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.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the `%tensorflow_version 1.x` magic: more info.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [2]:
```
# https://medium.com/@rushic24/mounting-google-drive-in-google-colab-5ecd1d3b735a
# https://towardsdatascience.com/3-ways-to-load-csv-files-into-colab-7c14fcbdcb92#targetText=To%20start%2C%20log%20into%20your,Colab%20has%20it%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%3Aoob&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_p
0	ca	mrs	grades_prek_2	53

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

	school_state	teacher_prefix	project_grade_category	teacher_number_of_previously_p
1	ut	ms	grades_3_5	4
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)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
Number of data points in train data (109248, 9)
---------------------------------------------------
The attributes of data : ['school_state' 'teacher_prefix' 'project_grad
e_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_p
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, appro
ved_project, test_size=0.33, stratify=approved_project)
```

In [7]:
```
tfidf_essay_vectorizer = TfidfVectorizer()
tfidf_essay_vectorizer.fit(X_train["essay"])
#X_Train_essay_tfidf = tfidf_essay_vectorizer.transform(X_train["essa
y"])
```

Out[7]:
```
TfidfVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.float64'>, encoding='utf-8'
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
                dtype=<class 'numpy.float64'>, encoding='utf-8',
                input='content', lowercase=True, max_df=1.0, max_featur
es=None,
                min_df=1, ngram_range=(1, 1), norm='l2', preprocessor=N
one,
                smooth_idf=True, stop_words=None, strip_accents=None,
                sublinear_tf=False, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, use_idf=True, vocabulary=None)
```

In [8]:
```
tfidf_essay_vectorizer.idf_
```

Out[8]:
```
array([ 7.15105371,  5.92992128, 11.50776253, ..., 11.50776253,
       11.50776253, 11.10229743])
```

In [0]:
```
# we are converting a dictionary with word as a key, and the idf as a v
alue

dictionary = dict(zip(tfidf_essay_vectorizer.get_feature_names(), list(
tfidf_essay_vectorizer.idf_)))
tfidf_words = set(tfidf_essay_vectorizer.get_feature_names())
```

In [0]:
```
tfidf_dictinary_df = pd.DataFrame(list(dictionary.items()), columns=['W
ord', 'Value'])
```

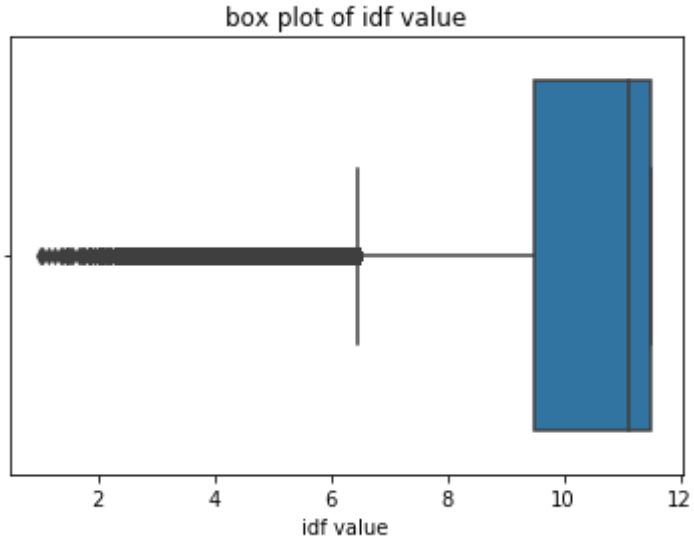
In [0]:
```
tfidf_dictinary_df = tfidf_dictinary_df.sort_values(by ='Value' )
```

In [12]:
```
print(tfidf_dictinary_df["Value"].min())
print(tfidf_dictinary_df["Value"].max())
```

```
1.0074459895205732
11.50776253494305
```

In [13]:
```
sns.boxplot(x="Value", data=tfidf_dictinary_df )
plt.xlabel("idf value")
plt.title("box plot of idf value")
plt.show()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



In [14]:
```
print("\nQuantiles:")
print(np.percentile(tfidf_dictinary_df['Value'],np.arange(0, 100, 10)))
```

```
Quantiles:
[ 1.00744599  7.48241084  8.94281318  9.89832462 10.5914718  11.1022974
3
 11.10229743 11.50776253 11.50776253 11.50776253]
```

In [15]:
```
final_tfidf = tfidf_dictinary_df[tfidf_dictinary_df["Value"] <=  np.per
centile(tfidf_dictinary_df['Value'],10)]
final_tfidf.shape
```

Out[15]:
```
(4848, 2)
```

In [16]:
```
#clearing the graph of tensorflow
tf.keras.backend.clear_session()

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

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:541: The name tf.placeholder is deprecated.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

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

In [0]:
```
final_word_list = final_tfidf["Word"].tolist()
```

In [18]:
```
# https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/

# prepare tokenizer
text_tokenizer = Tokenizer()
text_tokenizer.fit_on_texts(final_word_list)
vocab_size = len(text_tokenizer.word_index) + 1
vocab_size
```

Out[18]:
```
4849
```

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 [20]:
```
# 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)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

In [22]:
```
# 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)
```

```
(4849, 300)
```

In [23]:
```
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/ba
ckend/tensorflow_backend.py:66: The name tf.get_default_graph is deprec
ated. Please use tf.compat.v1.get_default_graph instead.
```

In [24]:
```
Emb_Text_Data = (Emb_Txt_Data)(input_seq_total_text_data)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:4432: The name tf.random_uniform is depreca
ted. Please use tf.random.uniform instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:190: The name tf.get_default_session is dep
recated. Please use tf.compat.v1.get_default_session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/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/ba
ckend/tensorflow_backend.py:203: The name tf.Session is deprecated. Ple
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ase use tf.compat.v1.Session instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:207: The name tf.global_variables is deprec
ated. Please use tf.compat.v1.global_variables instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/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/ba
ckend/tensorflow_backend.py:223: The name tf.variables_initializer is d
eprecated. Please use tf.compat.v1.variables_initializer instead.
```

In [25]:
```
Emb_Text_Data
```

Out[25]:
```
<tf.Tensor 'embedding_1/embedding_lookup/Identity:0' shape=(?, 300, 30
0) dtype=float32>
```

In [26]:
```
lstm = LSTM(64, recurrent_dropout=0.5, kernel_regularizer = regularizer
s.l2(0.001), return_sequences = True)(Emb_Text_Data)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:148: The name tf.placeholder_with_default i
s deprecated. Please use tf.compat.v1.placeholder_with_default instead.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:3733: calling dropout (from tensorflow.pyth
on.ops.nn_ops) with keep_prob is deprecated and will be removed in a fu
ture version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
```

In [27]:
```
lstm
```

Out[27]:
```
<tf.Tensor 'lstm_1/transpose_1:0' shape=(?, ?, 64) dtype=float32>
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

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

In [29]:
```
flatten
```

Out[29]:
```
<tf.Tensor 'flatten_1/Reshape:0' shape=(?, ?) dtype=float32>
```

In [0]:
```
# Vectorizing schoolstate on Train, Test

ssvectorizer = CountVectorizer()

# Fit only to train data
ssvectorizer.fit(X_train['school_state'].values)

# Transform to train,test and CV data
X_Train_ss_one_hot = ssvectorizer.transform(X_train['school_state'].val
ues)
X_Test_ss_one_hot = ssvectorizer.transform(X_test['school_state'].value
s)
```

In [0]:
```
# Vectorizing project grade category on Train, Test

pgcvectorizer = CountVectorizer()

# Fit only to train data
pgcvectorizer.fit(X_train['project_grade_category'].values)

# Transform to train,test and CV data
X_Train_pgc_one_hot = pgcvectorizer.transform(X_train['project_grade_ca
tegory'].values)
X_Test_pgc_one_hot = pgcvectorizer.transform(X_test['project_grade_cate
gory'].values)
```

In [0]:
```
# Vectorizing clean_categories on Train, Test

ccvectorizer = CountVectorizer()

# Fit only to train data
ccvectorizer.fit(X_train['clean_categories'].values)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Transform to train,test
X_Train_cc_one_hot = ccvectorizer.transform(X_train['clean_categories']
.values)
X_Test_cc_one_hot = ccvectorizer.transform(X_test['clean_categories'].v
alues)
```

In [0]:
```
# Vectorizing clean_subcategories on Train, Test

csvectorizer = CountVectorizer()

# Fit only to train data
csvectorizer.fit(X_train['clean_subcategories'].values)

# Transform to train,test
X_Train_cs_one_hot = ccvectorizer.transform(X_train['clean_subcategorie
s'].values)
X_Test_cs_one_hot = ccvectorizer.transform(X_test['clean_subcategories'
].values)
```

In [0]:
```
# Vectorizing teacher prefix on Train, Test

tpvectorizer = CountVectorizer()

# Fit only to train data
tpvectorizer.fit(X_train['teacher_prefix'].values)

# Transform to train,test
X_Train_tp_one_hot = ccvectorizer.transform(X_train['teacher_prefix'].v
alues)
X_Test_tp_one_hot = ccvectorizer.transform(X_test['teacher_prefix'].val
ues)
```

In [0]:
```
teacher_number_of_previously_posted_projects_train = X_train['teacher_n
umber_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_num
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ber_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_p
osted_projects_train,price_train),axis=1)
concat_numerical_test = np.concatenate((teacher_number_of_previously_po
sted_projects_test,price_test),axis=1)
```

In [0]:
```
from scipy.sparse import hstack

X_tr = hstack((X_Train_cc_one_hot,X_Train_cs_one_hot,X_Train_ss_one_hot
,X_Train_tp_one_hot,X_Train_pgc_one_hot,concat_numerical_train)).todens
e()
X_te = hstack((X_Test_cc_one_hot,X_Test_cs_one_hot,X_Test_ss_one_hot,X_
Test_tp_one_hot,X_Test_pgc_one_hot,concat_numerical_test)).todense()
```

In [0]:
```
X_tr_final = np.expand_dims(X_tr,2)
X_te_final = np.expand_dims(X_te,2)
```

In [38]:
```
print(X_tr_final.shape)
print(X_te_final.shape)
```

```
(73196, 84, 1)
(36052, 84, 1)
```

In [0]:
```
Other_than_text_data =  Input(shape=(84,1),name='Other_than_text_data')
```

In [40]:
```
Conv1D_1 = Conv1D(64, 3, activation='relu', kernel_initializer="he_norm
al")(Other_than_text_data)
Conv1D_2 = Conv1D(64, 3, activation='relu', kernel_initializer="he_norm
al")(Conv1D_1)
Conv1D_3 = Conv1D(32, 3, activation='relu', kernel_initializer="he_norm
al")(Conv1D_2)
Conv1D_4 = Conv1D(32, 3, activation='relu', kernel_initializer="he_norm
al")(Conv1D_3)
Conv1D_5 = Conv1D(16, 3, activation='relu', kernel_initializer="he_norm
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
al")(Conv1D_4)
flatten_1 = Flatten()(Conv1D_5)
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/ba
ckend/tensorflow_backend.py:4479: The name tf.truncated_normal is depre

cated. Please use tf.random.truncated_normal instead.
```

In [0]:
```
concatenate = concatenate(inputs=[flatten, flatten_1],name="concatenat
e")
```

In [42]:
```
Dense_layer1_after_concat = Dense(256,activation="relu", kernel_initial
izer="he_normal", kernel_regularizer=regularizers.l2(0.001))(concatenat
e)

dropout = Dropout(0.5)(Dense_layer1_after_concat)

Dense_layer2_after_concat = Dense(128,activation="relu", kernel_initial
izer="he_normal", kernel_regularizer=regularizers.l2(0.001))(dropout)

dropuout_1 = Dropout(0.3)(Dense_layer2_after_concat)

Dense_layer3_after_concat = Dense(64,activation='relu',kernel_initializ
er='he_normal', kernel_regularizer=regularizers.l2(0.001))(dropuout_1)

dropuout_2 = Dropout(0.7)(Dense_layer3_after_concat)

Dense_layer4_after_concat = Dense(32,activation='relu',kernel_initializ
er='he_normal', kernel_regularizer=regularizers.l2(0.001))(dropuout_2)

batchnormalization_1 = BatchNormalization()(Dense_layer4_after_concat)

output_layer_to_classify_with_softmax = Dense(2,activation='softmax',ke
rnel_initializer="he_normal",name="output")(batchnormalization_1)

model_1 = Model(inputs=[input_seq_total_text_data,Other_than_text_data
],outputs=[output_layer_to_classify_with_softmax])
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
model_1.summary()
```

```
WARNING:tensorflow:Large dropout rate: 0.7 (>0.5). In TensorFlow 2.x, d
ropout() uses dropout rate instead of keep_prob. Please ensure that thi
s is intended.
Model: "model_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   Connec
ted to
=================================================================
Other_than_text_data (InputLaye (None, 84, 1)             0

_________________________________________________________________
conv1d_1 (Conv1D)            (None, 82, 64)            256       Other_
than_text_data[0][0]
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 80, 64)            12352     conv1d
_1[0][0]
_________________________________________________________________
input_seq_total_text_data (Inpu (None, 300)               0

_________________________________________________________________
conv1d_3 (Conv1D)            (None, 78, 32)            6176      conv1d
_2[0][0]
_________________________________________________________________
embedding_1 (Embedding)      (None, 300, 300)          1454700   input_
seq_total_text_data[0][0]
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 76, 32)            3104      conv1d
_3[0][0]
_________________________________________________________________
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
_____
lstm_1 (LSTM)                (None, 300, 64)       93440       embedd
ing_1[0][0]
_____
_____
conv1d_5 (Conv1D)            (None, 74, 16)        1552        conv1d
_4[0][0]
_____
_____
flatten_1 (Flatten)          (None, 19200)         0           lstm_1
[0][0]
_____
_____
flatten_2 (Flatten)          (None, 1184)          0           conv1d
_5[0][0]
_____
_____
concatenate (Concatenate)    (None, 20384)         0           flatte
n_1[0][0]
                                                               flatte
n_2[0][0]
_____
_____
dense_1 (Dense)              (None, 256)           5218560     concat
enate[0][0]
_____
_____
dropout_1 (Dropout)          (None, 256)           0           dense_
1[0][0]
_____
_____
dense_2 (Dense)              (None, 128)           32896       dropou
t_1[0][0]
_____
_____
dropout_2 (Dropout)          (None, 128)           0           dense_
2[0][0]
_____
_____
dense_3 (Dense)              (None, 64)            8256        dropou
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
t_2[0][0]

__________________________________________________________________________
_______________________________
dropout_3 (Dropout)             (None, 64)           0           dense_
3[0][0]
__________________________________________________________________________
_______________________________
dense_4 (Dense)                 (None, 32)           2080        dropou
t_3[0][0]
__________________________________________________________________________
_______________________________
batch_normalization_1 (BatchNor (None, 32)           128         dense_
4[0][0]
__________________________________________________________________________
_______________________________
output (Dense)                  (None, 2)            66          batch_
normalization_1[0][0]
==========================================================================
============================
Total params: 6,833,566
Trainable params: 5,378,802
Non-trainable params: 1,454,764
__________________________________________________________________________
_______________________________
```

In [43]:
```
# https://github.com/taomanwai/tensorboardcolab/blob/master/README.md

tbc=TensorBoardColab()
```

```
Wait for 8 seconds...
TensorBoard link:
https://823175b0.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-yo
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
ur-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,X_tr_final]

model_1_test_data = [padded_text_test,X_te_final]
```

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 [48]:
```
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-47-2303c0155baf>:3: py_func (fro
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
m tensorflow.python.ops.script_ops) is deprecated and will be removed i
n 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 te
nsor to
    an ndarray (just call tensor.numpy()) but having access to eager te
nsors
    means `tf.py_function`s can use accelerators such as GPUs as well a
s
    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 [49]:
```
history = model_1.fit(model_1_train_data, Y_train, batch_size=512, epoc
hs=20, verbose=1, validation_data=(model_1_test_data, Y_test), callback
s=[TensorBoardColabCallback(tbc)])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorfl
ow_core/python/ops/math_grad.py:1424: where (from tensorflow.python.op
s.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/ba
ckend/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/ba
ckend/tensorflow_backend.py:1020: The name tf.assign is deprecated. Ple
ase use tf.compat.v1.assign instead.

Train on 73196 samples, validate on 36052 samples
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorbo
ardcolab/core.py:49: The name tf.summary.FileWriter is deprecated. Plea
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
se use tf.compat.v1.summary.FileWriter instead.

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

Epoch 1/20
73196/73196 [==============================] - 120s 2ms/step - loss: 1.
1107 - auroc: 0.5020 - val_loss: 0.7047 - val_auroc: 0.5247
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorbo
ardcolab/callbacks.py:51: The name tf.Summary is deprecated. Please use
tf.compat.v1.Summary instead.

Epoch 2/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
6265 - auroc: 0.5073 - val_loss: 0.5712 - val_auroc: 0.5892
Epoch 3/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
5415 - auroc: 0.5718 - val_loss: 0.5289 - val_auroc: 0.6975
Epoch 4/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
4967 - auroc: 0.6779 - val_loss: 0.5092 - val_auroc: 0.7190
Epoch 5/20
73196/73196 [==============================] - 110s 2ms/step - loss: 0.
4605 - auroc: 0.7159 - val_loss: 0.4795 - val_auroc: 0.7142
Epoch 6/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
4380 - auroc: 0.7304 - val_loss: 0.4575 - val_auroc: 0.7466
Epoch 7/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
4243 - auroc: 0.7360 - val_loss: 0.4325 - val_auroc: 0.7503
Epoch 8/20
73196/73196 [==============================] - 112s 2ms/step - loss: 0.
4143 - auroc: 0.7377 - val_loss: 0.4311 - val_auroc: 0.7520
Epoch 9/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
4094 - auroc: 0.7401 - val_loss: 0.4160 - val_auroc: 0.7502
Epoch 10/20
73196/73196 [==============================] - 112s 2ms/step - loss: 0.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

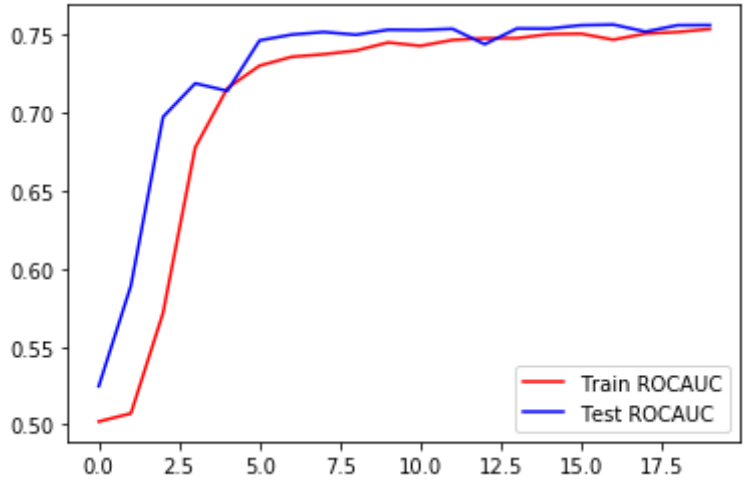
```
4043 - auroc: 0.7453 - val_loss: 0.4142 - val_auroc: 0.7534
Epoch 11/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
4029 - auroc: 0.7431 - val_loss: 0.4124 - val_auroc: 0.7532
Epoch 12/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
3995 - auroc: 0.7468 - val_loss: 0.4129 - val_auroc: 0.7540
Epoch 13/20
73196/73196 [==============================] - 112s 2ms/step - loss: 0.
3979 - auroc: 0.7480 - val_loss: 0.4367 - val_auroc: 0.7441
Epoch 14/20
73196/73196 [==============================] - 112s 2ms/step - loss: 0.
3970 - auroc: 0.7480 - val_loss: 0.4049 - val_auroc: 0.7542
Epoch 15/20
73196/73196 [==============================] - 113s 2ms/step - loss: 0.
3952 - auroc: 0.7506 - val_loss: 0.4172 - val_auroc: 0.7542
Epoch 16/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
3941 - auroc: 0.7509 - val_loss: 0.4106 - val_auroc: 0.7563
Epoch 17/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
3947 - auroc: 0.7470 - val_loss: 0.4201 - val_auroc: 0.7567
Epoch 18/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
3928 - auroc: 0.7510 - val_loss: 0.3972 - val_auroc: 0.7522
Epoch 19/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
3918 - auroc: 0.7520 - val_loss: 0.4049 - val_auroc: 0.7563
Epoch 20/20
73196/73196 [==============================] - 111s 2ms/step - loss: 0.
3911 - auroc: 0.7539 - val_loss: 0.4000 - val_auroc: 0.7564
```

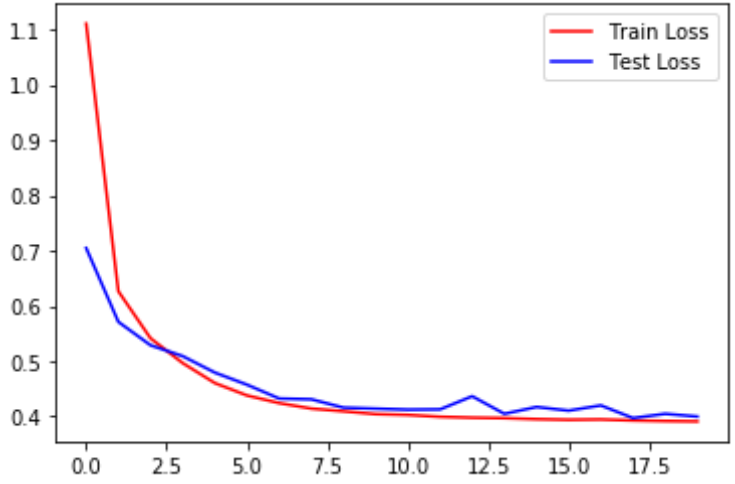
In [50]:

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

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
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 0.7564

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD