```python
pp=BGM.predict_proba(X)# Calcualting the probabilities of each prediction
df_new=pd.DataFrame(X,columns=feats)
df_new[[f'predict_proba_{i}' for i in range(7)]]=pp # creating new dataframe columns of probabilites
df_new['preds']=preds
df_new['predict_proba']=np.max(pp,axis=1)
df_new['predict']=np.argmax(pp,axis=1)

train_index=np.array([])
for n in range(7):
    n_inx=df_new[(df_new.preds==n) & (df_new.predict_proba > 0.68)].index
    train_index = np.concatenate((train_index, n_inx))
```

✓ 0.0s Python

```python
#ricopue's notebook's code snippet
from sklearn.model_selection import StratifiedKFold
X_new=df_new.loc[train_index][feats]
y=df_new.loc[train_index]['preds']

params_lgb = {'learning_rate': 0.06,'objective': 'multiclass','boosting': 'gbdt','n_jobs': -1,'verbosity': -1, 'num_classes':7}

model_list=[]

gkf = StratifiedKFold(11)
for fold, (train_idx, valid_idx) in enumerate(gkf.split(X_new,y)):

    tr_dataset = lgb.Dataset(X_new.iloc[train_idx],y.iloc[train_idx],feature_name = feats)
    vl_dataset = lgb.Dataset(X_new.iloc[valid_idx],y.iloc[valid_idx],feature_name = feats)

    model = lgb.train(params = params_lgb,
                train_set = tr_dataset,
                valid_sets =  vl_dataset,
                num_boost_round = 5000,
                callbacks=[ lgb.early_stopping(stopping_rounds=300, verbose=False), lgb.log_evaluation(period=200)])

    model_list.append(model)
```

✓ 5.7s Python

```
[200]   valid_0's multi_logloss: 0.000291562
[200]   valid_0's multi_logloss: 0.000291119
[200]   valid_0's multi_logloss: 0.00026959
[200]   valid_0's multi_logloss: 0.0127449
[200]   valid_0's multi_logloss: 0.000164035
[400]   valid_0's multi_logloss: 0.000162874
[600]   valid_0's multi_logloss: 0.000162571
[800]   valid_0's multi_logloss: 0.000162431
[1000]  valid_0's multi_logloss: 0.000162351
```

```python
labels=np.argmax(lgb_preds,axis=1)
```

✓ 0.0s Python

```python
fig = plt.figure(figsize=(8,6))
ax = plt.subplot(label="bla")
sns.scatterplot(df[feats], marker='o')
ax.set_title("Before clustering")
```

✓ 0.4s Python

```
Text(0.5, 1.0, 'Before clustering')
```