TensorBoard Observations

In all the 3 models, I vectorized the categorical, textual, and numerical features and then applied the LSTM models as per the given architectures.

Model-1

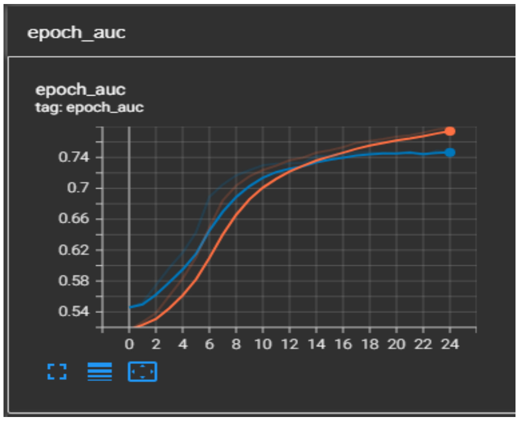
Here, I used ***OrdinalEncoder*** to vectorize the categorical features. I used ***Tokenizer*** to vectorize the essay and clean_subcategories features. I *normalized* the numerical vectors.

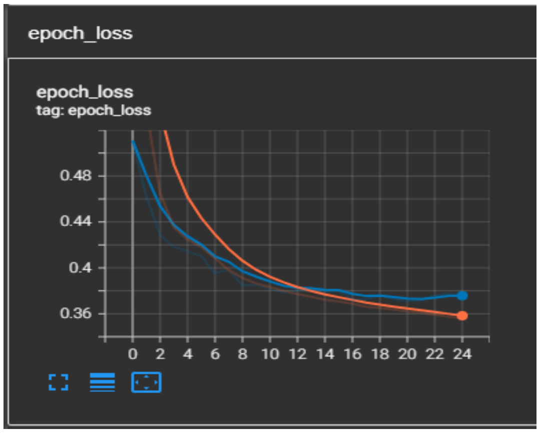
My model is set to train for 25 epochs. I have used the ***ModelCheckpoint*** and ***EarlyStopping*** callbacks. I used the Adam optimizer with 0.0001 learning rate. I used ***BatchNormalization*** after the latter Dense layers in the architecture.

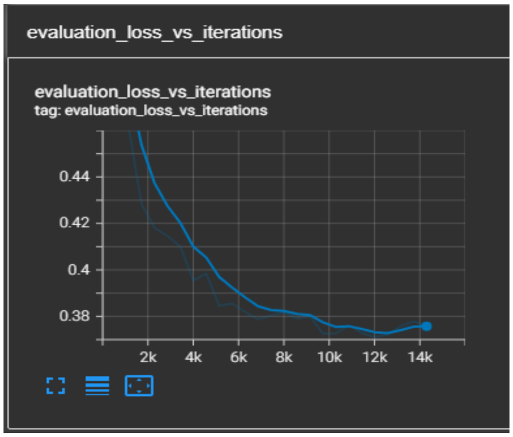
When we evaluate the model, the validation_auc is 0.7486. If I would have run the model for more epochs, I would have got a higher validation_auc.

Note: Red is the train curve and blue is the validation curve.

1. The auc is increasing substantially in every epoch and at the end of the 25th epoch, the validation_auc is 0.7486.
2. The epoch loss is steeply decreasing to 0.3621 after training for 20 epochs.
3. The evaluation_loss is decreasing significantly over iterations.







Model-2

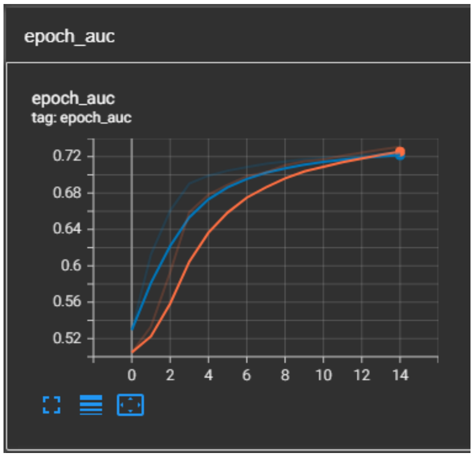
In this model, I fit the TF-IDF vectorizer on the train essay data. I got the IDF values for each word in the train data. The IDF scores range between 1 and 11. Data above 65%ile has an IDF value of nearly 11 and stopped increasing. I chose a minimum threshold of 1%ile and a maximum threshold of 65%ile as the desirable range of IDF scores. I removed the words having IDF scores lower than 1%ile and words having IDF scores more than 50%ile from both the train and test data.

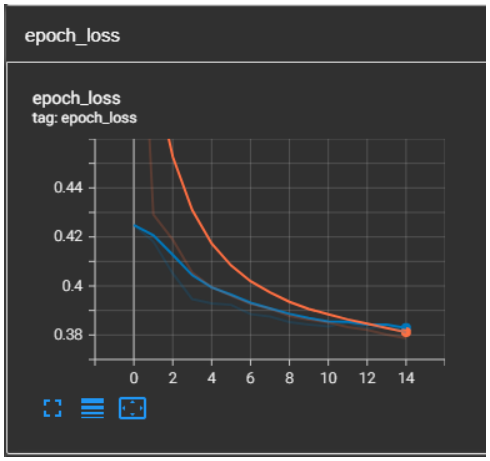
I used Tokenizer on the modified essay data and clean_subcategories. I used ***OrdinalEncoder*** to vectorize the rest of the categorical features. I *normalized* the numerical vectors.
My model is set to train for 25 epochs. I have used the ***ModelCheckpoint*** and ***EarlyStopping*** callbacks. I used the Adam optimizer with 0.0001 learning rate. I used ***BatchNormalization*** in the architecture to achieve the desirable validation_auc.

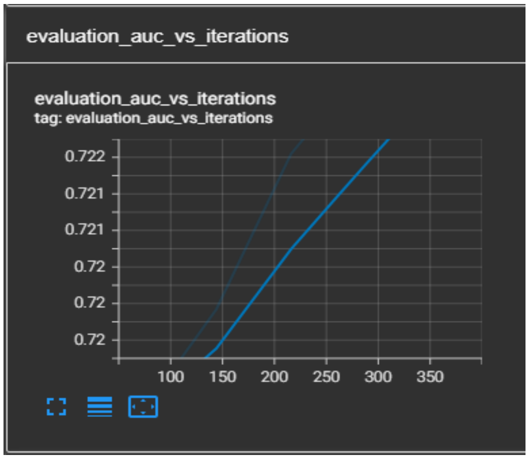
When we evaluate the model, the validation_auc is 0.7236. If I would have run the model for more epochs, I would have got a higher validation_auc.

Note: Red is the train curve and blue is the validation curve.

1. The epoch_auc is increasing and reaching 0.7307 at the end of training on the train data and 0.7236 on validation data over 25 epochs.
2. The loss is decreasing after training in the subsequent epochs.
3. The evaluation_auc is increasing in every iteration and it is reaching 0.7236.







Model-3

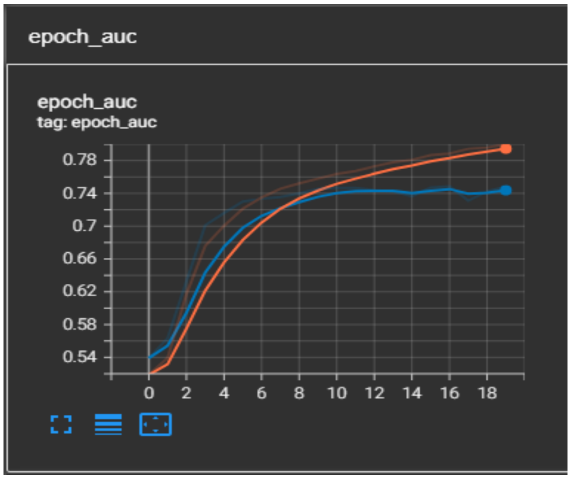
I used Keras Tokenizer on the essay and clean_subcategories features. I used ***OneHotEncoder*** to vectorize the rest of the categorical features. I normalized the numerical vectors.

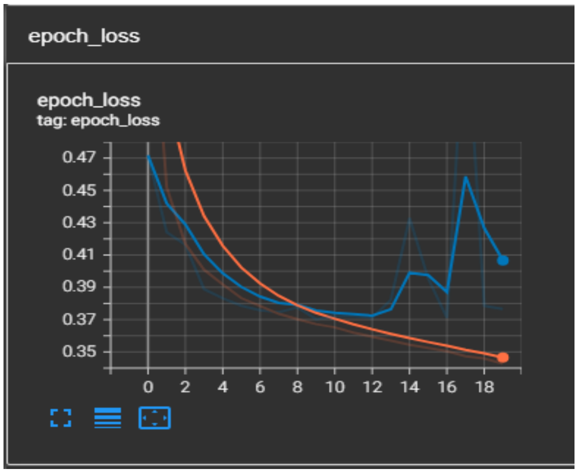
My model is set to train for 20 epochs. I have used the ***ModelCheckpoint*** and ***EarlyStopping*** callbacks. I used the Adam optimizer with 0.0001 learning rate. I used ***BatchNormalization*** after every Dense layer in the architecture.

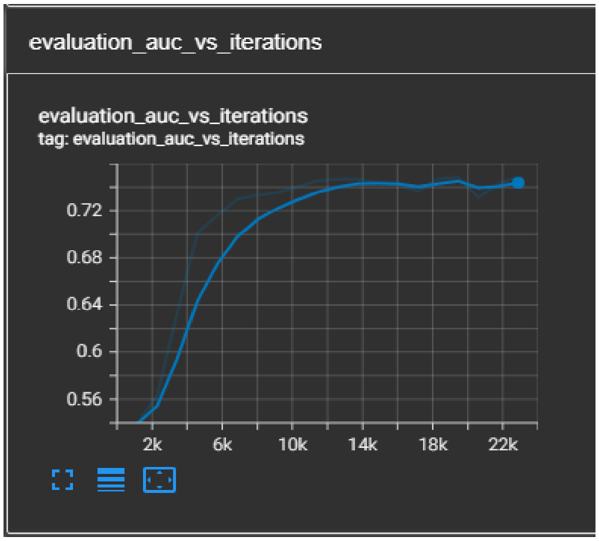
When we evaluate the model, the validation_auc is 0.7481. If I would have run the model for more epochs, I would have got a higher validation_auc.

Note: Red is the train curve and blue is the validation curve.

1. The auc is increasing substantially after every epoch where auc on train data is 0.7999 and validation_auc is 0.7481.
2. The loss is decreasing drastically and the loss on the validation set is highly erratic at the end of training for 20 epochs.
3. The evaluation_auc is increasing steeply after every iteration and is getting plateau.


epoch_auc
epoch_auc
tag: epoch_auc


epoch_loss
epoch_loss
tag: epoch_loss


evaluation_auc_vs_iterations
evaluation_auc_vs_iterations
tag: evaluation_auc_vs_iterations
2k
6k
10k
14k
18k
22k