

T5 - Grid Search

March 6, 2021

1 Grid search

1.1 Setting up

```
[ ]: import pandas as pd
import numpy as np
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# Load data
dataObj = load_breast_cancer()
X = dataObj.data
y = dataObj.target

# Splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    stratify=y,
                                                    test_size=0.20,
                                                    random_state=1)
```

```
[ ]: from sklearn.svm import SVC

pipe_svc = Pipeline([('scl', StandardScaler()),
                      ('clf', SVC(random_state=1))])
```

```
[ ]: # Get parameter names
for k, v in pipe_svc.get_params().items():
    print(f"{k:35.35s}: {str(v):35.35s}...")
```

```
[ ]: param_range = [0.0001, 0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0]

set1 = {'clf__C': param_range,
        'clf__kernel': ['linear']}

set2 = {'clf__C': param_range,
        'clf__gamma': param_range,
```

```
        'clf__kernel': ['rbf']}]

param_grid = [set1, set2]
```

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

gs = GridSearchCV(estimator=pipe_svc,
                  param_grid=param_grid,
                  scoring='accuracy',
                  cv=10,
                  n_jobs=-1)
```

```
[ ]: gs.fit(X_train, y_train)
```

```
[ ]: # Get parameter names
for k, v in gs.get_params().items():
    print(f"{k:35.35s}: {str(v):35.35}...")
```

```
[ ]: df = pd.DataFrame(gs.cv_results_)
display(df.head())
```

```
[ ]: df = df.sort_values(by=['rank_test_score'])
display(df.head())
```

```
[ ]: print(gs.best_score_)
print(gs.best_params_)
```

```
[ ]: #Note that grid search already refit the entire training data with the best
    ↪ parameters. You can check this from this setting.
gs.refit
```

```
[ ]: y_pred = gs.predict(X_test)
testing_accuracy = gs.score(X_test, y_test)
print(f"Testing accuracy: {testing_accuracy:6.3f}")
```

```
[ ]: # To do this manually
clf = gs.best_estimator_
clf.fit(X_train, y_train)
testing_accuracy = clf.score(X_test, y_test)
print(f"Testing accuracy: {testing_accuracy:6.3f}")
```