XGBoost

1. 西班牙数据集

train index: [6426, 10427] train_len: 4000 test index: [14389, 15390] test_len: 1000

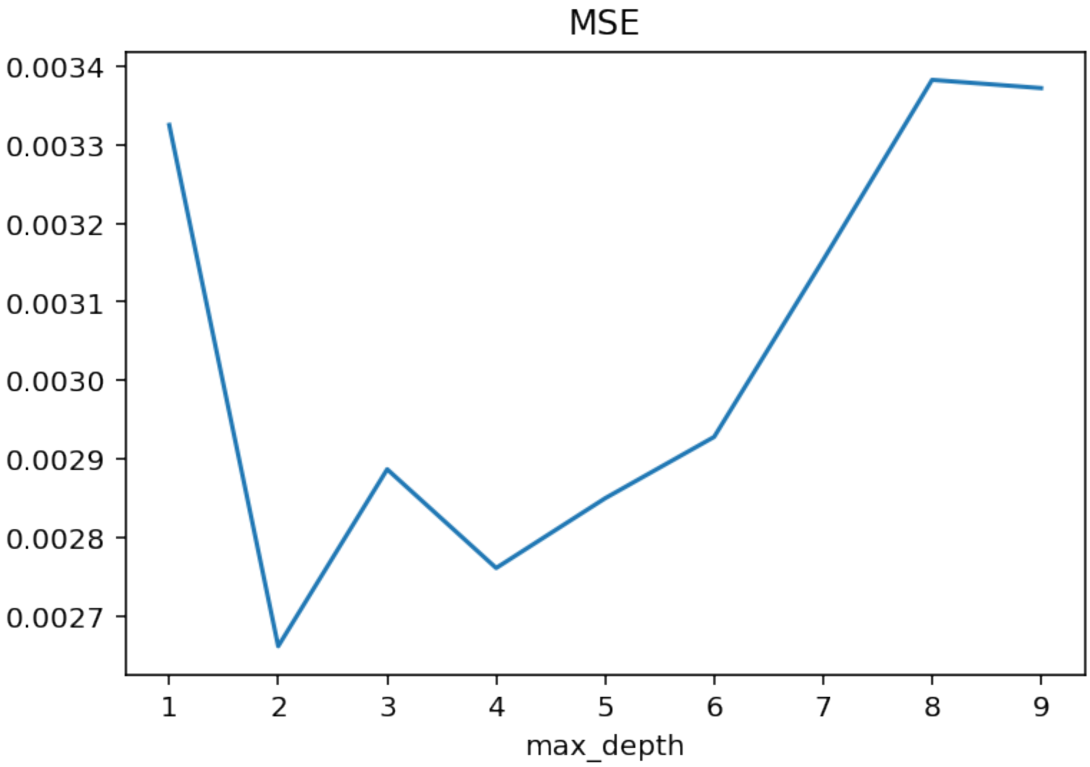
- **输入特征：**

```
'wind_speed', 'sin(wd)', 'cos(wd)', 【t期】
'wind_speed-1', 'sin(wd)-1','cos(wd)-1', 'wind_power-1' 【t-1期】
```

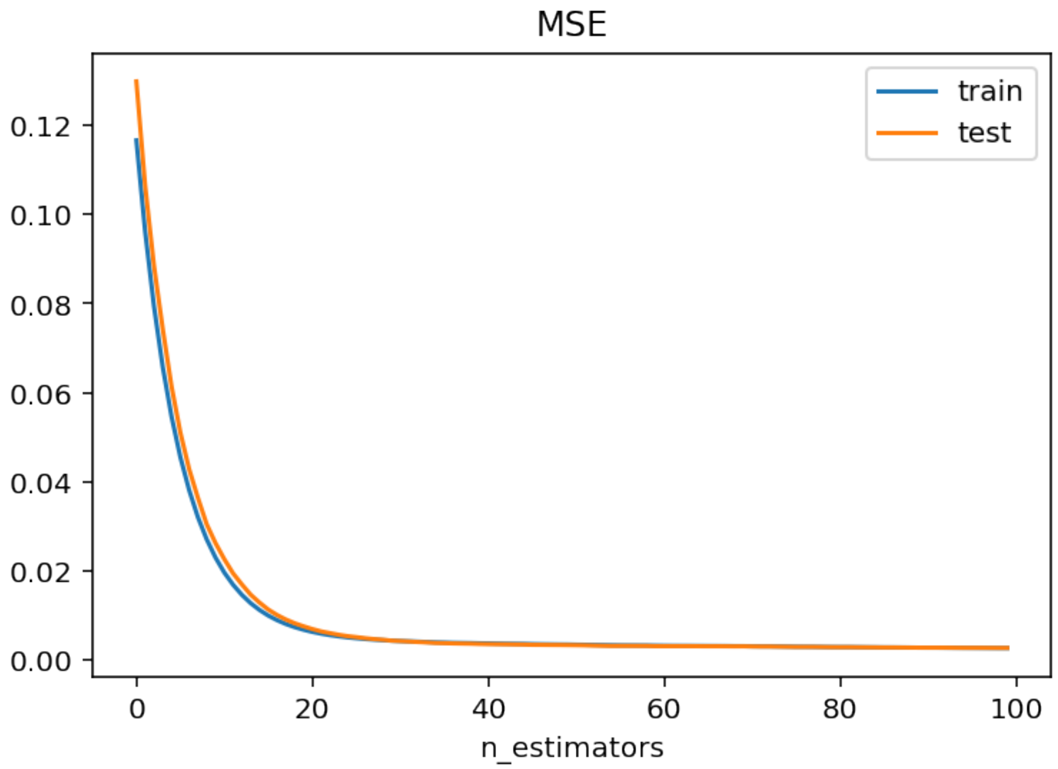
- **输出：** wind_power

1.1 寻找最大深度

max_depth = 2



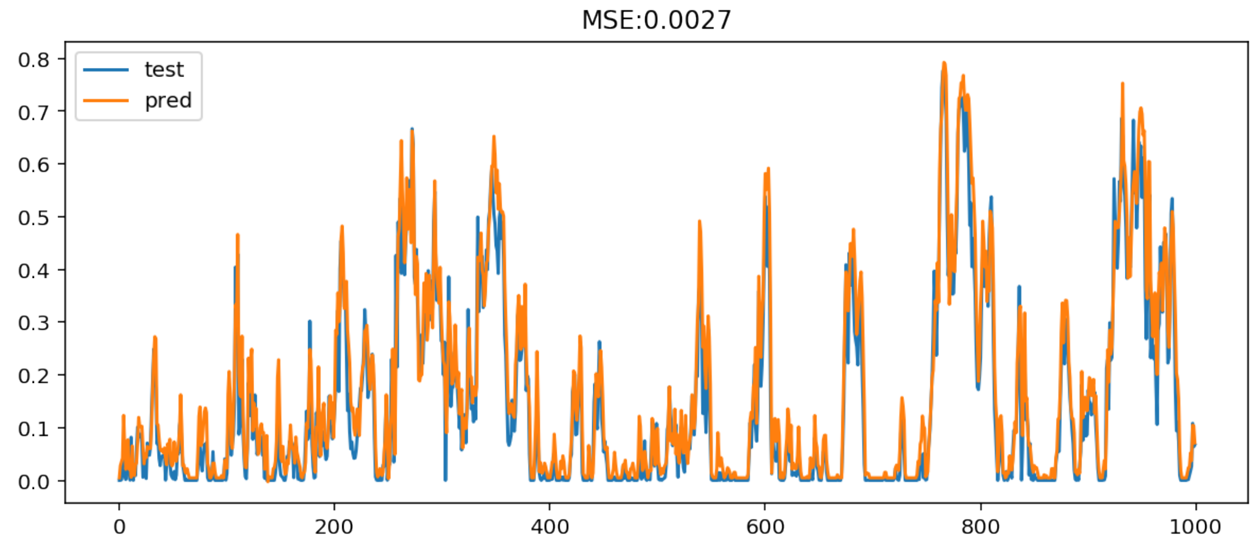
1.2 n_estimators



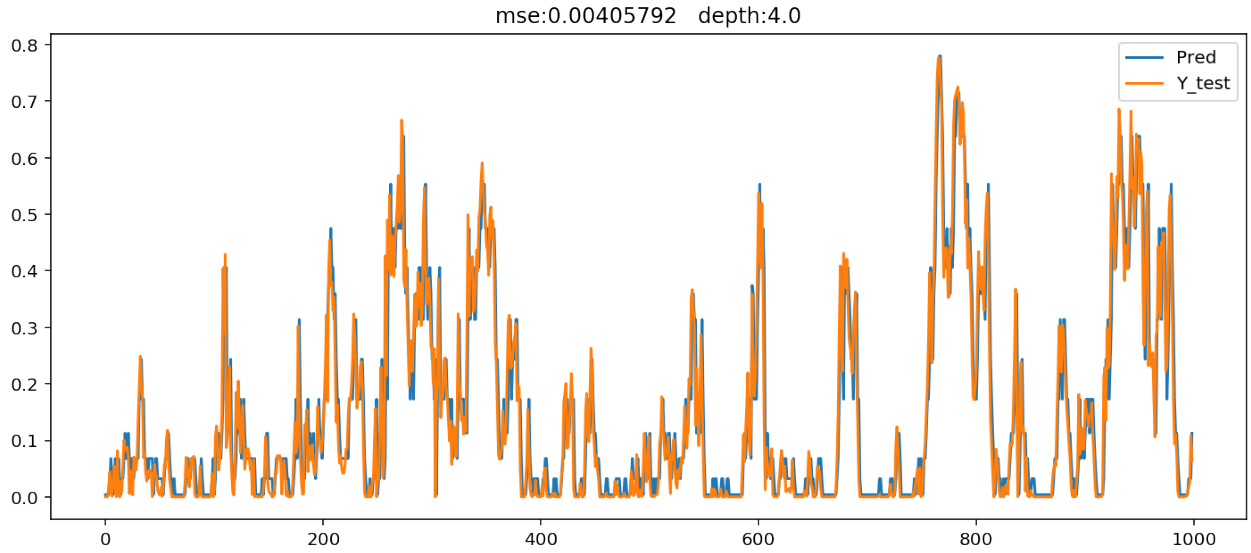
最终设置：

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0,
             importance_type='gain', learning_rate=0.1, max_delta_step=0,
             max_depth=2, min_child_weight=1, missing=None,
n_estimators=100,
             n_jobs=4, nthread=None, objective='reg:linear',
random_state=0,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
             silent=None, subsample=1, verbosity=1)
```

test mse: 0.0026614413208772537



与单纯决策树对比：



2. 美国数据集

train index: [3001, 7002] train_len: 4000 test index: [2000, 3001] test_len: 1000

- **输入特征：**

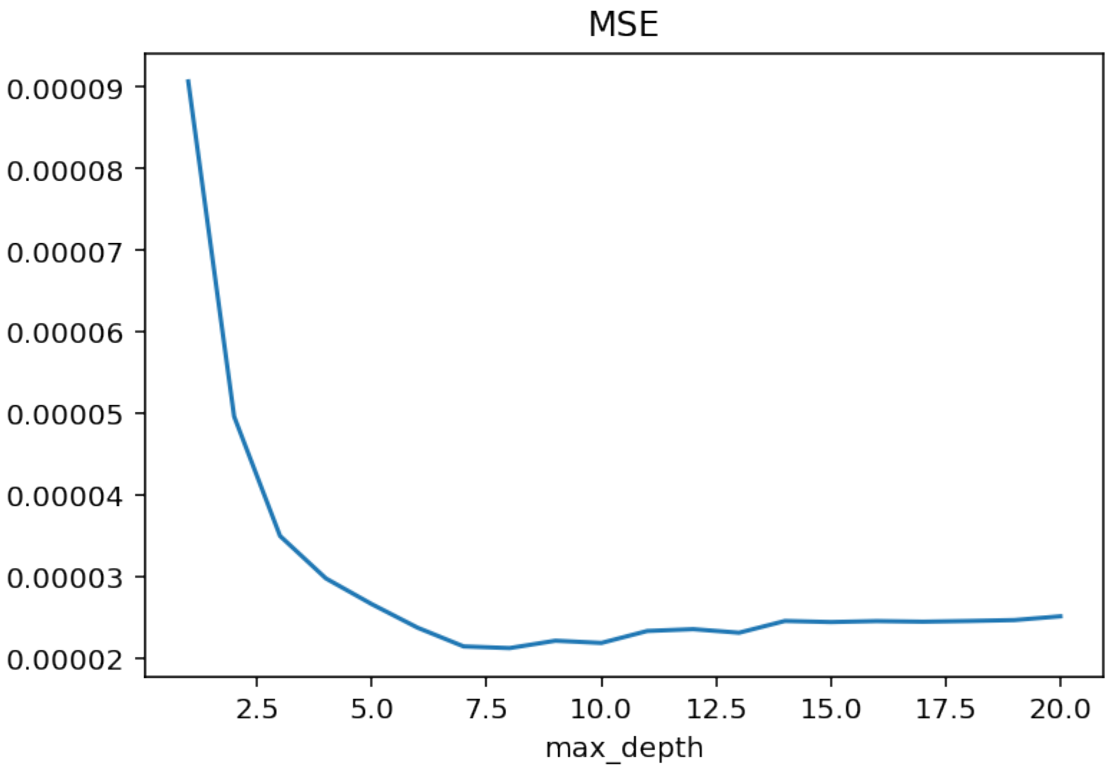
```
'wind_speed', 'sin(wd)', 'cos(wd)',  【t期】
'wind_speed-1', 'sin(wd)-1','cos(wd)-1', 'wind_power-1' 【t-1期】
```

- **输出：** wind_power

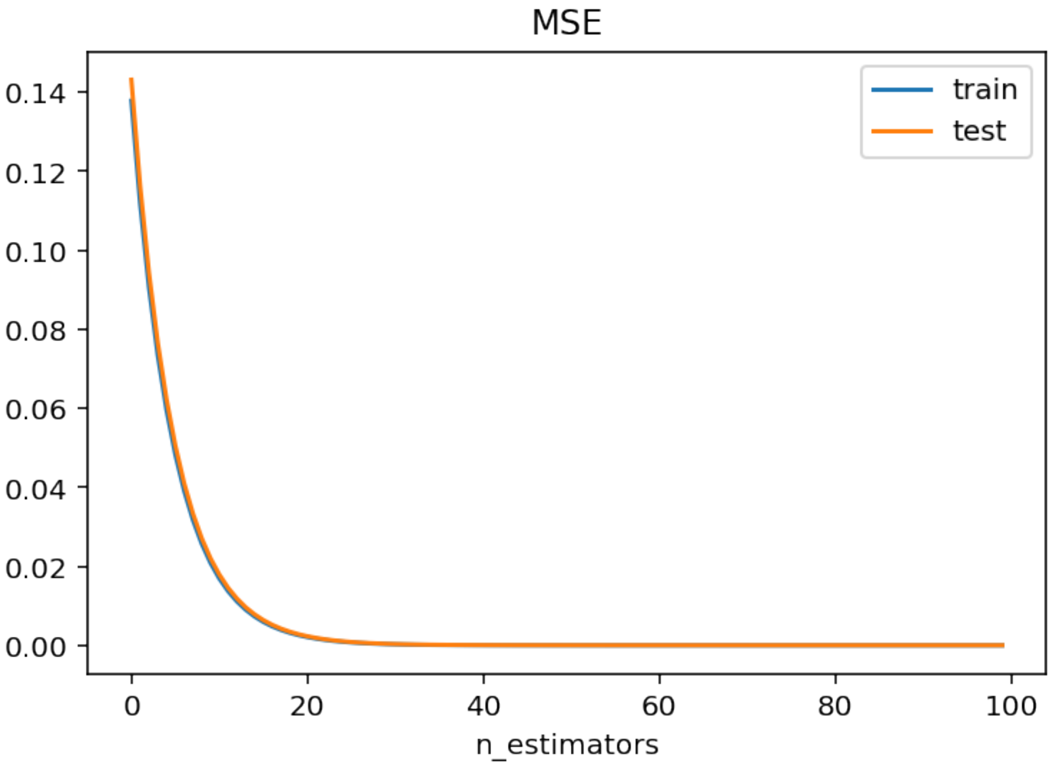
2.1 寻找最大深度

max depth = 7

最终设置：



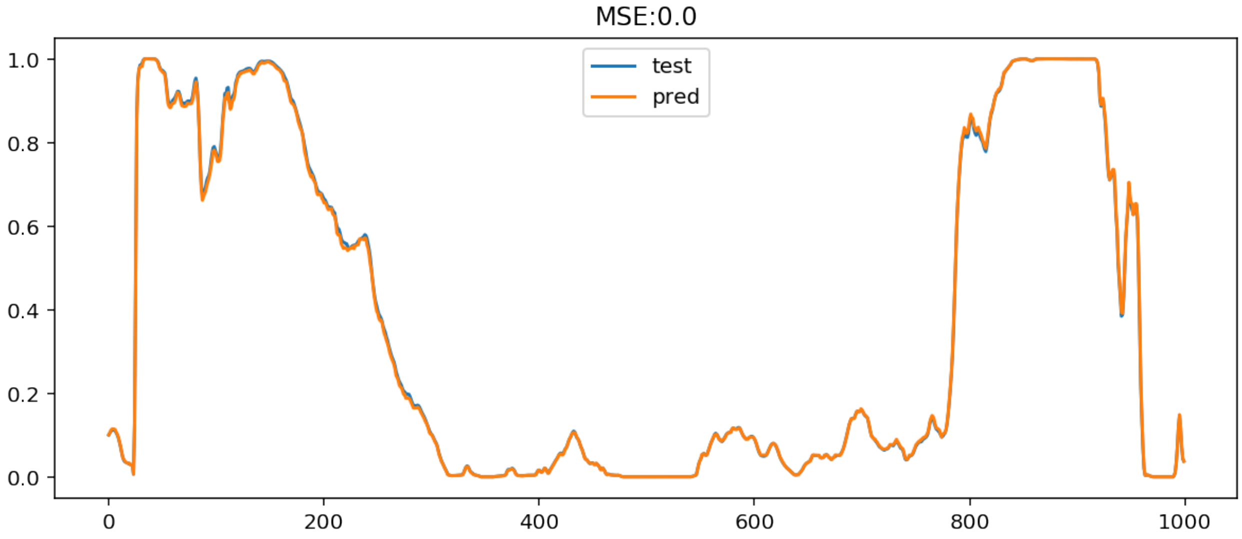
2.2 n_estimators



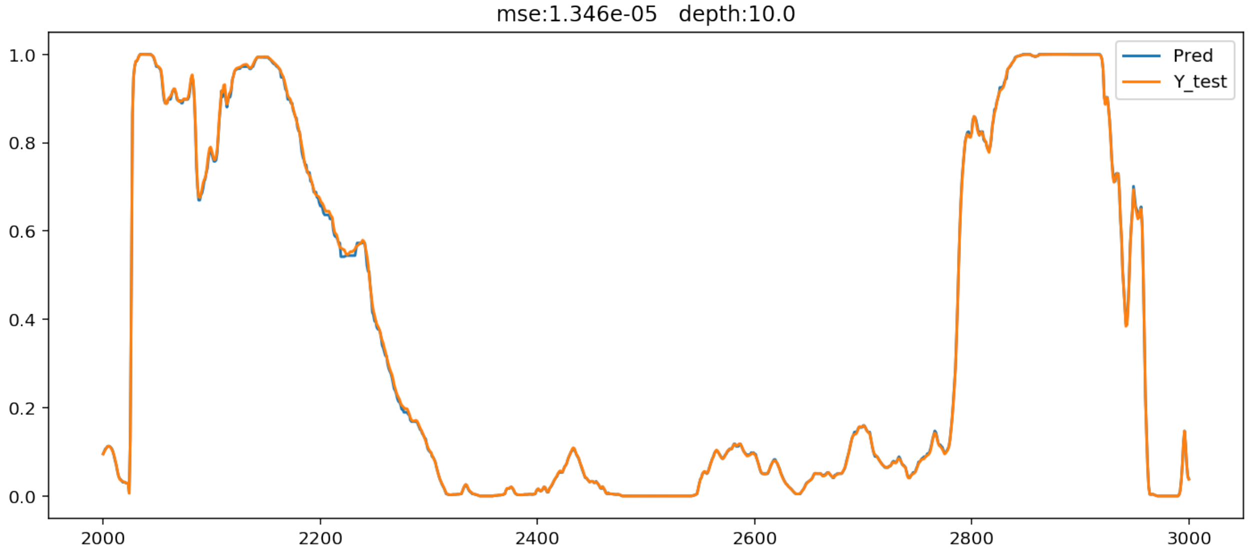
最终设置：

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0,
             importance_type='gain', learning_rate=0.1, max_delta_step=0,
             max_depth=7, min_child_weight=1, missing=None, n_estimators=100,
             n_jobs=4, nthread=None, objective='reg:linear', random_state=0,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
             silent=None, subsample=1, verbosity=1)
```

test mse : 2.1475904432718094e-05



与单纯决策树对比：


mse:1.346e-05   depth:10.0
Pred
Y_test