

Donors Choose - Model 3

In [1]:

```
# importing required libraries
import warnings
warnings.filterwarnings("ignore")
import pandas as pd
import numpy as np
from keras.layers import Input, Embedding, LSTM, Dropout, BatchNormalization, Dense, concat
from keras.preprocessing.text import Tokenizer, one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Model, load_model
from keras.utils import np_utils
from keras import regularizers
from keras.optimizers import *
from keras.callbacks import ModelCheckpoint, EarlyStopping, TensorBoard, ReduceLROnPlateau

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split

import tensorflow as tf
import matplotlib.pyplot as plt
%matplotlib inline
import re
from tqdm import tqdm
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
import pickle
from sklearn.preprocessing import StandardScaler
from scipy.sparse import hstack
```

Using TensorFlow backend.

In [2]:

```
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 (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("drive/My Drive/ML_data/preprocessed_data.csv")
project_data.head()
```

Out[3]:

	Unnamed: 0	Unnamed: 0.1	id	teacher_id	teacher_prefix	school_st
0	0	160221	p253737	c90749f5d961ff158d4b4d1e7dc665fc	mrs	
1	1	140945	p258326	897464ce9ddc600bced1151f324dd63a	mr	
2	2	21895	p182444	3465aaf82da834c0582ebd0ef8040ca0	ms	
3	3	45	p246581	f3cb9bffbbba169bef1a77b243e620b60	mrs	
4	4	172407	p104768	be1f7507a41f8479dc06f047086a39ec	mrs	

In [4]:

```
project_data["project_is_approved"].value_counts()
```

Out[4]:

```
1    92706
0    16542
Name: project_is_approved, dtype: int64
```

In [5]:

```
project_data.columns
```

Out[5]:

```
Index(['Unnamed: 0', 'Unnamed: 0.1', 'id', 'teacher_id', 'teacher_prefix',
      'school_state', 'project_submitted_datetime', 'project_grade_category',
      'project_subject_categories', 'project_subject_subcategories',
      'project_title', 'project_essay_1', 'project_essay_2',
      'project_essay_3', 'project_essay_4', 'project_resource_summary',
      'teacher_number_of_previously_posted_projects', 'project_is_approved',
      'essay', 'price', 'quantity', 'std_price', 'nrm_price',
      'project_summary_numerical'],
      dtype='object')
```

In [6]:

```
# checking for null values
project_data.isnull().sum()
```

Out[6]:

```
Unnamed: 0          0
Unnamed: 0.1        0
id                 0
teacher_id         0
teacher_prefix     0
school_state       0
project_submitted_datetime  0
project_grade_category  0
project_subject_categories  0
project_subject_subcategories  0
project_title      43
project_essay_1     0
project_essay_2     0
project_essay_3    105490
project_essay_4    105490
project_resource_summary  0
teacher_number_of_previously_posted_projects  0
project_is_approved  0
essay              0
price              0
quantity           0
std_price          0
nrm_price          0
project_summary_numerical  0
dtype: int64
```

In [0]:

```
# filling the null values with ''
project_data['project_title'] = project_data['project_title'].fillna('')
```

In [0]:

```
# combining essay and project_title columns
project_data['cleaned_text'] = project_data['essay'] + project_data['project_title']
```

In [0]:

```
# dropping unnecessary columns
project_data = project_data.drop(['Unnamed: 0', 'Unnamed: 0.1', 'id', 'teacher_id', 'project_id', 'project_resource_summary', 'std_price', 'nrm_price', 'project_essay_1', 'project_essay_2', 'project_essay_3', 'project_essay_4', 'essay', 'project_title'])
```

In [10]:

```
# columns left after dropping unnecessary columns
project_data.columns
```

Out[10]:

```
Index(['teacher_prefix', 'school_state', 'project_grade_category',
      'project_subject_categories', 'project_subject_subcategories',
      'teacher_number_of_previously_posted_projects', 'project_is_approved',
      'price', 'quantity', 'project_summary_numerical', 'cleaned_text'],
      dtype='object')
```

In [0]:

```
target = project_data['project_is_approved']
features = project_data.drop(['project_is_approved'], axis=1)
```

In [0]:

```
# splitting the dataset into train(75%) and test(25%) set
X_train, X_test, y_train, y_test = train_test_split(features, target, stratify=target, test_size=0.25)
```

In [13]:

```
print('Shape of Train data', X_train.shape)
print('Shape of Test data', X_test.shape)
```

```
Shape of Train data (81936, 10)
Shape of Test data (27312, 10)
```

Filtering Text Data (essays & project_title) based on idf values

In [0]:

```
tfidf = TfidfVectorizer()
combine_tfidf = tfidf.fit_transform(X_train['cleaned_text'])

# converting to dictionary
combine_dict = dict(zip(tfidf.get_feature_names(), list(tfidf.idf_)))
```

In [0]:

```
tfidf_df = pd.DataFrame(list(combine_dict.items()), columns=['Words', 'IDF Values'])
tfidf_df = tfidf_df.sort_values(by='IDF Values')
```

In [16]:

```
# finding the min & max idf values
print(tfidf_df['IDF Values'].min())
print(tfidf_df['IDF Values'].max())
```

```
1.0074234116042697
11.62055875771544
```

In [17]:

```
# based on the idf values we prepare the corpus, thereby leaving the words with lower idf v
corpus = tfidf_df[(tfidf_df['IDF Values'] >= 2) & (tfidf_df['IDF Values'] <=11)]
corpus.shape
```

Out[17]:

```
(27769, 2)
```

In [18]:

```
vocab = corpus["Words"].tolist()
vocab[:10]
```

Out[18]:

```
['new',
 'year',
 'one',
 'would',
 'time',
 'student',
 'want',
 'skills',
 'grade',
 'reading']
```

Tokenizing the Text

In [0]:

```
# convert the sentences (strings) into integers
tokenizer = Tokenizer()
tokenizer.fit_on_texts(vocab)
sequences_train = tokenizer.texts_to_sequences(X_train['cleaned_text'])
sequences_test = tokenizer.texts_to_sequences(X_test['cleaned_text'])
```

In [20]:

```
# get word -> integer mapping
word2idx = tokenizer.word_index
print('Found %s unique tokens.' % len(word2idx))
```

```
Found 27769 unique tokens.
```

Padding the sequences

In [0]:

```
EMBEDDING_DIM = 300
MAX_SEQUENCE_LENGTH = 100
```

In [22]:

```
encoded_train = pad_sequences(sequences_train, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', encoded_train.shape)
```

Shape of data tensor: (81936, 100)

In [23]:

```
encoded_test = pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', encoded_test.shape)
```

Shape of data tensor: (27312, 100)

Getting the vector representation using Glove vectors

In [0]:

```
# Loading Embedding File
pickle_in = open('drive/My Drive/ML_data/glove_vectors', 'rb')
glove_words = pickle.load(pickle_in)
```

In [0]:

```
num_words = len(word2idx) + 1
embedding_matrix = np.zeros((num_words, 300))
for word, i in word2idx.items():
    if i < len(vocab):
        embedding_vector = glove_words.get(word)
        if embedding_vector is not None:
            # words not found in embedding index will be all zeros.
            embedding_matrix[i] = embedding_vector
```

In [26]:

```
print(num_words)
print('-----')
print(embedding_matrix.shape)
```

27770

(27770, 300)

Vectorizing all the categorical features using CountVectorizer

In [0]:

```
vect = CountVectorizer(binary=True)

train_prefix = vect.fit_transform(X_train["teacher_prefix"])
test_prefix = vect.transform(X_test["teacher_prefix"])
```

In [0]:

```
vect = CountVectorizer(binary=True)

train_state = vect.fit_transform(X_train["school_state"])
test_state = vect.transform(X_test["school_state"])
```

In [0]:

```
vect = CountVectorizer(binary=True)

train_grade = vect.fit_transform(X_train["project_grade_category"])
test_grade = vect.transform(X_test["project_grade_category"])
```

In [0]:

```
vect = CountVectorizer(binary=True)

train_subcat = vect.fit_transform(X_train["project_subject_categories"])
test_subcat = vect.transform(X_test["project_subject_categories"])
```

In [0]:

```
vect = CountVectorizer(binary=True)

train_subcat_1 = vect.fit_transform(X_train["project_subject_subcategories"])
test_subcat_1 = vect.transform(X_test["project_subject_subcategories"])
```

Reshaping & Standardizing numerical features

In [0]:

```
num_train_1=X_train['project_summary_numerical'].values.reshape(-1, 1)
num_train_2=X_train['price'].values.reshape(-1, 1)
num_train_3=X_train['quantity'].values.reshape(-1, 1)
num_train_4=X_train['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1)

num_test_1=X_test['project_summary_numerical'].values.reshape(-1, 1)
num_test_2=X_test['price'].values.reshape(-1, 1)
num_test_3=X_test['quantity'].values.reshape(-1, 1)
num_test_4=X_test['teacher_number_of_previously_posted_projects'].values.reshape(-1, 1)
```

In [0]:

```
# concatenating train numerical features
num_train=np.concatenate((num_train_1,num_train_2,num_train_3,num_train_4),axis=1)

# concatenating test numerical features
num_test=np.concatenate((num_test_1,num_test_2,num_test_3,num_test_4),axis=1)

# Standardizing the features
norm=StandardScaler()
norm_train=norm.fit_transform(num_train)
norm_test=norm.transform(num_test)
```

In [0]:

```
# concatenating categorical features
cat_train = hstack([train_prefix,train_state,train_grade,train_subcat,train_subcat_1]).todense()
cat_test = hstack([test_prefix,test_state,test_grade,test_subcat,test_subcat_1]).todense()
```

In [0]:

```
# concatenating the numerical & categorical features
all_train = np.hstack((cat_train,norm_train))
all_test = np.hstack((cat_test,norm_test))
```

In [0]:

```
final_train = np.expand_dims(all_train,2)
final_test = np.expand_dims(all_test,2)
```

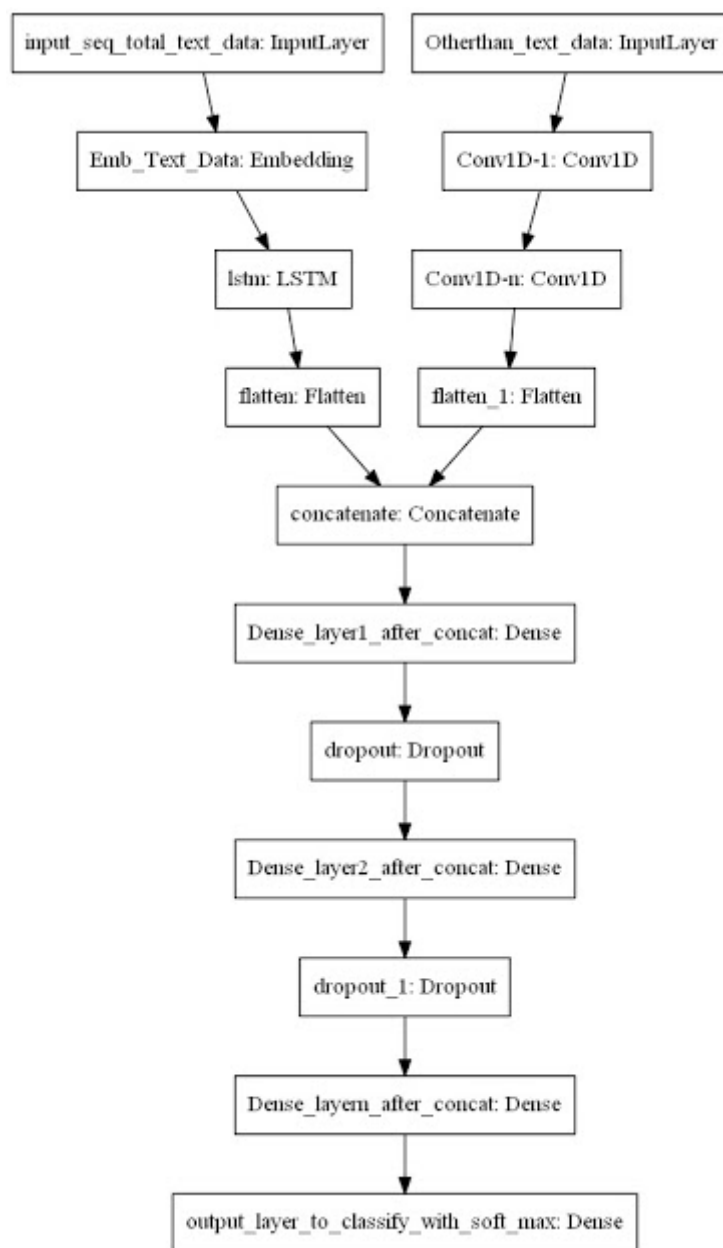
In [37]:

```
print(final_train.shape)
print('-----')
print(final_test.shape)
```

(81936, 512, 1)

(27312, 512, 1)

Defining model architecture



In [38]:

```
# Load pre-trained word embeddings into an Embedding layer
# note that we set trainable = False so as to keep the embeddings of fixed sized
embedding_layer = Embedding(
    num_words,
    300,
    weights=[embedding_matrix],
    input_length=MAX_SEQUENCE_LENGTH,
    trainable=False
)
input_text = Input(shape=(MAX_SEQUENCE_LENGTH,), name='input_text')
x = embedding_layer(input_text)
x = LSTM(256, dropout=0.5, kernel_regularizer=regularizers.l2(0.001), return_sequences=True)
flatten_1 = Flatten()(x)
```

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.

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.

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: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 [0]:

```
inp_conv = Input(shape=(512, 1), name='features_all')
x1 = Conv1D(filters=128, kernel_size=3, activation='relu', kernel_initializer="he_normal")(inp_conv)
x2 = Conv1D(filters=128, kernel_size=3, activation='relu', kernel_initializer="he_normal")(x1)
flatten_2 = Flatten()(x2)
```

In [0]:

```
x_concatenate = concatenate([flatten_1, flatten_2])

x = Dense(128,activation="relu",kernel_initializer="he_normal",kernel_regularizer=regularizer)

x = Dropout(0.5)(x)

x = Dense(64,activation="relu",kernel_initializer="he_normal",kernel_regularizer=regularizer)

x = Dropout(0.3)(x)

x = Dense(32,activation="relu",kernel_initializer="he_normal",kernel_regularizer=regularizer)

output = Dense(2, activation='softmax', name='output')(x)
model_3 = Model(inputs=[input_text,inp_conv],outputs=[output])
```

In [0]:

```
train_data_3 = [encoded_train, final_train]
test_data_3 = [encoded_test, final_test]

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

In [0]:

```
checkpoint_3 = ModelCheckpoint("model_3.h5",
                              monitor="val_auroc",
                              mode="max",
                              save_best_only = True,
                              verbose=1)

tensorboard_3 = TensorBoard(log_dir='graph_3', histogram_freq=0, batch_size=512, write_graph=True,
                             write_images=False, embeddings_freq=0, embeddings_layer_names=None,
                             embeddings_data=None, update_freq='epoch')

callbacks_3 = [tensorboard_3,checkpoint_3]
```

In [0]:

```
# Defining Custom ROC-AUC function
from sklearn.metrics import roc_auc_score

def auc1(y_true, y_pred):
    if len(np.unique(y_true[:,1])) == 1:
        return 0.5
    else:
        return roc_auc_score(y_true, y_pred)

def auroc(y_true, y_pred):
    return tf.py_func(auc1, (y_true, y_pred), tf.double)
```

In [0]:

```
adam = Adam(lr=0.001, beta_1=0.9, beta_2=0.999, epsilon=None, decay=0.0, amsgrad=False)
```

In [45]:

```
model_3.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 <ipython-input-43-a7e6cba44e56>:10: 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 [46]:

```
history_3 = model_3.fit(train_data_3, Y_train, batch_size=512,
                        epochs=30, validation_data=(test_data_3,Y_test), callbacks=callback
```

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250: add_dispatch_support.<locals>.wrapper (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

Train on 81936 samples, validate on 27312 samples

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.

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/callbacks.py:1125: The name tf.summary.FileWriter is deprecated. Please use tf.compat.v1.summary.FileWriter instead.

Epoch 1/30

81936/81936 [=====] - 42s 509us/step - loss: 0.8055
- auroc: 0.6348 - val_loss: 0.5749 - val_auroc: 0.7258

Epoch 00001: val_auroc improved from -inf to 0.72584, saving model to model_3.h5

Epoch 2/30

81936/81936 [=====] - 37s 452us/step - loss: 0.5246
- auroc: 0.7037 - val_loss: 0.4843 - val_auroc: 0.7395

Epoch 00002: val_auroc improved from 0.72584 to 0.73948, saving model to model_3.h5

Epoch 3/30

81936/81936 [=====] - 37s 450us/step - loss: 0.4700
- auroc: 0.7211 - val_loss: 0.4669 - val_auroc: 0.7468

Epoch 00003: val_auroc improved from 0.73948 to 0.74680, saving model to model_3.h5

Epoch 4/30

81936/81936 [=====] - 37s 452us/step - loss: 0.4422
- auroc: 0.7325 - val_loss: 0.4355 - val_auroc: 0.7492

Epoch 00004: val_auroc improved from 0.74680 to 0.74922, saving model to model_3.h5

Epoch 5/30

81936/81936 [=====] - 37s 454us/step - loss: 0.4253
- auroc: 0.7375 - val_loss: 0.4153 - val_auroc: 0.7548

Epoch 00005: val_auroc improved from 0.74922 to 0.75478, saving model to model_3.h5

Epoch 6/30

81936/81936 [=====] - 37s 449us/step - loss: 0.4130
- auroc: 0.7438 - val_loss: 0.4060 - val_auroc: 0.7491

Epoch 00006: val_auroc did not improve from 0.75478

Epoch 7/30

81936/81936 [=====] - 37s 448us/step - loss: 0.4064
- auroc: 0.7469 - val_loss: 0.4070 - val_auroc: 0.7570

Epoch 00007: val_auroc improved from 0.75478 to 0.75699, saving model to model_3.h5

Epoch 8/30

81936/81936 [=====] - 37s 448us/step - loss: 0.4011
- auroc: 0.7462 - val_loss: 0.4003 - val_auroc: 0.7579

Epoch 00008: val_auroc improved from 0.75699 to 0.75786, saving model to model_3.h5

Epoch 9/30

81936/81936 [=====] - 37s 449us/step - loss: 0.3987
- auroc: 0.7464 - val_loss: 0.3904 - val_auroc: 0.7592

Epoch 00009: val_auroc improved from 0.75786 to 0.75916, saving model to model_3.h5

Epoch 10/30

81936/81936 [=====] - 37s 447us/step - loss: 0.3925
- auroc: 0.7502 - val_loss: 0.3969 - val_auroc: 0.7578

Epoch 00010: val_auroc did not improve from 0.75916

Epoch 11/30

81936/81936 [=====] - 37s 448us/step - loss: 0.3926
- auroc: 0.7506 - val_loss: 0.3840 - val_auroc: 0.7618

Epoch 00011: val_auroc improved from 0.75916 to 0.76183, saving model to model_3.h5

Epoch 12/30

81936/81936 [=====] - 37s 449us/step - loss: 0.3898
- auroc: 0.7526 - val_loss: 0.3840 - val_auroc: 0.7636

Epoch 00012: val_auroc improved from 0.76183 to 0.76364, saving model to model_3.h5

Epoch 13/30

81936/81936 [=====] - 37s 451us/step - loss: 0.3891
- auroc: 0.7526 - val_loss: 0.3879 - val_auroc: 0.7611

Epoch 00013: val_auroc did not improve from 0.76364

Epoch 14/30

81936/81936 [=====] - 37s 447us/step - loss: 0.3886
- auroc: 0.7536 - val_loss: 0.3809 - val_auroc: 0.7650

Epoch 00014: val_auroc improved from 0.76364 to 0.76502, saving model to model_3.h5

Epoch 15/30

81936/81936 [=====] - 37s 450us/step - loss: 0.3859
- auroc: 0.7561 - val_loss: 0.3823 - val_auroc: 0.7652

Epoch 00015: val_auroc improved from 0.76502 to 0.76517, saving model to model_3.h5

Epoch 16/30

81936/81936 [=====] - 37s 448us/step - loss: 0.3869
- auroc: 0.7548 - val_loss: 0.3854 - val_auroc: 0.7649

Epoch 00016: val_auroc did not improve from 0.76517

Epoch 17/30

81936/81936 [=====] - 37s 447us/step - loss: 0.3859
- auroc: 0.7552 - val_loss: 0.3807 - val_auroc: 0.7649

Epoch 00017: val_auroc did not improve from 0.76517

Epoch 18/30

81936/81936 [=====] - 37s 449us/step - loss: 0.3839
- auroc: 0.7564 - val_loss: 0.3791 - val_auroc: 0.7640

Epoch 00018: val_auroc did not improve from 0.76517

Epoch 19/30

81936/81936 [=====] - 37s 447us/step - loss: 0.3836
- auroc: 0.7598 - val_loss: 0.3800 - val_auroc: 0.7656

Epoch 00019: val_auroc improved from 0.76517 to 0.76562, saving model to model_3.h5

Epoch 20/30

81936/81936 [=====] - 37s 447us/step - loss: 0.3831
- auroc: 0.7592 - val_loss: 0.3765 - val_auroc: 0.7685

Epoch 00020: val_auroc improved from 0.76562 to 0.76851, saving model to model_3.h5

Epoch 21/30

81936/81936 [=====] - 37s 448us/step - loss: 0.3809
- auroc: 0.7618 - val_loss: 0.3782 - val_auroc: 0.7654

Epoch 00021: val_auroc did not improve from 0.76851

Epoch 22/30

81936/81936 [=====] - 37s 455us/step - loss: 0.3843
- auroc: 0.7603 - val_loss: 0.3762 - val_auroc: 0.7682

Epoch 00022: val_auroc did not improve from 0.76851

Epoch 23/30

81936/81936 [=====] - 37s 449us/step - loss: 0.3814
- auroc: 0.7617 - val_loss: 0.3778 - val_auroc: 0.7669

Epoch 00023: val_auroc did not improve from 0.76851

Epoch 24/30

81936/81936 [=====] - 37s 450us/step - loss: 0.3817
- auroc: 0.7595 - val_loss: 0.3780 - val_auroc: 0.7694

Epoch 00024: val_auroc improved from 0.76851 to 0.76945, saving model to model_3.h5

Epoch 25/30

81936/81936 [=====] - 37s 448us/step - loss: 0.3820
- auroc: 0.7636 - val_loss: 0.3769 - val_auroc: 0.7692

Epoch 00025: val_auroc did not improve from 0.76945

Epoch 26/30

81936/81936 [=====] - 37s 449us/step - loss: 0.3813
- auroc: 0.7613 - val_loss: 0.3781 - val_auroc: 0.7688

Epoch 00026: val_auroc did not improve from 0.76945

Epoch 27/30

81936/81936 [=====] - 37s 448us/step - loss: 0.3775
- auroc: 0.7646 - val_loss: 0.3749 - val_auroc: 0.7705

Epoch 00027: val_auroc improved from 0.76945 to 0.77051, saving model to model_3.h5

Epoch 28/30

81936/81936 [=====] - 37s 449us/step - loss: 0.3786
- auroc: 0.7651 - val_loss: 0.3762 - val_auroc: 0.7686

Epoch 00028: val_auroc did not improve from 0.77051

Epoch 29/30

81936/81936 [=====] - 37s 452us/step - loss: 0.3798
- auroc: 0.7650 - val_loss: 0.3769 - val_auroc: 0.7682

Epoch 00029: val_auroc did not improve from 0.77051

Epoch 30/30

81936/81936 [=====] - 37s 455us/step - loss: 0.3784

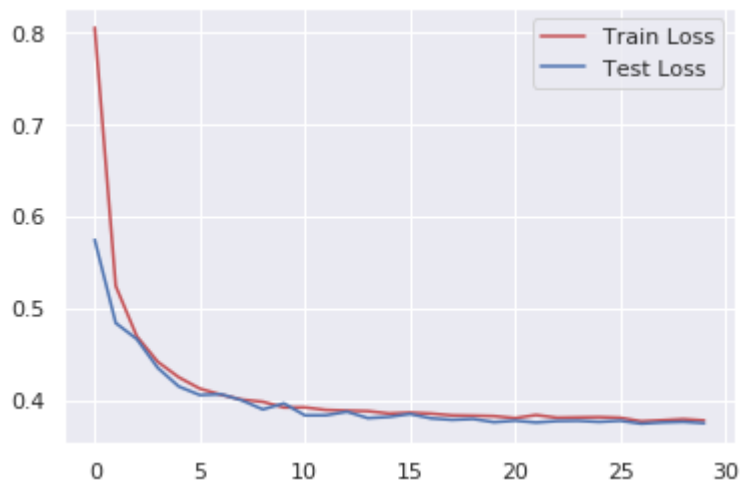
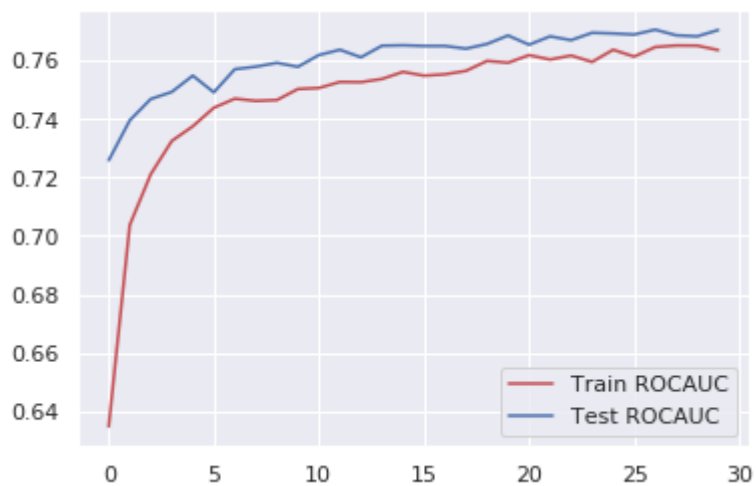
- auroc: 0.7635 - val_loss: 0.3755 - val_auroc: 0.7704

Epoch 00030: val_auroc did not improve from 0.77051

In [47]:

```
sns.set()
plt.plot(history_3.history['auroc'], 'r')
plt.plot(history_3.history['val_auroc'], 'b')
plt.legend({'Train ROCAUC': 'r', 'Test ROCAUC': 'b'})
plt.show()
```

```
plt.plot(history_3.history['loss'], 'r')
plt.plot(history_3.history['val_loss'], 'b')
plt.legend({'Train Loss': 'r', 'Test Loss': 'b'})
plt.show()
```



AUC Score for Model 3 - 0.77051