

# RandomForestRegressor\_geyser

July 12, 2022

```
[1]: import pandas as pd

df = pd.read_csv('geyser.csv')
df.head()
```

```
[1]: Unnamed: 0  waiting  duration
0           1       80  4.016667
1           2       71  2.150000
2           3       57  4.000000
3           4       80  4.000000
4           5       75  4.000000
```

```
[2]: from sklearn.ensemble import RandomForestRegressor
```

```
[3]: from sklearn.model_selection import train_test_split
```

```
[4]: data = df[['waiting', 'duration']]
```

```
[5]: data.head()
```

```
[5]: waiting  duration
0       80  4.016667
1       71  2.150000
2       57  4.000000
3       80  4.000000
4       75  4.000000
```

```
[27]: x_train,x_test,y_train,y_test = \
      ↪ train_test_split(data['waiting'],data['duration'],shuffle=False)
```

```
[28]: x_train.head()
```

```
[28]: 0    80
      1    71
      2    57
      3    80
      4    75
```

Name: waiting, dtype: int64

```
[29]: x_train.describe()
```

```
[29]: count      224.000000
      mean       72.084821
      std       13.783624
      min       43.000000
      25%       59.000000
      50%       76.000000
      75%       82.000000
      max      108.000000
      Name: waiting, dtype: float64
```

```
[30]: y_train.describe()
```

```
[30]: count      224.000000
      mean       3.446280
      std       1.161000
      min       0.833333
      25%       2.000000
      50%       4.000000
      75%       4.383333
      max       5.450000
      Name: duration, dtype: float64
```

```
[31]: x_train.values.reshape(-1,1)
```

```
[31]: array([[ 80],
          [ 71],
          [ 57],
          [ 80],
          [ 75],
          [ 77],
          [ 60],
          [ 86],
          [ 77],
          [ 56],
          [ 81],
          [ 50],
          [ 89],
          [ 54],
          [ 90],
          [ 73],
          [ 60],
          [ 83],
          [ 65],
```

[ 82],  
[ 84],  
[ 54],  
[ 85],  
[ 58],  
[ 79],  
[ 57],  
[ 88],  
[ 68],  
[ 76],  
[ 78],  
[ 74],  
[ 85],  
[ 75],  
[ 65],  
[ 76],  
[ 58],  
[ 91],  
[ 50],  
[ 87],  
[ 48],  
[ 93],  
[ 54],  
[ 86],  
[ 53],  
[ 78],  
[ 52],  
[ 83],  
[ 60],  
[ 87],  
[ 49],  
[ 80],  
[ 60],  
[ 92],  
[ 43],  
[ 89],  
[ 60],  
[ 84],  
[ 69],  
[ 74],  
[ 71],  
[108],  
[ 50],  
[ 77],  
[ 57],  
[ 80],  
[ 61],

[ 82],  
[ 48],  
[ 81],  
[ 73],  
[ 62],  
[ 79],  
[ 54],  
[ 80],  
[ 73],  
[ 81],  
[ 62],  
[ 81],  
[ 71],  
[ 79],  
[ 81],  
[ 74],  
[ 59],  
[ 81],  
[ 66],  
[ 87],  
[ 53],  
[ 80],  
[ 50],  
[ 87],  
[ 51],  
[ 82],  
[ 58],  
[ 81],  
[ 49],  
[ 92],  
[ 50],  
[ 88],  
[ 62],  
[ 93],  
[ 56],  
[ 89],  
[ 51],  
[ 79],  
[ 58],  
[ 82],  
[ 52],  
[ 88],  
[ 52],  
[ 78],  
[ 69],  
[ 75],  
[ 77],

[ 53],  
[ 80],  
[ 55],  
[ 87],  
[ 53],  
[ 85],  
[ 61],  
[ 93],  
[ 54],  
[ 76],  
[ 80],  
[ 81],  
[ 59],  
[ 86],  
[ 78],  
[ 71],  
[ 77],  
[ 76],  
[ 94],  
[ 75],  
[ 50],  
[ 83],  
[ 82],  
[ 72],  
[ 77],  
[ 75],  
[ 65],  
[ 79],  
[ 72],  
[ 78],  
[ 77],  
[ 79],  
[ 75],  
[ 78],  
[ 64],  
[ 80],  
[ 49],  
[ 88],  
[ 54],  
[ 85],  
[ 51],  
[ 96],  
[ 50],  
[ 80],  
[ 78],  
[ 81],  
[ 72],

[ 75],  
[ 78],  
[ 87],  
[ 69],  
[ 55],  
[ 83],  
[ 49],  
[ 82],  
[ 57],  
[ 84],  
[ 57],  
[ 84],  
[ 73],  
[ 78],  
[ 57],  
[ 79],  
[ 57],  
[ 90],  
[ 62],  
[ 87],  
[ 78],  
[ 52],  
[ 98],  
[ 48],  
[ 78],  
[ 79],  
[ 65],  
[ 84],  
[ 50],  
[ 83],  
[ 60],  
[ 80],  
[ 50],  
[ 88],  
[ 50],  
[ 84],  
[ 74],  
[ 76],  
[ 65],  
[ 89],  
[ 49],  
[ 88],  
[ 51],  
[ 78],  
[ 85],  
[ 65],  
[ 75],

```

[ 77],
[ 69],
[ 92],
[ 68],
[ 87],
[ 61],
[ 81],
[ 55],
[ 93],
[ 53],
[ 84],
[ 70],
[ 73],
[ 93],
[ 50],
[ 87],
[ 77]], dtype=int64)

```

```
[32]: y_train
```

```

[32]: 0      4.016667
      1      2.150000
      2      4.000000
      3      4.000000
      4      4.000000
      ...
     219     4.233333
     220     1.933333
     221     4.350000
     222     4.000000
     223     4.000000
      Name: duration, Length: 224, dtype: float64

```

```

[39]: rd = RandomForestRegressor()
      rf=rd.fit(x_train.values.reshape(-1,1),y_train)
      y_pred = rd.predict(x_test.values.reshape(-1,1))

```

```
[40]: y_pred
```

```

[40]: array([3.25686119, 4.13446786, 2.7378022 , 3.25686119, 2.63463245,
            4.51716151, 2.48789008, 4.38108222, 3.40460879, 4.51716151,
            3.11900726, 1.82423088, 2.62922646, 3.35080132, 4.02295278,
            2.63463245, 1.82423088, 4.28236665, 1.87870186, 4.13446786,
            3.44423385, 4.45270925, 3.11900726, 3.25686119, 4.60966665,
            3.25976 , 4.21472753, 2.62922646, 4.13446786, 2.85671156,
            4.20747725, 2.85671156, 4.21472753, 2.62922646, 4.32862501,
            3.40460879, 3.35254943, 3.52425404, 4.38108222, 2.7378022 ,

```

```
1.87870186, 2.97066578, 4.45270925, 2.42738888, 4.20747725,
1.82423088, 4.6321848 , 2.85671156, 3.35080132, 4.63179592,
2.3084695 , 4.57255896, 2.42692065, 3.25976 , 3.25976 ,
2.86054667, 3.25976 , 3.35080132, 3.25686119, 2.86054667,
4.22379021, 2.85671156, 4.20747725, 1.87870186, 4.20747725,
2.62922646, 4.63417759, 3.25976 , 4.45270925, 2.62922646,
4.31092302, 2.42692065, 4.15904516, 1.98762105, 3.11900726])
```

```
[41]: y_test
```

```
[41]: 224    4.000000
      225    4.216667
      226    4.000000
      227    4.133333
      228    1.883333
      ...
      294    4.083333
      295    2.066667
      296    4.000000
      297    4.000000
      298    2.000000
      Name: duration, Length: 75, dtype: float64
```

```
[46]: rf.score(x_test.values.reshape(-1,1),y_test)
```

```
[46]: 0.23961659983689
```

```
[48]: from sklearn.model_selection import GridSearchCV
```

```
params = { 'n_estimators' : [10, 100],
           'max_depth' : [6, 8, 10, 12],
           'min_samples_leaf' : [8, 12, 18],
           'min_samples_split' : [8, 16, 20]
         }
grid_cv = GridSearchCV(rf, param_grid = params, cv = 3, n_jobs = -1)
grid_cv.fit(x_train.values.reshape(-1,1), y_train)

print('      : ', grid_cv.best_params_)
print('      : {:.4f}'.format(grid_cv.best_score_))
```

```
      : {'max_depth': 8, 'min_samples_leaf': 18, 'min_samples_split': 8,
'n_estimators': 10}
      : 0.4367
```

```
[50]: rd_r = RandomForestRegressor(n_estimators = 10, max_depth = 8, min_samples_leaf=
      ↪= 18, min_samples_split = 8)
      rd_rf=rd_r.fit(x_train.values.reshape(-1,1),y_train)
```



```
y_pred = rd_rf.predict(x_test.values.reshape(-1,1))  
rd_rf.score(x_test.values.reshape(-1,1),y_test)
```

[50]: 0.2591348145542661