PIPELINES

Outline

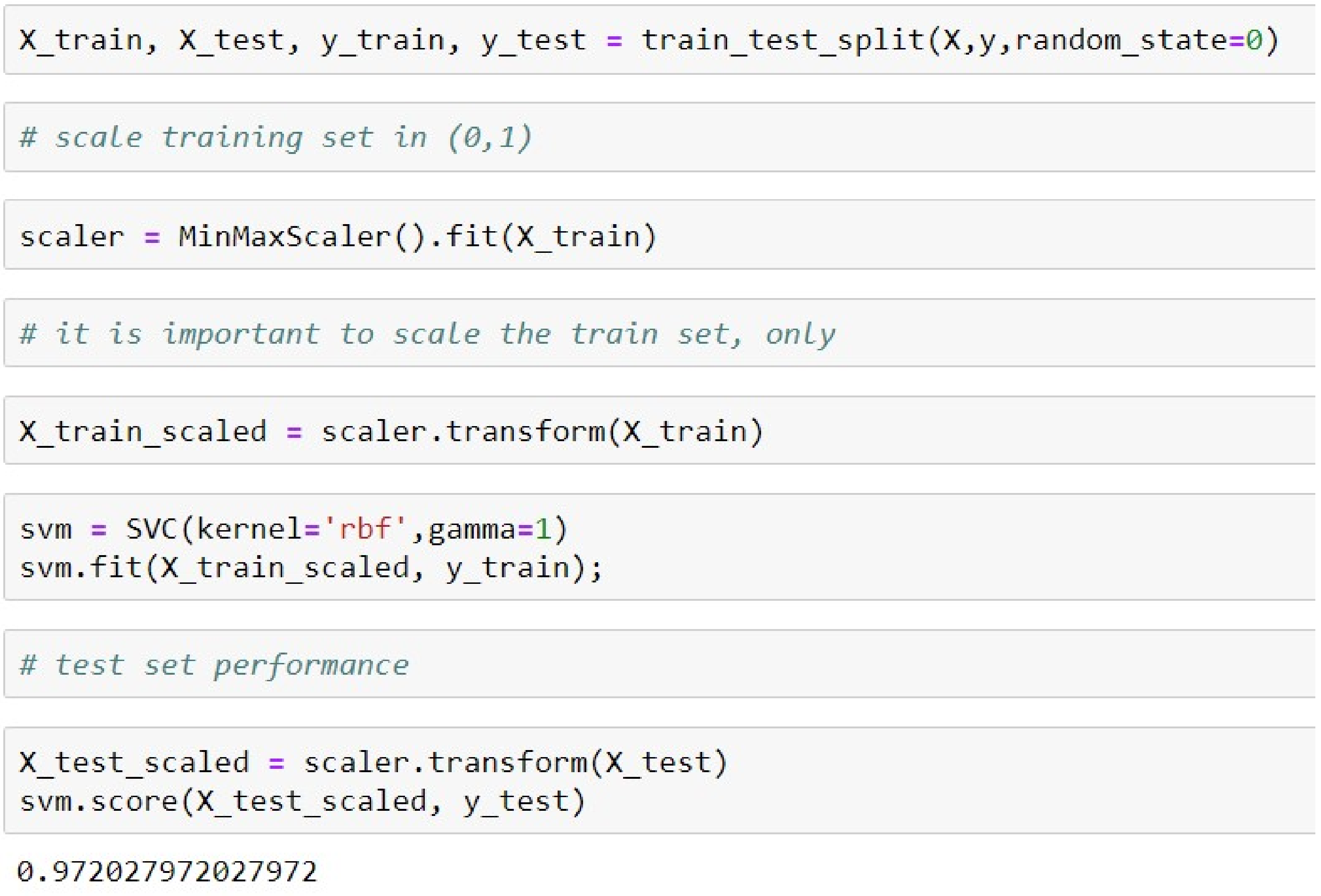
- Introduction
- Pipeline
- Pipeline with cross validation
- Pipeline GridSearchCV
- make_pipeline

Pipeline

A sequence of steps in the same command for

- scaling
- selecting features
- tuning parameters
- building models

Pipeline -Example



```
X_train, X_test, y_train, y_test = train_test_split(X,y,random_state=0)
```

```
# scale training set in (0,1)
```

```
scaler = MinMaxScaler().fit(X_train)
```

```
# it is important to scale the train set, only
```

```
X_train_scaled = scaler.transform(X_train)
```

```
svm = SVC(kernel='rbf',gamma=1)
svm.fit(X_train_scaled, y_train);
```

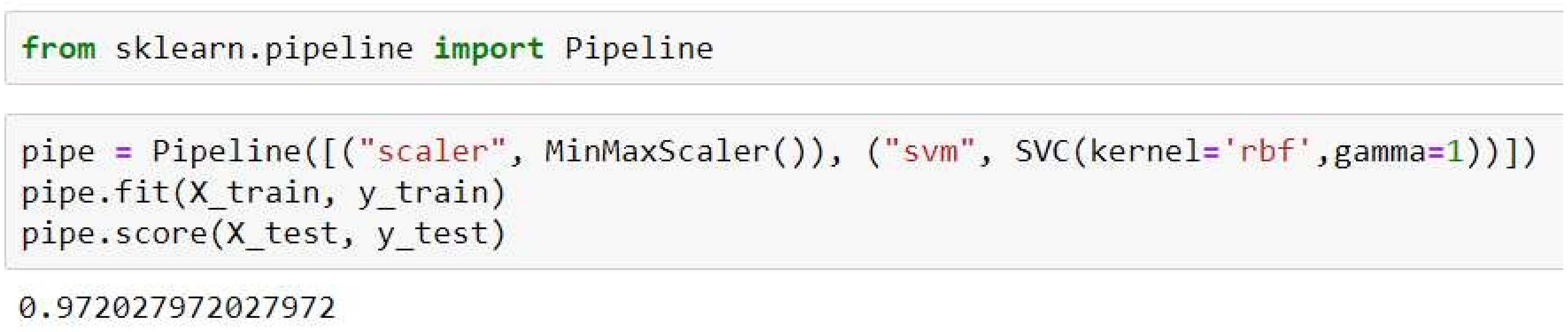
```
# test set performance
```

```
X_test_scaled = scaler.transform(X_test)
svm.score(X_test_scaled, y_test)
```

```
0.972027972027972
```

Pipeline -Example

pipeline



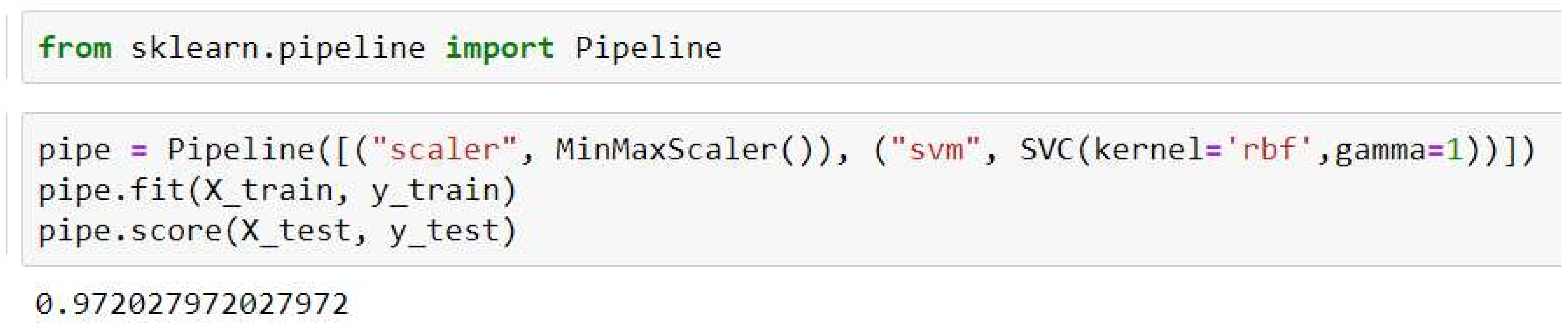
```
from sklearn.pipeline import Pipeline
```

```
pipe = Pipeline([("scaler", MinMaxScaler()), ("svm", SVC(kernel='rbf',gamma=1))])
pipe.fit(X_train, y_train)
pipe.score(X_test, y_test)
```

```
0.972027972027972
```

Pipeline -Example

pipeline



```
from sklearn.pipeline import Pipeline
```

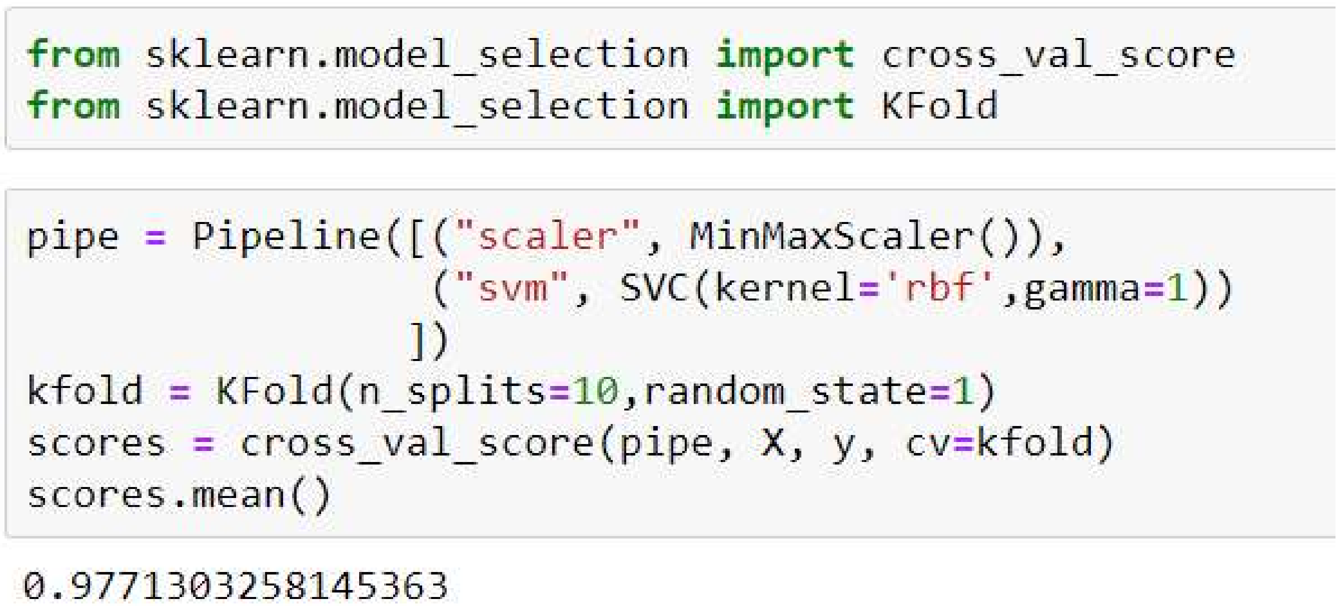
```
pipe = Pipeline([("scaler", MinMaxScaler()), ("svm", SVC(kernel='rbf',gamma=1))])
pipe.fit(X_train, y_train)
pipe.score(X_test, y_test)
```

```
0.972027972027972
```

two steps

- “scaler”
- “svm”

Pipeline with cross validation -Example

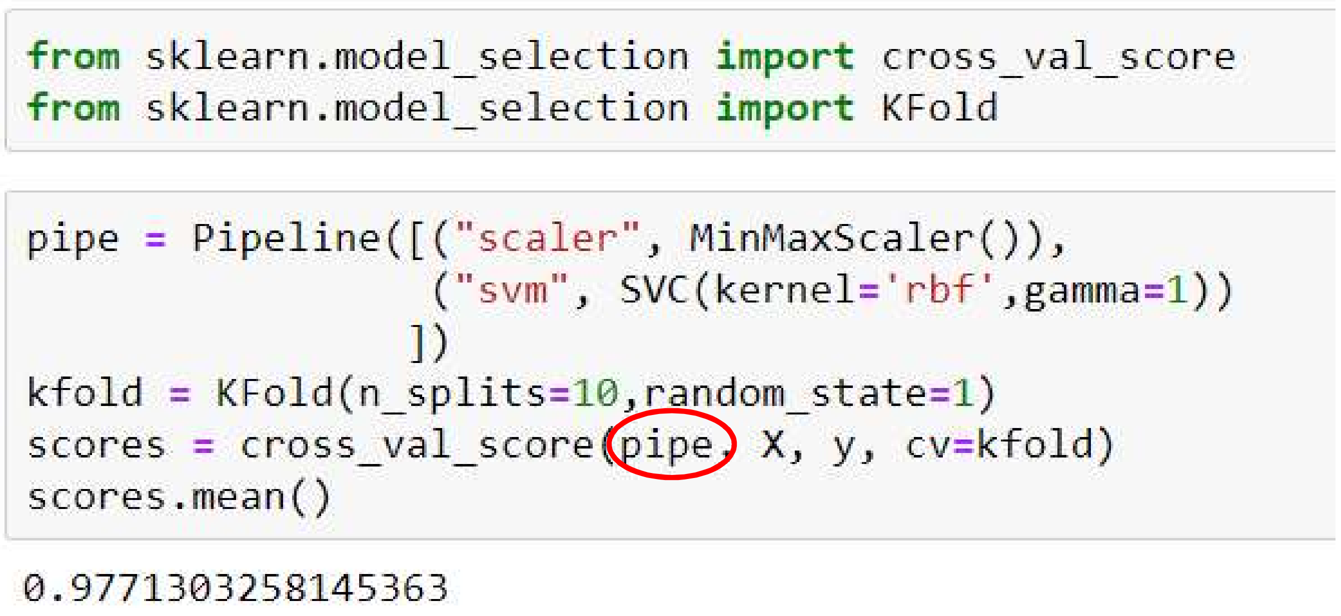


```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
```

```
pipe = Pipeline([("scaler", MinMaxScaler()),
                 ("svm", SVC(kernel='rbf',gamma=1))
                ])
kfold = KFold(n_splits=10,random_state=1)
scores = cross_val_score(pipe, X, y, cv=kfold)
scores.mean()
```

```
0.9771303258145363
```

Pipeline with cross validation -Example



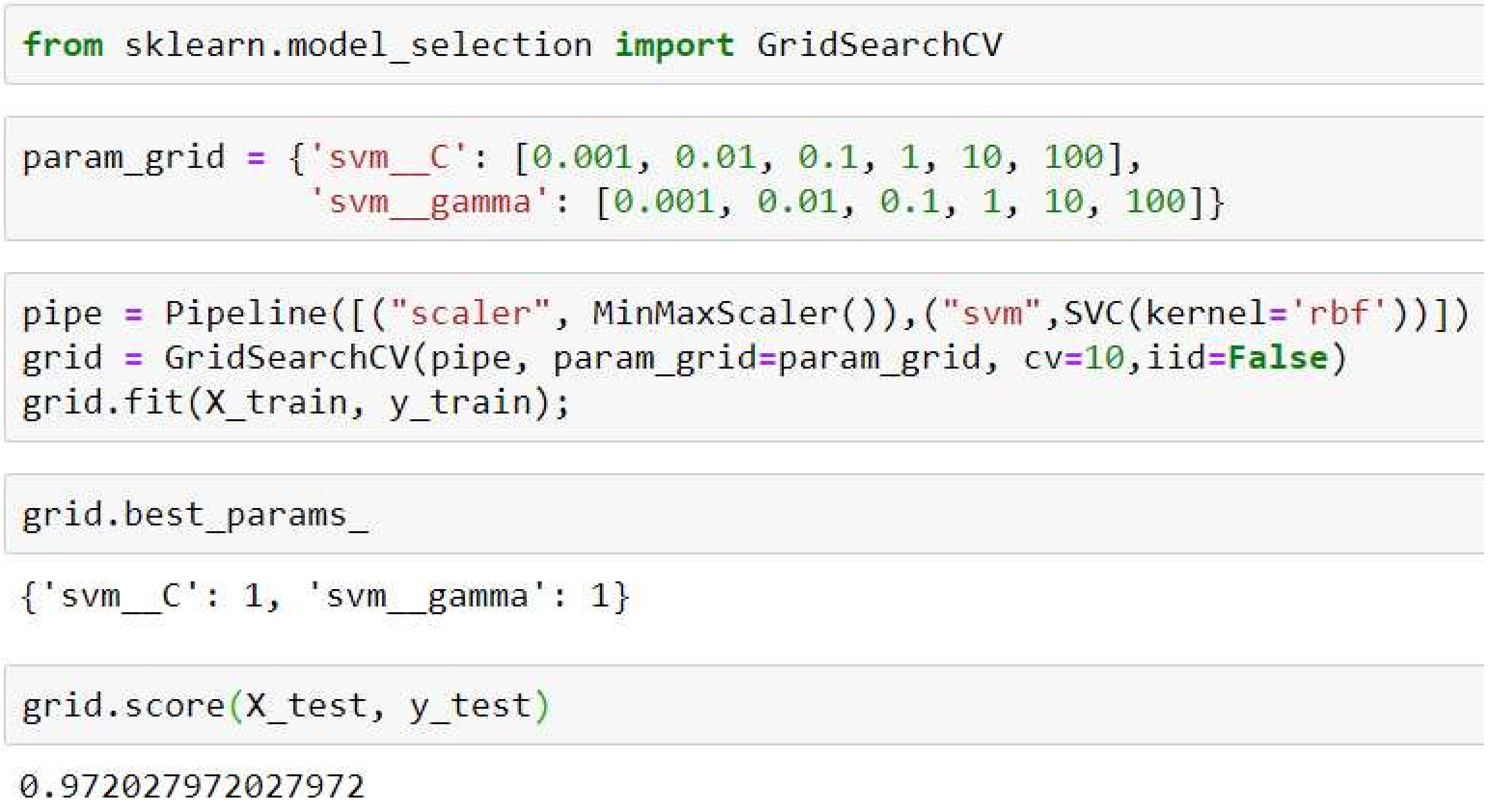
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
```

```
pipe = Pipeline([("scaler", MinMaxScaler()),
                 ("svm", SVC(kernel='rbf',gamma=1))
                ])
kfold = KFold(n_splits=10,random_state=1)
scores = cross_val_score(pipe, X, y, cv=kfold)
scores.mean()
```

```
0.9771303258145363
```

At each split only the train set is scaled with *MinMaxScaler*

Pipeline in Grid Search CV -Example



```
from sklearn.model_selection import GridSearchCV
```

```
param_grid = {'svm__C': [0.001, 0.01, 0.1, 1, 10, 100],
              'svm__gamma': [0.001, 0.01, 0.1, 1, 10, 100]}
```

```
pipe = Pipeline([("scaler", MinMaxScaler()),("svm",SVC(kernel='rbf'))])
grid = GridSearchCV(pipe, param_grid=param_grid, cv=10,iid=False)
grid.fit(X_train, y_train);
```

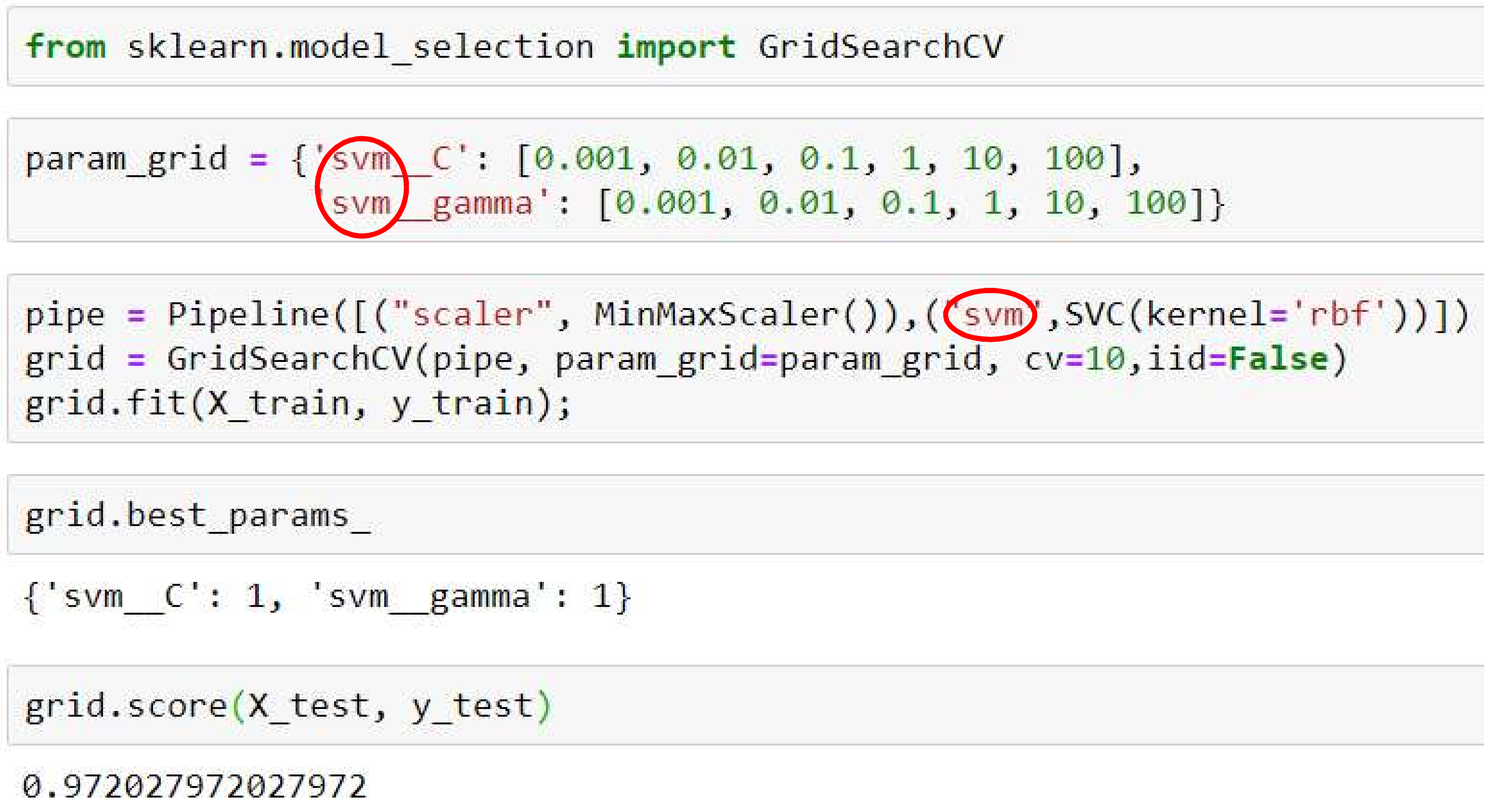
```
grid.best_params_
```

```
{'svm__C': 1, 'svm__gamma': 1}
```

```
grid.score(X_test, y_test)
```

```
0.972027972027972
```

Pipeline in Grid Search CV -Example



```python
from sklearn.model_selection import GridSearchCV
```

```python
param_grid = {'svm__C': [0.001, 0.01, 0.1, 1, 10, 100],
              'svm__gamma': [0.001, 0.01, 0.1, 1, 10, 100]}
```

```python
pipe = Pipeline([("scaler", MinMaxScaler()),('svm',SVC(kernel='rbf'))])
grid = GridSearchCV(pipe, param_grid=param_grid, cv=10,iid=False)
grid.fit(X_train, y_train);
```

```python
grid.best_params_
```

```
{'svm__C': 1, 'svm__gamma': 1}
```

```python
grid.score(X_test, y_test)
```

```
0.972027972027972
```

step name is followed by parameter name

make_pipeline

.

```
from sklearn.pipeline import make_pipeline
```

```
pipe1 = Pipeline([("scaler", MinMaxScaler()), ("svm", SVC(kernel='rbf',gamma=1))])
```

```
pipe2 = make_pipeline( MinMaxScaler(), SVC(kernel='rbf',gamma=1) )
```