pythonpractice

February 7, 2019

1 #Notebook for decision tree---Tiancheng

In [1]:
```
import pandas as pd
import os
os.chdir("C:/Users/Arche/graduate/R working directory/5505/project")# Set my working d
os.getcwd()

prac=pd.read_csv("prac.csv")#Read data
prac=prac.dropna(axis=0)
prac.head()
```

Out[1]:

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
10	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S

In [2]:
```
from sklearn import tree
model=tree.DecisionTreeClassifier()
input= prac.drop(["Survived","Name","Ticket","SibSp","Cabin","Embarked"],axis=1) # Get
target=prac["Survived"] # Get the result data(the value we want to perdict)
input.head()
```

Out[2]:

	Pclass	Sex	Age	Parch	Fare
1	1	female	38.0	0	71.2833
3	1	female	35.0	0	53.1000
6	1	male	54.0	0	51.8625
10	3	female	4.0	1	16.7000
11	1	female	58.0	0	26.5500

```
In [3]: from sklearn.preprocessing import LabelEncoder
        le=LabelEncoder()
        input["sex_new"]=le.fit_transform(prac["Sex"])# Transfor our label from string type to
        input_n=input.drop(["Sex"],axis=1)
        input_n.head()
```

Out[3]:

	Pclass	Age	Parch	Fare	sex_new
1	1	38.0	0	71.2833	0
3	1	35.0	0	53.1000	0
6	1	54.0	0	51.8625	1
10	3	4.0	1	16.7000	0
11	1	58.0	0	26.5500	0

```
In [4]: model.fit(input_n,target) # fit the model
```

```
Out[4]: DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                    max_features=None, max_leaf_nodes=None,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                    splitter='best')
```

```
In [5]: model.score(input_n,target)
```

```
Out[5]: 1.0
```

```
In [6]: model.predict([[3,20,0,50,1]])# useing the model for perdict
```

```
Out[6]: array([0], dtype=int64)
```