In [1]:
```
import pandas as pd
import numpy as np
%matplotlib inline
import seaborn as sns
import matplotlib.pyplot as plt
import os
from sklearn.preprocessing import LabelEncoder,StandardScaler, OneHotEncoder
from scipy.sparse import csr_matrix, hstack
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import StratifiedKFold
from sklearn.metrics import log_loss
from sklearn.feature_extraction.text import TfidfTransformer,TfidfVectorizer,C
ountVectorizer
from sklearn.cluster import KMeans
from xgboost import XGBClassifier
from sklearn.calibration import CalibratedClassifierCV
import warnings
warnings.filterwarnings("ignore")
```

In [2]:
```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation,BatchNormalization
from keras.wrappers.scikit_learn import KerasClassifier
from keras.utils import np_utils
from keras.optimizers import SGD,Adagrad
from keras.layers.advanced_activations import PReLU
from sklearn.model_selection import train_test_split
from keras.callbacks import EarlyStopping,TensorBoard
from statistics import mean
```

```
Using TensorFlow backend.
```

LOADING THE DATA

In [3]:
```
gatrain = pd.read_csv("gender_age_train.csv",index_col='device_id')
gatest = pd.read_csv("gender_age_test.csv",index_col='device_id')
phone=pd.read_csv("phone_brand_device_model.csv")
app_label=pd.read_csv('app_labels.csv')
label_cat=pd.read_csv("label_categories.csv")
app_events=pd.read_csv("app_events.csv", dtype={'is_active':bool})
events = pd.read_csv('events.csv',  parse_dates=['timestamp'],index_col='event
_id')
```

In [4]:
```
#removing duplicate device id's
phone = phone.drop_duplicates('device_id',keep='first').set_index('device_id')
```

In [5]:
```
print(gatrain.shape)
print(gatest.shape)
print(phone.shape)
print(app_label.shape)
print(label_cat.shape)
print(app_events.shape)
print(events.shape)
```

```
(74645, 3)
(112071, 0)
(186716, 2)
(459943, 2)
(930, 2)
(32473067, 4)
(3252950, 4)
```

SPLITTING THE DATA

SOME DEVICES HAVE EVENTS INFORMATION AND SOME DEVICES DOES NOT HAVE EVENT INFORMTION.

1. SO WE DIVIDE THE DATA INTO TRAIN AND TEST IN BOTH EVENTS AND NO EVENTS DATA.

In [6]:
```
#https://docs.scipy.org/doc/numpy/reference/generated/numpy.in1d.html
mask=np.in1d(gatrain.index,events["device_id"].values)
gatrain_events= gatrain[mask]

mask=np.in1d(gatest.index,events["device_id"].values)
gatest_events= gatest[mask]
```

In [7]:
```
#https://docs.scipy.org/doc/numpy/reference/generated/numpy.in1d.html
mask=np.in1d(gatrain.index,events["device_id"].values,invert=True)
gatrain_noevents= gatrain[mask]

mask=np.in1d(gatest.index,events["device_id"].values,invert=True)
gatest_noevents= gatest[mask]
```

In [8]:
```
#Each row of is given by a unique integer as an identifier

gatrain['trainrow'] = np.arange(gatrain.shape[0])
gatest['testrow'] = np.arange(gatest.shape[0])

gatrain_events['trainrow']=np.arange(gatrain_events.shape[0])
gatest_events['testrow']=np.arange(gatest_events.shape[0])

gatrain_noevents['trainrow']=np.arange(gatrain_noevents.shape[0])
gatest_noevents['testrow']=np.arange(gatest_noevents.shape[0])
```

In [9]:
```
print("train data with events information:",gatrain_events.shape)
print("train data without events information:",gatrain_noevents.shape)
print("test data with events information:",gatest_events.shape)
print("test data without events information:",gatest_noevents.shape)
```

```
train data with events information: (23309, 4)
train data without events information: (51336, 4)
test data with events information: (35194, 1)
test data without events information: (76877, 1)
```

VECTORIZING PHONE BRAND

In [10]:
```
brandencoder = LabelEncoder().fit(phone.phone_brand)
phone['brand'] = brandencoder.transform(phone['phone_brand'])
nbrand=len(brandencoder.classes_)
```

In [11]:
```
import pickle
with open('brandencoder','wb') as fp:
    pickle.dump(brandencoder,fp)
```

VECTORIZING PHONE MODEL

In [12]:
```
m = phone.phone_brand.str.cat(phone.device_model)
#m=phone['phone_brand'].str.cat(phone['device_model'])
modelencoder = LabelEncoder().fit(m)
phone['model'] = modelencoder.transform(m)
nmodel=len(modelencoder.classes_)
```

In [13]:
```
import pickle
with open('modelencoder','wb') as fp:
    pickle.dump(modelencoder,fp)
```

FEATURES USING APP ID'S

In [14]:
```
#https://www.kaggle.com/dvasyukova/a-linear-model-on-apps-and-labels
#number of times app used in a device id's
appencoder = LabelEncoder().fit(app_events['app_id'])
app_events['app'] = appencoder.transform(app_events['app_id'])


napps = len(appencoder.classes_)

deviceapps = (app_events.merge(events[['device_id']], how='left',left_on='even
t_id',right_index=True)
                       .groupby(['device_id','app'])['app'].agg(['size'])# gro
uping by device id and app and finding size of app
                       .merge(gatrain_events[['trainrow']], how='left', left_i
ndex=True, right_index=True)#finding trainrow
                       .merge(gatest_events[['testrow']], how='left', left_ind
ex=True, right_index=True)#finding testrow
                       .reset_index())
deviceapps.head()
```

Out[14]:

	device_id	app	size	trainrow	testrow
0	-9222956879900151005	548	18	5145.0	NaN
1	-9222956879900151005	1096	18	5145.0	NaN
2	-9222956879900151005	1248	26	5145.0	NaN
3	-9222956879900151005	1545	12	5145.0	NaN
4	-9222956879900151005	1664	18	5145.0	NaN

In [15]:
```
import pickle
with open('appencoder','wb') as fp:
    pickle.dump(appencoder,fp)
```

FEATURES USING APP LABELS

In [17]:
```
app_label = app_label.loc[app_label.app_id.isin(app_events.app_id.unique())]
app_label['app'] = appencoder.transform(app_label.app_id)
labelencoder = LabelEncoder().fit(app_label.label_id)
app_label['label'] = labelencoder.transform(app_label.label_id)
nlabels = len(labelencoder.classes_)
```

In [18]:
```
import pickle
with open('labelencoder','wb') as fp:
    pickle.dump(labelencoder,fp)
```

In [19]:
```
devicelabels = (deviceapps[['device_id','app']]
                .merge(app_label[['app','label']])
                .groupby(['device_id','label'])['app'].agg(['size'])
                .merge(gatrain_events[['trainrow']], how='left', left_index=True, right_index=True)
                .merge(gatest_events[['testrow']], how='left', left_index=True, right_index=True)
                .reset_index())
devicelabels.head()
```

Out[19]:

	device_id	label	size	trainrow	testrow
0	-9222956879900151005	117	1	5145.0	NaN
1	-9222956879900151005	120	1	5145.0	NaN
2	-9222956879900151005	126	1	5145.0	NaN
3	-9222956879900151005	138	2	5145.0	NaN
4	-9222956879900151005	147	2	5145.0	NaN

FEATURES USING TIME FEATURE

In [20]:
```
#we are processing timestamp feature to get hour and day and dividing into 4 bins
events['hour'] = events['timestamp'].map(lambda x:pd.to_datetime(x).hour)
events['hourbin'] = [1 if ((x>=1)&(x<=6)) else 2 if ((x>=7)&(x<=12)) else 3 if ((x>=13)&(x<=18)) else 4 for x in events['hour']]
```

In [21]:
```
events.hour=events.hour.astype(str)
events.hourbin=events.hourbin.astype(str)
```

In [22]:
```
hourjoin = events.groupby("device_id")["hour"].apply(lambda x: " ".join('0'+str(s) for s in x))
```

In [23]:
```
hourbinjoin=events.groupby("device_id")["hourbin"].apply(lambda x: " ".join('0'+str(s) for s in x))
```

In [24]:
```
daysjoin=events['timestamp'].dt.day_name()
events['day']=daysjoin.map({'Sunday':0,'Monday':1,'Tuesday':2,'Wednesday':3,'Thursday':4,'Friday':5,'Saturday':6})
```

In [25]:
```
daysjoin = events.groupby("device_id")["day"].apply(lambda x: " ".join("0"+str(s) for s in x))
```

FEATURES USING LATITUDE AND LONGITUDE

In [26]:
```
median_lat = events.groupby("device_id")["latitude"].agg('median')
```

In [27]:
```
median_lon=events.groupby("device_id")["longitude"].agg('median')
```

WE ARE CLUSTERING MEDIAN LATITUDES AND LONGITUDES IN TO 10 CLUSTERS

In [28]:
```
com=pd.concat([median_lat, median_lon], axis=1)
kmeans = KMeans(n_clusters=10, random_state=0).fit(com)
clustered_geo_features=pd.Series(kmeans.labels_)
clustered_geo_features.index=median_lon.index
```

FEATURES BASED ON ACTIVE APPS AND APP COUNT

In [29]:
```
apps = app_events.groupby("event_id")["is_active"].apply(lambda x: " ".join(st
r(s) for s in x))
```

In [30]:
```
events["apps_active"] = events.index.map(apps)
active_apps_events = events.groupby("device_id")["apps_active"].apply(lambda x
: " ".join(str(s) for s in x if str(s)!='nan'))
```

MODELLING

ONE HOT ENCODING OF PHONE BRAND

In [31]:
```
gatrain['brand'] = phone['brand']
gatest['brand'] = phone['brand']

Xtr_brand = csr_matrix((np.ones(gatrain.shape[0]),
                       (gatrain.trainrow, gatrain.brand)))
Xte_brand = csr_matrix((np.ones(gatest.shape[0]),
                       (gatest.testrow, gatest.brand)))
```

In [32]:
```
print(Xtr_brand.shape)
print(Xte_brand.shape)
```

```
(74645, 131)
(112071, 131)
```

ONE HOT ENCODING OF PHONE MODEL

In [33]:
```
gatrain['model'] = phone['model']
gatest['model'] = phone['model']

Xtr_model = csr_matrix((np.ones(gatrain.shape[0]),
                       (gatrain.trainrow, gatrain.model)))
Xte_model = csr_matrix((np.ones(gatest.shape[0]),
                       (gatest.testrow, gatest.model)))
print(Xtr_model.shape)
print(Xte_model.shape)
```

```
(74645, 1667)
(112071, 1667)
```

ONE HOT ENCODING OF DEVICE APPS

In [34]:
```
d = deviceapps.dropna(subset=['trainrow'])
Xtr_app = csr_matrix((np.ones(d.shape[0]), (d.trainrow, d.app)),
                      shape=(gatrain.shape[0],napps))
d = deviceapps.dropna(subset=['testrow'])
Xte_app = csr_matrix((np.ones(d.shape[0]), (d.testrow, d.app)),
                      shape=(gatest.shape[0],napps))

print(Xtr_app.shape)
print(Xte_app.shape)
```

```
(74645, 19237)
(112071, 19237)
```

ONE HOT ENCODING OF APP CATEGORY

In [35]:
```
d = devicelabels.dropna(subset=['trainrow'])
Xtr_label = csr_matrix((np.ones(d.shape[0]), (d.trainrow, d.label)),
                      shape=(gatrain.shape[0],nlabels))
d = devicelabels.dropna(subset=['testrow'])
Xte_label = csr_matrix((np.ones(d.shape[0]), (d.testrow, d.label)),
                      shape=(gatest.shape[0],nlabels))

print(Xtr_label.shape)
print(Xte_label.shape)
```

```
(74645, 492)
(112071, 492)
```

In [36]:
```
#hstacking all the features
Xtrain = hstack((Xtr_brand, Xtr_model, Xtr_app, Xtr_label), format='csr')
Xtest =  hstack((Xte_brand, Xte_model, Xte_app, Xte_label), format='csr')
print('Train data shape:',Xtrain.shape)
print('Test data shape:',Xtest.shape)
```

```
Train data shape: (74645, 21527)
Test data shape: (112071, 21527)
```

In [37]:
```
#applying applying label encoding on target variable
targetencoder = LabelEncoder().fit(gatrain.group)
y = targetencoder.transform(gatrain.group)
nclasses = len(targetencoder.classes_)
```

In [38]:
```
import pickle
with open('classlabel','wb') as fp:
    pickle.dump(y,fp)
```

In [39]:
```
#splitting data into train and validation
from sklearn.calibration import CalibratedClassifierCV
from sklearn.model_selection import train_test_split

xtr, xcv, ytr, ycv = train_test_split(Xtrain, y,stratify=y,test_size=0.15)
print(xtr.shape,ytr.shape)
print(xcv.shape,ycv.shape)
```

```
(63448, 21527) (63448,)
(11197, 21527) (11197,)
```

LOGISTIC REGRESSION

In [35]:
```
from sklearn.svm import LinearSVC
from sklearn.calibration import CalibratedClassifierCV
alpha = [0.00001,0.0001,0.001,0.01,0.1,1,10,100,1000]
cv_log_error = []
for i in alpha:
    print('for c = ',i)
    SGD  = LogisticRegression(class_weight = 'balanced',penalty = 'l2',C = i)
    clf =  SGD.fit(xtr,ytr)
    calib_sgd = CalibratedClassifierCV(clf,method = 'sigmoid')
    calib_sgd.fit(xtr,ytr)
    y_cv_pred = calib_sgd.predict_proba(xcv)
    #cv_log_error.append(log_loss(y_cv,y_cv_pred))
    print('for c = ',i ,'the log loss is :',log_loss(ycv,y_cv_pred))
```

```
for c =  1e-05
for c =  1e-05 the log loss is : 2.4240490121671416
for c =  0.0001
for c =  0.0001 the log loss is : 2.4239127140248673
for c =  0.001
for c =  0.001 the log loss is : 2.421898251247192
for c =  0.01
for c =  0.01 the log loss is : 2.4034478389221365
for c =  0.1
for c =  0.1 the log loss is : 2.408299979523952
for c =  1
for c =  1 the log loss is : 2.4181286899702306
for c =  10
for c =  10 the log loss is : 2.42395984079844
for c =  100
for c =  100 the log loss is : 2.4254481618915733
for c =  1000
for c =  1000 the log loss is : 2.425641919967717
```

WE CHOSE OUR BEST C TO BE 0.01

In [36]:
```
clf = LogisticRegression(C=0.01, class_weight='balanced', multi_class='multino
mial', solver='lbfgs')
clf.fit(xtr, ytr)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(xcv, ycv)

predict_y = sig_clf.predict_proba(xtr)
loss=log_loss(ytr, predict_y)
print("The train log loss for best C is:",loss)
predict_y = sig_clf.predict_proba(xcv)
loss=log_loss(ycv, predict_y)
print("The validation log loss for best C is:",loss)
```

```
The train log loss for best C is: 2.4145317866106
The validation log loss for best C is: 2.354060651561517
```

MODELLING USING DEVICES WITHOUT EVENTS

In [40]:
```
Xtrain_whole = hstack((Xtr_brand, Xtr_model), format='csr')

targetencoder = LabelEncoder().fit(gatrain.group)
y = targetencoder.transform(gatrain.group)
```

In [41]:
```
gatest_noevents['model']=phone['model']
gatest_noevents['brand']=phone['brand']
```

In [42]:
```
gatest_noevents_model = csr_matrix((np.ones(gatest_noevents.shape[0]),
                       (gatest_noevents.testrow, gatest_noevents.model)))

gatest_noevents_brand= csr_matrix((np.ones(gatest_noevents.shape[0]),
                       (gatest_noevents.testrow, gatest_noevents.brand)))
```

In [43]:
```
xtest_noevents=hstack((gatest_noevents_brand, gatest_noevents_model), format=
'csr')
```

In [44]:
```
print("xtrain shape:",Xtrain_whole.shape)
print("ytrain shape:",y.shape)

print("xtest shape:",xtest_noevents.shape)
```

```
xtrain shape: (74645, 1798)
ytrain shape: (74645,)
xtest shape: (76877, 1798)
```

In [42]:
```
xtr, xcv, ytr, ycv = train_test_split(Xtrain_whole, y,stratify=y,test_size=0.1
5,random_state=18)
```

LOGISTIC REGRESSION

In [43]:
```python
alpha = [0.001,0.01,0.02,0.1,0.15,1,10]


for i in alpha:
    clf = LogisticRegression(C=i, class_weight='balanced', multi_class='multin
omial',solver='lbfgs')
    clf.fit(xtr, ytr)
    #Using Model Calibration
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(xtr, ytr)
    predict_y = sig_clf.predict_proba(xcv)
    print('For values of C = ', i, "The validation log loss is:",log_loss(ycv,
predict_y))
```

```
For values of C =  0.001 The validation log loss is: 2.4030020103020036
For values of C =  0.01 The validation log loss is: 2.39641152579951
For values of C =  0.02 The validation log loss is: 2.3940695248817785
For values of C =  0.1 The validation log loss is: 2.3896730084069544
For values of C =  0.15 The validation log loss is: 2.3891201368497037
For values of C =  1 The validation log loss is: 2.39105667945638
For values of C =  10 The validation log loss is: 2.398847882073424
```

WE CHOSE OUR BEST C TO BE 0.15

In [44]:
```python
clf = LogisticRegression(C=0.15, class_weight='balanced', multi_class='multino
mial', solver='lbfgs')
clf.fit(xtr, ytr)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(xtr, ytr)

predict_y = sig_clf.predict_proba(xtr)
loss=log_loss(ytr, predict_y)
print("The train log loss for best C is:",loss)
predict_y = sig_clf.predict_proba(xcv)
loss=log_loss(ycv, predict_y)
print("The validation log loss for best C is:",loss)
```

```
The train log loss for best C is: 2.362802878894095
The validation log loss for best C is: 2.3891201368497037
```

In [45]:
```python
#predicting for test data
no_events_pred_lr=sig_clf.predict_proba(xtest_noevents)
```

In [46]:
```python
#saving the model
from sklearn.externals import joblib as jobl
from joblib import dump
np.save('lr_noevents',no_events_pred_lr)
```

OBSERVATIONS:

FOR LOGISTIC REGRESSION MODEL TRAIN LOGLOSS IS 2.3628 AND VALIDATION LOSS IS 2.3891

NEURAL NETWORKS

In [47]:
```
#https://www.kaggle.com/c/talkingdata-mobile-user-demographics/discussion/2342
4
def noevents_nn_model1(input_shape):
    model = Sequential()
    model.add(Dense(256, input_dim=input_shape))
    model.add(PReLU())
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(64))
    model.add(PReLU())
    model.add(BatchNormalization())
    model.add(Dropout(0.5))
    model.add(Dense(12))
    model.add(Activation('softmax'))
    model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
    return model
```

In [48]:
```
model_sum=noevents_nn_model1(xtr.shape[1])
model_sum.summary()
```

```
WARNING: Logging before flag parsing goes to stderr.
W0408 12:30:09.095427 11128 deprecation_wrapper.py:119] From c:\users\navee\a
ppdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorf
low_backend.py:74: The name tf.get_default_graph is deprecated. Please use t
f.compat.v1.get_default_graph instead.

W0408 12:30:09.523084 11128 deprecation_wrapper.py:119] From c:\users\navee\a
ppdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorf
low_backend.py:517: The name tf.placeholder is deprecated. Please use tf.comp
at.v1.placeholder instead.

W0408 12:30:09.580778 11128 deprecation_wrapper.py:119] From c:\users\navee\a
ppdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorf
low_backend.py:4138: The name tf.random_uniform is deprecated. Please use tf.
random.uniform instead.

W0408 12:30:09.785201 11128 deprecation_wrapper.py:119] From c:\users\navee\a
ppdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorf
low_backend.py:133: The name tf.placeholder_with_default is deprecated. Pleas
e use tf.compat.v1.placeholder_with_default instead.

W0408 12:30:09.803154 11128 deprecation.py:506] From c:\users\navee\appdata\l
ocal\programs\python\python37\lib\site-packages\keras\backend\tensorflow_back
end.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_pr
ob 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 - k
eep_prob`.
W0408 12:30:09.902594 11128 deprecation_wrapper.py:119] From c:\users\navee\a
ppdata\local\programs\python\python37\lib\site-packages\keras\optimizers.py:7
90: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.
Optimizer instead.

W0408 12:30:09.918372 11128 deprecation_wrapper.py:119] From c:\users\navee\a
ppdata\local\programs\python\python37\lib\site-packages\keras\backend\tensorf
low_backend.py:3295: The name tf.log is deprecated. Please use tf.math.log in
stead.
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 256)               460544    
_________________________________________________________________
p_re_lu_1 (PReLU)            (None, 256)               256       
_________________________________________________________________
batch_normalization_1 (Batch (None, 256)               1024      
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0         
_________________________________________________________________
dense_2 (Dense)              (None, 64)                16448     
_________________________________________________________________
p_re_lu_2 (PReLU)            (None, 64)                64        
_________________________________________________________________
batch_normalization_2 (Batch (None, 64)                256       
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0         
_________________________________________________________________
dense_3 (Dense)              (None, 12)                780       
_________________________________________________________________
activation_1 (Activation)    (None, 12)                0         
=================================================================
Total params: 479,372
Trainable params: 478,732
Non-trainable params: 640
_________________________________________________________________
```

In [49]:
```
early_stop=EarlyStopping(monitor='val_loss',patience=5,restore_best_weights=True)
```

In [50]:
```
def noevents_average_nn_1(state):
    """
    Takes a list of Random Seeds, splits the data into Train and CV based on S
eed, trains model and takes average of
    predictions while testing
    """
    model_list=[]
    loss_list=[]
    avg_cv_loss=0
    for i in range(len(state)):
        xtr, xcv, ytr, ycv = train_test_split(Xtrain_whole, y,stratify=y,test_
size=0.15,random_state=state[i])
        ytr=np_utils.to_categorical(ytr)
        ycv=np_utils.to_categorical(ycv)
        model=noevents_nn_model1(xtr.shape[1])
        model.fit(xtr, ytr, batch_size=256, epochs=20, verbose=1, validation_d
ata=(xcv, ycv),callbacks=[early_stop])
        model.save('saved_models/no_events/nn '+str(i+1))
        pred=model.predict_proba(xcv)
        cv_loss=log_loss(ycv, pred)
        print("Validation Log Loss of  Model in Current Run: ",cv_loss)
        model_list.append(model)
        loss_list.append(cv_loss)
    avg_cv_loss=mean(loss_list)
    print("Average CV Loss of "+str(len(state))+" Runs :",avg_cv_loss)
    return(model_list)
```

In [51]:
```
random_seeds=[9,18,42,86,103]
model_list_1= noevents_average_nn_1(random_seeds)
```

```
W0408 12:30:10.290519 11128 deprecation.py:323] From c:\users\navee\appdata\l
ocal\programs\python\python37\lib\site-packages\tensorflow\python\ops\math_gr
ad.py:1250: add_dispatch_support.<locals>.wrapper (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
```

```
Train on 63448 samples, validate on 11197 samples
Epoch 1/20
63448/63448 [==============================] - 5s 74us/step - loss: 2.8828 -
acc: 0.1041 - val_loss: 2.4166 - val_acc: 0.1458
Epoch 2/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.4785 -
acc: 0.1316 - val_loss: 2.3994 - val_acc: 0.1516
Epoch 3/20
63448/63448 [==============================] - 4s 56us/step - loss: 2.4185 -
acc: 0.1435 - val_loss: 2.3957 - val_acc: 0.1517
Epoch 4/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.3978 -
acc: 0.1521 - val_loss: 2.3947 - val_acc: 0.1516
Epoch 5/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.3890 -
acc: 0.1560 - val_loss: 2.3926 - val_acc: 0.1566
Epoch 6/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.3829 -
acc: 0.1593 - val_loss: 2.3907 - val_acc: 0.1553
Epoch 7/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.3777 -
acc: 0.1592 - val_loss: 2.3904 - val_acc: 0.1561
Epoch 8/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.3719 -
acc: 0.1642 - val_loss: 2.3926 - val_acc: 0.1551
Epoch 9/20
63448/63448 [==============================] - 5s 77us/step - loss: 2.3668 -
acc: 0.1660 - val_loss: 2.3933 - val_acc: 0.1521
Epoch 10/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.3624 -
acc: 0.1693 - val_loss: 2.3940 - val_acc: 0.1537
Epoch 11/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.3596 -
acc: 0.1686 - val_loss: 2.3960 - val_acc: 0.1566
Epoch 12/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.3553 -
acc: 0.1709 - val_loss: 2.3966 - val_acc: 0.1561
Validation Log Loss of  Model in Current Run:  2.3904335856160874
Train on 63448 samples, validate on 11197 samples
Epoch 1/20
63448/63448 [==============================] - 4s 68us/step - loss: 2.8901 -
acc: 0.1050 - val_loss: 2.4086 - val_acc: 0.1483
Epoch 2/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.4851 -
acc: 0.1304 - val_loss: 2.3965 - val_acc: 0.1503
Epoch 3/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.4164 -
acc: 0.1452 - val_loss: 2.3923 - val_acc: 0.1521
Epoch 4/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.3988 -
acc: 0.1511 - val_loss: 2.3910 - val_acc: 0.1545
Epoch 5/20
63448/63448 [==============================] - 4s 57us/step - loss: 2.3909 -
acc: 0.1551 - val_loss: 2.3895 - val_acc: 0.1581
Epoch 6/20
63448/63448 [==============================] - 7s 111us/step - loss: 2.3839 -
acc: 0.1602 - val_loss: 2.3875 - val_acc: 0.1590
```

```
Epoch 7/20
63448/63448 [==============================] - 14s 220us/step - loss: 2.3763
- acc: 0.1614 - val_loss: 2.3882 - val_acc: 0.1587
Epoch 8/20
63448/63448 [==============================] - 11s 173us/step - loss: 2.3712
- acc: 0.1650 - val_loss: 2.3906 - val_acc: 0.1540
Epoch 9/20
63448/63448 [==============================] - 13s 205us/step - loss: 2.3678
- acc: 0.1648 - val_loss: 2.3915 - val_acc: 0.1565
Epoch 10/20
63448/63448 [==============================] - 8s 119us/step - loss: 2.3617 -
acc: 0.1676 - val_loss: 2.3943 - val_acc: 0.1588
Epoch 11/20
63448/63448 [==============================] - 4s 70us/step - loss: 2.3578 -
acc: 0.1698 - val_loss: 2.3937 - val_acc: 0.1561
Validation Log Loss of  Model in Current Run:  2.387496891607927
Train on 63448 samples, validate on 11197 samples
Epoch 1/20
63448/63448 [==============================] - 5s 75us/step - loss: 2.9050 -
acc: 0.1012 - val_loss: 2.4104 - val_acc: 0.1475
Epoch 2/20
63448/63448 [==============================] - 5s 74us/step - loss: 2.4828 -
acc: 0.1306 - val_loss: 2.3929 - val_acc: 0.1573
Epoch 3/20
63448/63448 [==============================] - 5s 72us/step - loss: 2.4167 -
acc: 0.1447 - val_loss: 2.3902 - val_acc: 0.1551
Epoch 4/20
63448/63448 [==============================] - 4s 64us/step - loss: 2.3989 -
acc: 0.1521 - val_loss: 2.3881 - val_acc: 0.1549
Epoch 5/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3903 -
acc: 0.1558 - val_loss: 2.3872 - val_acc: 0.1529
Epoch 6/20
63448/63448 [==============================] - 4s 66us/step - loss: 2.3836 -
acc: 0.1597 - val_loss: 2.3859 - val_acc: 0.1531
Epoch 7/20
63448/63448 [==============================] - 4s 64us/step - loss: 2.3769 -
acc: 0.1631 - val_loss: 2.3872 - val_acc: 0.1536
Epoch 8/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3725 -
acc: 0.1632 - val_loss: 2.3876 - val_acc: 0.1551
Epoch 9/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3677 -
acc: 0.1644 - val_loss: 2.3884 - val_acc: 0.1557
Epoch 10/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3629 -
acc: 0.1673 - val_loss: 2.3891 - val_acc: 0.1536
Epoch 11/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3584 -
acc: 0.1707 - val_loss: 2.3897 - val_acc: 0.1515
Validation Log Loss of  Model in Current Run:  2.385876258398855
Train on 63448 samples, validate on 11197 samples
Epoch 1/20
63448/63448 [==============================] - 5s 77us/step - loss: 2.9246 -
acc: 0.1037 - val_loss: 2.4114 - val_acc: 0.1472
Epoch 2/20
63448/63448 [==============================] - 4s 60us/step - loss: 2.4911 -
```

```
acc: 0.1277 - val_loss: 2.3947 - val_acc: 0.1546
Epoch 3/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.4185 -
acc: 0.1449 - val_loss: 2.3922 - val_acc: 0.1599
Epoch 4/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3997 -
acc: 0.1498 - val_loss: 2.3902 - val_acc: 0.1589
Epoch 5/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3904 -
acc: 0.1559 - val_loss: 2.3872 - val_acc: 0.1592
Epoch 6/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3835 -
acc: 0.1580 - val_loss: 2.3859 - val_acc: 0.1575
Epoch 7/20
63448/63448 [==============================] - 4s 60us/step - loss: 2.3770 -
acc: 0.1630 - val_loss: 2.3859 - val_acc: 0.1604
Epoch 8/20
63448/63448 [==============================] - 4s 62us/step - loss: 2.3712 -
acc: 0.1643 - val_loss: 2.3878 - val_acc: 0.1576
Epoch 9/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3672 -
acc: 0.1657 - val_loss: 2.3892 - val_acc: 0.1540
Epoch 10/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3621 -
acc: 0.1685 - val_loss: 2.3910 - val_acc: 0.1550
Epoch 11/20
63448/63448 [==============================] - 4s 60us/step - loss: 2.3561 -
acc: 0.1702 - val_loss: 2.3938 - val_acc: 0.1551
Epoch 12/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3556 -
acc: 0.1705 - val_loss: 2.3955 - val_acc: 0.1526
Validation Log Loss of  Model in Current Run:  2.3858903664887627
Train on 63448 samples, validate on 11197 samples
Epoch 1/20
63448/63448 [==============================] - 5s 74us/step - loss: 2.8879 -
acc: 0.1052 - val_loss: 2.4110 - val_acc: 0.1477
Epoch 2/20
63448/63448 [==============================] - 4s 64us/step - loss: 2.4786 -
acc: 0.1328 - val_loss: 2.3959 - val_acc: 0.1496
Epoch 3/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.4162 -
acc: 0.1452 - val_loss: 2.3928 - val_acc: 0.1547
Epoch 4/20
63448/63448 [==============================] - 4s 62us/step - loss: 2.3994 -
acc: 0.1505 - val_loss: 2.3921 - val_acc: 0.1572
Epoch 5/20
63448/63448 [==============================] - 4s 61us/step - loss: 2.3888 -
acc: 0.1569 - val_loss: 2.3900 - val_acc: 0.1582
Epoch 6/20
63448/63448 [==============================] - 4s 60us/step - loss: 2.3838 -
acc: 0.1582 - val_loss: 2.3888 - val_acc: 0.1557
Epoch 7/20
63448/63448 [==============================] - 4s 64us/step - loss: 2.3790 -
acc: 0.1607 - val_loss: 2.3890 - val_acc: 0.1558
Epoch 8/20
63448/63448 [==============================] - 4s 64us/step - loss: 2.3711 -
acc: 0.1645 - val_loss: 2.3896 - val_acc: 0.1553
```

```
Epoch 9/20
63448/63448 [==============================] - 4s 60us/step - loss: 2.3689 -
acc: 0.1646 - val_loss: 2.3901 - val_acc: 0.1586
Epoch 10/20
63448/63448 [==============================] - 4s 62us/step - loss: 2.3624 -
acc: 0.1682 - val_loss: 2.3915 - val_acc: 0.1549
Epoch 11/20
63448/63448 [==============================] - 4s 58us/step - loss: 2.3586 -
acc: 0.1698 - val_loss: 2.3907 - val_acc: 0.1541
Validation Log Loss of  Model in Current Run:  2.3887528186292855
Average CV Loss of 5 Runs : 2.3876899841481833
```

In [52]:

```
avg_pred=np.zeros((xtr.shape[0],12))
for i in range(len(model_list_1)):
    train_pred=model_list_1[i].predict_proba(xtr)
    avg_pred+=train_pred
avg_pred/=len(model_list_1)
print("Train Average Log-Loss: ",log_loss(ytr, avg_pred))
```

```
Train Average Log-Loss:  2.3528577585587853
```

In [53]:

```
avg_pred=np.zeros((xcv.shape[0],12))
for i in range(len(model_list_1)):
    cv_pred=model_list_1[i].predict_proba(xcv)
    avg_pred+=cv_pred
avg_pred/=len(model_list_1)
print("Validation Average Log-Loss: ",log_loss(ycv, avg_pred))
```

```
Validation Average Log-Loss:  2.3577544682106013
```

In [54]:

```
avg_pred=np.zeros((xtest_noevents.shape[0],12))
for i in range(len(model_list_1)):
    test_pred=model_list_1[i].predict_proba(xtest_noevents)
    avg_pred+=test_pred
avg_pred/=len(model_list_1)
```

In [55]:

```
#saving the model
np.save('nn1_noevents_1',avg_pred)
```

OBSERVATIONS: USING NEURAL NETWORK WE GOT TRAIN LOSS OF 2.3528 AND TEST LOSS OF 2.3577

MODEL 2

In [56]:
```
#https://www.kaggle.com/c/talkingdata-mobile-user-demographics/discussion/2342
4
def noevents_nn_model2(input_dim,output_dim, learRate=0.0025):

    model = Sequential()
    model.add(Dense(500, input_shape=(input_dim,), init='uniform'))
    model.add(PReLU(init='zero'))
    model.add(Dropout(0.82))
    model.add(Dense(output_dim, init='uniform'))
    model.add(Activation('softmax'))
    opt = Adagrad(lr=learRate, epsilon=1e-08)
    model.compile(loss='categorical_crossentropy',
                  optimizer=opt,
                  metrics=['accuracy'])
    return model
```

In [57]:
```
model_sum=noevents_nn_model2(xtr.shape[1],12)
model_sum.summary()
```

```
W0408 12:35:07.644370 11128 nn_ops.py:4224] Large dropout rate: 0.82 (>0.5).
In TensorFlow 2.x, dropout() uses dropout rate instead of keep_prob. Please e
nsure that this is intended.

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_19 (Dense)             (None, 500)               899500
_________________________________________________________________
p_re_lu_13 (PReLU)           (None, 500)               500
_________________________________________________________________
dropout_13 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_20 (Dense)             (None, 12)                6012
_________________________________________________________________
activation_7 (Activation)    (None, 12)                0
=================================================================
Total params: 906,012
Trainable params: 906,012
Non-trainable params: 0
_________________________________________________________________
```

In [58]:
```
early_stop=EarlyStopping(monitor='val_loss',patience=5,restore_best_weights=Tr
ue)
```

In [59]:
```
def noevents_average_nn_2(state):
    """
    Takes a list of Random Seeds, splits the data into Train and CV based on S
eed, trains model and takes average of
    predictions while testing
    """
    model_list=[]
    loss_list=[]
    avg_cv_loss=0
    for i in range(len(state)):
        xtr, xcv, ytr, ycv = train_test_split(Xtrain_whole, y,stratify=y,test_
size=0.15,random_state=state[i])
        ytr=np_utils.to_categorical(ytr)
        ycv=np_utils.to_categorical(ycv)
        model=noevents_nn_model2(xtr.shape[1],12)
        #logdir = os.path.join("logs","noevents_nn1."+str(i+1))
        #t_callback=TensorBoard(log_dir=logdir)
        model.fit(xtr, ytr, batch_size=256, epochs=30, verbose=1, validation_d
ata=(xcv, ycv),callbacks=[early_stop])
        pred=model.predict_proba(xcv)
        cv_loss=log_loss(ycv, pred)
        print("Validation Log Loss of  Model in Current Run: ",cv_loss)
        model_list.append(model)
        loss_list.append(cv_loss)
    avg_cv_loss=mean(loss_list)
    print("Average CV Loss of "+str(len(state))+" Runs :",avg_cv_loss)
    return(model_list)
```

In [60]:
```
random_seeds=[9,18,42,86,103]
model_list_2= noevents_average_nn_2(random_seeds)
```

```
W0408 12:35:07.760055 11128 nn_ops.py:4224] Large dropout rate: 0.82 (>0.5).
In TensorFlow 2.x, dropout() uses dropout rate instead of keep_prob. Please e
nsure that this is intended.
```

```
Train on 63448 samples, validate on 11197 samples
Epoch 1/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.4315 -
acc: 0.1345 - val_loss: 2.4168 - val_acc: 0.1427
Epoch 2/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.4143 -
acc: 0.1416 - val_loss: 2.4114 - val_acc: 0.1440
Epoch 3/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4089 -
acc: 0.1452 - val_loss: 2.4081 - val_acc: 0.1444
Epoch 4/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4047 -
acc: 0.1462 - val_loss: 2.4059 - val_acc: 0.1466
Epoch 5/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4024 -
acc: 0.1482 - val_loss: 2.4042 - val_acc: 0.1478
Epoch 6/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4006 -
acc: 0.1497 - val_loss: 2.4030 - val_acc: 0.1485
Epoch 7/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3989 -
acc: 0.1504 - val_loss: 2.4020 - val_acc: 0.1499
Epoch 8/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.3977 -
acc: 0.1518 - val_loss: 2.4011 - val_acc: 0.1501
Epoch 9/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.3956 -
acc: 0.1528 - val_loss: 2.4004 - val_acc: 0.1501
Epoch 10/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3954 -
acc: 0.1534 - val_loss: 2.3998 - val_acc: 0.1496
Epoch 11/30
63448/63448 [==============================] - 4s 64us/step - loss: 2.3932 -
acc: 0.1557 - val_loss: 2.3993 - val_acc: 0.1508
Epoch 12/30
63448/63448 [==============================] - 4s 64us/step - loss: 2.3928 -
acc: 0.1564 - val_loss: 2.3988 - val_acc: 0.1516
Epoch 13/30
63448/63448 [==============================] - 4s 63us/step - loss: 2.3921 -
acc: 0.1562 - val_loss: 2.3984 - val_acc: 0.1527
Epoch 14/30
63448/63448 [==============================] - 4s 63us/step - loss: 2.3909 -
acc: 0.1576 - val_loss: 2.3980 - val_acc: 0.1525
Epoch 15/30
63448/63448 [==============================] - 4s 64us/step - loss: 2.3898 -
acc: 0.1600 - val_loss: 2.3976 - val_acc: 0.1522
Epoch 16/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.3885 -
acc: 0.1568 - val_loss: 2.3972 - val_acc: 0.1523
Epoch 17/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3881 -
acc: 0.1580 - val_loss: 2.3969 - val_acc: 0.1525
Epoch 18/30
63448/63448 [==============================] - 4s 71us/step - loss: 2.3882 -
acc: 0.1584 - val_loss: 2.3966 - val_acc: 0.1522
Epoch 19/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3876 -
```

```
acc: 0.1594 - val_loss: 2.3963 - val_acc: 0.1520
Epoch 20/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.3863 -
acc: 0.1595 - val_loss: 2.3961 - val_acc: 0.1520
Epoch 21/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3856 -
acc: 0.1593 - val_loss: 2.3958 - val_acc: 0.1523
Epoch 22/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.3851 -
acc: 0.1607 - val_loss: 2.3955 - val_acc: 0.1541
Epoch 23/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3848 -
acc: 0.1607 - val_loss: 2.3953 - val_acc: 0.1546
Epoch 24/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3851 -
acc: 0.1634 - val_loss: 2.3951 - val_acc: 0.1545
Epoch 25/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3834 -
acc: 0.1618 - val_loss: 2.3949 - val_acc: 0.1544
Epoch 26/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3830 -
acc: 0.1630 - val_loss: 2.3947 - val_acc: 0.1545
Epoch 27/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3818 -
acc: 0.1638 - val_loss: 2.3945 - val_acc: 0.1545
Epoch 28/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3822 -
acc: 0.1633 - val_loss: 2.3943 - val_acc: 0.1537
Epoch 29/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3822 -
acc: 0.1627 - val_loss: 2.3941 - val_acc: 0.1533
Epoch 30/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.3809 -
acc: 0.1627 - val_loss: 2.3939 - val_acc: 0.1535

W0408 12:37:17.340040 11128 nn_ops.py:4224] Large dropout rate: 0.82 (>0.5).
In TensorFlow 2.x, dropout() uses dropout rate instead of keep_prob. Please e
nsure that this is intended.
```

```
Validation Log Loss of  Model in Current Run:  2.393906689524874
Train on 63448 samples, validate on 11197 samples
Epoch 1/30
63448/63448 [==============================] - 5s 77us/step - loss: 2.4322 -
acc: 0.1355 - val_loss: 2.4170 - val_acc: 0.1474
Epoch 2/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.4155 -
acc: 0.1423 - val_loss: 2.4108 - val_acc: 0.1508
Epoch 3/30
63448/63448 [==============================] - 4s 71us/step - loss: 2.4097 -
acc: 0.1463 - val_loss: 2.4070 - val_acc: 0.1512
Epoch 4/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.4068 -
acc: 0.1459 - val_loss: 2.4043 - val_acc: 0.1525
Epoch 5/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4044 -
acc: 0.1461 - val_loss: 2.4023 - val_acc: 0.1523
Epoch 6/30
63448/63448 [==============================] - 5s 73us/step - loss: 2.4014 -
acc: 0.1494 - val_loss: 2.4007 - val_acc: 0.1519
Epoch 7/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3996 -
acc: 0.1514 - val_loss: 2.3995 - val_acc: 0.1533
Epoch 8/30
63448/63448 [==============================] - 4s 64us/step - loss: 2.3984 -
acc: 0.1524 - val_loss: 2.3984 - val_acc: 0.1539
Epoch 9/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3969 -
acc: 0.1536 - val_loss: 2.3975 - val_acc: 0.1550
Epoch 10/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3961 -
acc: 0.1511 - val_loss: 2.3968 - val_acc: 0.1567
Epoch 11/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3939 -
acc: 0.1546 - val_loss: 2.3962 - val_acc: 0.1563
Epoch 12/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3929 -
acc: 0.1534 - val_loss: 2.3956 - val_acc: 0.1554
Epoch 13/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3924 -
acc: 0.1554 - val_loss: 2.3951 - val_acc: 0.1560
Epoch 14/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3922 -
acc: 0.1543 - val_loss: 2.3946 - val_acc: 0.1568
Epoch 15/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3905 -
acc: 0.1583 - val_loss: 2.3942 - val_acc: 0.1560
Epoch 16/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3901 -
acc: 0.1568 - val_loss: 2.3938 - val_acc: 0.1558
Epoch 17/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3892 -
acc: 0.1580 - val_loss: 2.3935 - val_acc: 0.1567
Epoch 18/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3884 -
acc: 0.1583 - val_loss: 2.3931 - val_acc: 0.1566
Epoch 19/30
```

```
63448/63448 [==============================] - 4s 66us/step - loss: 2.3877 -
acc: 0.1578 - val_loss: 2.3928 - val_acc: 0.1570
Epoch 20/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3873 -
acc: 0.1599 - val_loss: 2.3926 - val_acc: 0.1569
Epoch 21/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3859 -
acc: 0.1611 - val_loss: 2.3923 - val_acc: 0.1565
Epoch 22/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3873 -
acc: 0.1582 - val_loss: 2.3920 - val_acc: 0.1565
Epoch 23/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3864 -
acc: 0.1599 - val_loss: 2.3918 - val_acc: 0.1566
Epoch 24/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3846 -
acc: 0.1597 - val_loss: 2.3916 - val_acc: 0.1572
Epoch 25/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3843 -
acc: 0.1605 - val_loss: 2.3914 - val_acc: 0.1581
Epoch 26/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3839 -
acc: 0.1634 - val_loss: 2.3912 - val_acc: 0.1582
Epoch 27/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3829 -
acc: 0.1626 - val_loss: 2.3910 - val_acc: 0.1579
Epoch 28/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3831 -
acc: 0.1630 - val_loss: 2.3908 - val_acc: 0.1576
Epoch 29/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3822 -
acc: 0.1622 - val_loss: 2.3906 - val_acc: 0.1583
Epoch 30/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3822 -
acc: 0.1615 - val_loss: 2.3904 - val_acc: 0.1580

W0408 12:39:25.408527 11128 nn_ops.py:4224] Large dropout rate: 0.82 (>0.5).
In TensorFlow 2.x, dropout() uses dropout rate instead of keep_prob. Please e
nsure that this is intended.
```

```
Validation Log Loss of  Model in Current Run:  2.390445421803921
Train on 63448 samples, validate on 11197 samples
Epoch 1/30
63448/63448 [==============================] - 5s 79us/step - loss: 2.4323 -
acc: 0.1345 - val_loss: 2.4160 - val_acc: 0.1410
Epoch 2/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4156 -
acc: 0.1418 - val_loss: 2.4098 - val_acc: 0.1418
Epoch 3/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.4095 -
acc: 0.1459 - val_loss: 2.4060 - val_acc: 0.1454
Epoch 4/30
63448/63448 [==============================] - 4s 64us/step - loss: 2.4062 -
acc: 0.1464 - val_loss: 2.4033 - val_acc: 0.1460
Epoch 5/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4031 -
acc: 0.1479 - val_loss: 2.4013 - val_acc: 0.1468
Epoch 6/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4017 -
acc: 0.1489 - val_loss: 2.3997 - val_acc: 0.1474
Epoch 7/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4003 -
acc: 0.1505 - val_loss: 2.3984 - val_acc: 0.1482
Epoch 8/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3980 -
acc: 0.1534 - val_loss: 2.3973 - val_acc: 0.1498
Epoch 9/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3969 -
acc: 0.1518 - val_loss: 2.3963 - val_acc: 0.1509
Epoch 10/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3955 -
acc: 0.1547 - val_loss: 2.3955 - val_acc: 0.1514
Epoch 11/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.3948 -
acc: 0.1541 - val_loss: 2.3948 - val_acc: 0.1516s: 2.3950 - acc: 0.
Epoch 12/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.3936 -
acc: 0.1543 - val_loss: 2.3942 - val_acc: 0.1518
Epoch 13/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3925 -
acc: 0.1568 - val_loss: 2.3936 - val_acc: 0.1534
Epoch 14/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.3918 -
acc: 0.1557 - val_loss: 2.3930 - val_acc: 0.1522
Epoch 15/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3915 -
acc: 0.1572 - val_loss: 2.3926 - val_acc: 0.1528
Epoch 16/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3897 -
acc: 0.1594 - val_loss: 2.3921 - val_acc: 0.1540
Epoch 17/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3894 -
acc: 0.1590 - val_loss: 2.3917 - val_acc: 0.1548
Epoch 18/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3887 -
acc: 0.1588 - val_loss: 2.3914 - val_acc: 0.1552
Epoch 19/30
```

```
63448/63448 [==============================] - 4s 65us/step - loss: 2.3883 -
acc: 0.1582 - val_loss: 2.3910 - val_acc: 0.1558
Epoch 20/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3873 -
acc: 0.1610 - val_loss: 2.3906 - val_acc: 0.1564
Epoch 21/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3872 -
acc: 0.1605 - val_loss: 2.3903 - val_acc: 0.1565
Epoch 22/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3861 -
acc: 0.1602 - val_loss: 2.3900 - val_acc: 0.1562
Epoch 23/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3858 -
acc: 0.1587 - val_loss: 2.3898 - val_acc: 0.1563
Epoch 24/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3856 -
acc: 0.1604 - val_loss: 2.3895 - val_acc: 0.1570
Epoch 25/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3845 -
acc: 0.1617 - val_loss: 2.3893 - val_acc: 0.1571
Epoch 26/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3846 -
acc: 0.1613 - val_loss: 2.3890 - val_acc: 0.1575
Epoch 27/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.3841 -
acc: 0.1627 - val_loss: 2.3888 - val_acc: 0.1587
Epoch 28/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3824 -
acc: 0.1614 - val_loss: 2.3886 - val_acc: 0.1587
Epoch 29/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3823 -
acc: 0.1623 - val_loss: 2.3884 - val_acc: 0.1588
Epoch 30/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3826 -
acc: 0.1621 - val_loss: 2.3882 - val_acc: 0.1591

W0408 12:41:34.231742 11128 nn_ops.py:4224] Large dropout rate: 0.82 (>0.5).
In TensorFlow 2.x, dropout() uses dropout rate instead of keep_prob. Please e
nsure that this is intended.
```

```
Validation Log Loss of  Model in Current Run:  2.3882187361524307
Train on 63448 samples, validate on 11197 samples
Epoch 1/30
63448/63448 [==============================] - 5s 81us/step - loss: 2.4325 -
acc: 0.1353 - val_loss: 2.4162 - val_acc: 0.1448
Epoch 2/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.4151 -
acc: 0.1429 - val_loss: 2.4099 - val_acc: 0.1463
Epoch 3/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4093 -
acc: 0.1448 - val_loss: 2.4060 - val_acc: 0.1465
Epoch 4/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4068 -
acc: 0.1465 - val_loss: 2.4033 - val_acc: 0.1468
Epoch 5/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4035 -
acc: 0.1489 - val_loss: 2.4012 - val_acc: 0.1498
Epoch 6/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4012 -
acc: 0.1497 - val_loss: 2.3996 - val_acc: 0.1523
Epoch 7/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3992 -
acc: 0.1519 - val_loss: 2.3983 - val_acc: 0.1525
Epoch 8/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3977 -
acc: 0.1509 - val_loss: 2.3973 - val_acc: 0.1539
Epoch 9/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3959 -
acc: 0.1526 - val_loss: 2.3964 - val_acc: 0.1549
Epoch 10/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3949 -
acc: 0.1536 - val_loss: 2.3956 - val_acc: 0.1552
Epoch 11/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3938 -
acc: 0.1560 - val_loss: 2.3950 - val_acc: 0.1560
Epoch 12/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3935 -
acc: 0.1548 - val_loss: 2.3944 - val_acc: 0.1553
Epoch 13/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3930 -
acc: 0.1545 - val_loss: 2.3939 - val_acc: 0.1563
Epoch 14/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3908 -
acc: 0.1554 - val_loss: 2.3934 - val_acc: 0.1558
Epoch 15/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3902 -
acc: 0.1581 - val_loss: 2.3930 - val_acc: 0.1560
Epoch 16/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3907 -
acc: 0.1572 - val_loss: 2.3926 - val_acc: 0.1565
Epoch 17/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3891 -
acc: 0.1579 - val_loss: 2.3923 - val_acc: 0.1558
Epoch 18/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3891 -
acc: 0.1571 - val_loss: 2.3920 - val_acc: 0.1569
Epoch 19/30
```

```
63448/63448 [==============================] - 4s 67us/step - loss: 2.3878 -
acc: 0.1584 - val_loss: 2.3917 - val_acc: 0.1570
Epoch 20/30
63448/63448 [==============================] - 4s 67us/step - loss: 2.3874 -
acc: 0.1569 - val_loss: 2.3914 - val_acc: 0.1567
Epoch 21/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3863 -
acc: 0.1608 - val_loss: 2.3911 - val_acc: 0.1574
Epoch 22/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3853 -
acc: 0.1600 - val_loss: 2.3909 - val_acc: 0.1583
Epoch 23/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3855 -
acc: 0.1595 - val_loss: 2.3907 - val_acc: 0.1596
Epoch 24/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3846 -
acc: 0.1606 - val_loss: 2.3904 - val_acc: 0.1591
Epoch 25/30
63448/63448 [==============================] - 4s 69us/step - loss: 2.3842 -
acc: 0.1606 - val_loss: 2.3902 - val_acc: 0.1595
Epoch 26/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3839 -
acc: 0.1598 - val_loss: 2.3900 - val_acc: 0.1597
Epoch 27/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3826 -
acc: 0.1624 - val_loss: 2.3898 - val_acc: 0.1591
Epoch 28/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3833 -
acc: 0.1607 - val_loss: 2.3897 - val_acc: 0.1592 loss
Epoch 29/30
63448/63448 [==============================] - 4s 68us/step - loss: 2.3827 -
acc: 0.1615 - val_loss: 2.3895 - val_acc: 0.1592
Epoch 30/30
63448/63448 [==============================] - 4s 70us/step - loss: 2.3817 -
acc: 0.1615 - val_loss: 2.3893 - val_acc: 0.1595
Validation Log Loss of  Model in Current Run:  2.389341872515844
Train on 63448 samples, validate on 11197 samples
Epoch 1/30
63448/63448 [==============================] - 5s 82us/step - loss: 2.4303 -
acc: 0.1354 - val_loss: 2.4152 - val_acc: 0.1402
Epoch 2/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.4133 -
acc: 0.1441 - val_loss: 2.4097 - val_acc: 0.1409
Epoch 3/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4086 -
acc: 0.1448 - val_loss: 2.4065 - val_acc: 0.1443
Epoch 4/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.4052 -
acc: 0.1470 - val_loss: 2.4043 - val_acc: 0.1445
Epoch 5/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4020 -
acc: 0.1481 - val_loss: 2.4025 - val_acc: 0.1454
Epoch 6/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.4000 -
acc: 0.1482 - val_loss: 2.4012 - val_acc: 0.1457
Epoch 7/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3986 -
```

```
acc: 0.1515 - val_loss: 2.4002 - val_acc: 0.1467
Epoch 8/30
63448/63448 [==============================] - 4s 65us/step - loss: 2.3976 -
acc: 0.1530 - val_loss: 2.3993 - val_acc: 0.1483
Epoch 9/30
63448/63448 [==============================] - 4s 66us/step - loss: 2.3965 -
acc: 0.1520 - val_loss: 2.3985 - val_acc: 0.1502
Epoch 10/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3947 -
acc: 0.1554 - val_loss: 2.3979 - val_acc: 0.1521
Epoch 11/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.3938 -
acc: 0.1554 - val_loss: 2.3973 - val_acc: 0.1528
Epoch 12/30
63448/63448 [==============================] - 5s 72us/step - loss: 2.3927 -
acc: 0.1571 - val_loss: 2.3967 - val_acc: 0.1532
Epoch 13/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.3918 -
acc: 0.1566 - val_loss: 2.3963 - val_acc: 0.1536
Epoch 14/30
63448/63448 [==============================] - 5s 77us/step - loss: 2.3916 -
acc: 0.1557 - val_loss: 2.3958 - val_acc: 0.1540
Epoch 15/30
63448/63448 [==============================] - 5s 73us/step - loss: 2.3899 -
acc: 0.1583 - val_loss: 2.3954 - val_acc: 0.1534
Epoch 16/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3889 -
acc: 0.1577 - val_loss: 2.3950 - val_acc: 0.1541
Epoch 17/30
63448/63448 [==============================] - 5s 76us/step - loss: 2.3883 -
acc: 0.1594 - val_loss: 2.3946 - val_acc: 0.1545
Epoch 18/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3878 -
acc: 0.1599 - val_loss: 2.3943 - val_acc: 0.1541
Epoch 19/30
63448/63448 [==============================] - 5s 80us/step - loss: 2.3873 -
acc: 0.1597 - val_loss: 2.3940 - val_acc: 0.1550
Epoch 20/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.3867 -
acc: 0.1590 - val_loss: 2.3937 - val_acc: 0.1548
Epoch 21/30
63448/63448 [==============================] - 5s 73us/step - loss: 2.3863 -
acc: 0.1595 - val_loss: 2.3934 - val_acc: 0.1543
Epoch 22/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3854 -
acc: 0.1615 - val_loss: 2.3931 - val_acc: 0.1544
Epoch 23/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3850 -
acc: 0.1609 - val_loss: 2.3929 - val_acc: 0.1545
Epoch 24/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3843 -
acc: 0.1603 - val_loss: 2.3926 - val_acc: 0.1546
Epoch 25/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.3838 -
acc: 0.1611 - val_loss: 2.3924 - val_acc: 0.1550
Epoch 26/30
63448/63448 [==============================] - 5s 76us/step - loss: 2.3832 -
```

```
acc: 0.1623 - val_loss: 2.3922 - val_acc: 0.1547
Epoch 27/30
63448/63448 [==============================] - 5s 74us/step - loss: 2.3831 -
acc: 0.1622 - val_loss: 2.3919 - val_acc: 0.1559
Epoch 28/30
63448/63448 [==============================] - 5s 73us/step - loss: 2.3824 -
acc: 0.1615 - val_loss: 2.3917 - val_acc: 0.1575
Epoch 29/30
63448/63448 [==============================] - 5s 71us/step - loss: 2.3817 -
acc: 0.1609 - val_loss: 2.3915 - val_acc: 0.1576
Epoch 30/30
63448/63448 [==============================] - 5s 75us/step - loss: 2.3814 -
acc: 0.1625 - val_loss: 2.3913 - val_acc: 0.1578
Validation Log Loss of  Model in Current Run:  2.391342581548126
Average CV Loss of 5 Runs : 2.390651060309039
```

In [61]:

```
avg_pred=np.zeros((xtr.shape[0],12))
for i in range(len(model_list_2)):
    train_pred=model_list_2[i].predict_proba(xtr)
    avg_pred+=train_pred
avg_pred/=len(model_list_2)
print("Train Average Log-Loss: ",log_loss(ytr, avg_pred))
```

```
Train Average Log-Loss:  2.3770776429186817
```

In [62]:

```
avg_pred=np.zeros((xcv.shape[0],12))
for i in range(len(model_list_2)):
    cv_pred=model_list_2[i].predict_proba(xcv)
    avg_pred+=cv_pred
avg_pred/=len(model_list_2)
print("Validation Average Log-Loss: ",log_loss(ycv, avg_pred))
```

```
Validation Average Log-Loss:  2.3788470152956647
```

In [63]:

```
avg_pred=np.zeros((xtest_noevents.shape[0],12))
for i in range(len(model_list_2)):
    test_pred=model_list_2[i].predict_proba(xtest_noevents)
    avg_pred+=test_pred
avg_pred/=len(model_list_2)
```

In [64]:

```
#saving the model
np.save('nn2_noevents_1',avg_pred)
```

OBSERVATIONS: 1.THE TRAIN AND VALIDATION LOSS FOR THE MODEL ARE 2.377 AND 2.378 RESPEECTIVELY.

XGBOOST

In [65]:
```
#https://www.kaggle.com/c/talkingdata-mobile-user-demographics/discussion/2342
4
xgb = XGBClassifier(n_estimators=350, n_jobs=-1,learning_rate=0.05, colsample_
bytree=0.7, max_depth=5,subsample=0.7,objective='multi:softprob',num_class=12,
eval_metric='mlogloss')
xgb.fit(xtr, ytr)
#Using Model Calibration
clf = CalibratedClassifierCV(xgb, method="sigmoid")
clf.fit(xtr, ytr)

pred_y=clf.predict_proba(xtr)
print("Train Log Loss :",log_loss(ytr, pred_y))


pred_y=clf.predict_proba(xcv)
print("Validation Log Loss :",log_loss(ycv, pred_y))
```

```
Train Log Loss : 2.3718148085004658
Validation Log Loss : 2.3929110310146204
```

In [66]:
```
no_events_pred_lr=clf.predict_proba(xtest_noevents)
```

In [67]:
```
#saving the model
np.save('xgb_noevents_1.npy',no_events_pred_lr)
```

OBSERVATIONS: THE TRAIN AND VALIDATION LOSS ARE 2.3718 AND 2.3929 RESPECTIVELY. THESE ARE NOT AS GOOD AS THE NEURAL NETWORK MODEL.

MODELLING USING DEVICES WITH EVENTS

ONE HOT ENCODING OF PHONE BRAND

In [45]:
```
gatrain_events['brand']=phone['brand']
gatest_events['brand']=phone['brand']


#https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csr_matrix.html
Xtr_events_brand = csr_matrix((np.ones(gatrain_events.shape[0]), # Number of Rows/Devices
                       (gatrain_events.trainrow, gatrain_events.brand)),shape=(gatrain_events.shape[0],nbrand))
Xte_events_brand = csr_matrix((np.ones(gatest_events.shape[0]), # Number of Rows/Devices
                       (gatest_events.testrow, gatest_events.brand)),shape=(gatest_events.shape[0],nbrand))

print("Train Brand One-hot Shape: ",Xtr_events_brand.shape)
print("Test Brand One-hot Shape: ",Xte_events_brand.shape)
```

```
Train Brand One-hot Shape:  (23309, 131)
Test Brand One-hot Shape:  (35194, 131)
```

ONE HOT ENCODING OF PHONE MODEL

In [46]:
```
gatrain_events['model']=phone['model']
gatest_events['model']=phone['model']

Xtr_events_model = csr_matrix((np.ones(gatrain_events.shape[0]),
                       (gatrain_events.trainrow, gatrain_events.model)),shape=(gatrain_events.shape[0],nmodel))

Xte_events_model = csr_matrix((np.ones(gatest_events.shape[0]),
                       (gatest_events.testrow, gatest_events.model)),shape=(gatest_events.shape[0],nmodel))
print("Train Brand One-hot Shape: ",Xtr_events_model.shape)
print("Test Brand One-hot Shape: ",Xte_events_model.shape)
```

```
Train Brand One-hot Shape:  (23309, 1667)
Test Brand One-hot Shape:  (35194, 1667)
```

ONE HOT ENCODING OF DEVICE APPS

In [47]:
```
#Since the Deviceapps has both train and test columns merged to create Train A
pps One-Hot we will Drop all Nan of Train Row
#Once we remove Nan in Train Rows we will get the Apps in Train Data and we cr
eate CSR Matrix for those rows
d = deviceapps.dropna(subset=['trainrow'])
Xtr_events_app = csr_matrix((np.ones(d.shape[0]), (d.trainrow, d.app)),
                      shape=(gatrain_events.shape[0],napps))

#Since the Deviceapps has both train and test columns merged to create Test Ap
ps One-Hot we will Drop all Nan of Test Row
#Once we remove Nan in Test Rows we will get the Apps in Test Data and we crea
te CSR Matrix for those rows
d = deviceapps.dropna(subset=['testrow'])
Xte_events_app = csr_matrix((np.ones(d.shape[0]), (d.testrow, d.app)),
                      shape=(gatest_events.shape[0],napps))
print("Train Event Apps One-hot Shape: ",Xtr_events_app.shape)
print("Test Event Apps One-hot Shape: ",Xte_events_app.shape)
```

```
Train Event Apps One-hot Shape:  (23309, 19237)
Test Event Apps One-hot Shape:  (35194, 19237)
```

ONE HOT ENCODING OF DEVICE LABELS

In [48]:
```
#Since the Devicelabels has both train and test columns merged to create Train
Labels One-Hot we will Drop all Nan of Train Row
#Once we remove Nan in Train Rows we will get the Labels in Train Data and we
 create CSR Matrix for those rows
d = devicelabels.dropna(subset=['trainrow'])
Xtr_events_labels = csr_matrix((np.ones(d.shape[0]), (d.trainrow, d.label)),
                      shape=(gatrain_events.shape[0],nlabels))
#Since the Devicelabels has both train and test columns merged to create Test
 Labels One-Hot we will Drop all Nan of Test Row
#Once we remove Nan in Test Rows we will get the Labels in Test Data and we cr
eate CSR Matrix for those rows
d = devicelabels.dropna(subset=['testrow'])
Xte_events_labels = csr_matrix((np.ones(d.shape[0]), (d.testrow, d.label)),
                      shape=(gatest_events.shape[0],nlabels))
print("Train Event Labels One-hot Shape: ",Xtr_events_labels.shape)
print("Test Event Labels One-hot Shape: ",Xte_events_labels.shape)
```

```
Train Event Labels One-hot Shape:  (23309, 492)
Test Event Labels One-hot Shape:  (35194, 492)
```

TFIDF FEATURES FOR HOURS

In [49]:
```
gatrain_events["hourjoin"]=gatrain_events.index.map(hourjoin)
gatest_events["hourjoin"]=gatest_events.index.map(hourjoin)

vectorizer=TfidfVectorizer()
vectorizer.fit(gatrain_events['hourjoin'].values)

X_tr_hourjoin_tfidf = vectorizer.transform(gatrain_events['hourjoin'].values)
X_te_hourjoin_tfidf = vectorizer.transform(gatest_events['hourjoin'].values)

print("Train Event Hours TF-IDF Shape: ",X_tr_hourjoin_tfidf.shape)
print("Test Event Hours TF-IDF Shape: ",X_te_hourjoin_tfidf.shape)
```

```
Train Event Hours TF-IDF Shape:  (23309, 24)
Test Event Hours TF-IDF Shape:  (35194, 24)
```

In [50]:
```
import pickle
with open('hour_tfidf','wb') as fp:
    pickle.dump(vectorizer,fp)
```

BOW FOR HOURS

In [51]:
```
gatrain_events["hourjoin"]=gatrain_events.index.map(hourjoin)
gatest_events["hourjoin"]=gatest_events.index.map(hourjoin)

vectorizer=CountVectorizer()
vectorizer.fit(gatrain_events['hourjoin'].values)

X_tr_hourjoin_onehot = vectorizer.transform(gatrain_events['hourjoin'].values)
X_te_hourjoin_onehot = vectorizer.transform(gatest_events['hourjoin'].values)
print("After vectorizations")
print("Train Event Hours One-hot Shape: ",X_tr_hourjoin_onehot.shape)
print("Test Event Hours One-hot Shape: ",X_te_hourjoin_onehot.shape)
```

```
After vectorizations
Train Event Hours One-hot Shape:  (23309, 24)
Test Event Hours One-hot Shape:  (35194, 24)
```

In [52]:
```
import pickle
with open('hour_bow','wb') as fp:
    pickle.dump(vectorizer,fp)
```

ONE HOT ENCODING OF HOUR BIN

In [53]:
```
gatrain_events["hourbinjoin"]=gatrain_events.index.map(hourbinjoin)
gatest_events["hourbinjoin"]=gatest_events.index.map(hourbinjoin)

vectorizer=CountVectorizer(binary=True)
vectorizer.fit(gatrain_events['hourbinjoin'].values)

X_tr_hourbinjoin_onehot = vectorizer.transform(gatrain_events['hourbinjoin'].v
alues)
X_te_hourbinjoin_onehot = vectorizer.transform(gatest_events['hourbinjoin'].va
lues)

print("Train Event Hours One-hot Shape: ",X_tr_hourbinjoin_onehot.shape)
print("Test Event Hours One-hot Shape: ",X_te_hourbinjoin_onehot.shape)
```

```
Train Event Hours One-hot Shape:  (23309, 4)
Test Event Hours One-hot Shape:  (35194, 4)
```

In [54]:
```
import pickle
with open('hour_bin_bow','wb') as fp:
    pickle.dump(vectorizer,fp)
```

TFIDF FEATURES FOR DAY

In [55]:
```
gatrain_events["daysjoin"]=gatrain_events.index.map(daysjoin)
gatest_events["daysjoin"]=gatest_events.index.map(daysjoin)

vectorizer=TfidfVectorizer()
vectorizer.fit(gatrain_events['daysjoin'].values)

X_tr_daysjoin_tfidf = vectorizer.transform(gatrain_events['daysjoin'].values)
X_te_daysjoin_tfidf = vectorizer.transform(gatest_events['daysjoin'].values)
print("After vectorizations")
print("Train Event days TF-IDF Shape: ",X_tr_daysjoin_tfidf.shape)
print("Test Event days TF-IDF Shape: ",X_te_daysjoin_tfidf.shape)
```

```
After vectorizations
Train Event days TF-IDF Shape:  (23309, 7)
Test Event days TF-IDF Shape:  (35194, 7)
```

In [56]:
```
import pickle
with open('day_tfidf','wb') as fp:
    pickle.dump(vectorizer,fp)
```

STANDARDIZING LATITUDE AND LONGITUDE

In [57]:
```
gatrain_events["latitude"]=gatrain_events.index.map(median_lat)
gatest_events["latitude"]=gatest_events.index.map(median_lat)

scaler=StandardScaler()
scaler.fit(gatrain_events['latitude'].values.reshape(-1,1))

X_tr_event_lat = scaler.transform(gatrain_events['latitude'].values.reshape(-1
,1))
X_te_event_lat = scaler.transform(gatest_events['latitude'].values.reshape(-1,
1))

print("Train Event Latitude Standardized Shape: ",X_tr_event_lat.shape)
print("Test Event Latitude Standardized  Shape: ",X_te_event_lat.shape)
```

```
Train Event Latitude Standardized Shape:  (23309, 1)
Test Event Latitude Standardized  Shape:  (35194, 1)
```

In [58]:
```
import pickle
with open('lat_scaler','wb') as fp:
    pickle.dump(scaler,fp)
```

In [59]:
```
gatrain_events["longitude"]=gatrain_events.index.map(median_lon)
gatest_events["longitude"]=gatest_events.index.map(median_lon)

scaler=StandardScaler()
scaler.fit(gatrain_events['longitude'].values.reshape(-1,1))

X_tr_event_lon = scaler.transform(gatrain_events['longitude'].values.reshape(-
1,1))
X_te_event_lon = scaler.transform(gatest_events['longitude'].values.reshape(-1
,1))

print("Train Event longitude Standardized Shape: ",X_tr_event_lon.shape)
print("Test Event longitude Standardized  Shape: ",X_te_event_lon.shape)
```

```
Train Event longitude Standardized Shape:  (23309, 1)
Test Event longitude Standardized  Shape:  (35194, 1)
```

In [60]:
```
import pickle
with open('lon_scaler','wb') as fp:
    pickle.dump(scaler,fp)
```

ONE HOT ENCODING OF CLUSTERED FEATURES

In [61]:
```
gatrain_events["locationbin"]=gatrain_events.index.map(clustered_geo_features)
gatest_events["locationbin"]=gatest_events.index.map(clustered_geo_features)

#gatrain_events.locationbin=gatrain_events.locationbin.astype(str)
#gatest_events.locationbin=gatest_events.locationbin.astype(str)

vectorizer= OneHotEncoder()
vectorizer.fit(gatrain_events['locationbin'].values.reshape(-1,1))

X_tr_clus = vectorizer.transform(gatrain_events['locationbin'].values.reshape(
-1,1))
X_te_clus = vectorizer.transform(gatest_events['locationbin'].values.reshape(-
1,1))

print("Train Event locationbin Shape: ",X_tr_clus.shape)
print("Test Event locationbin Shape: ",X_te_clus.shape)
```

```
Train Event locationbin Shape:  (23309, 10)
Test Event locationbin Shape:  (35194, 10)
```

In [62]:
```
import pickle
with open('clustered_features','wb') as fp:
    pickle.dump(vectorizer,fp)
```

TFIDF FEATURE FOR APP IS_ACTIVE

In [63]:
```
#Mapping The Values the values to train and test dataframes
gatrain_events['apps_active']=gatrain_events.index.map(active_apps_events)
gatest_events['apps_active']=gatest_events.index.map(active_apps_events)

vectorizer=TfidfVectorizer()
vectorizer.fit(gatrain_events['apps_active'].values)

X_tr_active = vectorizer.transform(gatrain_events['apps_active'].values)
X_te_active = vectorizer.transform(gatest_events['apps_active'].values)

print("Train Apps Active TF-IDF Shape: ",X_tr_active.shape)
print("Test Apps Active TF-IDF Shape: ",X_te_active.shape)
```

```
Train Apps Active TF-IDF Shape:  (23309, 2)
Test Apps Active TF-IDF Shape:  (35194, 2)
```

In [64]:
```
import pickle
with open('isactive_tfidf','wb') as fp:
    pickle.dump(vectorizer,fp)
```

In [65]:
```
#creating final data matrix
X_train_events=hstack((Xtr_events_brand,Xtr_events_model,Xtr_events_labels,X_tr_hourjoin_tfidf,X_tr_hourbinjoin_onehot,X_tr_daysjoin_tfidf,X_tr_event_lat,X_tr_event_lon,Xtr_events_app,X_tr_active,X_tr_clus),format='csr')

X_test_events =hstack((Xte_events_brand,Xte_events_model,Xte_events_labels,X_te_hourjoin_tfidf,X_te_hourbinjoin_onehot,X_te_daysjoin_tfidf,X_te_event_lat,X_te_event_lon,Xte_events_app,X_te_active,X_te_clus),format='csr')

print(X_train_events.shape)
print(X_test_events.shape)
```

```
(23309, 21576)
(35194, 21576)
```

In [81]:
```
#label encoding target variable
targetencoder = LabelEncoder().fit(gatrain_events.group)
y = targetencoder.transform(gatrain_events.group)
```

In [82]:
```
print("xtrain shape:",X_train_events.shape)
print("ytrain shape:",y.shape)

print("xtest shape:",X_test_events.shape)
```

```
xtrain shape: (23309, 21576)
ytrain shape: (23309,)
xtest shape: (35194, 21576)
```

In [83]:
```
xtr, xcv, ytr, ycv = train_test_split(X_train_events, y,stratify=y,test_size=0.2,random_state=9)
```

In [84]:
```
#one hot encoding target variable
ytr=np_utils.to_categorical(ytr)
ycv=np_utils.to_categorical(ycv)
```

NEURAL NETWORK 1

In [85]:
```
def events_nn_model1(input_dim,output_dim):
    model = Sequential()
    model.add(Dropout(0.15, input_shape=(input_dim,)))
    model.add(Dense(240, init='uniform'))
    model.add(PReLU(init='zero'))
    model.add(Dropout(0.8))
    model.add(Dense(240, init='uniform'))
    model.add(PReLU(init='zero', weights=None))
    model.add(Dropout(0.35))
    model.add(Dense(260, init='uniform'))
    model.add(PReLU(init='zero', weights=None))
    model.add(Dropout(0.40))
    model.add(Dense(output_dim, init='uniform'))
    model.add(Activation('softmax'))

    opt = Adagrad(lr=0.008, epsilon=1e-08)
    model.compile(loss='categorical_crossentropy',
                  optimizer=opt,
                  metrics=['accuracy'])
    return model
```

In [86]:
```
model_sum=events_nn_model1(xtr.shape[1],12)
model_sum.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dropout_19 (Dropout)         (None, 21576)             0         
_________________________________________________________________
dense_31 (Dense)             (None, 240)               5178480   
_________________________________________________________________
p_re_lu_19 (PReLU)           (None, 240)               240       
_________________________________________________________________
dropout_20 (Dropout)         (None, 240)               0         
_________________________________________________________________
dense_32 (Dense)             (None, 240)               57840     
_________________________________________________________________
p_re_lu_20 (PReLU)           (None, 240)               240       
_________________________________________________________________
dropout_21 (Dropout)         (None, 240)               0         
_________________________________________________________________
dense_33 (Dense)             (None, 260)               62660     
_________________________________________________________________
p_re_lu_21 (PReLU)           (None, 260)               260       
_________________________________________________________________
dropout_22 (Dropout)         (None, 260)               0         
_________________________________________________________________
dense_34 (Dense)             (None, 12)                3132      
_________________________________________________________________
activation_13 (Activation)   (None, 12)                0         
=================================================================
Total params: 5,302,852
Trainable params: 5,302,852
Non-trainable params: 0
_________________________________________________________________
```

In [87]:
```
early_stop=EarlyStopping(monitor='val_loss',patience=5,restore_best_weights=True)
```

In [88]:
```
def events_average_nn_1(state):
    """
    Takes a list of Random Seeds, splits the data into Train and CV based on S
eed, trains model and takes average of
    predictions while testing
    """
    model_list=[]
    loss_list=[]
    avg_cv_loss=0
    for i in range((state)):
        model=events_nn_model1(xtr.shape[1],12)
        model.fit(xtr, ytr, batch_size=149, epochs=20, verbose=1, validation_d
ata=(xcv, ycv),callbacks=[early_stop])
        model.save('saved_models/events/nn1'+str(i+1))
        pred=model.predict_proba(xcv)
        cv_loss=log_loss(ycv, pred)
        print("Validation Log Loss of  Model in Current Run: ",cv_loss)
        model_list.append(model)
        loss_list.append(cv_loss)
    avg_cv_loss=mean(loss_list)
    print("Average CV Loss of "+str((state))+" Runs :",avg_cv_loss)
    return(model_list)
```

In [89]:
```
model_list_2=events_average_nn_1(10)
```

```
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 16s 854us/step - loss: 2.2832
- acc: 0.1883 - val_loss: 2.1078 - val_acc: 0.2602
Epoch 2/20
18647/18647 [==============================] - 15s 821us/step - loss: 2.1294
- acc: 0.2401 - val_loss: 2.0377 - val_acc: 0.2969
Epoch 3/20
18647/18647 [==============================] - 15s 804us/step - loss: 2.0580
- acc: 0.2699 - val_loss: 2.0073 - val_acc: 0.3012
Epoch 4/20
18647/18647 [==============================] - 15s 787us/step - loss: 2.0219
- acc: 0.2880 - val_loss: 1.9709 - val_acc: 0.3215
Epoch 5/20
18647/18647 [==============================] - 15s 792us/step - loss: 1.9912
- acc: 0.2924 - val_loss: 1.9530 - val_acc: 0.3220
Epoch 6/20
18647/18647 [==============================] - 15s 805us/step - loss: 1.9563
- acc: 0.3013 - val_loss: 1.9505 - val_acc: 0.3168
Epoch 7/20
18647/18647 [==============================] - 15s 791us/step - loss: 1.9350
- acc: 0.3127 - val_loss: 1.9464 - val_acc: 0.3220
Epoch 8/20
18647/18647 [==============================] - 15s 798us/step - loss: 1.9187
- acc: 0.3130 - val_loss: 1.9306 - val_acc: 0.3271
Epoch 9/20
18647/18647 [==============================] - 15s 785us/step - loss: 1.8983
- acc: 0.3241 - val_loss: 1.9241 - val_acc: 0.3265
Epoch 10/20
18647/18647 [==============================] - 14s 763us/step - loss: 1.8741
- acc: 0.3339 - val_loss: 1.9207 - val_acc: 0.3275
Epoch 11/20
18647/18647 [==============================] - 14s 767us/step - loss: 1.8543
- acc: 0.3383 - val_loss: 1.9211 - val_acc: 0.3235
Epoch 12/20
18647/18647 [==============================] - 15s 778us/step - loss: 1.8433
- acc: 0.3423 - val_loss: 1.9186 - val_acc: 0.3241
Epoch 13/20
18647/18647 [==============================] - 14s 767us/step - loss: 1.8259
- acc: 0.3439 - val_loss: 1.9126 - val_acc: 0.3263
Epoch 14/20
18647/18647 [==============================] - 14s 765us/step - loss: 1.8118
- acc: 0.3502 - val_loss: 1.9149 - val_acc: 0.3271
Epoch 15/20
18647/18647 [==============================] - 14s 764us/step - loss: 1.7944
- acc: 0.3559 - val_loss: 1.9189 - val_acc: 0.3239
Epoch 16/20
18647/18647 [==============================] - 14s 774us/step - loss: 1.7882
- acc: 0.3608 - val_loss: 1.9151 - val_acc: 0.3275
Epoch 17/20
18647/18647 [==============================] - 14s 769us/step - loss: 1.7690
- acc: 0.3661 - val_loss: 1.9180 - val_acc: 0.3265
Epoch 18/20
18647/18647 [==============================] - 15s 778us/step - loss: 1.7641
- acc: 0.3677 - val_loss: 1.9137 - val_acc: 0.3299
Validation Log Loss of  Model in Current Run:  1.9136059269573138
Train on 18647 samples, validate on 4662 samples
```

```
Epoch 1/20
18647/18647 [==============================] - 16s 876us/step - loss: 2.3051
- acc: 0.1749 - val_loss: 2.1252 - val_acc: 0.2578
Epoch 2/20
18647/18647 [==============================] - 15s 778us/step - loss: 2.1361
- acc: 0.2389 - val_loss: 2.0454 - val_acc: 0.2898
Epoch 3/20
18647/18647 [==============================] - 14s 772us/step - loss: 2.0704
- acc: 0.2671 - val_loss: 2.0079 - val_acc: 0.2997
Epoch 4/20
18647/18647 [==============================] - 14s 776us/step - loss: 2.0264
- acc: 0.2822 - val_loss: 1.9727 - val_acc: 0.3104
Epoch 5/20
18647/18647 [==============================] - 14s 770us/step - loss: 1.9958
- acc: 0.2914 - val_loss: 1.9737 - val_acc: 0.2992
Epoch 6/20
18647/18647 [==============================] - 14s 771us/step - loss: 1.9618
- acc: 0.3011 - val_loss: 1.9551 - val_acc: 0.3142
Epoch 7/20
18647/18647 [==============================] - 15s 787us/step - loss: 1.9387
- acc: 0.3040 - val_loss: 1.9462 - val_acc: 0.3187
Epoch 8/20
18647/18647 [==============================] - 15s 784us/step - loss: 1.9280
- acc: 0.3159 - val_loss: 1.9417 - val_acc: 0.3142
Epoch 9/20
18647/18647 [==============================] - 14s 760us/step - loss: 1.9011
- acc: 0.3202 - val_loss: 1.9305 - val_acc: 0.3263
Epoch 10/20
18647/18647 [==============================] - 14s 774us/step - loss: 1.8765
- acc: 0.3278 - val_loss: 1.9242 - val_acc: 0.3230
Epoch 11/20
18647/18647 [==============================] - 14s 778us/step - loss: 1.8635
- acc: 0.3388 - val_loss: 1.9216 - val_acc: 0.3213
Epoch 12/20
18647/18647 [==============================] - 14s 764us/step - loss: 1.8497
- acc: 0.3354 - val_loss: 1.9225 - val_acc: 0.3230
Epoch 13/20
18647/18647 [==============================] - 14s 767us/step - loss: 1.8411
- acc: 0.3428 - val_loss: 1.9212 - val_acc: 0.3248
Epoch 14/20
18647/18647 [==============================] - 15s 781us/step - loss: 1.8222
- acc: 0.3436 - val_loss: 1.9160 - val_acc: 0.3237
Epoch 15/20
18647/18647 [==============================] - 14s 772us/step - loss: 1.8042
- acc: 0.3552 - val_loss: 1.9194 - val_acc: 0.3256
Epoch 16/20
18647/18647 [==============================] - 14s 763us/step - loss: 1.7896
- acc: 0.3569 - val_loss: 1.9148 - val_acc: 0.3267
Epoch 17/20
18647/18647 [==============================] - 16s 861us/step - loss: 1.7753
- acc: 0.3638 - val_loss: 1.9220 - val_acc: 0.3263
Epoch 18/20
18647/18647 [==============================] - 14s 773us/step - loss: 1.7692
- acc: 0.3698 - val_loss: 1.9197 - val_acc: 0.3245
Epoch 19/20
18647/18647 [==============================] - 14s 769us/step - loss: 1.7475
- acc: 0.3698 - val_loss: 1.9213 - val_acc: 0.3280
```

```
Epoch 20/20
18647/18647 [==============================] - 15s 810us/step - loss: 1.7461
- acc: 0.3712 - val_loss: 1.9354 - val_acc: 0.3170
Validation Log Loss of  Model in Current Run:  1.9367125339164373
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 18s 945us/step - loss: 2.2979
- acc: 0.1819 - val_loss: 2.1559 - val_acc: 0.2606
Epoch 2/20
18647/18647 [==============================] - 16s 863us/step - loss: 2.1412
- acc: 0.2369 - val_loss: 2.0460 - val_acc: 0.2821
Epoch 3/20
18647/18647 [==============================] - 16s 834us/step - loss: 2.0747
- acc: 0.2664 - val_loss: 2.0131 - val_acc: 0.3039
Epoch 4/20
18647/18647 [==============================] - 16s 832us/step - loss: 2.0286
- acc: 0.2797 - val_loss: 1.9821 - val_acc: 0.3091
Epoch 5/20
18647/18647 [==============================] - 15s 816us/step - loss: 1.9968
- acc: 0.2885 - val_loss: 1.9650 - val_acc: 0.3183
Epoch 6/20
18647/18647 [==============================] - 15s 828us/step - loss: 1.9620
- acc: 0.3041 - val_loss: 1.9557 - val_acc: 0.3117
Epoch 7/20
18647/18647 [==============================] - 15s 826us/step - loss: 1.9425
- acc: 0.3070 - val_loss: 1.9472 - val_acc: 0.3164
Epoch 8/20
18647/18647 [==============================] - 16s 837us/step - loss: 1.9183
- acc: 0.3133 - val_loss: 1.9398 - val_acc: 0.3179
Epoch 9/20
18647/18647 [==============================] - 15s 819us/step - loss: 1.8951
- acc: 0.3286 - val_loss: 1.9343 - val_acc: 0.3196
Epoch 10/20
18647/18647 [==============================] - 15s 820us/step - loss: 1.8788
- acc: 0.3281 - val_loss: 1.9219 - val_acc: 0.3237
Epoch 11/20
18647/18647 [==============================] - 15s 818us/step - loss: 1.8619
- acc: 0.3326 - val_loss: 1.9303 - val_acc: 0.3224
Epoch 12/20
18647/18647 [==============================] - 15s 820us/step - loss: 1.8488
- acc: 0.3402 - val_loss: 1.9159 - val_acc: 0.3252
Epoch 13/20
18647/18647 [==============================] - 16s 833us/step - loss: 1.8431
- acc: 0.3404 - val_loss: 1.9160 - val_acc: 0.3256
Epoch 14/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.8144
- acc: 0.3530 - val_loss: 1.9152 - val_acc: 0.3228
Epoch 15/20
18647/18647 [==============================] - 15s 799us/step - loss: 1.8013
- acc: 0.3559 - val_loss: 1.9149 - val_acc: 0.3252
Epoch 16/20
18647/18647 [==============================] - 15s 783us/step - loss: 1.7890
- acc: 0.3582 - val_loss: 1.9145 - val_acc: 0.3280
Epoch 17/20
18647/18647 [==============================] - 15s 792us/step - loss: 1.7784
- acc: 0.3673 - val_loss: 1.9227 - val_acc: 0.3207
Epoch 18/20
```

```
18647/18647 [==============================] - 15s 812us/step - loss: 1.7676
- acc: 0.3645 - val_loss: 1.9206 - val_acc: 0.3248
Epoch 19/20
18647/18647 [==============================] - 15s 816us/step - loss: 1.7520
- acc: 0.3730 - val_loss: 1.9182 - val_acc: 0.3256
Epoch 20/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.7487
- acc: 0.3700 - val_loss: 1.9221 - val_acc: 0.3241
Validation Log Loss of  Model in Current Run:  1.9237789996264
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 16s 877us/step - loss: 2.2848
- acc: 0.1854 - val_loss: 2.1432 - val_acc: 0.2492
Epoch 2/20
18647/18647 [==============================] - 15s 817us/step - loss: 2.1296
- acc: 0.2450 - val_loss: 2.0251 - val_acc: 0.2846
Epoch 3/20
18647/18647 [==============================] - 15s 818us/step - loss: 2.0554
- acc: 0.2733 - val_loss: 1.9900 - val_acc: 0.3140
Epoch 4/20
18647/18647 [==============================] - 15s 824us/step - loss: 2.0230
- acc: 0.2837 - val_loss: 1.9835 - val_acc: 0.3050
Epoch 5/20
18647/18647 [==============================] - 15s 812us/step - loss: 1.9856
- acc: 0.2962 - val_loss: 1.9566 - val_acc: 0.3097
Epoch 6/20
18647/18647 [==============================] - 14s 759us/step - loss: 1.9666
- acc: 0.3058 - val_loss: 1.9504 - val_acc: 0.3164
Epoch 7/20
18647/18647 [==============================] - 14s 770us/step - loss: 1.9340
- acc: 0.3111 - val_loss: 1.9408 - val_acc: 0.3218
Epoch 8/20
18647/18647 [==============================] - 14s 764us/step - loss: 1.9148
- acc: 0.3185 - val_loss: 1.9395 - val_acc: 0.3170
Epoch 9/20
18647/18647 [==============================] - 14s 772us/step - loss: 1.8887
- acc: 0.3255 - val_loss: 1.9258 - val_acc: 0.3263
Epoch 10/20
18647/18647 [==============================] - 14s 752us/step - loss: 1.8752
- acc: 0.3336 - val_loss: 1.9244 - val_acc: 0.3215
Epoch 11/20
18647/18647 [==============================] - 15s 792us/step - loss: 1.8615
- acc: 0.3321 - val_loss: 1.9246 - val_acc: 0.3260
Epoch 12/20
18647/18647 [==============================] - 15s 799us/step - loss: 1.8423
- acc: 0.3433 - val_loss: 1.9187 - val_acc: 0.3271
Epoch 13/20
18647/18647 [==============================] - 15s 778us/step - loss: 1.8385
- acc: 0.3447 - val_loss: 1.9206 - val_acc: 0.3297
Epoch 14/20
18647/18647 [==============================] - 14s 775us/step - loss: 1.8177
- acc: 0.3446 - val_loss: 1.9190 - val_acc: 0.3303
Epoch 15/20
18647/18647 [==============================] - 16s 834us/step - loss: 1.8023
- acc: 0.3534 - val_loss: 1.9267 - val_acc: 0.3286
Epoch 16/20
18647/18647 [==============================] - 15s 807us/step - loss: 1.7851
```

```
- acc: 0.3625 - val_loss: 1.9182 - val_acc: 0.3286
Epoch 17/20
18647/18647 [==============================] - 15s 828us/step - loss: 1.7784
- acc: 0.3649 - val_loss: 1.9229 - val_acc: 0.3265
Epoch 18/20
18647/18647 [==============================] - 16s 859us/step - loss: 1.7729
- acc: 0.3613 - val_loss: 1.9256 - val_acc: 0.3280
Epoch 19/20
18647/18647 [==============================] - 16s 839us/step - loss: 1.7505
- acc: 0.3694 - val_loss: 1.9255 - val_acc: 0.3269
Epoch 20/20
18647/18647 [==============================] - 16s 840us/step - loss: 1.7471
- acc: 0.3700 - val_loss: 1.9282 - val_acc: 0.3290
Validation Log Loss of  Model in Current Run:  1.9296700733280108
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 17s 894us/step - loss: 2.2951
- acc: 0.1793 - val_loss: 2.1393 - val_acc: 0.2435
Epoch 2/20
18647/18647 [==============================] - 15s 805us/step - loss: 2.1336
- acc: 0.2361 - val_loss: 2.0347 - val_acc: 0.2958
Epoch 3/20
18647/18647 [==============================] - 15s 808us/step - loss: 2.0728
- acc: 0.2644 - val_loss: 2.0083 - val_acc: 0.3014
Epoch 4/20
18647/18647 [==============================] - 15s 808us/step - loss: 2.0254
- acc: 0.2768 - val_loss: 1.9843 - val_acc: 0.3093
Epoch 5/20
18647/18647 [==============================] - 15s 812us/step - loss: 1.9930
- acc: 0.2908 - val_loss: 1.9700 - val_acc: 0.3054
Epoch 6/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.9691
- acc: 0.3003 - val_loss: 1.9472 - val_acc: 0.3220
Epoch 7/20
18647/18647 [==============================] - 15s 814us/step - loss: 1.9395
- acc: 0.3113 - val_loss: 1.9389 - val_acc: 0.3220
Epoch 8/20
18647/18647 [==============================] - 15s 815us/step - loss: 1.9196
- acc: 0.3163 - val_loss: 1.9315 - val_acc: 0.3185
Epoch 9/20
18647/18647 [==============================] - 15s 814us/step - loss: 1.9017
- acc: 0.3158 - val_loss: 1.9310 - val_acc: 0.3200
Epoch 10/20
18647/18647 [==============================] - 15s 821us/step - loss: 1.8796
- acc: 0.3274 - val_loss: 1.9207 - val_acc: 0.3284
Epoch 11/20
18647/18647 [==============================] - 16s 846us/step - loss: 1.8728
- acc: 0.3352 - val_loss: 1.9191 - val_acc: 0.3239
Epoch 12/20
18647/18647 [==============================] - 15s 814us/step - loss: 1.8577
- acc: 0.3363 - val_loss: 1.9195 - val_acc: 0.3245
Epoch 13/20
18647/18647 [==============================] - 16s 836us/step - loss: 1.8376
- acc: 0.3419 - val_loss: 1.9163 - val_acc: 0.3220
Epoch 14/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.8172
- acc: 0.3492 - val_loss: 1.9152 - val_acc: 0.3256
```

```
Epoch 15/20
18647/18647 [==============================] - 15s 789us/step - loss: 1.8079
- acc: 0.3524 - val_loss: 1.9180 - val_acc: 0.3245
Epoch 16/20
18647/18647 [==============================] - 15s 797us/step - loss: 1.7937
- acc: 0.3543 - val_loss: 1.9127 - val_acc: 0.3252
Epoch 17/20
18647/18647 [==============================] - 15s 808us/step - loss: 1.7825
- acc: 0.3616 - val_loss: 1.9160 - val_acc: 0.3198
Epoch 18/20
18647/18647 [==============================] - 15s 819us/step - loss: 1.7666
- acc: 0.3665 - val_loss: 1.9178 - val_acc: 0.3224
Epoch 19/20
18647/18647 [==============================] - 15s 827us/step - loss: 1.7566
- acc: 0.3685 - val_loss: 1.9203 - val_acc: 0.3237
Epoch 20/20
18647/18647 [==============================] - 15s 804us/step - loss: 1.7386
- acc: 0.3699 - val_loss: 1.9194 - val_acc: 0.3181
Validation Log Loss of  Model in Current Run:  1.9203065615975894
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 17s 891us/step - loss: 2.2866
- acc: 0.1833 - val_loss: 2.1179 - val_acc: 0.2531
Epoch 2/20
18647/18647 [==============================] - 14s 759us/step - loss: 2.1303
- acc: 0.2436 - val_loss: 2.0570 - val_acc: 0.2999
Epoch 3/20
18647/18647 [==============================] - 14s 757us/step - loss: 2.0682
- acc: 0.2684 - val_loss: 2.0122 - val_acc: 0.2956
Epoch 4/20
18647/18647 [==============================] - 15s 788us/step - loss: 2.0270
- acc: 0.2828 - val_loss: 1.9793 - val_acc: 0.3085
Epoch 5/20
18647/18647 [==============================] - 14s 768us/step - loss: 1.9897
- acc: 0.2916 - val_loss: 1.9699 - val_acc: 0.3074
Epoch 6/20
18647/18647 [==============================] - 15s 791us/step - loss: 1.9645
- acc: 0.3028 - val_loss: 1.9496 - val_acc: 0.3175
Epoch 7/20
18647/18647 [==============================] - 15s 804us/step - loss: 1.9400
- acc: 0.3117 - val_loss: 1.9357 - val_acc: 0.3226
Epoch 8/20
18647/18647 [==============================] - 15s 809us/step - loss: 1.9176
- acc: 0.3162 - val_loss: 1.9444 - val_acc: 0.3202
Epoch 9/20
18647/18647 [==============================] - 16s 839us/step - loss: 1.8963
- acc: 0.3256 - val_loss: 1.9258 - val_acc: 0.3220
Epoch 10/20
18647/18647 [==============================] - 16s 849us/step - loss: 1.8884
- acc: 0.3282 - val_loss: 1.9217 - val_acc: 0.3260
Epoch 11/20
18647/18647 [==============================] - 16s 842us/step - loss: 1.8682
- acc: 0.3336 - val_loss: 1.9173 - val_acc: 0.3220
Epoch 12/20
18647/18647 [==============================] - 16s 856us/step - loss: 1.8451
- acc: 0.3406 - val_loss: 1.9168 - val_acc: 0.3260
Epoch 13/20
```

```
18647/18647 [==============================] - 16s 836us/step - loss: 1.8263
- acc: 0.3428 - val_loss: 1.9159 - val_acc: 0.3299
Epoch 14/20
18647/18647 [==============================] - 16s 842us/step - loss: 1.8232
- acc: 0.3477 - val_loss: 1.9157 - val_acc: 0.3310
Epoch 15/20
18647/18647 [==============================] - 16s 877us/step - loss: 1.8113
- acc: 0.3492 - val_loss: 1.9182 - val_acc: 0.3248
Epoch 16/20
18647/18647 [==============================] - 16s 841us/step - loss: 1.7985
- acc: 0.3532 - val_loss: 1.9158 - val_acc: 0.3299
Epoch 17/20
18647/18647 [==============================] - 16s 835us/step - loss: 1.7862
- acc: 0.3604 - val_loss: 1.9173 - val_acc: 0.3252
Epoch 18/20
18647/18647 [==============================] - 16s 839us/step - loss: 1.7741
- acc: 0.3606 - val_loss: 1.9219 - val_acc: 0.3318
Epoch 19/20
18647/18647 [==============================] - 15s 820us/step - loss: 1.7576
- acc: 0.3723 - val_loss: 1.9194 - val_acc: 0.3312
Validation Log Loss of  Model in Current Run:  1.9162446622131852
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 17s 905us/step - loss: 2.2878
- acc: 0.1842 - val_loss: 2.1195 - val_acc: 0.2492
Epoch 2/20
18647/18647 [==============================] - 15s 813us/step - loss: 2.1343
- acc: 0.2474 - val_loss: 2.0474 - val_acc: 0.2846
Epoch 3/20
18647/18647 [==============================] - 15s 826us/step - loss: 2.0727
- acc: 0.2675 - val_loss: 2.0180 - val_acc: 0.3050
Epoch 4/20
18647/18647 [==============================] - 15s 805us/step - loss: 2.0303
- acc: 0.2795 - val_loss: 1.9816 - val_acc: 0.3093
Epoch 5/20
18647/18647 [==============================] - 15s 806us/step - loss: 1.9940
- acc: 0.2919 - val_loss: 1.9607 - val_acc: 0.3181
Epoch 6/20
18647/18647 [==============================] - 15s 824us/step - loss: 1.9651
- acc: 0.3009 - val_loss: 1.9576 - val_acc: 0.3160
Epoch 7/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.9469
- acc: 0.3098 - val_loss: 1.9380 - val_acc: 0.3183
Epoch 8/20
18647/18647 [==============================] - 15s 827us/step - loss: 1.9204
- acc: 0.3166 - val_loss: 1.9344 - val_acc: 0.3177
Epoch 9/20
18647/18647 [==============================] - 15s 821us/step - loss: 1.9028
- acc: 0.3209 - val_loss: 1.9388 - val_acc: 0.3196
Epoch 10/20
18647/18647 [==============================] - 15s 818us/step - loss: 1.8788
- acc: 0.3278 - val_loss: 1.9261 - val_acc: 0.3226
Epoch 11/20
18647/18647 [==============================] - 15s 798us/step - loss: 1.8763
- acc: 0.3238 - val_loss: 1.9327 - val_acc: 0.3192
Epoch 12/20
18647/18647 [==============================] - 15s 798us/step - loss: 1.8506
```

```
- acc: 0.3381 - val_loss: 1.9199 - val_acc: 0.3198
Epoch 13/20
18647/18647 [==============================] - 16s 859us/step - loss: 1.8400
- acc: 0.3446 - val_loss: 1.9195 - val_acc: 0.3278
Epoch 14/20
18647/18647 [==============================] - 15s 799us/step - loss: 1.8241
- acc: 0.3457 - val_loss: 1.9163 - val_acc: 0.3258
Epoch 15/20
18647/18647 [==============================] - 15s 807us/step - loss: 1.8058
- acc: 0.3570 - val_loss: 1.9186 - val_acc: 0.3190
Epoch 16/20
18647/18647 [==============================] - 15s 822us/step - loss: 1.8049
- acc: 0.3513 - val_loss: 1.9140 - val_acc: 0.3258
Epoch 17/20
18647/18647 [==============================] - 15s 793us/step - loss: 1.7871
- acc: 0.3599 - val_loss: 1.9201 - val_acc: 0.3280
Epoch 18/20
18647/18647 [==============================] - 15s 807us/step - loss: 1.7747
- acc: 0.3600 - val_loss: 1.9189 - val_acc: 0.3280
Epoch 19/20
18647/18647 [==============================] - 15s 794us/step - loss: 1.7660
- acc: 0.3639 - val_loss: 1.9219 - val_acc: 0.3241
Epoch 20/20
18647/18647 [==============================] - 15s 798us/step - loss: 1.7468
- acc: 0.3744 - val_loss: 1.9277 - val_acc: 0.3288
Validation Log Loss of  Model in Current Run:  1.9291087886768443
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 16s 867us/step - loss: 2.2869
- acc: 0.1868 - val_loss: 2.1600 - val_acc: 0.2544
Epoch 2/20
18647/18647 [==============================] - 15s 806us/step - loss: 2.1389
- acc: 0.2411 - val_loss: 2.0484 - val_acc: 0.2924
Epoch 3/20
18647/18647 [==============================] - 15s 801us/step - loss: 2.0696
- acc: 0.2671 - val_loss: 2.0009 - val_acc: 0.3012
Epoch 4/20
18647/18647 [==============================] - 15s 808us/step - loss: 2.0255
- acc: 0.2752 - val_loss: 1.9805 - val_acc: 0.3130
Epoch 5/20
18647/18647 [==============================] - 15s 808us/step - loss: 1.9912
- acc: 0.2919 - val_loss: 1.9656 - val_acc: 0.3142
Epoch 6/20
18647/18647 [==============================] - 15s 806us/step - loss: 1.9720
- acc: 0.2947 - val_loss: 1.9542 - val_acc: 0.3138
Epoch 7/20
18647/18647 [==============================] - 15s 791us/step - loss: 1.9457
- acc: 0.3078 - val_loss: 1.9513 - val_acc: 0.3172
Epoch 8/20
18647/18647 [==============================] - 15s 804us/step - loss: 1.9199
- acc: 0.3176 - val_loss: 1.9395 - val_acc: 0.3179
Epoch 9/20
18647/18647 [==============================] - 15s 808us/step - loss: 1.9046
- acc: 0.3221 - val_loss: 1.9298 - val_acc: 0.3237
Epoch 10/20
18647/18647 [==============================] - 15s 809us/step - loss: 1.8891
- acc: 0.3229 - val_loss: 1.9223 - val_acc: 0.3263
```

```
Epoch 11/20
18647/18647 [==============================] - 15s 820us/step - loss: 1.8708
- acc: 0.3328 - val_loss: 1.9291 - val_acc: 0.3205
Epoch 12/20
18647/18647 [==============================] - 16s 839us/step - loss: 1.8489
- acc: 0.3430 - val_loss: 1.9208 - val_acc: 0.3252
Epoch 13/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.8441
- acc: 0.3397 - val_loss: 1.9176 - val_acc: 0.3241
Epoch 14/20
18647/18647 [==============================] - 15s 818us/step - loss: 1.8205
- acc: 0.3475 - val_loss: 1.9153 - val_acc: 0.3273
Epoch 15/20
18647/18647 [==============================] - 17s 885us/step - loss: 1.8067
- acc: 0.3520 - val_loss: 1.9165 - val_acc: 0.3275
Epoch 16/20
18647/18647 [==============================] - 15s 789us/step - loss: 1.7972
- acc: 0.3559 - val_loss: 1.9136 - val_acc: 0.3258
Epoch 17/20
18647/18647 [==============================] - 15s 792us/step - loss: 1.7898
- acc: 0.3539 - val_loss: 1.9172 - val_acc: 0.3273
Epoch 18/20
18647/18647 [==============================] - 16s 839us/step - loss: 1.7747
- acc: 0.3601 - val_loss: 1.9187 - val_acc: 0.3288
Epoch 19/20
18647/18647 [==============================] - 15s 819us/step - loss: 1.7580
- acc: 0.3718 - val_loss: 1.9194 - val_acc: 0.3243
Epoch 20/20
18647/18647 [==============================] - 15s 818us/step - loss: 1.7521
- acc: 0.3698 - val_loss: 1.9172 - val_acc: 0.3254
Validation Log Loss of  Model in Current Run:  1.9185632223832525
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 17s 919us/step - loss: 2.2961
- acc: 0.1805 - val_loss: 2.1186 - val_acc: 0.2585
Epoch 2/20
18647/18647 [==============================] - 16s 855us/step - loss: 2.1372
- acc: 0.2459 - val_loss: 2.0371 - val_acc: 0.2932
Epoch 3/20
18647/18647 [==============================] - 16s 846us/step - loss: 2.0676
- acc: 0.2646 - val_loss: 2.0051 - val_acc: 0.3042
Epoch 4/20
18647/18647 [==============================] - 15s 809us/step - loss: 2.0292
- acc: 0.2790 - val_loss: 1.9902 - val_acc: 0.3003
Epoch 5/20
18647/18647 [==============================] - 15s 805us/step - loss: 1.9961
- acc: 0.2886 - val_loss: 1.9915 - val_acc: 0.3093
Epoch 6/20
18647/18647 [==============================] - 15s 808us/step - loss: 1.9647
- acc: 0.2961 - val_loss: 1.9557 - val_acc: 0.3119
Epoch 7/20
18647/18647 [==============================] - 16s 834us/step - loss: 1.9419
- acc: 0.3049 - val_loss: 1.9508 - val_acc: 0.3117
Epoch 8/20
18647/18647 [==============================] - 15s 809us/step - loss: 1.9206
- acc: 0.3195 - val_loss: 1.9425 - val_acc: 0.3179
Epoch 9/20
```

```
18647/18647 [==============================] - 15s 829us/step - loss: 1.9037
- acc: 0.3202 - val_loss: 1.9295 - val_acc: 0.3245
Epoch 10/20
18647/18647 [==============================] - 16s 843us/step - loss: 1.8851
- acc: 0.3247 - val_loss: 1.9276 - val_acc: 0.3233
Epoch 11/20
18647/18647 [==============================] - 15s 828us/step - loss: 1.8686
- acc: 0.3282 - val_loss: 1.9285 - val_acc: 0.3202
Epoch 12/20
18647/18647 [==============================] - 16s 871us/step - loss: 1.8613
- acc: 0.3369 - val_loss: 1.9211 - val_acc: 0.3235
Epoch 13/20
18647/18647 [==============================] - 16s 858us/step - loss: 1.8450
- acc: 0.3390 - val_loss: 1.9177 - val_acc: 0.3325
Epoch 14/20
18647/18647 [==============================] - 16s 852us/step - loss: 1.8212
- acc: 0.3487 - val_loss: 1.9146 - val_acc: 0.3293
Epoch 15/20
18647/18647 [==============================] - 15s 825us/step - loss: 1.8110
- acc: 0.3525 - val_loss: 1.9188 - val_acc: 0.3228
Epoch 16/20
18647/18647 [==============================] - 15s 822us/step - loss: 1.7998
- acc: 0.3501 - val_loss: 1.9145 - val_acc: 0.3327
Epoch 17/20
18647/18647 [==============================] - 15s 827us/step - loss: 1.7845
- acc: 0.3560 - val_loss: 1.9213 - val_acc: 0.3218
Epoch 18/20
18647/18647 [==============================] - 16s 840us/step - loss: 1.7684
- acc: 0.3637 - val_loss: 1.9174 - val_acc: 0.3312
Epoch 19/20
18647/18647 [==============================] - 15s 830us/step - loss: 1.7573
- acc: 0.3664 - val_loss: 1.9251 - val_acc: 0.3267
Epoch 20/20
18647/18647 [==============================] - 15s 791us/step - loss: 1.7377
- acc: 0.3765 - val_loss: 1.9242 - val_acc: 0.3250
Validation Log Loss of  Model in Current Run:  1.9258740875150422
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 17s 901us/step - loss: 2.2759
- acc: 0.1882 - val_loss: 2.1619 - val_acc: 0.2218
Epoch 2/20
18647/18647 [==============================] - 15s 810us/step - loss: 2.1346
- acc: 0.2425 - val_loss: 2.0396 - val_acc: 0.2928
Epoch 3/20
18647/18647 [==============================] - 15s 818us/step - loss: 2.0685
- acc: 0.2666 - val_loss: 2.0111 - val_acc: 0.2939
Epoch 4/20
18647/18647 [==============================] - 15s 821us/step - loss: 2.0228
- acc: 0.2806 - val_loss: 1.9822 - val_acc: 0.3035
Epoch 5/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.9909
- acc: 0.2895 - val_loss: 1.9584 - val_acc: 0.3140
Epoch 6/20
18647/18647 [==============================] - 15s 815us/step - loss: 1.9623
- acc: 0.3015 - val_loss: 1.9478 - val_acc: 0.3218
Epoch 7/20
18647/18647 [==============================] - 16s 842us/step - loss: 1.9396
```

```
- acc: 0.3083 - val_loss: 1.9474 - val_acc: 0.3172
Epoch 8/20
18647/18647 [==============================] - 15s 817us/step - loss: 1.9139
- acc: 0.3167 - val_loss: 1.9349 - val_acc: 0.3168
Epoch 9/20
18647/18647 [==============================] - 15s 814us/step - loss: 1.8933
- acc: 0.3262 - val_loss: 1.9282 - val_acc: 0.3230
Epoch 10/20
18647/18647 [==============================] - 15s 827us/step - loss: 1.8764
- acc: 0.3325 - val_loss: 1.9207 - val_acc: 0.3245
Epoch 11/20
18647/18647 [==============================] - 15s 813us/step - loss: 1.8670
- acc: 0.3251 - val_loss: 1.9225 - val_acc: 0.3280
Epoch 12/20
18647/18647 [==============================] - 15s 814us/step - loss: 1.8539
- acc: 0.3355 - val_loss: 1.9233 - val_acc: 0.3194
Epoch 13/20
18647/18647 [==============================] - 15s 811us/step - loss: 1.8352
- acc: 0.3401 - val_loss: 1.9188 - val_acc: 0.3230
Epoch 14/20
18647/18647 [==============================] - 15s 792us/step - loss: 1.8162
- acc: 0.3439 - val_loss: 1.9197 - val_acc: 0.3256
Epoch 15/20
18647/18647 [==============================] - 15s 806us/step - loss: 1.8127
- acc: 0.3522 - val_loss: 1.9379 - val_acc: 0.3220
Epoch 16/20
18647/18647 [==============================] - 15s 830us/step - loss: 1.7812
- acc: 0.3601 - val_loss: 1.9163 - val_acc: 0.3237
Epoch 17/20
18647/18647 [==============================] - 15s 801us/step - loss: 1.7845
- acc: 0.3550 - val_loss: 1.9128 - val_acc: 0.3218
Epoch 18/20
18647/18647 [==============================] - 15s 788us/step - loss: 1.7699
- acc: 0.3607 - val_loss: 1.9208 - val_acc: 0.3192
Epoch 19/20
18647/18647 [==============================] - 15s 815us/step - loss: 1.7546
- acc: 0.3693 - val_loss: 1.9202 - val_acc: 0.3209
Epoch 20/20
18647/18647 [==============================] - 15s 819us/step - loss: 1.7456
- acc: 0.3757 - val_loss: 1.9192 - val_acc: 0.3228
Validation Log Loss of  Model in Current Run:  1.9209416287566417
Average CV Loss of 10 Runs : 1.9234806484970717
```

In [90]:

```
avg_pred=np.zeros((xtr.shape[0],12))
for i in range(len(model_list_2)):
    train_pred=model_list_2[i].predict_proba(xtr)
    avg_pred+=train_pred
avg_pred/=len(model_list_2)
print("Train Average Log-Loss: ",log_loss(ytr, avg_pred))
```

```
Train Average Log-Loss:  1.5406584936512548
```

In [91]:
```
avg_pred=np.zeros((xcv.shape[0],12))
for i in range(len(model_list_2)):
    cv_pred=model_list_2[i].predict_proba(xcv)
    avg_pred+=cv_pred
avg_pred/=len(model_list_2)
print("Validation Average Log-Loss: ",log_loss(ycv, avg_pred))
```

```
Validation Average Log-Loss:  1.9074406784271256
```

In [92]:
```
avg_pred=np.zeros((X_test_events.shape[0],12))
for i in range(len(model_list_2)):
    test_pred=model_list_2[i].predict_proba(X_test_events)
    avg_pred+=test_pred
avg_pred/=len(model_list_2)
```

In [93]:
```
np.save('nn1_events_1',avg_pred)
```

OBSERVATIONS: 1.THE TRAIN AND VALIDATION LOSSES ARE 1.5406 AND 1.9074 RESPECTIVELY.

1. BOTH TRAIN AND VALIDATIONN LOSSES DECREASED AS WE ADDED MORE FEATURES.

NEURAL NETWORK 2

In [94]:
```
def events_nn_model2(input_dim,output_dim):
    model = Sequential()
    model.add(Dropout(0.4, input_shape=(input_dim,)))
    model.add(Dense(75))
    model.add(PReLU())
    model.add(Dropout(0.30))
    model.add(Dense(50, init='normal', activation='tanh'))
    model.add(PReLU())
    model.add(Dropout(0.20))
    model.add(Dense(output_dim, init='normal', activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adadelta', metri
cs=['accuracy'])
    return model
```

In [95]:
```
model_sum=events_nn_model2(xtr.shape[1],12)
model_sum.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dropout_63 (Dropout)         (None, 21576)             0         
_________________________________________________________________
dense_75 (Dense)             (None, 75)                1618275   
_________________________________________________________________
p_re_lu_52 (PReLU)           (None, 75)                75        
_________________________________________________________________
dropout_64 (Dropout)         (None, 75)                0         
_________________________________________________________________
dense_76 (Dense)             (None, 50)                3800      
_________________________________________________________________
p_re_lu_53 (PReLU)           (None, 50)                50        
_________________________________________________________________
dropout_65 (Dropout)         (None, 50)                0         
_________________________________________________________________
dense_77 (Dense)             (None, 12)                612       
=================================================================
Total params: 1,622,812
Trainable params: 1,622,812
Non-trainable params: 0
_________________________________________________________________
```

In [96]:
```
def events_average_nn_2(state):

    model_list=[]
    loss_list=[]
    avg_cv_loss=0
    for i in range((state)):
        model=events_nn_model2(xtr.shape[1],12)
        model.fit(xtr, ytr, batch_size=149, epochs=20, verbose=1, validation_d
ata=(xcv, ycv),callbacks=[early_stop])
        model.save('saved_models/events/nn2'+str(i+1))
        pred=model.predict_proba(xcv)
        cv_loss=log_loss(ycv, pred)
        print("Validation Log Loss of  Model in Current Run: ",cv_loss)
        model_list.append(model)
        loss_list.append(cv_loss)
    avg_cv_loss=mean(loss_list)
    print("Average CV Loss of "+str((state))+" Runs :",avg_cv_loss)
    return(model_list)
```

In [97]:
```
model_list_2=events_average_nn_2(10)
```

```
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 13s 706us/step - loss: 2.3350
- acc: 0.1686 - val_loss: 2.1600 - val_acc: 0.2454
Epoch 2/20
18647/18647 [==============================] - 12s 641us/step - loss: 2.1608
- acc: 0.2397 - val_loss: 2.0982 - val_acc: 0.2683
Epoch 3/20
18647/18647 [==============================] - 13s 689us/step - loss: 2.0919
- acc: 0.2597 - val_loss: 2.0512 - val_acc: 0.2861
Epoch 4/20
18647/18647 [==============================] - 13s 685us/step - loss: 2.0469
- acc: 0.2799 - val_loss: 1.9799 - val_acc: 0.3166
Epoch 5/20
18647/18647 [==============================] - 12s 632us/step - loss: 2.0245
- acc: 0.2851 - val_loss: 1.9927 - val_acc: 0.3027
Epoch 6/20
18647/18647 [==============================] - 12s 635us/step - loss: 1.9889
- acc: 0.2986 - val_loss: 1.9650 - val_acc: 0.3162
Epoch 7/20
18647/18647 [==============================] - 12s 650us/step - loss: 1.9787
- acc: 0.3034 - val_loss: 1.9700 - val_acc: 0.3142
Epoch 8/20
18647/18647 [==============================] - 13s 712us/step - loss: 1.9568
- acc: 0.3110 - val_loss: 1.9337 - val_acc: 0.3233
Epoch 9/20
18647/18647 [==============================] - 12s 657us/step - loss: 1.9395
- acc: 0.3102 - val_loss: 1.9785 - val_acc: 0.3140
Epoch 10/20
18647/18647 [==============================] - 12s 658us/step - loss: 1.9292
- acc: 0.3182 - val_loss: 1.9467 - val_acc: 0.3164
Epoch 11/20
18647/18647 [==============================] - 12s 634us/step - loss: 1.9116
- acc: 0.3227 - val_loss: 1.9328 - val_acc: 0.3187
Epoch 12/20
18647/18647 [==============================] - 12s 623us/step - loss: 1.8988
- acc: 0.3324 - val_loss: 1.9169 - val_acc: 0.3278
Epoch 13/20
18647/18647 [==============================] - 12s 636us/step - loss: 1.8862
- acc: 0.3372 - val_loss: 1.9572 - val_acc: 0.3102
Epoch 14/20
18647/18647 [==============================] - 12s 622us/step - loss: 1.8677
- acc: 0.3404 - val_loss: 1.9572 - val_acc: 0.3190
Epoch 15/20
18647/18647 [==============================] - 12s 626us/step - loss: 1.8570
- acc: 0.3413 - val_loss: 1.9490 - val_acc: 0.3181
Epoch 16/20
18647/18647 [==============================] - 12s 627us/step - loss: 1.8447
- acc: 0.3522 - val_loss: 1.9303 - val_acc: 0.3301
Epoch 17/20
18647/18647 [==============================] - 12s 637us/step - loss: 1.8272
- acc: 0.3567 - val_loss: 1.9427 - val_acc: 0.3260
Validation Log Loss of  Model in Current Run:  1.9168982927262757
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 13s 707us/step - loss: 2.3279
- acc: 0.1774 - val_loss: 2.1513 - val_acc: 0.2424
```

```
Epoch 2/20
18647/18647 [==============================] - 12s 617us/step - loss: 2.1482
- acc: 0.2451 - val_loss: 2.1517 - val_acc: 0.2480
Epoch 3/20
18647/18647 [==============================] - 11s 612us/step - loss: 2.0857
- acc: 0.2651 - val_loss: 2.0919 - val_acc: 0.2662
Epoch 4/20
18647/18647 [==============================] - 12s 618us/step - loss: 2.0402
- acc: 0.2813 - val_loss: 2.0846 - val_acc: 0.2656
Epoch 5/20
18647/18647 [==============================] - 12s 620us/step - loss: 2.0140
- acc: 0.2892 - val_loss: 1.9836 - val_acc: 0.2969
Epoch 6/20
18647/18647 [==============================] - 11s 616us/step - loss: 1.9926
- acc: 0.2990 - val_loss: 1.9386 - val_acc: 0.3202
Epoch 7/20
18647/18647 [==============================] - 12s 621us/step - loss: 1.9673
- acc: 0.3070 - val_loss: 1.9354 - val_acc: 0.3256
Epoch 8/20
18647/18647 [==============================] - 11s 616us/step - loss: 1.9498
- acc: 0.3137 - val_loss: 1.9509 - val_acc: 0.3149
Epoch 9/20
18647/18647 [==============================] - 12s 618us/step - loss: 1.9301
- acc: 0.3227 - val_loss: 1.9435 - val_acc: 0.3267
Epoch 10/20
18647/18647 [==============================] - 11s 611us/step - loss: 1.9183
- acc: 0.3233 - val_loss: 1.9459 - val_acc: 0.3164
Epoch 11/20
18647/18647 [==============================] - 11s 603us/step - loss: 1.9008
- acc: 0.3277 - val_loss: 1.9223 - val_acc: 0.3271
Epoch 12/20
18647/18647 [==============================] - 11s 610us/step - loss: 1.8960
- acc: 0.3343 - val_loss: 1.9331 - val_acc: 0.3237
Epoch 13/20
18647/18647 [==============================] - 12s 645us/step - loss: 1.8700
- acc: 0.3377 - val_loss: 1.9246 - val_acc: 0.3256
Epoch 14/20
18647/18647 [==============================] - 11s 604us/step - loss: 1.8607
- acc: 0.3458 - val_loss: 1.9147 - val_acc: 0.3194
Epoch 15/20
18647/18647 [==============================] - 11s 603us/step - loss: 1.8494
- acc: 0.3460 - val_loss: 1.9315 - val_acc: 0.3239
Epoch 16/20
18647/18647 [==============================] - 11s 611us/step - loss: 1.8367
- acc: 0.3515 - val_loss: 1.9360 - val_acc: 0.3166
Epoch 17/20
18647/18647 [==============================] - 12s 617us/step - loss: 1.8187
- acc: 0.3571 - val_loss: 1.9337 - val_acc: 0.3175
Epoch 18/20
18647/18647 [==============================] - 12s 620us/step - loss: 1.8112
- acc: 0.3591 - val_loss: 1.9263 - val_acc: 0.3192
Epoch 19/20
18647/18647 [==============================] - 12s 624us/step - loss: 1.7877
- acc: 0.3687 - val_loss: 1.9415 - val_acc: 0.3243
Validation Log Loss of  Model in Current Run:  1.9146637250839105
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
```

```
18647/18647 [==============================] - 13s 715us/step - loss: 2.3170
- acc: 0.1779 - val_loss: 2.1846 - val_acc: 0.2179
Epoch 2/20
18647/18647 [==============================] - 11s 591us/step - loss: 2.1505
- acc: 0.2411 - val_loss: 2.1458 - val_acc: 0.2432
Epoch 3/20
18647/18647 [==============================] - 11s 593us/step - loss: 2.0837
- acc: 0.2660 - val_loss: 2.0299 - val_acc: 0.2906
Epoch 4/20
18647/18647 [==============================] - 11s 588us/step - loss: 2.0430
- acc: 0.2783 - val_loss: 1.9921 - val_acc: 0.3048
Epoch 5/20
18647/18647 [==============================] - 11s 575us/step - loss: 2.0140
- acc: 0.2928 - val_loss: 1.9691 - val_acc: 0.3190
Epoch 6/20
18647/18647 [==============================] - 12s 626us/step - loss: 1.9916
- acc: 0.2929 - val_loss: 1.9847 - val_acc: 0.3085
Epoch 7/20
18647/18647 [==============================] - 11s 604us/step - loss: 1.9701
- acc: 0.3035 - val_loss: 1.9432 - val_acc: 0.3269
Epoch 8/20
18647/18647 [==============================] - 12s 627us/step - loss: 1.9516
- acc: 0.3111 - val_loss: 1.9360 - val_acc: 0.3243
Epoch 9/20
18647/18647 [==============================] - 12s 621us/step - loss: 1.9357
- acc: 0.3190 - val_loss: 1.9513 - val_acc: 0.3166
Epoch 10/20
18647/18647 [==============================] - 11s 613us/step - loss: 1.9228
- acc: 0.3220 - val_loss: 1.9388 - val_acc: 0.3318
Epoch 11/20
18647/18647 [==============================] - 12s 630us/step - loss: 1.9185
- acc: 0.3197 - val_loss: 1.9370 - val_acc: 0.3190
Epoch 12/20
18647/18647 [==============================] - 12s 649us/step - loss: 1.8954
- acc: 0.3329 - val_loss: 1.9213 - val_acc: 0.3260
Epoch 13/20
18647/18647 [==============================] - 12s 632us/step - loss: 1.8787
- acc: 0.3355 - val_loss: 1.9198 - val_acc: 0.3284
Epoch 14/20
18647/18647 [==============================] - 12s 656us/step - loss: 1.8706
- acc: 0.3384 - val_loss: 1.9149 - val_acc: 0.3282
Epoch 15/20
18647/18647 [==============================] - 12s 634us/step - loss: 1.8602
- acc: 0.3399 - val_loss: 2.0072 - val_acc: 0.3027
Epoch 16/20
18647/18647 [==============================] - 12s 631us/step - loss: 1.8425
- acc: 0.3463 - val_loss: 1.9377 - val_acc: 0.3213
Epoch 17/20
18647/18647 [==============================] - 12s 618us/step - loss: 1.8324
- acc: 0.3504 - val_loss: 1.9421 - val_acc: 0.3185
Epoch 18/20
18647/18647 [==============================] - 11s 614us/step - loss: 1.8147
- acc: 0.3583 - val_loss: 1.9414 - val_acc: 0.3183
Epoch 19/20
18647/18647 [==============================] - 11s 616us/step - loss: 1.8030
- acc: 0.3615 - val_loss: 1.9322 - val_acc: 0.3248
Validation Log Loss of  Model in Current Run:  1.9148988088457843
```

```
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 14s 727us/step - loss: 2.3361
- acc: 0.1656 - val_loss: 2.2334 - val_acc: 0.2186
Epoch 2/20
18647/18647 [==============================] - 11s 616us/step - loss: 2.1667
- acc: 0.2360 - val_loss: 2.0562 - val_acc: 0.2891
Epoch 3/20
18647/18647 [==============================] - 11s 617us/step - loss: 2.0897
- acc: 0.2592 - val_loss: 2.0043 - val_acc: 0.3052
Epoch 4/20
18647/18647 [==============================] - 12s 620us/step - loss: 2.0462
- acc: 0.2803 - val_loss: 2.0205 - val_acc: 0.2855
Epoch 5/20
18647/18647 [==============================] - 12s 661us/step - loss: 2.0215
- acc: 0.2885 - val_loss: 1.9720 - val_acc: 0.3106
Epoch 6/20
18647/18647 [==============================] - 12s 642us/step - loss: 1.9983
- acc: 0.2969 - val_loss: 1.9922 - val_acc: 0.3087
Epoch 7/20
18647/18647 [==============================] - 12s 650us/step - loss: 1.9755
- acc: 0.3036 - val_loss: 1.9450 - val_acc: 0.3222
Epoch 8/20
18647/18647 [==============================] - 12s 627us/step - loss: 1.9554
- acc: 0.3114 - val_loss: 1.9403 - val_acc: 0.3260
Epoch 9/20
18647/18647 [==============================] - 12s 620us/step - loss: 1.9409
- acc: 0.3160 - val_loss: 1.9493 - val_acc: 0.3192
Epoch 10/20
18647/18647 [==============================] - 11s 609us/step - loss: 1.9268
- acc: 0.3214 - val_loss: 1.9250 - val_acc: 0.3254
Epoch 11/20
18647/18647 [==============================] - 11s 610us/step - loss: 1.9093
- acc: 0.3275 - val_loss: 1.9665 - val_acc: 0.3119
Epoch 12/20
18647/18647 [==============================] - 11s 605us/step - loss: 1.9063
- acc: 0.3238 - val_loss: 1.9375 - val_acc: 0.3222
Epoch 13/20
18647/18647 [==============================] - 11s 610us/step - loss: 1.8821
- acc: 0.3334 - val_loss: 1.9965 - val_acc: 0.3033
Epoch 14/20
18647/18647 [==============================] - 11s 606us/step - loss: 1.8725
- acc: 0.3372 - val_loss: 1.9428 - val_acc: 0.3198
Epoch 15/20
18647/18647 [==============================] - 11s 613us/step - loss: 1.8524
- acc: 0.3452 - val_loss: 1.9319 - val_acc: 0.3250
Validation Log Loss of  Model in Current Run:  1.924992392238362
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 13s 717us/step - loss: 2.3289
- acc: 0.1720 - val_loss: 2.2337 - val_acc: 0.2199
Epoch 2/20
18647/18647 [==============================] - 11s 615us/step - loss: 2.1578
- acc: 0.2379 - val_loss: 2.0511 - val_acc: 0.2840
Epoch 3/20
18647/18647 [==============================] - 11s 614us/step - loss: 2.0906
- acc: 0.2662 - val_loss: 2.0087 - val_acc: 0.3014
```

```
Epoch 4/20
18647/18647 [==============================] - 11s 607us/step - loss: 2.0399
- acc: 0.2805 - val_loss: 1.9674 - val_acc: 0.3134
Epoch 5/20
18647/18647 [==============================] - 11s 605us/step - loss: 2.0149
- acc: 0.2923 - val_loss: 1.9848 - val_acc: 0.3054
Epoch 6/20
18647/18647 [==============================] - 11s 602us/step - loss: 1.9946
- acc: 0.2985 - val_loss: 1.9581 - val_acc: 0.3119
Epoch 7/20
18647/18647 [==============================] - 12s 622us/step - loss: 1.9732
- acc: 0.3022 - val_loss: 1.9736 - val_acc: 0.3072
Epoch 8/20
18647/18647 [==============================] - 12s 625us/step - loss: 1.9574
- acc: 0.3069 - val_loss: 1.9595 - val_acc: 0.3106
Epoch 9/20
18647/18647 [==============================] - 12s 662us/step - loss: 1.9385
- acc: 0.3132 - val_loss: 1.9356 - val_acc: 0.3220
Epoch 10/20
18647/18647 [==============================] - 14s 754us/step - loss: 1.9191
- acc: 0.3230 - val_loss: 1.9294 - val_acc: 0.3263
Epoch 11/20
18647/18647 [==============================] - 12s 666us/step - loss: 1.9104
- acc: 0.3273 - val_loss: 1.9277 - val_acc: 0.3245
Epoch 12/20
18647/18647 [==============================] - 11s 603us/step - loss: 1.8938
- acc: 0.3307 - val_loss: 1.9205 - val_acc: 0.3256
Epoch 13/20
18647/18647 [==============================] - 11s 609us/step - loss: 1.8837
- acc: 0.3382 - val_loss: 1.9471 - val_acc: 0.3097
Epoch 14/20
18647/18647 [==============================] - 11s 611us/step - loss: 1.8668
- acc: 0.3381 - val_loss: 1.9324 - val_acc: 0.3220
Epoch 15/20
18647/18647 [==============================] - 11s 592us/step - loss: 1.8502
- acc: 0.3446 - val_loss: 1.9301 - val_acc: 0.3200
Epoch 16/20
18647/18647 [==============================] - 11s 593us/step - loss: 1.8359
- acc: 0.3487 - val_loss: 1.9358 - val_acc: 0.3215
Epoch 17/20
18647/18647 [==============================] - 11s 603us/step - loss: 1.8271
- acc: 0.3550 - val_loss: 1.9568 - val_acc: 0.3192
Validation Log Loss of  Model in Current Run:  1.9205034403254269
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 16s 845us/step - loss: 2.3284
- acc: 0.1669 - val_loss: 2.2163 - val_acc: 0.2059
Epoch 2/20
18647/18647 [==============================] - 14s 726us/step - loss: 2.1591
- acc: 0.2406 - val_loss: 2.0946 - val_acc: 0.2604
Epoch 3/20
18647/18647 [==============================] - 14s 727us/step - loss: 2.0845
- acc: 0.2653 - val_loss: 2.0344 - val_acc: 0.2962
Epoch 4/20
18647/18647 [==============================] - 13s 712us/step - loss: 2.0455
- acc: 0.2804 - val_loss: 2.0642 - val_acc: 0.2786
Epoch 5/20
```

```
18647/18647 [==============================] - 13s 717us/step - loss: 2.0113
- acc: 0.2901 - val_loss: 1.9781 - val_acc: 0.3115
Epoch 6/20
18647/18647 [==============================] - 13s 708us/step - loss: 1.9870
- acc: 0.2989 - val_loss: 1.9830 - val_acc: 0.2973
Epoch 7/20
18647/18647 [==============================] - 12s 637us/step - loss: 1.9720
- acc: 0.3011 - val_loss: 1.9472 - val_acc: 0.3218
Epoch 8/20
18647/18647 [==============================] - 11s 597us/step - loss: 1.9535
- acc: 0.3077 - val_loss: 1.9288 - val_acc: 0.3301
Epoch 9/20
18647/18647 [==============================] - 11s 607us/step - loss: 1.9335
- acc: 0.3180 - val_loss: 1.9576 - val_acc: 0.3222
Epoch 10/20
18647/18647 [==============================] - 12s 668us/step - loss: 1.9179
- acc: 0.3243 - val_loss: 1.9331 - val_acc: 0.3220
Epoch 11/20
18647/18647 [==============================] - 12s 658us/step - loss: 1.9006
- acc: 0.3272 - val_loss: 1.9319 - val_acc: 0.3260
Epoch 12/20
18647/18647 [==============================] - 12s 661us/step - loss: 1.8932
- acc: 0.3313 - val_loss: 1.9712 - val_acc: 0.2984
Epoch 13/20
18647/18647 [==============================] - 12s 659us/step - loss: 1.8769
- acc: 0.3369 - val_loss: 1.9516 - val_acc: 0.3106
Validation Log Loss of  Model in Current Run:  1.928770428193403
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 14s 767us/step - loss: 2.3246
- acc: 0.1728 - val_loss: 2.1944 - val_acc: 0.2411
Epoch 2/20
18647/18647 [==============================] - 11s 576us/step - loss: 2.1514
- acc: 0.2447 - val_loss: 2.1698 - val_acc: 0.2282
Epoch 3/20
18647/18647 [==============================] - 11s 595us/step - loss: 2.0869
- acc: 0.2667 - val_loss: 2.0212 - val_acc: 0.2979
Epoch 4/20
18647/18647 [==============================] - 12s 638us/step - loss: 2.0465
- acc: 0.2830 - val_loss: 1.9823 - val_acc: 0.3140
Epoch 5/20
18647/18647 [==============================] - 13s 703us/step - loss: 2.0152
- acc: 0.2899 - val_loss: 1.9797 - val_acc: 0.3052
Epoch 6/20
18647/18647 [==============================] - 12s 662us/step - loss: 1.9951
- acc: 0.2960 - val_loss: 1.9542 - val_acc: 0.3196
Epoch 7/20
18647/18647 [==============================] - 12s 670us/step - loss: 1.9707
- acc: 0.3051 - val_loss: 1.9474 - val_acc: 0.3125
Epoch 8/20
18647/18647 [==============================] - 11s 601us/step - loss: 1.9533
- acc: 0.3098 - val_loss: 1.9606 - val_acc: 0.3082
Epoch 9/20
18647/18647 [==============================] - 11s 581us/step - loss: 1.9370
- acc: 0.3130 - val_loss: 1.9435 - val_acc: 0.3241
Epoch 10/20
18647/18647 [==============================] - 11s 578us/step - loss: 1.9249
```

```
- acc: 0.3176 - val_loss: 1.9461 - val_acc: 0.3175
Epoch 11/20
18647/18647 [==============================] - 11s 580us/step - loss: 1.9060
- acc: 0.3251 - val_loss: 2.0473 - val_acc: 0.2718
Epoch 12/20
18647/18647 [==============================] - 11s 571us/step - loss: 1.8954
- acc: 0.3305 - val_loss: 1.9139 - val_acc: 0.3303
Epoch 13/20
18647/18647 [==============================] - 11s 579us/step - loss: 1.8790
- acc: 0.3320 - val_loss: 1.9116 - val_acc: 0.3260
Epoch 14/20
18647/18647 [==============================] - 11s 577us/step - loss: 1.8626
- acc: 0.3397 - val_loss: 1.9259 - val_acc: 0.3192
Epoch 15/20
18647/18647 [==============================] - 11s 576us/step - loss: 1.8457
- acc: 0.3463 - val_loss: 1.9211 - val_acc: 0.3230
Epoch 16/20
18647/18647 [==============================] - 11s 583us/step - loss: 1.8344
- acc: 0.3506 - val_loss: 1.9176 - val_acc: 0.3237
Epoch 17/20
18647/18647 [==============================] - 11s 580us/step - loss: 1.8264
- acc: 0.3564 - val_loss: 1.9421 - val_acc: 0.3164
Epoch 18/20
18647/18647 [==============================] - 11s 576us/step - loss: 1.8152
- acc: 0.3565 - val_loss: 1.9458 - val_acc: 0.3149
Validation Log Loss of  Model in Current Run:  1.9115930246656696
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 13s 687us/step - loss: 2.3201
- acc: 0.1744 - val_loss: 2.1997 - val_acc: 0.2201
Epoch 2/20
18647/18647 [==============================] - 11s 600us/step - loss: 2.1535
- acc: 0.2438 - val_loss: 2.1573 - val_acc: 0.2544
Epoch 3/20
18647/18647 [==============================] - 11s 577us/step - loss: 2.0855
- acc: 0.2650 - val_loss: 2.0144 - val_acc: 0.2883
Epoch 4/20
18647/18647 [==============================] - 11s 576us/step - loss: 2.0416
- acc: 0.2830 - val_loss: 1.9886 - val_acc: 0.3097
Epoch 5/20
18647/18647 [==============================] - 11s 580us/step - loss: 2.0178
- acc: 0.2884 - val_loss: 1.9824 - val_acc: 0.3127
Epoch 6/20
18647/18647 [==============================] - 11s 581us/step - loss: 1.9873
- acc: 0.3021 - val_loss: 1.9594 - val_acc: 0.3164
Epoch 7/20
18647/18647 [==============================] - 11s 577us/step - loss: 1.9680
- acc: 0.3092 - val_loss: 1.9480 - val_acc: 0.3228
Epoch 8/20
18647/18647 [==============================] - 11s 576us/step - loss: 1.9524
- acc: 0.3129 - val_loss: 1.9532 - val_acc: 0.3125
Epoch 9/20
18647/18647 [==============================] - 11s 580us/step - loss: 1.9372
- acc: 0.3221 - val_loss: 1.9327 - val_acc: 0.3218
Epoch 10/20
18647/18647 [==============================] - 11s 580us/step - loss: 1.9251
- acc: 0.3187 - val_loss: 1.9317 - val_acc: 0.3196
```

```
Epoch 11/20
18647/18647 [==============================] - 11s 575us/step - loss: 1.9032
- acc: 0.3277 - val_loss: 1.9359 - val_acc: 0.3196
Epoch 12/20
18647/18647 [==============================] - 11s 577us/step - loss: 1.8931
- acc: 0.3313 - val_loss: 1.9373 - val_acc: 0.3183
Epoch 13/20
18647/18647 [==============================] - 11s 579us/step - loss: 1.8835
- acc: 0.3375 - val_loss: 1.9235 - val_acc: 0.3235
Epoch 14/20
18647/18647 [==============================] - 11s 580us/step - loss: 1.8695
- acc: 0.3381 - val_loss: 1.9196 - val_acc: 0.3284
Epoch 15/20
18647/18647 [==============================] - 11s 583us/step - loss: 1.8513
- acc: 0.3430 - val_loss: 1.9149 - val_acc: 0.3263
Epoch 16/20
18647/18647 [==============================] - 11s 584us/step - loss: 1.8350
- acc: 0.3504 - val_loss: 1.9246 - val_acc: 0.3226
Epoch 17/20
18647/18647 [==============================] - 11s 587us/step - loss: 1.8234
- acc: 0.3532 - val_loss: 1.9464 - val_acc: 0.3157
Epoch 18/20
18647/18647 [==============================] - 11s 581us/step - loss: 1.8142
- acc: 0.3595 - val_loss: 1.9433 - val_acc: 0.3121
Epoch 19/20
18647/18647 [==============================] - 11s 583us/step - loss: 1.7976
- acc: 0.3649 - val_loss: 1.9347 - val_acc: 0.3207
Epoch 20/20
18647/18647 [==============================] - 11s 583us/step - loss: 1.7913
- acc: 0.3654 - val_loss: 1.9322 - val_acc: 0.3220
Validation Log Loss of  Model in Current Run:  1.9148608428341785
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 13s 690us/step - loss: 2.3379
- acc: 0.1681 - val_loss: 2.1805 - val_acc: 0.2250
Epoch 2/20
18647/18647 [==============================] - 11s 616us/step - loss: 2.1597
- acc: 0.2419 - val_loss: 2.0904 - val_acc: 0.2692
Epoch 3/20
18647/18647 [==============================] - 11s 585us/step - loss: 2.0887
- acc: 0.2621 - val_loss: 2.0280 - val_acc: 0.2932
Epoch 4/20
18647/18647 [==============================] - 11s 572us/step - loss: 2.0379
- acc: 0.2840 - val_loss: 1.9859 - val_acc: 0.3121
Epoch 5/20
18647/18647 [==============================] - 11s 612us/step - loss: 2.0161
- acc: 0.2846 - val_loss: 2.1284 - val_acc: 0.2486
Epoch 6/20
18647/18647 [==============================] - 11s 573us/step - loss: 1.9886
- acc: 0.2977 - val_loss: 1.9518 - val_acc: 0.3093
Epoch 7/20
18647/18647 [==============================] - 11s 604us/step - loss: 1.9681
- acc: 0.3061 - val_loss: 1.9466 - val_acc: 0.3213
Epoch 8/20
18647/18647 [==============================] - 11s 580us/step - loss: 1.9488
- acc: 0.3130 - val_loss: 1.9381 - val_acc: 0.3205
Epoch 9/20
```

```
18647/18647 [==============================] - 11s 613us/step - loss: 1.9387
- acc: 0.3175 - val_loss: 1.9406 - val_acc: 0.3226
Epoch 10/20
18647/18647 [==============================] - 11s 586us/step - loss: 1.9150
- acc: 0.3276 - val_loss: 1.9258 - val_acc: 0.3239
Epoch 11/20
18647/18647 [==============================] - 11s 590us/step - loss: 1.9101
- acc: 0.3251 - val_loss: 1.9281 - val_acc: 0.3338
Epoch 12/20
18647/18647 [==============================] - 12s 652us/step - loss: 1.8939
- acc: 0.3299 - val_loss: 1.9254 - val_acc: 0.3267
Epoch 13/20
18647/18647 [==============================] - 11s 596us/step - loss: 1.8781
- acc: 0.3359 - val_loss: 1.9725 - val_acc: 0.3157
Epoch 14/20
18647/18647 [==============================] - 12s 656us/step - loss: 1.8598
- acc: 0.3433 - val_loss: 1.9183 - val_acc: 0.3269
Epoch 15/20
18647/18647 [==============================] - 11s 615us/step - loss: 1.8502
- acc: 0.3471 - val_loss: 1.9429 - val_acc: 0.3102
Epoch 16/20
18647/18647 [==============================] - 11s 595us/step - loss: 1.8336
- acc: 0.3536 - val_loss: 1.9477 - val_acc: 0.3119
Epoch 17/20
18647/18647 [==============================] - 10s 561us/step - loss: 1.8240
- acc: 0.3521 - val_loss: 2.0130 - val_acc: 0.2992
Epoch 18/20
18647/18647 [==============================] - 11s 592us/step - loss: 1.8125
- acc: 0.3610 - val_loss: 1.9446 - val_acc: 0.3170
Epoch 19/20
18647/18647 [==============================] - 12s 619us/step - loss: 1.7925
- acc: 0.3641 - val_loss: 1.9429 - val_acc: 0.3200
Validation Log Loss of  Model in Current Run:  1.918339006581988
Train on 18647 samples, validate on 4662 samples
Epoch 1/20
18647/18647 [==============================] - 13s 696us/step - loss: 2.3280
- acc: 0.1693 - val_loss: 2.2421 - val_acc: 0.1918
Epoch 2/20
18647/18647 [==============================] - 11s 585us/step - loss: 2.1578
- acc: 0.2400 - val_loss: 2.1214 - val_acc: 0.2576
Epoch 3/20
18647/18647 [==============================] - 11s 576us/step - loss: 2.0890
- acc: 0.2640 - val_loss: 1.9978 - val_acc: 0.3016
Epoch 4/20
18647/18647 [==============================] - 11s 576us/step - loss: 2.0478
- acc: 0.2780 - val_loss: 1.9853 - val_acc: 0.3115
Epoch 5/20
18647/18647 [==============================] - 11s 577us/step - loss: 2.0227
- acc: 0.2827 - val_loss: 1.9697 - val_acc: 0.3177
Epoch 6/20
18647/18647 [==============================] - 11s 569us/step - loss: 1.9916
- acc: 0.2980 - val_loss: 1.9509 - val_acc: 0.3198
Epoch 7/20
18647/18647 [==============================] - 11s 566us/step - loss: 1.9759
- acc: 0.3025 - val_loss: 1.9436 - val_acc: 0.3226
Epoch 8/20
18647/18647 [==============================] - 11s 566us/step - loss: 1.9566
```

```
- acc: 0.3095 - val_loss: 1.9513 - val_acc: 0.3123
Epoch 9/20
18647/18647 [==============================] - 11s 575us/step - loss: 1.9447
- acc: 0.3139 - val_loss: 1.9394 - val_acc: 0.3263
Epoch 10/20
18647/18647 [==============================] - 11s 585us/step - loss: 1.9347
- acc: 0.3160 - val_loss: 1.9418 - val_acc: 0.3215
Epoch 11/20
18647/18647 [==============================] - 11s 602us/step - loss: 1.9052
- acc: 0.3258 - val_loss: 1.9427 - val_acc: 0.3138
Epoch 12/20
18647/18647 [==============================] - 11s 578us/step - loss: 1.8930
- acc: 0.3323 - val_loss: 1.9222 - val_acc: 0.3297
Epoch 13/20
18647/18647 [==============================] - 11s 575us/step - loss: 1.8862
- acc: 0.3368 - val_loss: 1.9412 - val_acc: 0.3198
Epoch 14/20
18647/18647 [==============================] - 11s 593us/step - loss: 1.8644
- acc: 0.3409 - val_loss: 1.9385 - val_acc: 0.3175
Epoch 15/20
18647/18647 [==============================] - 11s 571us/step - loss: 1.8588
- acc: 0.3436 - val_loss: 1.9255 - val_acc: 0.3271
Epoch 16/20
18647/18647 [==============================] - 11s 573us/step - loss: 1.8497
- acc: 0.3465 - val_loss: 1.9400 - val_acc: 0.3226
Epoch 17/20
18647/18647 [==============================] - 11s 592us/step - loss: 1.8285
- acc: 0.3562 - val_loss: 1.9394 - val_acc: 0.3275
Validation Log Loss of  Model in Current Run:  1.9222252802920925
Average CV Loss of 10 Runs : 1.918774524178709
```

In [98]:

```
avg_pred=np.zeros((xtr.shape[0],12))
for i in range(len(model_list_2)):
    train_pred=model_list_2[i].predict_proba(xtr)
    avg_pred+=train_pred
avg_pred/=len(model_list_2)
print("Train Average Log-Loss: ",log_loss(ytr, avg_pred))
```

```
Train Average Log-Loss:  1.7068032853037864
```

In [99]:

```
avg_pred=np.zeros((xcv.shape[0],12))
for i in range(len(model_list_2)):
    cv_pred=model_list_2[i].predict_proba(xcv)
    avg_pred+=cv_pred
avg_pred/=len(model_list_2)
print("Validation Average Log-Loss: ",log_loss(ycv, avg_pred))
```

```
Validation Average Log-Loss:  1.9012935504651483
```

In [100]:

```
avg_pred=np.zeros((X_test_events.shape[0],12))
for i in range(len(model_list_2)):
    test_pred=model_list_2[i].predict_proba(X_test_events)
    avg_pred+=test_pred
avg_pred/=len(model_list_2)
```

In [101]:
```
np.save('nn2_events_1',avg_pred)
```

OBSERVATIONS:

1. THE TRAIN AND VALIDATION LOSSES ARE 1.7068 AND 1.9012 RESPECTIVELY.

XGBOOST

In [102]:
```
ytr.shape
```

Out[102]:
```
(18647, 12)
```

In [103]:
```
xtr, xcv, ytr, ycv = train_test_split(X_train_events, y,stratify=y,test_size=0.2,random_state=9)
```

In [104]:
```
ytr.shape
```

Out[104]:
```
(18647,)
```

In [105]:
```
xgb = XGBClassifier(n_estimators=350, n_jobs=-1,learning_rate=0.05, colsample_bytree=0.7, max_depth=5,subsample=0.7,objective='multi:softprob',num_class=12,eval_metric='mlogloss')
xgb.fit(xtr, ytr)
#Using Model Calibration
clf = CalibratedClassifierCV(xgb, method="sigmoid")
clf.fit(xtr, ytr)

pred_y=clf.predict_proba(xtr)
print("Train Log Loss :",log_loss(ytr, pred_y))


pred_y=clf.predict_proba(xcv)
print("Validation Log Loss :",log_loss(ycv, pred_y))
```

```
Train Log Loss : 1.2839666002195145
Validation Log Loss : 2.057339870807861
```

In [106]:
```
events_pred_xgb=clf.predict_proba(X_test_events)
```

In [107]:
```
np.save('xgb_events_1.npy',events_pred_xgb)
```

OBSERVATIONS: THE TRAIN AND VALIDATION LOSSES ARE 1.2839 AND 2.0573 RESPECTIVELY.

LOGISTIC REGRESSION

In [108]:
```
# Train a Logistic regression+Calibration model using text features whicha re
 on-hot encoded
alpha = [0.001,0.01,0.02,0.1,0.15,1,10]


for i in alpha:
    clf = LogisticRegression(C=i, class_weight='balanced', multi_class='multin
omial',solver='lbfgs')
    clf.fit(xtr, ytr)
    #Using Model Calibration
    sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
    sig_clf.fit(xtr, ytr)
    predict_y = sig_clf.predict_proba(xcv)
    print('For values of C = ', i, "The validation log loss is:",log_loss(ycv,
predict_y))
```

```
For values of C =  0.001 The validation log loss is: 2.0982561817893224
For values of C =  0.01 The validation log loss is: 2.019844834880824
For values of C =  0.02 The validation log loss is: 2.0160115578199274
For values of C =  0.1 The validation log loss is: 2.043084585773688
For values of C =  0.15 The validation log loss is: 2.055436921152977
For values of C =  1 The validation log loss is: 2.1074612177933965
For values of C =  10 The validation log loss is: 2.13818998200252
```

WE CHOSE OUR BEST C TO BE 0.02

In [109]:
```
clf = LogisticRegression(C=0.02, class_weight='balanced', multi_class='multino
mial', solver='lbfgs')
clf.fit(xtr, ytr)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(xtr, ytr)

predict_y = sig_clf.predict_proba(xtr)
loss=log_loss(ytr, predict_y)
print("The train log loss for best C is:",loss)
predict_y = sig_clf.predict_proba(xcv)
loss=log_loss(ycv, predict_y)
print("The validation log loss for best C is:",loss)
```

```
The train log loss for best C is: 1.840631737548809
The validation log loss for best C is: 2.0160115578199274
```

In [110]:
```
events_pred_lr=clf.predict_proba(X_test_events)
```

In [111]:
```
#saving the model
np.save('lr_events_1.npy',events_pred_lr)
```

OBSERVATIONS: WE GOT TRAIN AND VALIDATION LOSS AS 1.84 AND 2.0160 RESPECTIVELY.

MODEL ENSEMBLING

MACHINE LEARNING MODELS

WE USE LOGISTIC REGRESSION AND XGBOOST WITH EVENTS AND WITHOUTS DATA AND WE CONCATENATE THE RESULTS.

In [112]:
```
lr1=np.load("lr_noevents.npy")
lr2=np.load("lr_events_1.npy")

xgb1=np.load("xgb_noevents_1.npy")
xgb2=np.load("xgb_events_1.npy")
```

In [113]:
```
w1=0.5
w2=0.5
w3=0.3
w4=0.5

test1=(w1*lr1)+(w2*xgb1)

test2=(w3*lr2)+(w4*xgb2)
```

In [114]:
```
gatrain=pd.read_csv('gender_age_train.csv',index_col = 'device_id')
targetencoder = LabelEncoder().fit(gatrain.group)
y = targetencoder.transform(gatrain.group)
nclasses = len(targetencoder.classes_)
```

In [115]:
```
pred_1 = pd.DataFrame(test1, index = gatest_noevents.index, columns=targetencoder.classes_)
pred_2 = pd.DataFrame(test2, index = gatest_events.index, columns=targetencoder.classes_)
final_pred=pd.concat([pred_1,pred_2], axis=0)
final_pred.shape
```

Out[115]:
```
(112071, 12)
```

In [116]:
```
final_pred.to_csv('ml_final.csv',index=True)
```

ENSEMBLING NEURAL NETS

In [117]:
```
noevents_nn1=np.load("nn1_noevents_1.npy")
noevents_nn2=np.load("nn2_noevents_1.npy")

events_nn1=np.load("nn1_events_1.npy")
events_nn2=np.load("nn2_events_1.npy")
```

WE ARE TAKING ONLY NEURAL NETWORK 1 FOR DEVICES WITHOUT EVENTS AND FOR DEVICES WITH EVENTS WE ARE TAKING AVERAGE OF BOTH NETWORKS.

In [118]:
```
w1=0.5
w2=0.5

test1=(1*noevents_nn1)

test2=(0.5*events_nn1)+(0.5*events_nn2)
```

In [119]:
```
gatrain=pd.read_csv('gender_age_train.csv',index_col = 'device_id')
```

In [120]:
```
targetencoder = LabelEncoder().fit(gatrain.group)
y = targetencoder.transform(gatrain.group)
nclasses = len(targetencoder.classes_)
```

In [121]:
```
pred_1 = pd.DataFrame(test1, index = gatest_noevents.index, columns=targetencoder.classes_)
pred_2 = pd.DataFrame(test2, index = gatest_events.index, columns=targetencoder.classes_)
final_pred_1=pd.concat([pred_1,pred_2], axis=0)
final_pred_1.shape
```

Out[121]:
```
(112071, 12)
```

In [122]:
```
final_pred_1.to_csv('dl_sub_1.csv',index=True)
```

RESULT

In [124]:
```python
from prettytable import PrettyTable

Result = PrettyTable()
Result.field_names = ["Model", "Data", "TRAIN LOSS"," Validation loss"]
Result.add_row(["Logistic Regression", "without events",  2.3628,2.3891])
Result.add_row(["XGboost", "without events",  2.3718,2.3929])
Result.add_row(["Avg Neural Network-1", "without events",  2.3528,2.3577])
Result.add_row(["Avg Neural Network-2", "without events",  2.3770,2.3788])

Result.add_row(["Logistic Regression", "WITH events",  1.8406,2.0160])
Result.add_row(["XGboost", "WITH events", 1.2839,2.0573])
Result.add_row(["Avg Neural Network-1", "WITH events",  1.5406,1.9074])
Result.add_row(["Avg Neural Network-2", "WITH events",  1.7068,1.9012])
Result.add_row(['LOGISTIC REGRESSION','FULL DATA',2.4145,2.3540])

print(Result)
```

```
+----------------------+----------------+------------+------------------+
|        Model         |      Data      | TRAIN LOSS |  Validation loss |
+----------------------+----------------+------------+------------------+
| Logistic Regression  | without events |   2.3628   |      2.3891      |
|       XGboost        | without events |   2.3718   |      2.3929      |
| Avg Neural Network-1 | without events |   2.3528   |      2.3577      |
| Avg Neural Network-2 | without events |   2.377    |      2.3788      |
| Logistic Regression  |  WITH events   |   1.8406   |      2.016       |
|       XGboost        |  WITH events   |   1.2839   |      2.0573      |
| Avg Neural Network-1 |  WITH events   |   1.5406   |      1.9074      |
| Avg Neural Network-2 |  WITH events   |   1.7068   |      1.9012      |
| LOGISTIC REGRESSION  |   FULL DATA    |   2.4145   |      2.354       |
+----------------------+----------------+------------+------------------+
```

REFERENCES

1.https://www.kaggle.com/dvasyukova/a-linear-model-on-apps-and-labels
(https://www.kaggle.com/dvasyukova/a-linear-model-on-apps-and-labels)
2.https://www.kaggle.com/c/talkingdata-mobile-user-demographics/discussion/23424
(https://www.kaggle.com/c/talkingdata-mobile-user-demographics/discussion/23424)

In [ ]:

In [ ]: