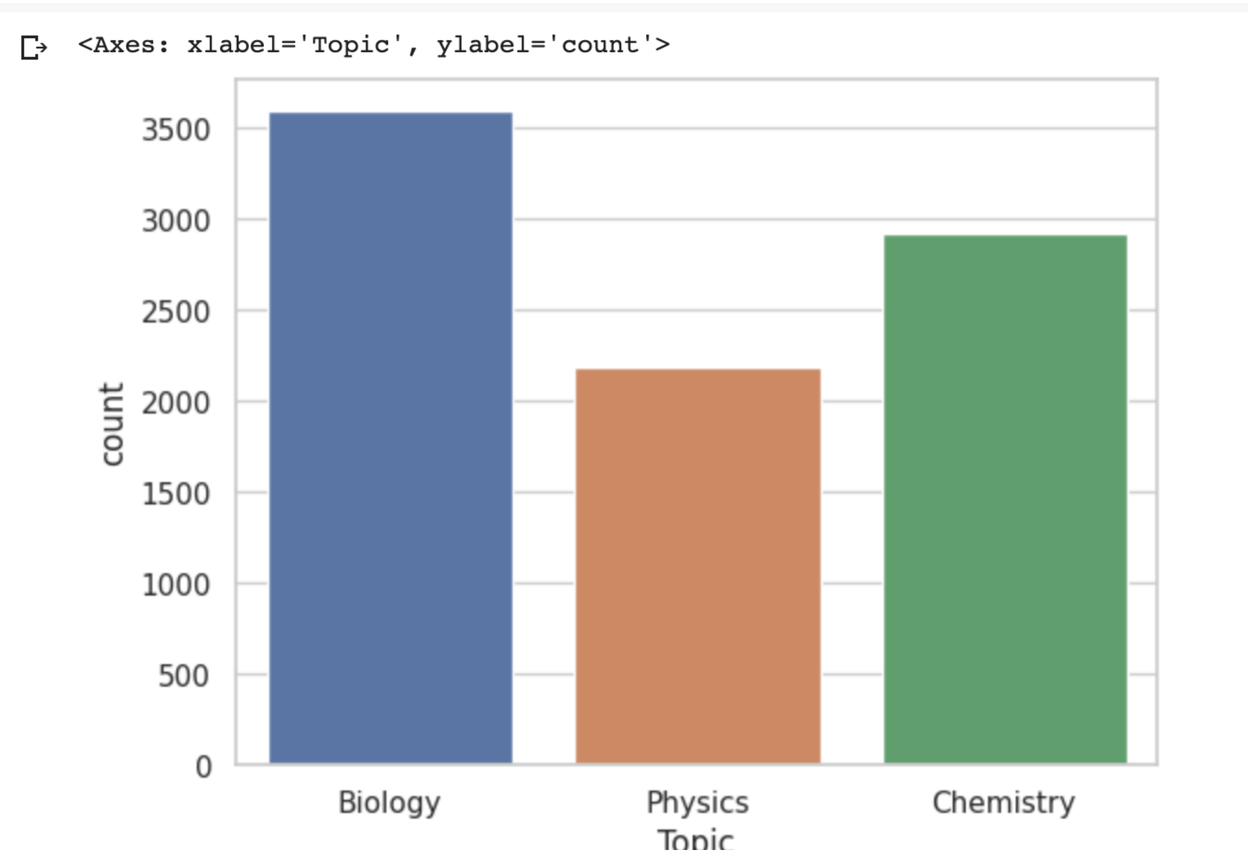
TEXT CLASSIFICATION WITH DEEP LEARNING MODELS

KERAS and TensorFlow are two API that are used to build deep learning models and has some prebuilt pipelines and features that aid in ML model creation. Here we start with a simple sequential model, then introduce the Deep Learning Long Short Term Memory RNN and Gradient Recurrent Unit (GRU) with embedding. We return to the original sequential model with and embedding Layer.

DATA- The Dataset was obtained on Kaggle, and be obtained through the following website:
https://www.kaggle.com/datasets/vivmankar/physics-vs-chemistry-vs-biology

This is a simple text classification model in which small education reviews are made on classes of three of the largest pure sciences: Biology, Chemistry and Biology. The data is already pre-split into training and testing split. There is slight deference in the categories, where biology has a slightly more reviews than chemistry, and chemistry has a few more than physics, (approximately 1.75 more biology than physics and approximate 1.2 more biology than chemistry). The distribution is seen below:



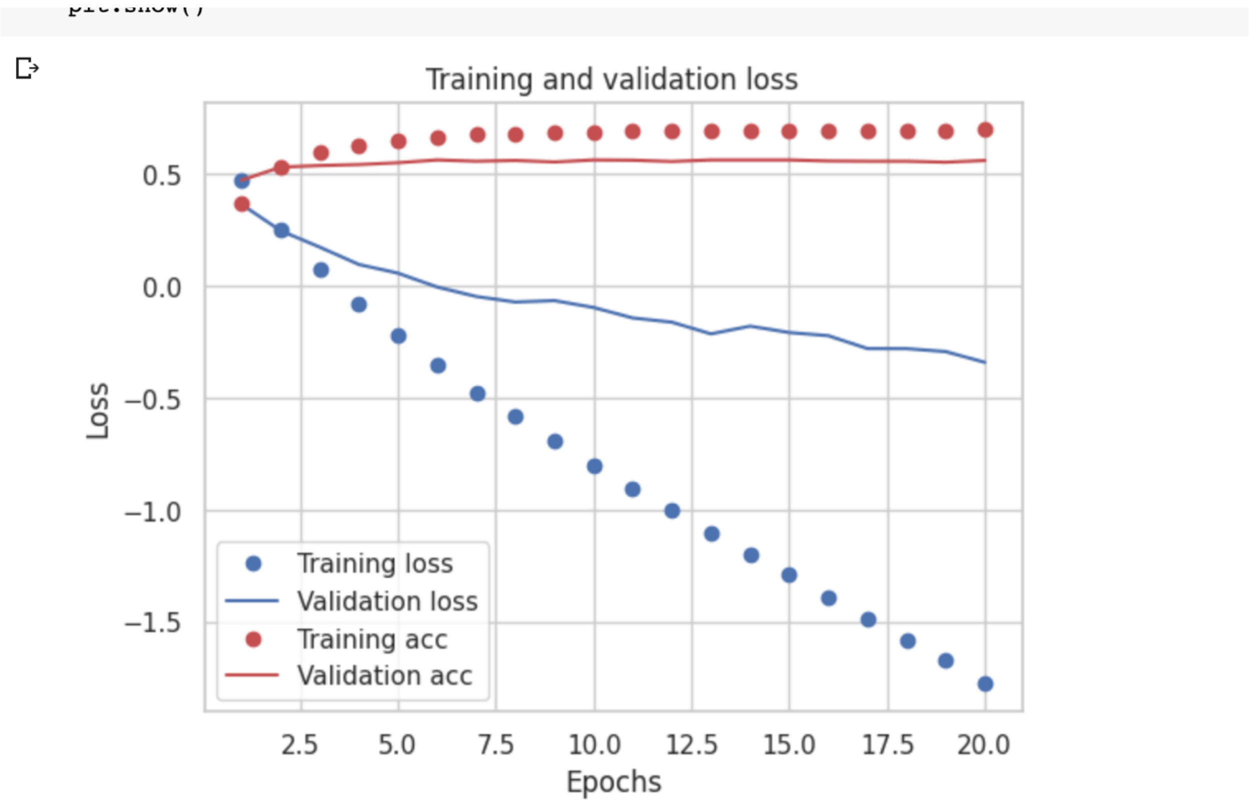
```
<Axes: xlabel='Topic', ylabel='count'>
```

SEQUENTIAL MODEL

A simple sequential model was created with bottlenecking dense layers with filters slowly reducing at each layer by half staring at 128. A dropout layer was added between the top three dense layers to avoid any possible overfitting. All layers accept the last two utilize ‘relu’ activation functions. The last two layers is a SoftMax and sigmoid activated dense layer. The data was passed through 20 epochs; however, it would seem that with 10 it would have been sufficient to obtained the same results.

RESULTS

This was the best results that I was able to obtain with the models with 56% precision. The loss in the training continually went down and accuracy went up with all epochs.



```
    plt.show()
```

KERAS LSTM/GRU

RESULTS

My initial thought that the LSTM would output better accuracy, however only 30 % accuracy was obtained. I switched over to a GRU, however accuracy only increased to 32% with the training group. However, with the testing group, 56% was obtained.

EMBEDDINGS

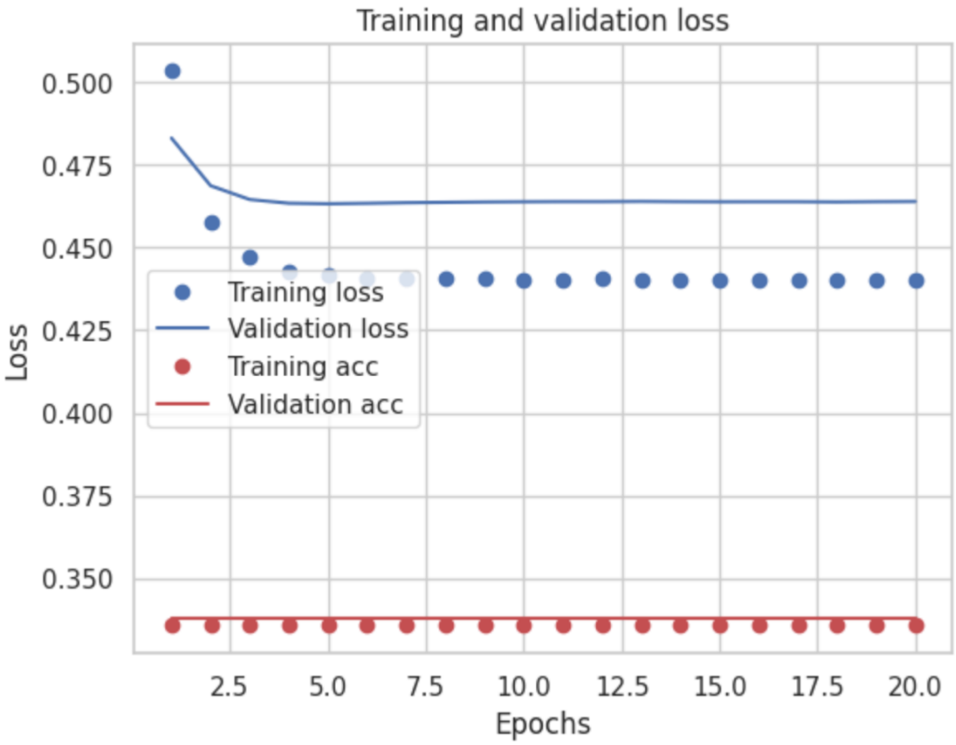
Finally, the use of embeddings were used on the original sequential with the hope that I might obtain better results. However, similar results to the LSTM and GRU were seen.

RESTULTS

Only 32 % accuracy was obtained with the training group. Although loss continually went down, the accuracy never seemed to go past the 32% threshold, please see the accuracy/loss plot below. However, 56% accuracy was seen with the testing group, which indicates that the model is random at best at guessing the correct category.

CONCLUSIONS

Clearly the parameters for the LSTM and GRU could use to be optimized, and although not shown several modifications were made however, I did not see much change. Even with the sequential model several modifications and changes to the layers were made, and this seemed to be the best results that were obtained. Had time permitted it was my plan to include an autoencoder, to see if better results could be seen.


Training and validation loss
0.500
0.475
0.450
0.425
0.400
0.375
0.350
Loss
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
Epochs
● Training loss
— Validation loss
● Training acc
— Validation acc