

```
In [ ]: import pandas as pd
        from myPackage import module #自己写的包
        import warnings
        warnings.filterwarnings("ignore")
```

1. 我们使用 task3 中用 KNN 填补得到的数据

首先将pm2.5数据分为三类：当 $pm2.5 \leq 35$ ，我们记为1类；当 $35 < pm2.5 \leq 150$ ，我们记为2类；当 $pm2.5 > 150$ ，我们记为3类

```
In [ ]: pm25_knn=pd.read_csv('pm25_data_knn.csv')
        # label pm2.5
        for i in range(len(pm25_knn)):
            if pm25_knn.loc[i,'pm2.5'] <= 35:
                pm25_knn.loc[i,'pm2.5'] = 1
            elif pm25_knn.loc[i,'pm2.5'] > 150:
                pm25_knn.loc[i,'pm2.5'] = 3
            else:
                pm25_knn.loc[i,'pm2.5'] = 2
```

smooth the time series over 3-hour moving windows

```
In [ ]: # smooth data
        pm25_knn_smoothed=pm25_knn
        for i in range(len(pm25_knn)-2):
            pm25_knn_smoothed.loc[i+1,'DEWP']=(pm25_knn.loc[i,'DEWP']+pm25_knn.loc[i+1,'DEWP']+pm25_knn.loc[i+2,'DEWP'])/3
            pm25_knn_smoothed.loc[i+1,'TEMP']=(pm25_knn.loc[i,'TEMP']+pm25_knn.loc[i+1,'TEMP']+pm25_knn.loc[i+2,'TEMP'])/3
            pm25_knn_smoothed.loc[i+1,'PRES']=(pm25_knn.loc[i,'PRES']+pm25_knn.loc[i+1,'PRES']+pm25_knn.loc[i+2,'PRES'])/3
            pm25_knn_smoothed.loc[i+1,'Iws']=(pm25_knn.loc[i,'Iws']+pm25_knn.loc[i+1,'Iws']+pm25_knn.loc[i+2,'Iws'])/3
            pm25_knn_smoothed.loc[i+1,'Is']=(pm25_knn.loc[i,'Is']+pm25_knn.loc[i+1,'Is']+pm25_knn.loc[i+2,'Is'])/3
            pm25_knn_smoothed.loc[i+1,'Ir']=(pm25_knn.loc[i,'Ir']+pm25_knn.loc[i+1,'Ir']+pm25_knn.loc[i+2,'Ir'])/3
```

获得训练集和测试集

```
In [ ]: test_data1, train_data1 = module.train_test_split(pm25_knn_smoothed)
```

我们用 XGBClassifier 来建模。经过调参，最终使用 XGBClassifier(learning_rate=0.1, n_estimators=600, random_state=0)

```
In [ ]: import xgboost as xgb

        var=['year','month','day','hour','DEWP','TEMP','PRES','Iws','Is','Ir','cbwd_NE','cbwd_NW','cbwd_SE','cbwd_cv']
        X_train1 = train_data1[var]
        X_test1 = test_data1[var]
```

```

y_train1 = train_data1['pm2.5']
y_test1 = test_data1['pm2.5']

modell=xgb.XGBClassifier(learning_rate=0.1, n_estimators=600,max_depth=5)
modell.fit(X_train1, y_train1)
print("the classification precision: %.4f" % modell.score(X_test1,y_test1))

```

[12:07:46] WARNING: D:\bld\xgboost-split_1645118015404\work\src\learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
the classification precision: 0.7460

模型的分类准确率为74.6%

1. 我们使用 task3 中用 mean 填补得到的数据

首先将pm2.5数据分为三类：当 $pm2.5 \leq 35$ ，我们记为1类；当 $35 < pm2.5 \leq 150$ ，我们记为2类；当 $pm2.5 > 150$ ，我们记为3类

```

In [ ]: pm25_mean=pd.read_csv('pm25_data_mean.csv')
# label pm2.5
for i in range(len(pm25_mean)):
    if pm25_mean.loc[i,'pm2.5'] <= 35:
        pm25_mean.loc[i,'pm2.5'] = 1
    elif pm25_mean.loc[i,'pm2.5'] > 150:
        pm25_mean.loc[i,'pm2.5'] = 3
    else:
        pm25_mean.loc[i,'pm2.5'] = 2

```

smooth the time series over 3-hour moving windows

```

In [ ]: # smooth data
pm25_mean_smoothed=pm25_mean
for i in range(len(pm25_mean)-2):
    pm25_mean_smoothed.loc[i+1,'DEWP']=(pm25_mean.loc[i,'DEWP']+pm25_mean.loc[i+1,'DEWP']+pm25_mean.loc[i+2,'DEWP'])/3
    pm25_mean_smoothed.loc[i+1,'TEMP']=(pm25_mean.loc[i,'TEMP']+pm25_mean.loc[i+1,'TEMP']+pm25_mean.loc[i+2,'TEMP'])/3
    pm25_mean_smoothed.loc[i+1,'PRES']=(pm25_mean.loc[i,'PRES']+pm25_mean.loc[i+1,'PRES']+pm25_mean.loc[i+2,'PRES'])/3
    pm25_mean_smoothed.loc[i+1,'Iws']=(pm25_mean.loc[i,'Iws']+pm25_mean.loc[i+1,'Iws']+pm25_mean.loc[i+2,'Iws'])/3
    pm25_mean_smoothed.loc[i+1,'Is']=(pm25_mean.loc[i,'Is']+pm25_mean.loc[i+1,'Is']+pm25_mean.loc[i+2,'Is'])/3
    pm25_mean_smoothed.loc[i+1,'Ir']=(pm25_mean.loc[i,'Ir']+pm25_mean.loc[i+1,'Ir']+pm25_mean.loc[i+2,'Ir'])/3

```

获得训练集和测试集

```

In [ ]: test_data2, train_data2 = module.train_test_split(pm25_mean_smoothed)

```

我们用 XGBClassifier 来建模。经过调参，最终使用 XGBClassifier(learning_rate=0.1, n_estimators=600, random_state=0)

```
In [ ]: import xgboost as xgb

var=['year', 'month', 'day', 'hour', 'DEWP', 'TEMP', 'PRES', 'Iws', 'Is', 'Ir', 'cbwd_NE', 'cbwd_NW', 'cbwd_SE', 'cbwd_cv']
X_train2 = train_data2[var]
X_test2 = test_data2[var]
y_train2 = train_data2['pm2.5']
y_test2 = test_data2['pm2.5']

model2=xgb.XGBClassifier(learning_rate=0.1, n_estimators=600,max_depth=5)
model2.fit(X_train2, y_train2)
print("the classification precision: %.4f" % model2.score(X_test2,y_test2))
```

```
[12:10:09] WARNING: D:\bld\xgboost-split_1645118015404\work\src\learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.
the classification precision: 0.7455
```

模型的分类准确率为74.55%