random-forest

December 19, 2023

0.1 This is a project of predicting quality of wine which is depended on various factors such as acidity, density, pH, alcohol etc. using Random forest classification

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df = pd.read_csv('redwine.csv')
df.head()
```

[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5

[3]:
```
df.isnull().sum()
```

[3]:
```
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
```

```
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

0.1.1 Model fitting

[5]:
```
from sklearn.ensemble import RandomForestClassifier
```

[6]:
```
inputs = df.drop(['quality'],axis=1)
```

[8]:
```
targets = df['quality']
```

[9]:
```
model = RandomForestClassifier()
```

[10]:
```
model.fit(inputs,targets)
```

[10]:
```
RandomForestClassifier()
```

[11]:
```
model.score(inputs,targets)
```

[11]:
```
1.0
```

[13]:
```
from sklearn.model_selection import train_test_split
```

[14]:
```
xtrain,xtest,ytrain,ytest = train_test_split(inputs,targets,test_size = 0.2)
```

[15]:
```
model.fit(xtrain,ytrain)
```

[15]:
```
RandomForestClassifier()
```

[20]:
```
model.score(xtest,ytest)
```

[20]:
```
0.671875
```

[18]:
```
model.predict (xtest)
```

[18]:
```
array([5, 6, 5, 6, 4, 6, 7, 5, 6, 5, 6, 6, 5, 5, 5, 6, 7, 6, 5, 5, 5, 6,
       6, 5, 5, 5, 5, 6, 6, 6, 5, 7, 6, 5, 5, 6, 5, 6, 6, 5, 6, 6, 5, 5,
       5, 7, 6, 7, 5, 6, 5, 6, 6, 5, 5, 5, 6, 7, 6, 7, 5, 5, 6, 5, 5, 6,
       5, 5, 6, 6, 5, 6, 7, 6, 5, 6, 6, 5, 6, 7, 5, 5, 6, 5, 6, 5, 5, 6,
       6, 7, 6, 6, 7, 5, 5, 6, 5, 5, 5, 6, 7, 6, 5, 6, 5, 5, 5, 6, 6, 5,
       5, 6, 6, 6, 5, 6, 5, 7, 5, 6, 5, 5, 5, 6, 6, 5, 5, 6, 5, 6, 5, 5,
```

```
6, 5, 6, 6, 5, 6, 6, 6, 5, 5, 6, 5, 5, 5, 7, 6, 7, 6, 5, 5, 5, 6,
7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 6, 7, 5, 6, 7, 6, 5,
6, 5, 5, 6, 5, 5, 6, 6, 6, 5, 5, 6, 5, 6, 5, 7, 6, 6, 5, 7, 6, 5,
6, 6, 6, 5, 6, 7, 5, 5, 6, 6, 5, 6, 6, 7, 5, 6, 5, 5, 5, 5, 5, 6,
6, 5, 5, 5, 5, 7, 6, 6, 5, 5, 6, 5, 5, 5, 7, 5, 5, 5, 5, 5, 5, 6,
5, 5, 6, 5, 5, 5, 6, 6, 6, 5, 7, 5, 7, 5, 5, 5, 5, 5, 6, 5, 6, 5,
7, 7, 6, 6, 5, 6, 5, 5, 5, 6, 5, 6, 6, 6, 5, 6, 6, 7, 7, 5, 7, 6,
7, 6, 6, 5, 7, 6, 5, 6, 6, 5, 5, 5, 7, 5, 5, 5, 5, 5, 6, 6, 5, 6,
5, 6, 5, 5, 7, 5, 7, 5, 5, 6, 5, 6], dtype=int64)
```

[19]:
```
model.predict(inputs)
```

[19]:
```
array([5, 5, 5, …, 6, 5, 6], dtype=int64)
```

0.1.2 Here we got score 0.671875 by splitting the data set into testing and training which is low compared to the score without splitting with value 1. So we use data without splitting and model is fitted. Using the fitted model we can predict the quality of wine with the desired independent facors related to wine quality

[ ]:
```

```