# bayesian_decision_tree

**bayesian_decision_tree 参数设置：**

```
mu = Y_train.mean()
sd_prior = Y_train.std() / 10
prior_pseudo_observations = 10
kappa = prior_pseudo_observations
alpha = prior_pseudo_observations / 2
var_prior = sd_prior**2
tau_prior = 1/var_prior
beta = alpha/tau_prior
prior = np.array([mu, kappa, alpha, beta])

# model
model = HyperplaneRegressionTree(
        partition_prior=0.9,
        prior=prior,
        delta=0,
        optimizer=SimulatedAnnealingOptimizer(10, 10, 0.9, 666))
```

**box-cox 变换：**

设 $wp \sim N(\mu, \sigma^2)$

则 $wp_{ln} = ln(wp + 0.01)$

$wp_{pred} = exp(f(X, wp_{ln})) - 0.01$

## 1. 西班牙数据集

train index: [6426, 10427] train_len: 4000

test index: [14389, 15390] test_len: 1000

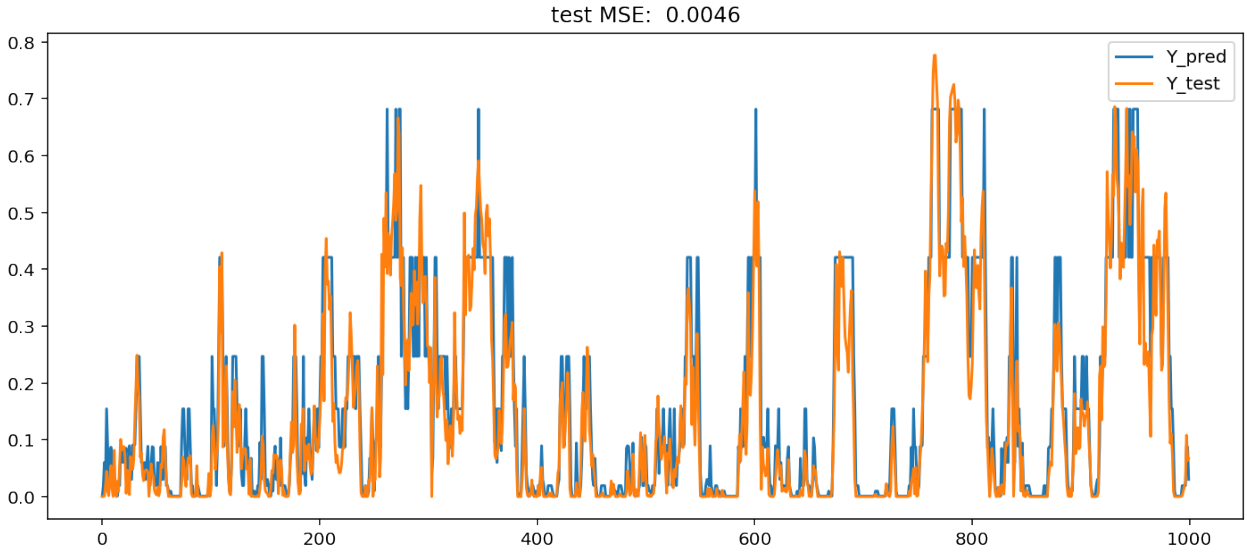
- **输入特征：**

```
'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

结果：

```
Tree depth and number of leaves: 14, 70
Feature importance: [0.54675919 0.00501782 0.00825381 0.04094665 0.01131947
0.0094344
 0.37826866]
 test mse: 0.0046
```




# 2. 美国数据集

train index: [3001, 7002] train_len: 4000

test index: [2000, 3001] test_len: 1000

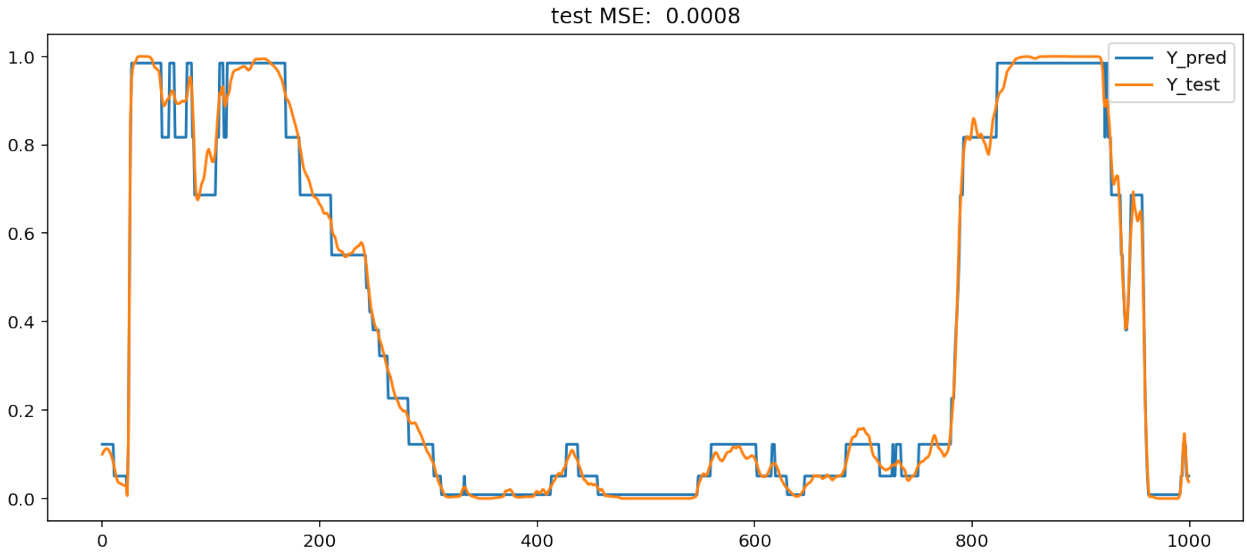
- **输入特征：**

```
'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

结果：

```
Tree depth and number of leaves: 5, 12
Feature importance: [9.97125044e-01 1.23482365e-04 1.73856521e-06
1.90291324e-05
 5.03447188e-05 9.01217196e-05 2.59023945e-03]
 test MSE:  0.0008
```


test MSE:  0.0008
Y_pred
Y_test
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000