

```
# -*- coding: utf-8 -*-  
"""
```

Created on Sat Oct 28 16:16:39 2017

```
@author: 麦芽的香气  
"""
```

```
import pandas as pd  
import numpy as np  
from sklearn import preprocessing  
import xgboost as xgb  
import lightgbm as lgb  
path='./'  
df=pd.read_csv(path+u'训练数据-ccf_first_round_user_shop_behavior.csv')  
shop=pd.read_csv(path+u'训练数据-ccf_first_round_shop_info.csv')  
test=pd.read_csv(path+u'AB榜测试集-evaluation_public.csv')  
df=pd.merge(df,shop[['shop_id','mall_id']],how='left',on='shop_id')  
df['time_stamp']=pd.to_datetime(df['time_stamp'])  
train=pd.concat([df,test])  
mall_list=list(set(list(shop.mall_id)))  
result=pd.DataFrame()  
for mall in mall_list:  
    train1=train[train.mall_id==mall].reset_index(drop=True)  
    l=[]  
    wifi_dict = {}  
    for index,row in train1.iterrows():  
        r = {}  
        wifi_list = [wifi.split('|') for wifi in row['wifi_infos'].split(';')]  
        for i in wifi_list:  
            r[i[0]]=int(i[1])  
            if i[0] not in wifi_dict:  
                wifi_dict[i[0]]=1  
            else:  
                wifi_dict[i[0]]+=1  
        l.append(r)  
    delate_wifi=[]  
    for i in wifi_dict:  
        if wifi_dict[i]<20:  
            delate_wifi.append(i)  
    m=[]  
    for row in l:  
        new={}  
        for n in row.keys():  
            if n not in delate_wifi:  
                new[n]=row[n]  
        m.append(new)  
    train1 = pd.concat([train1,pd.DataFrame(m)], axis=1)  
    df_train=train1[train1.shop_id.notnull()]  
    df_test=train1[train1.shop_id.isnull()]
```

```

lbl = preprocessing.LabelEncoder()
lbl.fit(list(df_train['shop_id'].values))
df_train['label'] = lbl.transform(list(df_train['shop_id'].values))
num_class=df_train['label'].max()+1
params = {
    'objective': 'multi:softmax',
    'eta': 1,
    'max_depth': 9,
    'eval_metric': 'merror',
    'seed': 0,
    'missing': -999,
    'num_class':num_class,
    'silent' : 1
}
feature=[x for x in train1.columns if x not in
['user_id','label','shop_id','time_stamp','mall_id','wifi_infos']]
xgbtrain = xgb.DMatrix(df_train[feature], df_train['label'])
xgbtest = xgb.DMatrix(df_test[feature])
watchlist = [ (xgbtrain,'train'), (xgbtrain, 'test') ]
num_rounds=1
model = xgb.train(params, xgbtrain, num_rounds, watchlist, early_stopping_rounds=15)
df_test['label']=model.predict(xgbtest)
df_test['shop_id']=df_test['label'].apply(lambda x:lbl.inverse_transform(int(x)))
r=df_test[['row_id','shop_id']]
result=pd.concat([result,r])
result['row_id']=result['row_id'].astype('int')
result.to_csv(path+'sub.csv',index=False)

```