In this notebook based on the previous notebook's results I am trying to use all the features and also adding pilot feature in it and trying out the models.

In [ ]:

```
!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0 (X11; Li
nux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36" --
header="Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9" --header="Accept-Language: en-GB,en-
US;q=0.9,en;q=0.8" --header="Referer: https://www.kaggle.com/" "https://storage.googleapi
s.com/kaggle-competitions-data/kaggle-v2/11835/224935/compressed/train.csv.zip?GoogleAcce
ssId=web-data@kaggle-161607.iam.gserviceaccount.com&Expires=1597542972&Signature=pJwkpuWj
RxVEL%2F0vyManv3qHj5dPZ35W75c3ZwCzo47KhBhdAPqrk5K5FCBtD12Rs9wlkaEhgvFwDXhaR00w9wauMYY0jr%
2BVnzG85DF%2F0mQ52xpD7RD8qjebQxsKHTyojKwGUMjM87qYGQDZid4xIs9djE15BY3zmkDsQLE8sagMYYW9%2Bq
r2gKvkVnM5FxW4NOtjkVEATINLCSBBUOD3L8y%2BZdXWtBDu9iUxKF4r9voOUZJfAQZiTRITrxX%2F1O8iqZRPCXe
6Ca0CoNXj9nyWB9GH4AcSGTIGop9r4YVDZOxQ9TIc8J8sNaw%2Br7F19C5R0R1Y338zEkqkFoGarVAAEw%3D%3D&r
esponse-content-disposition=attachment%3B+filename%3Dtrain.csv.zip" -c -O 'train.csv.zip'
```

```
--2020-08-13 01:56:45--  https://storage.googleapis.com/kaggle-competitions-data/kaggle-v
2/11835/224935/compressed/train.csv.zip?GoogleAccessId=web-data@kaggle-161607.iam.gservic
eaccount.com&Expires=1597542972&Signature=pJwkpuWjRxVEL%2F0vyManv3qHj5dPZ35W75c3ZwCzo47Kh
BhdAPqrk5K5FCBtD12Rs9wlkaEhgvFwDXhaR00w9wauMYY0jr%2BVnzG85DF%2F0mQ52xpD7RD8qjebQxsKHTyojK
wGUMjM87qYGQDZid4xIs9djE15BY3zmkDsQLE8sagMYYW9%2Bqr2gKvkVnM5FxW4NOtjkVEATINLCSBBUOD3L8y%2
BZdXWtBDu9iUxKF4r9voOUZJfAQZiTRITrxX%2F1O8iqZRPCXe6Ca0CoNXj9nyWB9GH4AcSGTIGop9r4YVDZOxQ9T
Ic8J8sNaw%2Br7F19C5R0R1Y338zEkqkFoGarVAAEw%3D%3D&response-content-disposition=attachment%
3B+filename%3Dtrain.csv.zip
Resolving storage.googleapis.com (storage.googleapis.com)... 74.125.203.128, 74.125.204.1
28, 64.233.189.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|74.125.203.128|:443... conn
ected.
HTTP request sent, awaiting response... 200 OK
Length: 456337398 (435M) [application/zip]
Saving to: ‘train.csv.zip’

train.csv.zip       100%[===================>] 435.20M  24.5MB/s    in 20s

2020-08-13 01:57:05 (22.1 MB/s) - ‘train.csv.zip’ saved [456337398/456337398]
```

In [ ]:

```
!unzip train.csv.zip
```

```
Archive:  train.csv.zip
  inflating: train.csv
```

In [ ]:

```
!wget --header="Host: storage.googleapis.com" --header="User-Agent: Mozilla/5.0 (X11; Li
nux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/84.0.4147.105 Safari/537.36" --
header="Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/ap
ng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9" --header="Accept-Language: en-GB,en-
US;q=0.9,en;q=0.8" --header="Referer: https://www.kaggle.com/" "https://storage.googleapi
s.com/kaggle-competitions-data/kaggle-v2/11835/224935/compressed/test.csv.zip?GoogleAcces
sId=web-data@kaggle-161607.iam.gserviceaccount.com&Expires=1596891757&Signature=bjWisHQtx
8jjPi2pj6S4wSnRT%2FhKcPZW4wjJWGbh8gYHBeSqI1jKA0B4DYZuf7VfkaiRAzUcUcPIBJjOrzPVU3pvPHjrwzYU
9VgWOcWfQF4vm7zHnjxpnfcwCWO8BenLgGLwK9%2B9BmNCHKBV5R6DE%2BwZvfaraCVHKM5mUwne9MXhR7VoaFVAH
Ah4T%2B3W7ibgqgzaU2ycBxSA8eE3nWZBCuPcXts9XyLAE8ZKqvQhCATBjE3hsz8eKDCwTUtbBU9oxc8e5WE%2Fn0
ZkLw6pd9OYFXZSDlNj5Rje%2BVTSJIiH9Vuln7AihRSObRvFRoIfd2V0fAgQz3LncqEukE1bGAkXuA%3D%3D&resp
onse-content-disposition=attachment%3B+filename%3Dtest.csv.zip" -c -O 'test.csv.zip'
```

```
--2020-08-05 13:03:02--  https://storage.googleapis.com/kaggle-competitions-data/kaggle-v
2/11835/224935/compressed/test.csv.zip?GoogleAccessId=web-data@kaggle-161607.iam.gservice
account.com&Expires=1596891757&Signature=bjWisHQtx8jjPi2pj6S4wSnRT%2FhKcPZW4wjJWGbh8gYHBe
SqI1jKA0B4DYZuf7VfkaiRAzUcUcPIBJjOrzPVU3pvPHjrwzYU9VgWOcWfQF4vm7zHnjxpnfcwCWO8BenLgGLwK9%
2B9BmNCHKBV5R6DE%2BwZvfaraCVHKM5mUwne9MXhR7VoaFVAHAh4T%2B3W7ibgqgzaU2ycBxSA8eE3nWZBCuPcXt
s9XyLAE8ZKqvQhCATBjE3hsz8eKDCwTUtbBU9oxc8e5WE%2Fn0ZkLw6pd9OYFXZSDlNj5Rje%2BVTSJIiH9Vuln7A
ihRSObRvFRoIfd2V0fAgQz3LncqEukE1bGAkXuA%3D%3D&response-content-disposition=attachment%3B+
filename%3Dtest.csv.zip
Resolving storage.googleapis.com (storage.googleapis.com)... 173.194.202.128, 74.125.20.1
28, 74.125.197.128, ...
Connecting to storage.googleapis.com (storage.googleapis.com)|173.194.202.128|:443... con
nected.
HTTP request sent, awaiting response... 200 OK
Length: 1791131386 (1.7G) [application/zip]
Saving to: ‘test.csv.zip’

test.csv.zip        100%[===================>]   1.67G  65.5MB/s    in 26s

2020-08-05 13:03:29 (64.6 MB/s) - ‘test.csv.zip’ saved [1791131386/1791131386]
```

In [ ]:

```
! unzip test.csv
```

```
Archive:  test.csv.zip
  inflating: test.csv
```

In [ ]:

```
import warnings
import itertools
import numpy as np
import pandas as pd
import seaborn as sns
import lightgbm as lgb
import matplotlib.pyplot as plt
from tqdm import tqdm_notebook as tqdm
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, log_loss
import dask.dataframe as dd
import dask
import gc

from yellowbrick.text import TSNEVisualizer

%matplotlib inline
plt.style.use("fivethirtyeight")

# import os
# print(os.listdir("../input"))

warnings.filterwarnings(action='ignore')
sns.set_style('whitegrid')
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: p
andas.util.testing is deprecated. Use the functions in the public API at pandas.testing i
nstead.
  import pandas.util.testing as tm
/usr/local/lib/python3.6/dist-packages/sklearn/utils/deprecation.py:144: FutureWarning: T
he sklearn.metrics.classification module is  deprecated in version 0.22 and will be remov
ed in version 0.24. The corresponding classes / functions should instead be imported from
sklearn.metrics. Anything that cannot be imported from sklearn.metrics is now part of the
private API.
  warnings.warn(message, FutureWarning)
```

In [ ]:

```
# This is to be used for memory optimization because the data is very large.
def reduce_mem_usage(df):
    """ iterate through all the columns of a dataframe and modify the data type
        to reduce memory usage.
    """
    start_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage of dataframe is {:.2f} MB'.format(start_mem))

    for col in df.columns:
        col_type = df[col].dtype
        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max
:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).m
ax:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
        else:
            df[col] = df[col].astype('category')

    end_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage after optimization is: {:.2f} MB'.format(end_mem))
    print('Decreased by {:.1f}%'.format(100 * (start_mem - end_mem) / start_mem))

    return df
def featureModify(isTrain, numRows):
    if isTrain:
        df = dd.read_csv('train.csv',nrows=numRows)
        df = df.compute()
        # df['pilot'] = 100*df['crew']+df['seat']
        df = reduce_mem_usage(df)
        df['event'] = df['event'].map({
            'A':0,
            'B':1,
            'C':2,
            'D':3
        })
    else:
        df = dd.read_csv('test.csv',nrows=numRows)
        df = df.compute()
        # df['pilot'] = 100*df['crew']+df['seat']
        df = reduce_mem_usage(df)

    return df
train = featureModify(True, None)
y = train['event']
# train = train.drop('event',axis=1)
print(train.shape)
print(train.columns)
```

In [ ]:

```
train['pilot'] = 100*train['seat']+train['crew']
```

In [ ]:

```
train = train[['crew', 'experiment', 'time', 'seat', 'eeg_fp1', 'eeg_f7', 'eeg_f8',
       'eeg_t4', 'eeg_t6', 'eeg_t5', 'eeg_t3', 'eeg_fp2', 'eeg_o1', 'eeg_p3',
       'eeg_pz', 'eeg_f3', 'eeg_fz', 'eeg_f4', 'eeg_c4', 'eeg_p4', 'eeg_poz',
       'eeg_c3', 'eeg_cz', 'eeg_o2', 'ecg', 'r', 'gsr','pilot']]
```

In [ ]:

```
train, train_test, y, y_test = train_test_split(train, y, test_size=0.2, shuffle=True)
train = lgb.Dataset(train, label=y,categorical_feature=[1])
del y
```

```
gc.collect()

train_test = lgb.Dataset(train_test, label=y_test,categorical_feature=[1])
del y_test
gc.collect()
```

Out[ ]:

```
0
```

In [ ]:

```
params = {
        "objective" : "multiclass",
        "metric" : "multi_error",
        'num_class':4,
        "num_leaves" : 30,
        "learning_rate" : 0.01,
        "bagging_fraction" : 0.9,
        "bagging_seed" : 0,
        "num_threads" : 4,
        'min_data_in_leaf':100,
        'min_split_gain':0.00019
}

model = lgb.train(  params,
                    train_set = train,
                    num_boost_round=1000,
                    early_stopping_rounds=200,
                    verbose_eval=100,
                    valid_sets=[train,train_test]
                  )
```

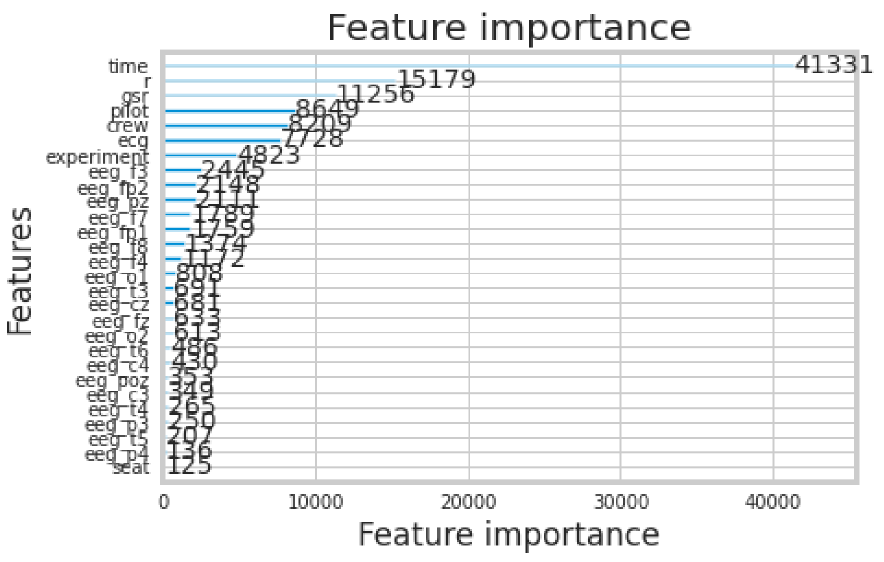
```
Training until validation scores don't improve for 200 rounds.
[100] training's multi_error: 0.0499315 valid_1's multi_error: 0.0505062
[200] training's multi_error: 0.0404403 valid_1's multi_error: 0.040618
[300] training's multi_error: 0.0342679 valid_1's multi_error: 0.0343549
[400] training's multi_error: 0.0262159 valid_1's multi_error: 0.0262089
[500] training's multi_error: 0.0204462 valid_1's multi_error: 0.0205386
[600] training's multi_error: 0.0170064 valid_1's multi_error: 0.0170316
[700] training's multi_error: 0.0141571 valid_1's multi_error: 0.0142529
[800] training's multi_error: 0.0126792 valid_1's multi_error: 0.0127829
[900] training's multi_error: 0.0115937 valid_1's multi_error: 0.0116633
[1000] training's multi_error: 0.0106848 valid_1's multi_error: 0.0108086
Did not meet early stopping. Best iteration is:
[1000] training's multi_error: 0.0106848 valid_1's multi_error: 0.0108086
```

In [ ]:

```
lgb.plot_importance(model)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9cca670860>
```



In [ ]:

```
lgb.create_tree_digraph(model)
```

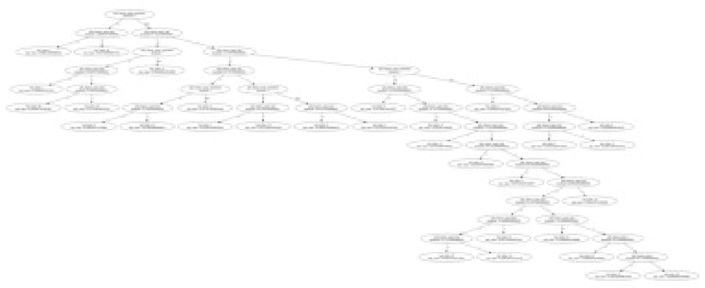
Out[ ]:

In [ ]:

```
lgb.plot_tree(model)
```

Out[ ]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f9cca62f518>
```



In [ ]:

```
import os
os.chdir('/content/drive/My Drive/ML case study/models')
import joblib
model = joblib.load('all_feature_model_with_pilot.pkl')
```

In [ ]:

```
import os

os.chdir('/content')
```

In [ ]:

```
test = featureModify(False, None)
print("Done test read")
```

```
Memory usage of dataframe is 3974.83 MB
Memory usage after optimization is: 1079.37 MB
Decreased by 72.8%
Done test read
```

In [ ]:

```
test.head()
```

Out[ ]:

	id	crew	experiment	time	seat	eeg_fp1	eeg_f7	eeg_f8	eeg_t4	eeg_t6	eeg_t5	eeg_t3	eeg
0	0	1	LOFT	0.000000	0	17.906250	6.128906	0.994629	-28.203125	-47.687500	-187.125000	-33.187500	-4.22
1	1	1	LOFT	0.000000	1	45.875000	94.750000	23.296875	1.391602	2.060547	-5.144531	6.394531	33.40
2	2	1	LOFT	0.003906	0	33.125000	28.359375	-7.238281	-7.691406	-25.828125	-107.250000	12.843750	1.21
3	3	1	LOFT	0.003906	1	43.281250	95.875000	18.703125	-1.432617	-4.234375	-8.023438	7.425781	27.34
4	4	1	LOFT	0.007812	0	7.929688	3.460938	-10.859375	-26.359375	-25.890625	37.000000	-50.343750	11.67

In [ ]:

```
df_sub = pd.DataFrame()
df_sub['id'] = test['id']
test = test.drop('id',axis=1)
```

In [ ]:

```
test['pilot']= 100*test['seat']+test['crew']
```

In [ ]:

```
test = test[['crew', 'experiment', 'time', 'seat', 'eeg_fp1', 'eeg_f7', 'eeg_f8',
        'eeg_t4', 'eeg_t6', 'eeg_t5', 'eeg_t3', 'eeg_fp2', 'eeg_o1', 'eeg_p3',
        'eeg_pz', 'eeg_f3', 'eeg_fz', 'eeg_f4', 'eeg_c4', 'eeg_p4', 'eeg_poz',
        'eeg_c3', 'eeg_cz', 'eeg_o2', 'ecg', 'r', 'gsr','pilot']]
```

In [ ]:

```
test.head()
```

Out[ ]:

	crew	experiment	time	seat	eeg_fp1	eeg_f7	eeg_f8	eeg_t4	eeg_t6	eeg_t5	eeg_t3	eeg_fp2
0	1	LOFT	0.000000	0	17.906250	6.128906	0.994629	-28.203125	-47.687500	-187.125000	-33.187500	-4.22265
1	1	LOFT	0.000000	1	45.875000	94.750000	23.296875	1.391602	2.060547	-5.144531	6.394531	33.40625
2	1	LOFT	0.003906	0	33.125000	28.359375	-7.238281	-7.691406	-25.828125	-107.250000	12.843750	1.21484
3	1	LOFT	0.003906	1	43.281250	95.875000	18.703125	-1.432617	-4.234375	-8.023438	7.425781	27.34375
4	1	LOFT	0.007812	0	7.929688	3.460938	-10.859375	-26.359375	-25.890625	37.000000	-50.343750	-11.67968

In [ ]:

```
y_pred = model.predict(test, num_iteration=model.best_iteration)
```

In [ ]:

```
y_pred
```

Out[ ]:

```
array([[9.99926747e-01, 5.67305914e-06, 5.64473263e-05, 1.11321791e-05],
       [9.99926818e-01, 5.65439506e-06, 5.62616175e-05, 1.12658535e-05],
       [9.99925448e-01, 5.77371003e-06, 5.74488096e-05, 1.13296852e-05],
       ...,
       [9.99862109e-01, 1.06516382e-05, 1.05984529e-04, 2.12546447e-05],
       [9.99822433e-01, 1.37691356e-05, 1.37003840e-04, 2.67943210e-05],
       [9.99862623e-01, 1.06223633e-05, 1.05693241e-04, 2.10615035e-05]])
```

In [ ]:

```
import os
os.chdir('/content/drive/My Drive/ML case study/results')
```

In [ ]:

```
import joblib
joblib.dump(y_pred,'all_feature_model_with_pilot_output.pkl')
```

Out[ ]:

```
['all_feature_model_with_pilot_output.pkl']
```

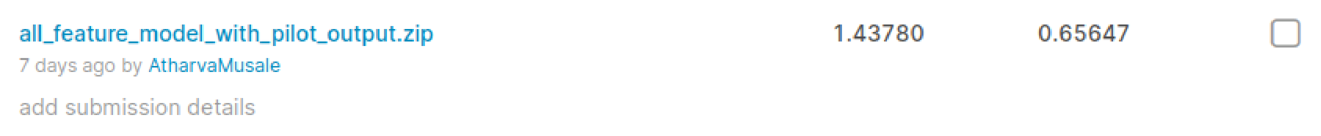
In [ ]:

```
df_sub = pd.DataFrame(np.concatenate((np.arange(len(test))[:, np.newaxis], y_pred), axis
=1), columns=['id', 'A', 'B', 'C', 'D'])
df_sub['id'] = df_sub['id'].astype(int)
print(df_sub)
df_sub.to_csv("all_feature_model_with_pilot_output.csv", index=False)
```

```
                id         A         B         C         D
0                0  0.999927  0.000006  0.000056  0.000011
1                1  0.999927  0.000006  0.000056  0.000011
2                2  0.999925  0.000006  0.000057  0.000011
3                3  0.999927  0.000006  0.000056  0.000011
4                4  0.999928  0.000006  0.000055  0.000011
...            ...       ...       ...       ...       ...
17965138  17965138  0.999863  0.000011  0.000105  0.000021
17965139  17965139  0.999822  0.000014  0.000137  0.000027
17965140  17965140  0.999862  0.000011  0.000106  0.000021
17965141  17965141  0.999822  0.000014  0.000137  0.000027
17965142  17965142  0.999863  0.000011  0.000106  0.000021

[17965143 rows x 5 columns]
```

After creating pilot feature



all_feature_model_with_pilot_output.zip   1.43780   0.65647
7 days ago by AtharvaMusale
add submission details

In [ ]: