# 02.ModelSelection

December 21, 2021

## 1 Model Selection

- /

[ ]:
```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
```

[ ]:
```
iris = load_iris()
dtc = DecisionTreeClassifier(random_state = 2021)
dtc.fit(iris.data, iris.target)
dtc.score(iris.data, iris.target)
```

[ ]:
```
1.0
```

- cross_validate()

[ ]:
```
from sklearn.model_selection import cross_validate
dtc = DecisionTreeClassifier()
cross_validate(dtc, iris.data, iris.target)
```

[ ]:
```
{'fit_time': array([0.00130796, 0.0004518 , 0.00041485, 0.00044489,
0.00047302]),
 'score_time': array([0.0002718 , 0.00022411, 0.00021887, 0.00038028,
0.00024414]),
 'test_score': array([0.96666667, 0.96666667, 0.9       , 0.96666667, 1.
])}
```

- cross_val_scroe()

[ ]:
```
from sklearn.model_selection import cross_val_score
cross_val_score(dtc, iris.data, iris.target, scoring = "accuracy", cv = 5)
```

[ ]:
```
array([0.96666667, 0.96666667, 0.9       , 0.93333333, 1.        ])
```

[ ]:
```
cross_val_score(dtc, iris.data, iris.target, scoring = "accuracy", cv = 3)
```

[ ]:
```
array([0.98, 0.92, 0.96])
```

- GridSearchCV : + .

[ ]:

```
# Train/Test dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =  train_test_split(
    iris.data, iris.target, stratify = iris.target,test_size = 0.2,␣
 ↪random_state = 2021
)
```

[ ]:

```
#
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(random_state = 2021)
dtc.get_params()
```

[ ]:

```
{'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': None,
 'max_leaf_nodes': None,
 'min_impurity_decrease': 0.0,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'random_state': 2021,
 'splitter': 'best'}
```

[ ]:

```
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_train, y_train)
```

[ ]:

```
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': 4, 'min_samples_split': 2}
```

[ ]:
```
#
best_clf = grid_dt.best_estimator_
```

[ ]:
```
#
best_clf.score(X_test, y_test)
```

[ ]:
```
0.9
```

- Support Vector Machine

[ ]:
```
from sklearn.svm import SVC
svc = SVC(random_state = 2021)
svc.get_params()
```

[ ]:
```
{'C': 1.0,
 'break_ties': False,
 'cache_size': 200,
 'class_weight': None,
 'coef0': 0.0,
 'decision_function_shape': 'ovr',
 'degree': 3,
 'gamma': 'scale',
 'kernel': 'rbf',
 'max_iter': -1,
 'probability': False,
 'random_state': 2021,
 'shrinking': True,
 'tol': 0.001,
 'verbose': False}
```

[ ]:
```
params = {'C' : [0.01, 0.1, 1, 10, 100]}
```

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

[ ]:
```
GridSearchCV(cv=5, estimator=SVC(random_state=2021),
             param_grid={'C': [0.01, 0.1, 1, 10, 100]}, scoring='accuracy')
```

[ ]:
```
grid_sv.best_params_
```

[ ]:
```
{'C': 10}
```

[ ]:
```
params = {'C' : [3, 6, 10, 15, 20]}
grid_sv = GridSearchCV(svc, param_grid = params, scoring = "accuracy", cv = 5)
grid_sv.fit(X_train, y_train)
grid_sv.best_params_
```

[ ]: {'C': 3}

[ ]:
```
params = {'C' : [2, 3, 4, 5]}
grid_sv = GridSearchCV(svc, param_grid = params, scoring = "accuracy", cv = 5)
grid_sv.fit(X_train, y_train)
grid_sv.best_params_
```

[ ]: {'C': 4}

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

[ ]: 1.0