ex12

September 18, 2024

[49]:
```
from ucimlrepo import fetch_ucirepo
import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV,␣
 ↪RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
```

[50]:
```
heart_disease = fetch_ucirepo(id=45)

X = heart_disease.data.features
y = heart_disease.data.targets

# print(heart_disease.metadata)
# print(heart_disease.variables)

data = pd.DataFrame(X).join(pd.DataFrame(y))

print(data.isnull().sum())
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
num         0
dtype: int64
```

[51]:
```
data.dropna(inplace=True)

print(data.isnull().sum())
```

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
num         0
dtype: int64
```

[52]:
```
data.head()
```

[52]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	\
0	63	1	1	145	233	1	2	150	0	2.3	3	
1	67	1	4	160	286	0	2	108	1	1.5	2	
2	67	1	4	120	229	0	2	129	1	2.6	2	
3	37	1	3	130	250	0	0	187	0	3.5	3	
4	41	0	2	130	204	0	2	172	0	1.4	1	

	ca	thal	num
0	0.0	6.0	0
1	3.0	3.0	2
2	2.0	7.0	1
3	0.0	3.0	0
4	0.0	3.0	0

[53]:
```
X = data.drop('num', axis=1)
y = data['num']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

[54]:
```
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
rf_accuracy = accuracy_score(y_test, rf_pred)
print(f"Random Forest Accuracy: {rf_accuracy:.2f}")

svc_model = SVC(random_state=42)
svc_model.fit(X_train, y_train)
svc_pred = svc_model.predict(X_test)
svc_accuracy = accuracy_score(y_test, svc_pred)
print(f"SVC Accuracy: {svc_accuracy:.2f}")
```

```
Random Forest Accuracy: 0.60
SVC Accuracy: 0.65
```

[55]:
```
rf_params = {
    'n_estimators': [50, 100, 200],
    'max_depth': [None, 5, 2],
    'min_samples_split': [2, 5, 3]
}

svc_params = {
    'C': [0.1, 1, 10],
    'kernel': ['linear', 'rbf'],
    'gamma': ['scale', 'auto']
}

rf_grid = GridSearchCV(estimator=rf_model, param_grid=rf_params,␣
 ↪scoring='accuracy', cv=5)
rf_grid.fit(X_train, y_train)
print("Best parameters for Random Forest (GridSearchCV):", rf_grid.best_params_)
print("Best accuracy for Random Forest (GridSearchCV):", rf_grid.best_score_)

svc_grid = GridSearchCV(estimator=svc_model, param_grid=svc_params,␣
 ↪scoring='accuracy', cv=5)
svc_grid.fit(X_train, y_train)
print("Best parameters for SVC (GridSearchCV):", svc_grid.best_params_)
print("Best accuracy for SVC (GridSearchCV):", svc_grid.best_score_)
```

```
Best parameters for Random Forest (GridSearchCV): {'max_depth': None,
'min_samples_split': 3, 'n_estimators': 50}
Best accuracy for Random Forest (GridSearchCV): 0.5613475177304965
Best parameters for SVC (GridSearchCV): {'C': 0.1, 'gamma': 'scale', 'kernel':
'linear'}
Best accuracy for SVC (GridSearchCV): 0.5823581560283688
```

[56]:
```
```

```
rf_random = RandomizedSearchCV(estimator=rf_model,␣
 ↪param_distributions=rf_params, n_iter=10, scoring='accuracy', cv=5,␣
 ↪random_state=42)
rf_random.fit(X_train, y_train)
print("Best parameters for Random Forest (RandomizedSearchCV):", rf_random.
 ↪best_params_)
print("Best accuracy for Random Forest (RandomizedSearchCV):", rf_random.
 ↪best_score_)

svc_random = RandomizedSearchCV(estimator=svc_model,␣
 ↪param_distributions=svc_params, n_iter=10, scoring='accuracy', cv=5,␣
 ↪random_state=42)
svc_random.fit(X_train, y_train)
print("Best parameters for SVC (RandomizedSearchCV):", svc_random.best_params_)
print("Best accuracy for SVC (RandomizedSearchCV):", svc_random.best_score_)
```

```
Best parameters for Random Forest (RandomizedSearchCV): {'n_estimators': 50,
'min_samples_split': 5, 'max_depth': 5}
Best accuracy for Random Forest (RandomizedSearchCV): 0.5611702127659575
Best parameters for SVC (RandomizedSearchCV): {'kernel': 'linear', 'gamma':
'scale', 'C': 0.1}
Best accuracy for SVC (RandomizedSearchCV): 0.5823581560283688
```

[57]:
```
print("\nComparison of GridSearchCV and RandomizedSearchCV:")
print("Random Forest - GridSearchCV:", rf_grid.best_params_, "Accuracy:",␣
 ↪rf_grid.best_score_)
print("Random Forest - RandomizedSearchCV:", rf_random.best_params_, "Accuracy:
 ↪", rf_random.best_score_)
print("SVC - GridSearchCV:", svc_grid.best_params_, "Accuracy:", svc_grid.
 ↪best_score_)
print("SVC - RandomizedSearchCV:", svc_random.best_params_, "Accuracy:",␣
 ↪svc_random.best_score_)
```

```
Comparison of GridSearchCV and RandomizedSearchCV:
Random Forest - GridSearchCV: {'max_depth': None, 'min_samples_split': 3,
'n_estimators': 50} Accuracy: 0.5613475177304965
Random Forest - RandomizedSearchCV: {'n_estimators': 50, 'min_samples_split': 5,
'max_depth': 5} Accuracy: 0.5611702127659575
SVC - GridSearchCV: {'C': 0.1, 'gamma': 'scale', 'kernel': 'linear'} Accuracy:
0.5823581560283688
SVC - RandomizedSearchCV: {'kernel': 'linear', 'gamma': 'scale', 'C': 0.1}
Accuracy: 0.5823581560283688
```