temperature-prediction-with-rf

December 12, 2023

[1]:
```
import pandas as pd
import numpy as np
from sklearn.tree import export_graphviz
import pydot
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:
```
data=pd.read_csv(r"F:\Documents\temps.csv")
```

[3]:
```
data
```

[3]:

	year	month	day	week	temp_2	temp_1	average	actual	friend
0	2019	1	1	Fri	45	45	45.6	45	29
1	2019	1	2	Sat	44	45	45.7	44	61
2	2019	1	3	Sun	45	44	45.8	41	56
3	2019	1	4	Mon	44	41	45.9	40	53
4	2019	1	5	Tues	41	40	46.0	44	41
..	…	…	…	…	…	…	…	…	…
343	2019	12	27	Tues	42	42	45.2	47	47
344	2019	12	28	Wed	42	47	45.3	48	58
345	2019	12	29	Thurs	47	48	45.3	48	65
346	2019	12	30	Fri	48	48	45.4	57	42
347	2019	12	31	Sat	48	57	45.5	40	57

[348 rows x 9 columns]

[4]:
```
data.columns
```

[4]:
```
Index(['year', 'month', 'day', 'week', 'temp_2', 'temp_1', 'average', 'actual',
       'friend'],
      dtype='object')
```

[5]:
```
data.isnull().sum()
```

[5]:
```
year       0
month      0
day        0
```

```
week       0
temp_2     0
temp_1     0
average    0
actual     0
friend     0
dtype: int64
```

[6]:
```
data=pd.get_dummies(data)
data
```

[6]:
```
     year  month  day  temp_2  temp_1  average  actual  friend  week_Fri  \
0    2019      1    1      45      45     45.6      45      29      True
1    2019      1    2      44      45     45.7      44      61     False
2    2019      1    3      45      44     45.8      41      56     False
3    2019      1    4      44      41     45.9      40      53     False
4    2019      1    5      41      40     46.0      44      41     False
..    …    …  …      …       …       …      …       …       …
343  2019     12   27      42      42     45.2      47      47     False
344  2019     12   28      42      47     45.3      48      58     False
345  2019     12   29      47      48     45.3      48      65     False
346  2019     12   30      48      48     45.4      57      42      True
347  2019     12   31      48      57     45.5      40      57     False

     week_Mon  week_Sat  week_Sun  week_Thurs  week_Tues  week_Wed
0       False     False     False       False      False     False
1       False      True     False       False      False     False
2       False     False      True       False      False     False
3        True     False     False       False      False     False
4       False     False     False       False       True     False
..        …       …       …          …          …       …
343     False     False     False       False       True     False
344     False     False     False       False      False      True
345     False     False     False        True      False     False
346     False     False     False       False      False     False
347     False      True     False       False      False     False

[348 rows x 15 columns]
```

[7]:
```
data.shape
```

[7]:
```
(348, 15)
```

[8]:
```
label=data['actual']
```

[9]:
```
data=data.drop('actual',axis=1)
```

[10]:
```
data_list=list(data.columns)
```

[11]:
```
data.columns
```

[11]:
```
Index(['year', 'month', 'day', 'temp_2', 'temp_1', 'average', 'friend',
       'week_Fri', 'week_Mon', 'week_Sat', 'week_Sun', 'week_Thurs',
       'week_Tues', 'week_Wed'],
      dtype='object')
```

[12]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(data,label,test_size=0.
 ↪33,random_state=42)
```

[13]:
```
from sklearn.ensemble import RandomForestRegressor
rf=RandomForestRegressor(n_estimators=1000,random_state=42)
rf.fit(x_train,y_train)
```

[13]:
```
RandomForestRegressor(n_estimators=1000, random_state=42)
```

[14]:
```
pred=rf.predict(x_test)
errors=abs(pred-y_test)
```

[15]:
```
pred
```

[15]:
```
array([68.545, 60.812, 51.336, 60.733, 66.216, 69.894, 79.749, 79.169,
       62.073, 73.072, 63.163, 73.261, 40.225, 62.642, 71.352, 55.795,
       61.401, 56.721, 56.827, 75.568, 64.805, 54.469, 64.971, 62.757,
       59.601, 53.137, 66.689, 46.708, 61.735, 77.37 , 73.309, 62.753,
       55.588, 78.842, 73.672, 61.128, 53.847, 50.836, 67.702, 43.932,
       67.59 , 57.301, 75.485, 42.801, 60.968, 72.251, 53.505, 77.142,
       54.334, 42.414, 46.126, 42.789, 65.168, 65.539, 74.987, 60.998,
       55.398, 59.761, 53.31 , 60.674, 66.107, 51.367, 59.999, 69.015,
       59.588, 59.224, 72.203, 69.325, 74.584, 40.396, 76.272, 57.386,
       59.953, 50.53 , 54.388, 65.067, 43.946, 77.604, 48.043, 52.814,
       53.287, 67.22 , 72.373, 72.887, 63.28 , 59.186, 46.988, 68.722,
       60.186, 83.93 , 64.952, 50.685, 52.764, 53.781, 76.064, 39.237,
       41.641, 44.913, 74.354, 74.425, 40.002, 75.322, 70.617, 52.645,
       75.335, 58.243, 41.494, 51.707, 57.755, 65.537, 63.392, 49.99 ,
       42.078, 65.45 , 60.261])
```

[16]:
```
print('Mean Absolute Error:', round(np.mean(errors),2),'degress. ')
```

```
Mean Absolute Error: 3.87 degress.
```

[17]:
```
mape= 100*(errors/y_test)
accuracy = 100-np.mean(mape)
print('Accuracy:', round(accuracy,2),'%')
```

```
Accuracy: 93.95 %
```

[18]:
```
tree=rf.estimators_[5]
export_graphviz(tree, out_file='tree.
  ↪dot',feature_names=data_list,rounded=True,precision=1)
(graph, )=pydot.graph_from_dot_file('tree.dot')
graph.write_png('tree.png');
```

[19]:
```
print('depth of tree is',tree.tree_.max_depth)
```

```
depth of tree is 13
```

[ ]:
```

```