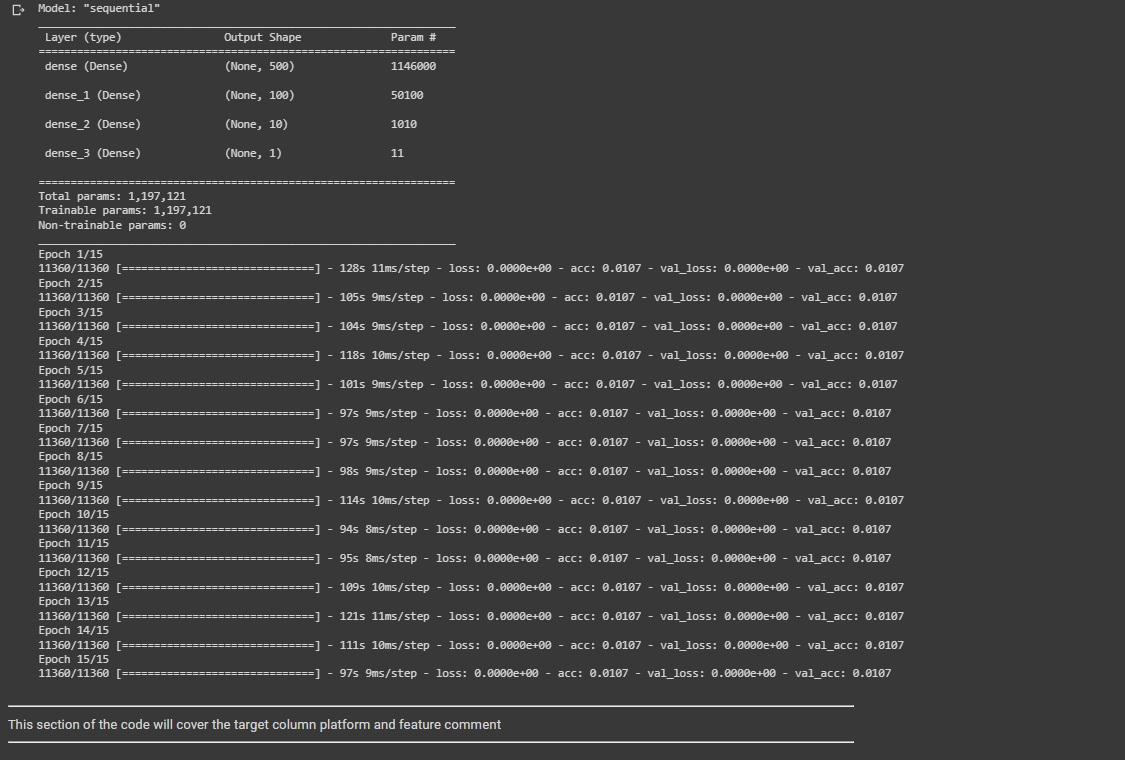
I have utilized Naïve Bayes as a standard comparison for the different architectures of deep learning models and their embeddings.

In my first run in with Keras, I was able to successfully compute a sequential model using activation function ‘softmax’. This first model is composed of four layers in descending order of: 500, 100, 10, 1. When it came time to construct the fit model, each iteration of epoch only obtained an accuracy score of 1%.

Text

Description automatically generated



On my second attempt, I decided on implementing the activation function ‘relu’ and ‘sigmoid’. The layering increased to eight total in the following order: 750, 500, 200, 100, 50, 10, 1. The compile network will be : loss=’binary\_crossentropy’. But in the model.fit, changed the epoch size to 10, increased batch\_size to 45, and used a validation split of 0.3 rather than validation data.

Graphical user interface, text

Description automatically generated

Moving on to Convolutional Neural Networks, I switched up the target and features from the previous method, since implementing for any correlation. When it comes to Naïve Bayes, the algorithm produced a low accuracy percentage of 7.4%. In comparison, I constructed a CNN model the first time it calculated the following:

Graphical user interface, text

Description automatically generated

Below is the 1st model I constructed.

Text

Description automatically generated

On the second construction of CNN, I added in more layers and increased the max features up to the limitations of Google Colab. As far as the model fit, I only switched to a validation data. Text

Description automatically generated

This new model produced the following accuracy:

Graphical user interface, text

Description automatically generated