03.Wine

December 21, 2021

1 Wine

1)

[ ]:
```
from sklearn.datasets import load_wine
wine = load_wine()
```

[ ]:
```
import pandas as pd
df = pd.DataFrame(wine.data, columns = wine.feature_names)
df["target"] = wine.target
df.head()
```

[ ]:

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0

[ ]:
```
df.shape
```

[ ]:
```
(178, 14)
```

[ ]:
```
df.target.value_counts()
```

[ ]:
```
1    71
0    59
2    48
Name: target, dtype: int64
```

[ ]:
```
wine.target_names
```

[ ]:
```
array(['class_0', 'class_1', 'class_2'], dtype='<U7')
```

2) /

[ ]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
```

[ ]:
```
X_wine, X_test, y_wine, y_test = train_test_split(
    wine.data, wine.target, stratify = wine.target,test_size = 0.2,␣
 ↪random_state = 2021
)
X_wine.shape, X_test.shape, y_wine.shape, y_test.shape
```

[ ]:
```
((142, 13), (36, 13), (142,), (36,))
```

[ ]:
```
import numpy as np
```

[ ]:
```
X_wine, X_test, y_wine, y_test = train_test_split(
    wine.data, wine.target, stratify = wine.target, test_size = 0.2,␣
 ↪random_state = 2021
)
np.unique(y_wine,return_counts = True)
```

[ ]:
```
(array([0, 1, 2]), array([47, 57, 38]))
```

3) GridSearchCV

- DecisionTreeClassifier SVC
-
-

[ ]:
```
import warnings
warnings.filterwarnings("ignore")
```

[ ]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV
```

[ ]:
```
dtc = DecisionTreeClassifier(random_state = 2021)
params = {
    'max_depth' : [2, 3, 4, 5, 6],
```

```
    "min_samples_split" : [2, 3, 4]
}
```

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

grid_dt = GridSearchCV(
    dtc, param_grid = params, scoring = "accuracy", cv = 5
)
```

[ ]:
```
grid_dt.fit(X_wine, y_wine)
```

[ ]:
```
GridSearchCV(cv=5, estimator=DecisionTreeClassifier(random_state=2021),
             param_grid={'max_depth': [2, 3, 4, 5, 6],
                         'min_samples_split': [2, 3, 4]},
             scoring='accuracy')
```

[ ]:
```
grid_dt.best_params_
```

[ ]:
```
{'max_depth': 3, 'min_samples_split': 2}
```

[ ]:
```
best_dt = grid_dt.best_estimator_
best_dt.score(X_test, y_test)
```

[ ]:
```
0.9722222222222222
```

- Support Vector Machine

[ ]:
```
from sklearn.svm import SVC
svc = SVC(random_state = 2021)
params = {
    "C" : [0.01, 0.1, 1, 10, 100]
}
```

[ ]:
```
grid_sv = GridSearchCV(svc, param_grid = params, scoring = "accuracy", cv = 5)
grid_sv.fit(X_wine, y_wine)
grid_sv.best_params_
```

[ ]:
```
{'C': 100}
```

[ ]:
```
params = {'C' : [30, 60, 100, 300, 500]}
grid_sv = GridSearchCV(svc, param_grid = params, scoring = "accuracy", cv = 5)
grid_sv.fit(X_wine, y_wine)
grid_sv.best_params_
```

[ ]:
```
{'C': 500}
```

[ ]:
```
params = {'C' : [400, 500, 700, 1000]}
grid_sv = GridSearchCV(svc, param_grid = params, scoring = "accuracy", cv = 5)
```

```
grid_sv.fit(X_wine, y_wine)
grid_sv.best_params_
```

[ ]: {'C': 1000}

[ ]:
```
params = {'C' : [1000, 5000, 7000, 10000]}
grid_sv = GridSearchCV(svc, param_grid = params, scoring = "accuracy", cv = 5)
grid_sv.fit(X_wine, y_wine)
grid_sv.best_params_
```

[ ]: {'C': 10000}

[ ]:
```
best_sv = grid_sv.best_estimator_
best_sv.score(X_test, y_test)
```

[ ]: 1.0