

```
In [1]: # importing libraries

import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
from keras.layers import Input, Embedding, LSTM, Dropout, BatchNormaliz
ation, Dense, concatenate, Flatten, Conv1D, MaxPool1D, LeakyReLU, ELU,
SpatialDropout1D, MaxPooling1D, GlobalAveragePooling1D, GlobalMaxPoolin
g1D
from keras.preprocessing.text import Tokenizer, one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Model, load_model
from keras import regularizers
from keras.optimizers import *
from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard
, ReduceLRonPlateau
from sklearn.feature_extraction.text import TfidfVectorizer, CountVecto
rizer
from sklearn.metrics import roc_auc_score
import tensorflow as tf
from tensorboardcolab import *
import matplotlib.pyplot as plt
%matplotlib inline
import re
from tqdm import tqdm
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import pickle
```

Using TensorFlow backend.

```
In [2]: # https://medium.com/@rushic24/mounting-google-drive-in-google-colab-5e
cd1d3b735a
# https://towardsdatascience.com/3-ways-to-load-csv-files-into-colab-7c
14fcbdc92#targetText=To%20start%2C%20log%20into%20your,Colab%20has%20i
```

```
t%20installed%20already).
```

```
from google.colab import drive  
drive.mount('/content/drive')
```

Go to this URL in a browser: [https://accounts.google.com/o/oauth2/auth?client\\_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect\\_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aaob&scope=email%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fpeopleapi.readonly&response\\_type=code](https://accounts.google.com/o/oauth2/auth?client_id=947318989803-6bn6qk8qdgf4n4g3pfee6491hc0brc4i.apps.googleusercontent.com&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aaob&scope=email%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdocs.test%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fdrive.photos.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fpeopleapi.readonly&response_type=code)

Enter your authorization code:  
.....  
Mounted at /content/drive

In [3]: *# reading datasets*

```
project_data = pd.read_csv("/content/drive/My Drive/Data/preprocessed_data.csv")  
#pd.read_csv("preprocessed_data.csv")  
project_data.head()
```

Out[3]:

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_t
0	ca	mrs	grades_prek_2	53
1	ut	ms	grades_3_5	4

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_p
2	ca	mrs	grades_prek_2	10
3	ga	mrs	grades_prek_2	2
4	wa	mrs	grades_3_5	2

```
In [4]: print("Number of data points in train data", project_data.shape)
print('- '*50)
print("The attributes of data :", project_data.columns.values)
```

Number of data points in train data (109248, 9)

-----  
The attributes of data : ['school\_state' 'teacher\_prefix' 'project\_grade\_category'  
'teacher\_number\_of\_previously\_posted\_projects' 'project\_is\_approved'  
'clean\_categories' 'clean\_subcategories' 'essay' 'price']

```
In [5]: approved_project = project_data['project_is_approved'].values
project_data.drop(['project_is_approved'], axis=1, inplace=True)
project_data.head(1)
```

Out[5]:

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_t
0	ca	mrs	grades_prek_2	53

```
In [0]: # Data splitting

from sklearn.model_selection import train_test_split

# Splitting in train and test
X_train, X_test, y_train, y_test = train_test_split(project_data, approved_project, test_size=0.33, stratify=approved_project)
```

```
In [7]: #clearing the graph of tensorflow
tf.keras.backend.clear_session()

input_seq_total_text_data = Input(shape=(300,), name="input_seq_total_text_data")
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:541: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

```
In [8]: # https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/
```

```
# prepare tokenizer
text_tokenizer = Tokenizer()
text_tokenizer.fit_on_texts(X_train["essay"])
vocab_size = len(text_tokenizer.word_index) + 1
vocab_size
```

Out[8]: 48319

```
In [0]: # integer encode the data

encoded_essay_train = text_tokenizer.texts_to_sequences(X_train["essay"]
)
encoded_essay_test = text_tokenizer.texts_to_sequences(X_test["essay"])
```

```
In [10]: # Padding data

padded_text_train = pad_sequences(encoded_essay_train, maxlen=300, padding='post', truncating='post')
padded_text_test = pad_sequences(encoded_essay_test, maxlen=300, padding='post', truncating='post')

print(padded_text_train.shape)
print(padded_text_test.shape)

(73196, 300)
(36052, 300)
```

```
In [0]: f = open("/content/drive/My Drive/Data/glove_vectors", "rb")
glove_words = pickle.load(f)
```

```
In [12]: # create a weight matrix for words in training docs

embedding_matrix = np.zeros((vocab_size, 300))
for word, i in text_tokenizer.word_index.items():

    embedding_vector = glove_words.get(word)
    if embedding_vector is not None:
```

```
embedding_matrix[i] = embedding_vector  
  
print(embedding_matrix.shape)  
  
(48319, 300)
```

```
In [13]: Emb_Txt_Data = Embedding(vocab_size, 300, weights = [embedding_matrix],  
input_length = 300, trainable=False)
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:66: The name tf.get\_default\_graph is deprecated. Please use tf.compat.v1.get\_default\_graph instead.

```
In [14]: Emb_Text_Data = (Emb_Txt_Data)(input_seq_total_text_data)
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:4432: The name tf.random\_uniform is deprecated. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:190: The name tf.get\_default\_session is deprecated. Please use tf.compat.v1.get\_default\_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:197: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:203: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:207: The name tf.global\_variables is deprecated. Please use tf.compat.v1.global\_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:216: The name tf.is\_variable\_initialized is

deprecated. Please use `tf.compat.v1.is_variable_initialized` instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:223: The name `tf.variables_initializer` is deprecated. Please use `tf.compat.v1.variables_initializer` instead.

```
In [15]: Emb_Text_Data
```

```
Out[15]: <tf.Tensor 'embedding_1/embedding_lookup/Identity:0' shape=(?, 300, 300) dtype=float32>
```

```
In [16]: lstm = LSTM(64, recurrent_dropout=0.5, kernel_regularizer = regularizers.l2(0.001), return_sequences = True)(Emb_Text_Data)
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:148: The name `tf.placeholder_with_default` is deprecated. Please use `tf.compat.v1.placeholder_with_default` instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:3733: calling `dropout` (from `tensorflow.python.ops.nn_ops`) with `keep_prob` is deprecated and will be removed in a future version.

Instructions for updating:

Please use ``rate`` instead of ``keep_prob``. Rate should be set to ``rate = 1 - keep_prob``.

```
In [17]: lstm
```

```
Out[17]: <tf.Tensor 'lstm_1/transpose_1:0' shape=(?, ?, 64) dtype=float32>
```

```
In [0]: flatten = Flatten()(lstm)
```

```
In [19]: flatten
```

```
Out[19]: <tf.Tensor 'flatten_1/Reshape:0' shape=(?, ?) dtype=float32>
```

```
In [0]: # processing for school state
```

```
input_school_state = Input(shape=(50,), name="input_school_state")
```

```
In [21]: unique_school_state = X_train["school_state"].nunique()
print(unique_school_state)

Emb_State_Data = Embedding(unique_school_state, 300, input_length = 50)
(input_school_state)

51
```

```
In [0]: flatten_1 = Flatten()(Emb_State_Data)
```

```
In [0]: encoded_school_state_train = [one_hot(d, unique_school_state) for d in
X_train['school_state']]
encoded_school_state_test = [one_hot(d, unique_school_state) for d in X
_test['school_state']]
```

```
In [24]: padded_school_state_train = pad_sequences(encoded_school_state_train, m
axlen=50, padding='post')
padded_school_state_test = pad_sequences(encoded_school_state_test, max
len=50, padding='post')

print(padded_school_state_train.shape)
print(padded_school_state_test.shape)

(73196, 50)
(36052, 50)
```

```
In [25]: # processing for project grade category

input_project_grade_category = Input(shape=(50,), name="input_project_gr
ade_category")

unique_project_grade = X_train["project_grade_category"].nunique()
print(unique_project_grade)

Emb_PGC_Data = Embedding(unique_project_grade, 300, input_length = 50)(
```



```
input_project_grade_category)
flatten_2 = Flatten()(Emb_PGC_Data)
```

4

```
In [26]: encoded_project_grade_train = [one_hot(d, unique_project_grade) for d in
n X_train['project_grade_category']]
encoded_project_grade_test = [one_hot(d, unique_project_grade) for d in
X_test['project_grade_category']]

padded_project_grade_train = pad_sequences(encoded_project_grade_train,
maxlen=50, padding='post')
padded_project_grade_test = pad_sequences(encoded_project_grade_test, m
axlen=50, padding='post')

print(padded_project_grade_train.shape)
print(padded_project_grade_test.shape)

(73196, 50)
(36052, 50)
```

```
In [27]: # processing for clean categories

input_clean_categories = Input(shape=(50,), name="input_clean_categorie
s")

unique_clean_categories = X_train["clean_categories"].nunique()
print(unique_clean_categories)

Emb_clean_categories_Data = Embedding(unique_clean_categories, 300, inp
ut_length = 50)(input_clean_categories)
flatten_3 = Flatten()(Emb_clean_categories_Data)

51
```

```
In [28]: encoded_clean_categories_train = [one_hot(d, unique_clean_categories) f
or d in X_train['clean_categories']]
encoded_clean_categories_test = [one_hot(d, unique_clean_categories) fo
r d in X_test['clean_categories']]
```

```

padded_clean_categories_train = pad_sequences(encoded_clean_categories_train, maxlen=50, padding='post')
padded_clean_categories_test = pad_sequences(encoded_clean_categories_test, maxlen=50, padding='post')

print(padded_clean_categories_train.shape)
print(padded_clean_categories_test.shape)

(73196, 50)
(36052, 50)

```

```

In [29]: # processing for clean subcategories

input_clean_subcategories = Input(shape=(50,), name="input_clean_subcategories")

unique_clean_subcategories = X_train["clean_subcategories"].nunique()
print(unique_clean_subcategories)

Emb_clean_subcategories_Data = Embedding(unique_clean_subcategories, 300, input_length = 50)(input_clean_subcategories)
flatten_4 = Flatten()(Emb_clean_subcategories_Data)

394

```

```

In [30]: encoded_clean_subcategories_train = [one_hot(d, unique_clean_subcategories) for d in X_train['clean_subcategories']]
encoded_clean_subcategories_test = [one_hot(d, unique_clean_subcategories) for d in X_test['clean_subcategories']]

padded_clean_subcategories_train = pad_sequences(encoded_clean_subcategories_train, maxlen=50, padding='post')
padded_clean_subcategories_test = pad_sequences(encoded_clean_subcategories_test, maxlen=50, padding='post')

print(padded_clean_subcategories_train.shape)
print(padded_clean_subcategories_test.shape)

```

```
(73196, 50)
(36052, 50)
```

```
In [31]: # processing for teacher prefix

input_teacher_prefix = Input(shape=(50,),name="input_teacher_prefix")

unique_teacher_prefix = X_train["teacher_prefix"].nunique()
print(unique_teacher_prefix)

Emb_teacher_prefix_Data = Embedding(unique_teacher_prefix, 300, input_length = 50)(input_teacher_prefix)
flatten_5 = Flatten()(Emb_teacher_prefix_Data)
```

```
5
```

```
In [32]: encoded_teacher_prefix_train = [one_hot(d, unique_teacher_prefix) for d
      in X_train['teacher_prefix']]
encoded_teacher_prefix_test = [one_hot(d, unique_teacher_prefix) for d
      in X_test['teacher_prefix']]

padded_teacher_prefix_train = pad_sequences(encoded_teacher_prefix_train, maxlen=50, padding='post')
padded_teacher_prefix_test = pad_sequences(encoded_teacher_prefix_test, maxlen=50, padding='post')

print(padded_teacher_prefix_train.shape)
print(padded_teacher_prefix_test.shape)
```

```
(73196, 50)
(36052, 50)
```

```
In [33]: # Processing numerical features

input_numerical_data = Input(shape=(2,),name="input_numerical_data")

Dense_for_rem_input = Dense(units=32,activation='relu',kernel_initializer='he_normal',name="Dense_for_rem_input")(input_numerical_data)
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:4479: The name tf.truncated\_normal is deprecated. Please use tf.random.truncated\_normal instead.

```
In [0]: teacher_number_of_previously_posted_projects_train = X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1)
price_train = X_train['price'].values.reshape(-1, 1)

teacher_number_of_previously_posted_projects_test = X_test['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1)
price_test = X_test['price'].values.reshape(-1, 1)

concat_numerical_train = np.concatenate((teacher_number_of_previously_posted_projects_train, price_train), axis=1)
concat_numerical_test = np.concatenate((teacher_number_of_previously_posted_projects_test, price_test), axis=1)
```

```
In [0]: concatenate = concatenate(inputs=[flatten, flatten_1, flatten_2, flatten_3, flatten_4, flatten_5, input_numerical_data], name="concatenate")
```

```
In [36]: Dense_layer1_after_concat = Dense(128, activation="relu", kernel_initializer="he_normal", kernel_regularizer=regularizers.l2(0.001))(concatenate)

dropout = Dropout(0.5)(Dense_layer1_after_concat)

Dense_layer2_after_concat = Dense(64, activation="relu", kernel_initializer="he_normal", kernel_regularizer=regularizers.l2(0.001))(dropout)

dropout_1 = Dropout(0.5)(Dense_layer2_after_concat)

Dense_layer3_after_concat = Dense(32, activation='relu', kernel_initializer='he_normal', kernel_regularizer=regularizers.l2(0.001))(dropout_1)

Dense_layer4_after_concat = Dense(16, activation='relu', kernel_initializer='he_normal', kernel_regularizer=regularizers.l2(0.001))(Dense_layer3_after_concat)
```

```

output_layer_to_classify_with_softmax = Dense(2,activation='softmax',kernel_initializer="he_normal",name="output")(Dense_layer4_after_concat)

model_1 = Model(inputs=[input_seq_total_text_data,input_school_state,input_project_grade_category,input_clean_categories,input_clean_subcategories,input_teacher_prefix,input_numerical_data],outputs=[output_layer_to_classify_with_softmax])

model_1.summary()

```

Model: "model\_1"

Layer (type)	Output Shape	Param #	Connected to
=====			
input_seq_total_text_data (InputLayer)	(None, 300)	0	
=====			
embedding_1 (Embedding)	(None, 300, 300)	14495700	input_seq_total_text_data[0][0]
=====			
input_school_state (InputLayer)	(None, 50)	0	
=====			
input_project_grade_category (InputLayer)	(None, 50)	0	
=====			
input_clean_categories (InputLayer)	(None, 50)	0	
=====			
input_clean_subcategories (InputLayer)	(None, 50)	0	

input_teacher_prefix (InputLaye	(None, 50)	0	
lstm_1 (LSTM)	(None, 300, 64)	93440	embedd
ing_1[0][0]			
embedding_2 (Embedding)	(None, 50, 300)	15300	input_
school_state[0][0]			
embedding_3 (Embedding)	(None, 50, 300)	1200	input_
project_grade_category[0][0]			
embedding_4 (Embedding)	(None, 50, 300)	15300	input_
clean_categories[0][0]			
embedding_5 (Embedding)	(None, 50, 300)	118200	input_
clean_subcategories[0][0]			
embedding_6 (Embedding)	(None, 50, 300)	1500	input_
teacher_prefix[0][0]			
flatten_1 (Flatten)	(None, 19200)	0	lstm_1
[0][0]			
flatten_2 (Flatten)	(None, 15000)	0	embedd
ing_2[0][0]			
flatten_3 (Flatten)	(None, 15000)	0	embedd
ing_3[0][0]			

ing\_5[0][0]

flatten_4 (Flatten) ing_4[0][0]	(None, 15000)	0	embedd
------------------------------------	---------------	---	--------

flatten_5 (Flatten) ing_5[0][0]	(None, 15000)	0	embedd
------------------------------------	---------------	---	--------

flatten_6 (Flatten) ing_6[0][0]	(None, 15000)	0	embedd
------------------------------------	---------------	---	--------

input_numerical_data (InputLaye	(None, 2)	0	
---------------------------------	-----------	---	--

concatenate (Concatenate) n_1[0][0]	(None, 94202)	0	flatte
n_2[0][0]			flatte
n_3[0][0]			flatte
n_4[0][0]			flatte
n_5[0][0]			flatte
n_6[0][0]			flatte
numerical_data[0][0]			input_

dense_1 (Dense) enate[0][0]	(None, 128)	12057984	concat
--------------------------------	-------------	----------	--------

dropout_1 (Dropout)	(None, 128)	0	dense
---------------------	-------------	---	-------

dropout_1 (Dropout) 1[0][0]	(None, 128)	0	dense_1[0][0]
<hr/>			
dense_2 (Dense) t_1[0][0]	(None, 64)	8256	dropout_1[0][0]
<hr/>			
dropout_2 (Dropout) 2[0][0]	(None, 64)	0	dense_2[0][0]
<hr/>			
dense_3 (Dense) t_2[0][0]	(None, 32)	2080	dropout_2[0][0]
<hr/>			
dense_4 (Dense) 3[0][0]	(None, 16)	528	dense_3[0][0]
<hr/>			
output (Dense) 4[0][0]	(None, 2)	34	dense_4[0][0]
<hr/>			
=====			
Total params: 26,809,522			
Trainable params: 12,313,822			
Non-trainable params: 14,495,700			
<hr/>			

In [37]: `# https://github.com/taomanwai/tensorboardcolab/blob/master/README.md`

```
tbc=TensorBoardColab()
```

Wait for 8 seconds...

TensorBoard link:

<https://6291a79d.ngrok.io>



```
In [0]: # https://machinelearningmastery.com/check-point-deep-learning-models-keras/
# https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/
# https://medium.com/singlestone/keras-callbacks-monitor-and-improve-your-deep-learning-205a8a27e91c
# https://www.tensorflow.org/tensorboard/get\_started
# https://keras.rstudio.com/reference/callback\_tensorboard.html
# https://colab.research.google.com/drive/1afN2SALDooZIHbBGmWZMT6cZ8ccVElWk#scrollTo=4pxUfiLhbS4Y&forceEdit=true&sandboxMode=true

#tensorboard_model_1 = TensorBoard(log_dir='./log', histogram_freq=1, write_graph=True, write_grads=True, batch_size=512, write_images=True)

#callbacks_1 = [tensorboard_model_1]
```

```
In [0]: model_1_train_data = [padded_text_train,padded_school_state_train,padded_project_grade_train,padded_clean_categories_train,padded_clean_subcategories_train,padded_teacher_prefix_train,concat_numerical_train]

model_1_test_data = [padded_text_test,padded_school_state_test,padded_project_grade_test,padded_clean_categories_test,padded_clean_subcategories_test,padded_teacher_prefix_test,concat_numerical_test]
```

```
In [0]: from keras.utils import np_utils

Y_train = np_utils.to_categorical(y_train, 2)
Y_test = np_utils.to_categorical(y_test, 2)
```

```
In [0]: # https://stackoverflow.com/questions/41032551/how-to-compute-receiving-operating-characteristic-roc-and-auc-in-keras

def auroc(y_true, y_pred):
    return tf.py_func(roc_auc_score, (y_true, y_pred), tf.double)
```

```
In [42]: model_1.compile(optimizer='adam', loss='categorical_crossentropy', metrics=[auroc])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:3576: The name tf.log is deprecated. Please use tf.math.log instead.
```

```
WARNING:tensorflow:From <ipython-input-41-2303c0155baf>:3: py_func (from tensorflow.python.ops.script_ops) is deprecated and will be removed in a future version.
```

Instructions for updating:

tf.py\_func is deprecated in TF V2. Instead, there are two options available in V2.

- tf.py\_function takes a python function which manipulates tf eager tensors instead of numpy arrays. It's easy to convert a tf eager tensor to

- an ndarray (just call tensor.numpy()) but having access to eager tensors

- means `tf.py\_function`s can use accelerators such as GPUs as well as

- being differentiable using a gradient tape.

- tf.numpy\_function maintains the semantics of the deprecated tf.py\_func

- (it is not differentiable, and manipulates numpy arrays). It drops the

- stateful argument making all functions stateful.

```
In [43]: history = model_1.fit(model_1_train_data, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(model_1_test_data, Y_test), callbacks=[TensorBoardColabCallback(tbc)])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/core/python/ops/math_grad.py:1424: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
```

Instructions for updating:

Use tf.where in 2.0, which has the same broadcast rule as np.where

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:1033: The name tf.assign_add is deprecated.
```

Please use `tf.compat.v1.assign_add` instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow\_backend.py:1020: The name `tf.assign` is deprecated. Please use `tf.compat.v1.assign` instead.

Train on 73196 samples, validate on 36052 samples

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard/core.py:49: The name `tf.summary.FileWriter` is deprecated. Please use `tf.compat.v1.summary.FileWriter` instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1122: The name `tf.summary.merge_all` is deprecated. Please use `tf.compat.v1.summary.merge_all` instead.

Epoch 1/20

73196/73196 [=====] - 115s 2ms/step - loss: 0.7977 - auroc: 0.6039 - val\_loss: 0.6384 - val\_auroc: 0.7193

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorboard/core.py:51: The name `tf.Summary` is deprecated. Please use `tf.compat.v1.Summary` instead.

Epoch 2/20

73196/73196 [=====] - 113s 2ms/step - loss: 0.5497 - auroc: 0.6927 - val\_loss: 0.5446 - val\_auroc: 0.7346

Epoch 3/20

73196/73196 [=====] - 113s 2ms/step - loss: 0.5031 - auroc: 0.7088 - val\_loss: 0.5081 - val\_auroc: 0.7440

Epoch 4/20

73196/73196 [=====] - 115s 2ms/step - loss: 0.4750 - auroc: 0.7186 - val\_loss: 0.4785 - val\_auroc: 0.7344

Epoch 5/20

73196/73196 [=====] - 116s 2ms/step - loss: 0.4534 - auroc: 0.7271 - val\_loss: 0.4611 - val\_auroc: 0.7533

Epoch 6/20

73196/73196 [=====] - 112s 2ms/step - loss: 0.4377 - auroc: 0.7346 - val\_loss: 0.4467 - val\_auroc: 0.7550

Epoch 7/20

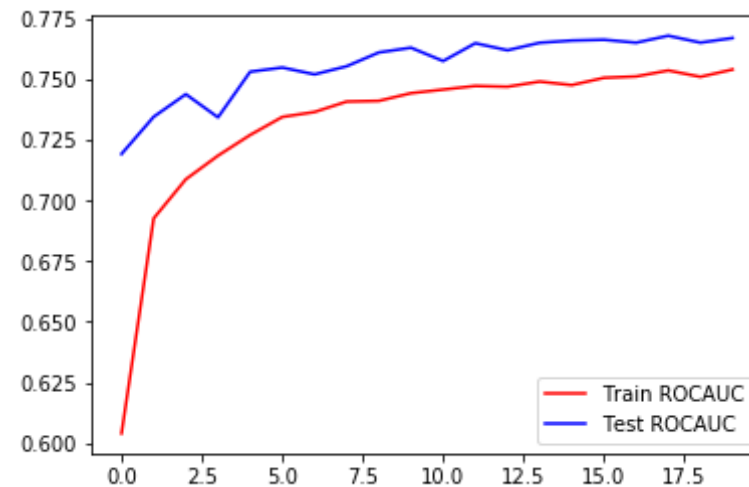
73196/73196 [=====] - 109s 1ms/step - loss: 0.

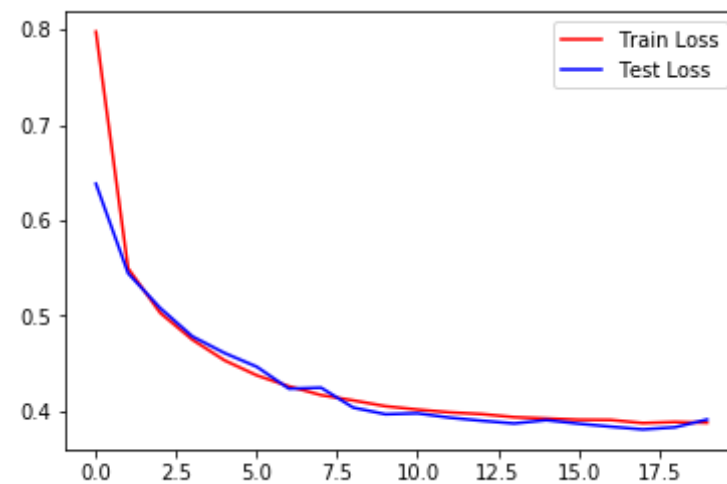
```
4262 - auroc: 0.7366 - val_loss: 0.4237 - val_auroc: 0.7522
Epoch 8/20
73196/73196 [=====] - 110s 1ms/step - loss: 0.
4170 - auroc: 0.7409 - val_loss: 0.4248 - val_auroc: 0.7555
Epoch 9/20
73196/73196 [=====] - 116s 2ms/step - loss: 0.
4114 - auroc: 0.7412 - val_loss: 0.4039 - val_auroc: 0.7612
Epoch 10/20
73196/73196 [=====] - 116s 2ms/step - loss: 0.
4054 - auroc: 0.7444 - val_loss: 0.3970 - val_auroc: 0.7632
Epoch 11/20
73196/73196 [=====] - 116s 2ms/step - loss: 0.
4018 - auroc: 0.7459 - val_loss: 0.3979 - val_auroc: 0.7577
Epoch 12/20
73196/73196 [=====] - 115s 2ms/step - loss: 0.
3989 - auroc: 0.7474 - val_loss: 0.3933 - val_auroc: 0.7650
Epoch 13/20
73196/73196 [=====] - 115s 2ms/step - loss: 0.
3971 - auroc: 0.7471 - val_loss: 0.3901 - val_auroc: 0.7621
Epoch 14/20
73196/73196 [=====] - 114s 2ms/step - loss: 0.
3939 - auroc: 0.7492 - val_loss: 0.3873 - val_auroc: 0.7652
Epoch 15/20
73196/73196 [=====] - 113s 2ms/step - loss: 0.
3925 - auroc: 0.7478 - val_loss: 0.3908 - val_auroc: 0.7662
Epoch 16/20
73196/73196 [=====] - 112s 2ms/step - loss: 0.
3912 - auroc: 0.7508 - val_loss: 0.3871 - val_auroc: 0.7665
Epoch 17/20
73196/73196 [=====] - 111s 2ms/step - loss: 0.
3910 - auroc: 0.7513 - val_loss: 0.3839 - val_auroc: 0.7653
Epoch 18/20
73196/73196 [=====] - 114s 2ms/step - loss: 0.
3877 - auroc: 0.7538 - val_loss: 0.3810 - val_auroc: 0.7681
Epoch 19/20
73196/73196 [=====] - 115s 2ms/step - loss: 0.
3888 - auroc: 0.7512 - val_loss: 0.3833 - val_auroc: 0.7653
Epoch 20/20
```

```
73196/73196 [=====] - 116s 2ms/step - loss: 0.3880 - auroc: 0.7542 - val_loss: 0.3914 - val_auroc: 0.7672
```

```
In [44]: plt.plot(history.history['auroc'], 'r')
plt.plot(history.history['val_auroc'], 'b')
plt.legend({'Train ROCAUC': 'r', 'Test ROCAUC': 'b'})
plt.show()

plt.plot(history.history['loss'], 'r')
plt.plot(history.history['val_loss'], 'b')
plt.legend({'Train Loss': 'r', 'Test Loss': 'b'})
plt.show()
```





Model ROCAUC Value is 0.7672.