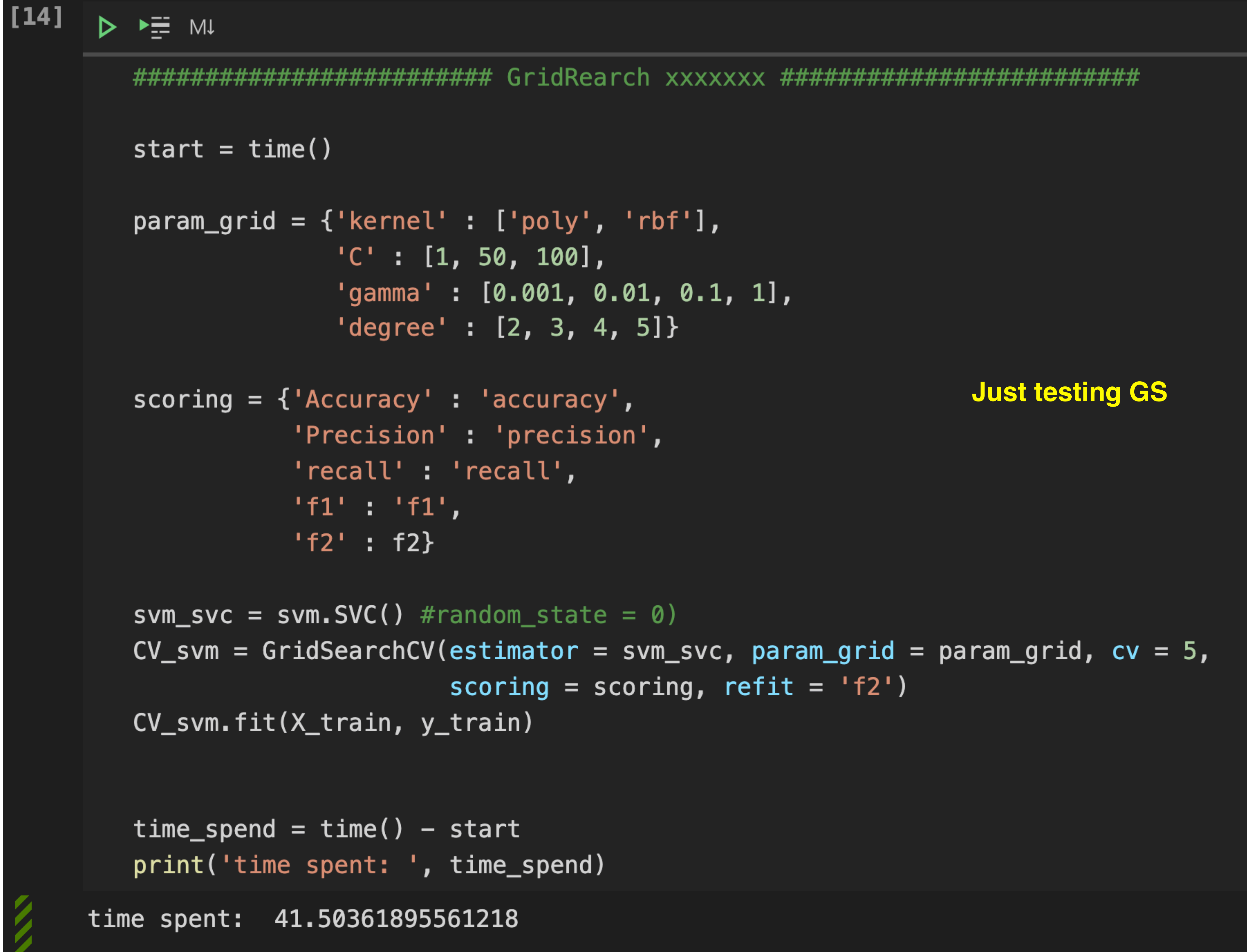
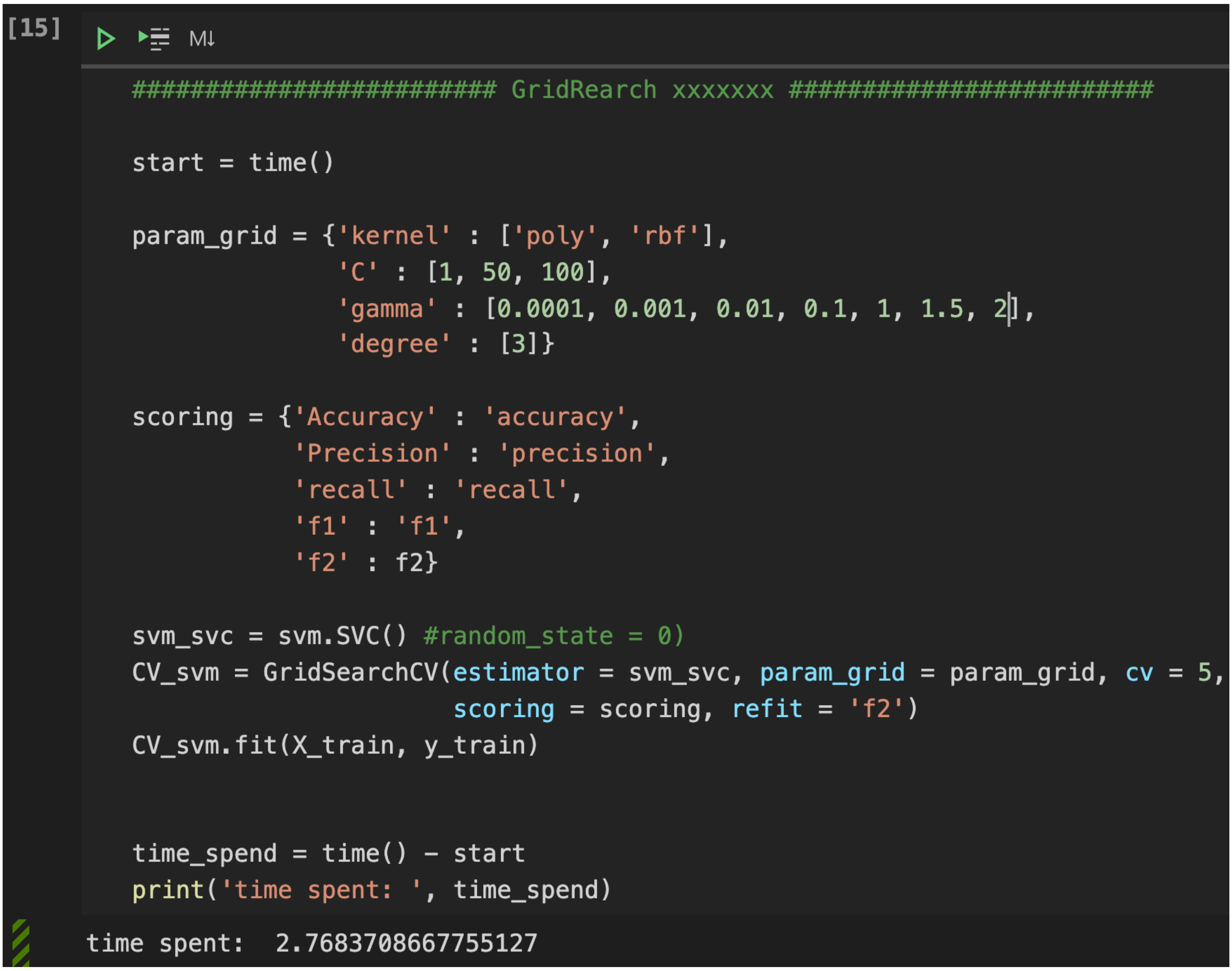


```
########################## GridRearch xxxxxxx ##########################

start = time()

param_grid = {'kernel' : ['poly', 'rbf'],
              'C' : [1, 50, 100],
              'gamma' : [0.001, 0.01, 0.1, 1],
              'degree' : [2, 3, 4, 5]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC() #random_state = 0)
CV_svm = GridSearchCV(estimator = svm_svc, param_grid = param_grid, cv = 5,
                      scoring = scoring, refit = 'f2')
CV_svm.fit(X_train, y_train)


time_spend = time() - start
print('time spent: ', time_spend)
```

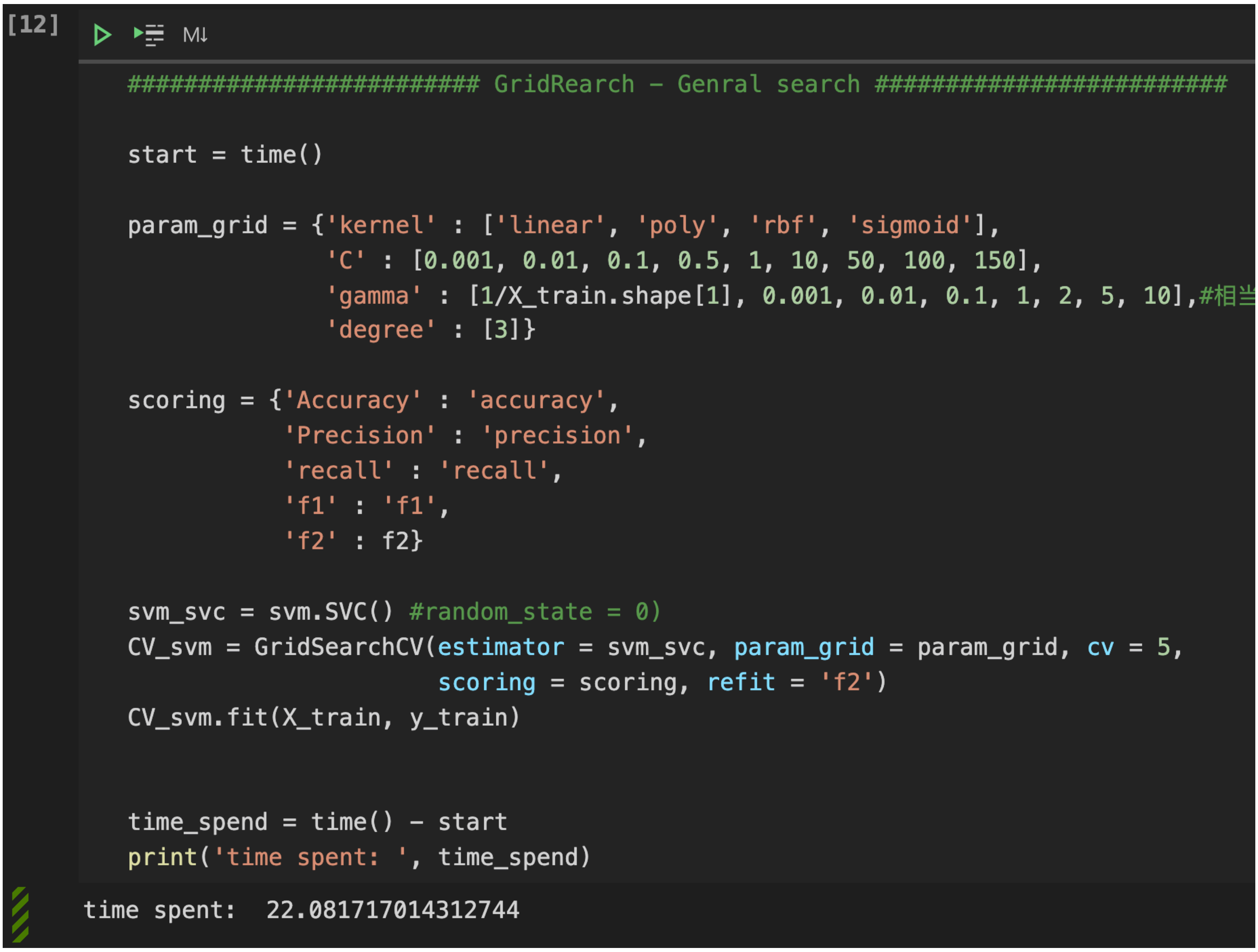
```
time spent:  41.50361895561218
```

Just testing GS



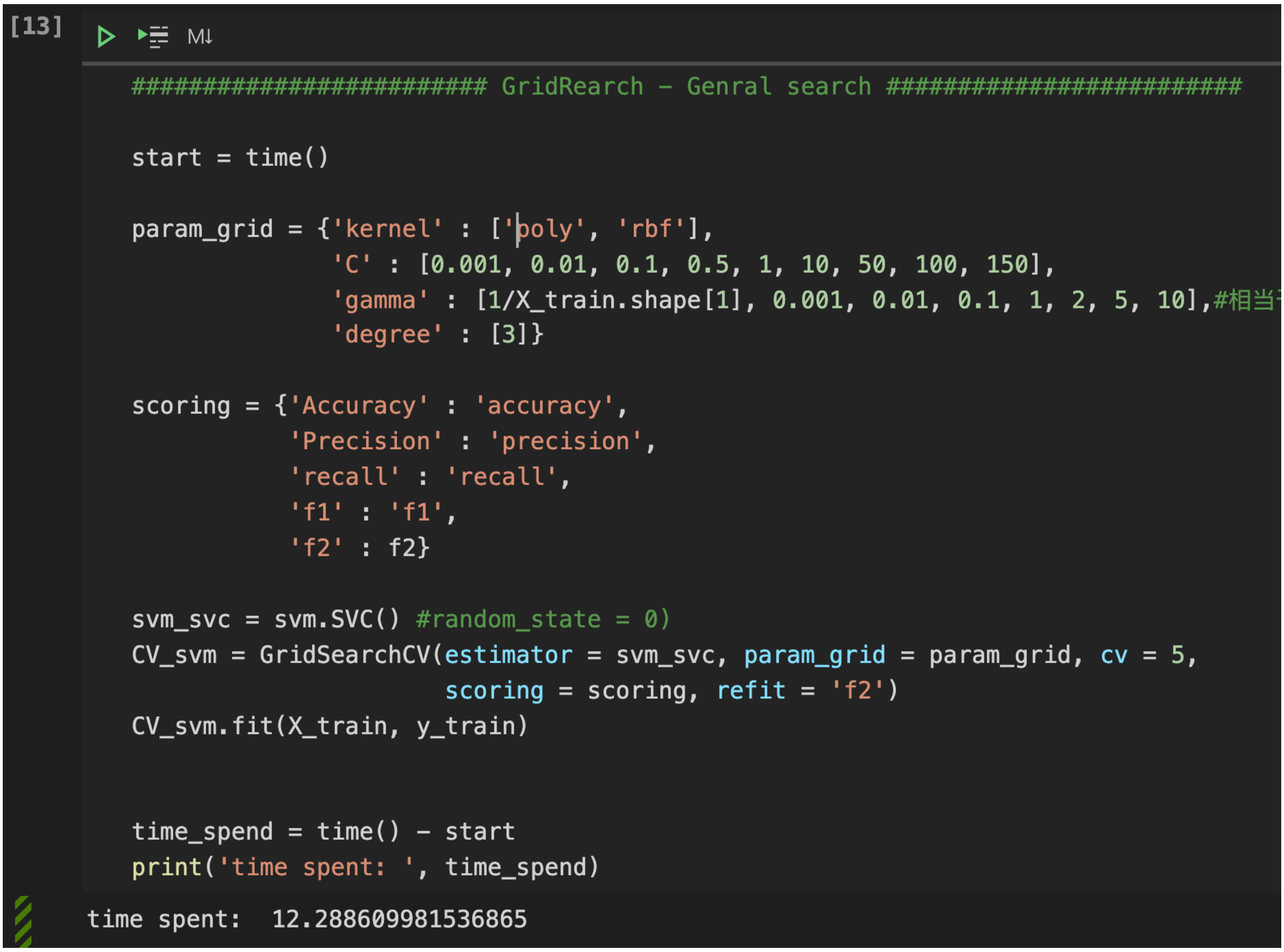
```
######################### GridRearch xxxxxxx #########################

start = time()

param_grid = {'kernel' : ['poly', 'rbf'],
              'C' : [1, 50, 100],
              'gamma' : [0.0001, 0.001, 0.01, 0.1, 1, 1.5, 2],
              'degree' : [3]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC() #random_state = 0)
CV_svm = GridSearchCV(estimator = svm_svc, param_grid = param_grid, cv = 5,
                      scoring = scoring, refit = 'f2')
CV_svm.fit(X_train, y_train)


time_spend = time() - start
print('time spent: ', time_spend)
```

```
time spent:  2.7683708667755127
```



```
######################### GridRearch - Genral search #########################

start = time()

param_grid = {'kernel' : ['linear', 'poly', 'rbf', 'sigmoid'],
              'C' : [0.001, 0.01, 0.1, 0.5, 1, 10, 50, 100, 150],
              'gamma' : [1/X_train.shape[1], 0.001, 0.01, 0.1, 1, 2, 5, 10],#相当
              'degree' : [3]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC() #random_state = 0)
CV_svm = GridSearchCV(estimator = svm_svc, param_grid = param_grid, cv = 5,
                      scoring = scoring, refit = 'f2')
CV_svm.fit(X_train, y_train)


time_spend = time() - start
print('time spent: ', time_spend)
```

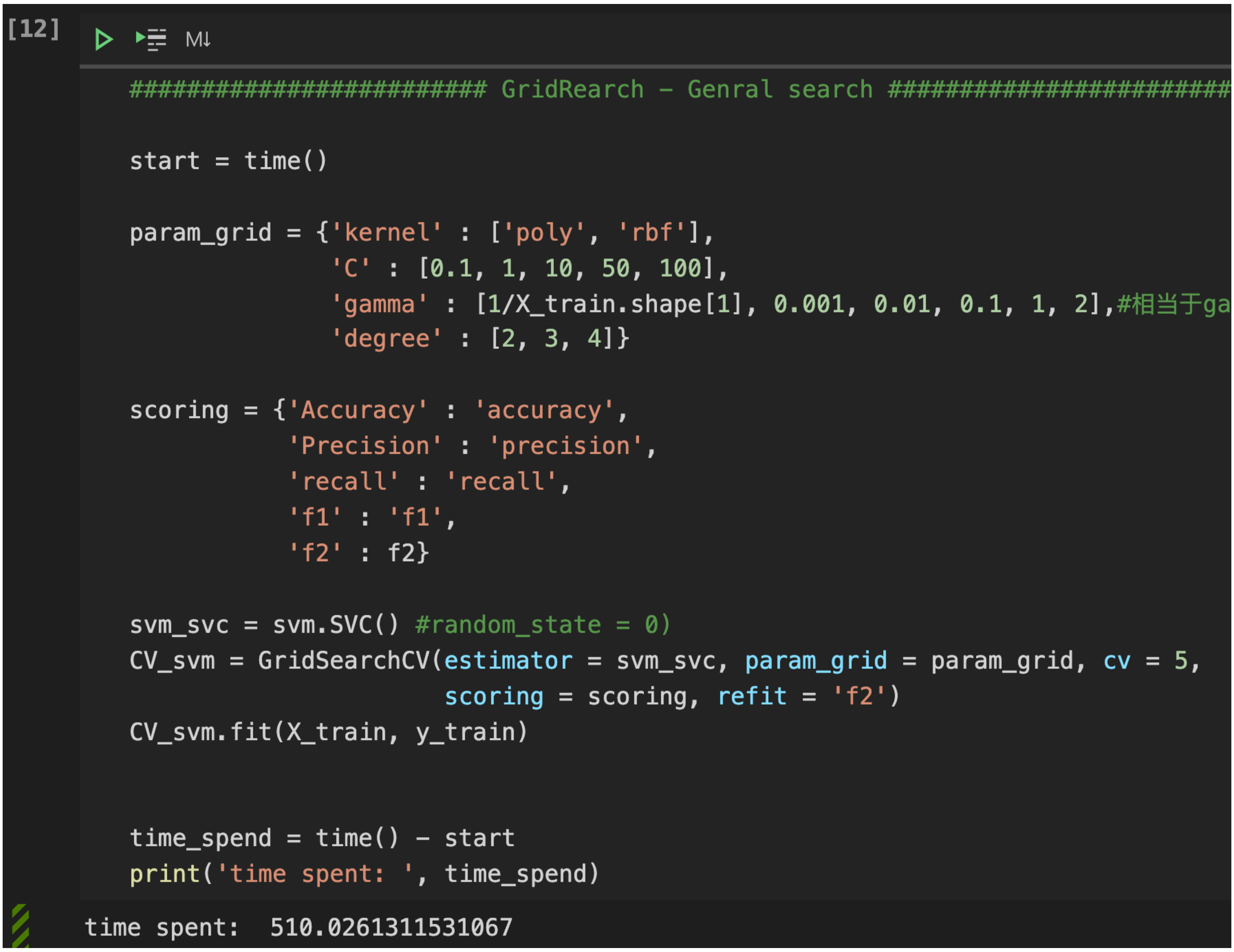
```
time spent:  22.081717014312744
```



```
######################### GridRearch - Genral search #########################

start = time()

param_grid = {'kernel' : ['poly', 'rbf'],
              'C' : [0.001, 0.01, 0.1, 0.5, 1, 10, 50, 100, 150],
              'gamma' : [1/X_train.shape[1], 0.001, 0.01, 0.1, 1, 2, 5, 10],#相当
              'degree' : [3]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC() #random_state = 0)
CV_svm = GridSearchCV(estimator = svm_svc, param_grid = param_grid, cv = 5,
                      scoring = scoring, refit = 'f2')
CV_svm.fit(X_train, y_train)


time_spend = time() - start
print('time spent: ', time_spend)
```

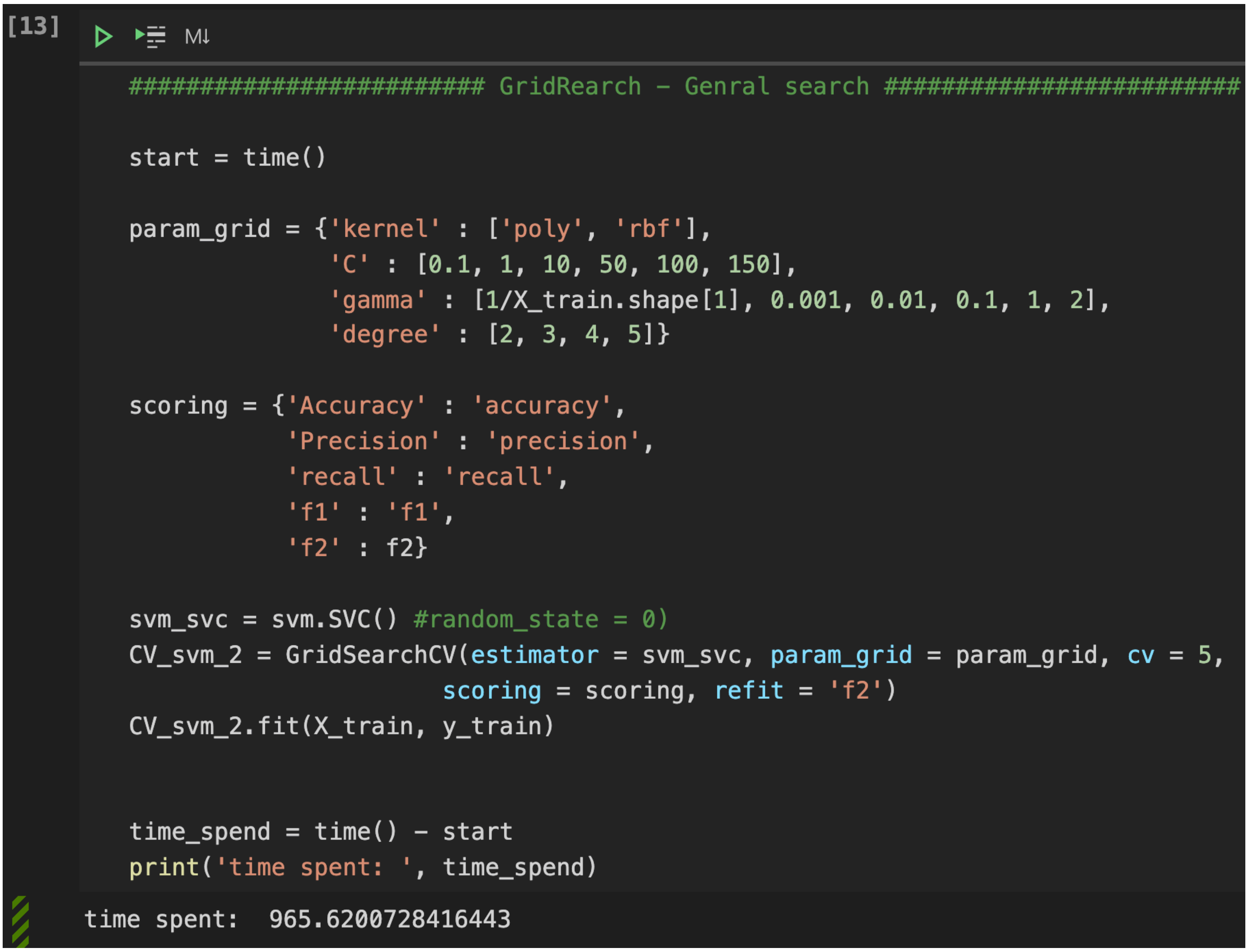
```
time spent:  12.288609981536865
```

Using the results from this search



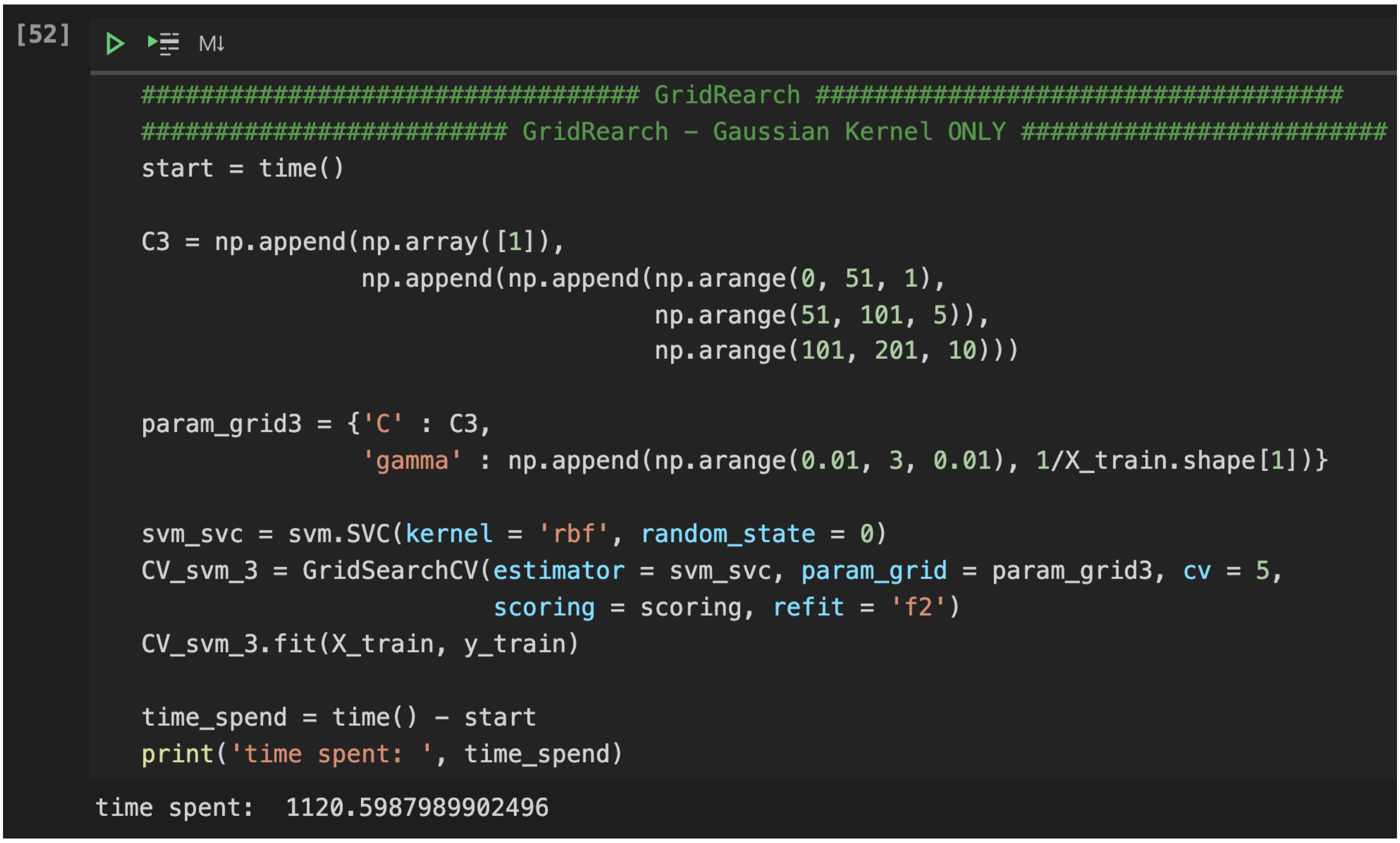
```
######################### GridRearch - Genral search ########################

start = time()

param_grid = {'kernel' : ['poly', 'rbf'],
              'C' : [0.1, 1, 10, 50, 100],
              'gamma' : [1/X_train.shape[1], 0.001, 0.01, 0.1, 1, 2],#相当于ga
              'degree' : [2, 3, 4]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC() #random_state = 0)
CV_svm = GridSearchCV(estimator = svm_svc, param_grid = param_grid, cv = 5,
                      scoring = scoring, refit = 'f2')
CV_svm.fit(X_train, y_train)


time_spend = time() - start
print('time spent: ', time_spend)
```

```
time spent:  510.0261311531067
```



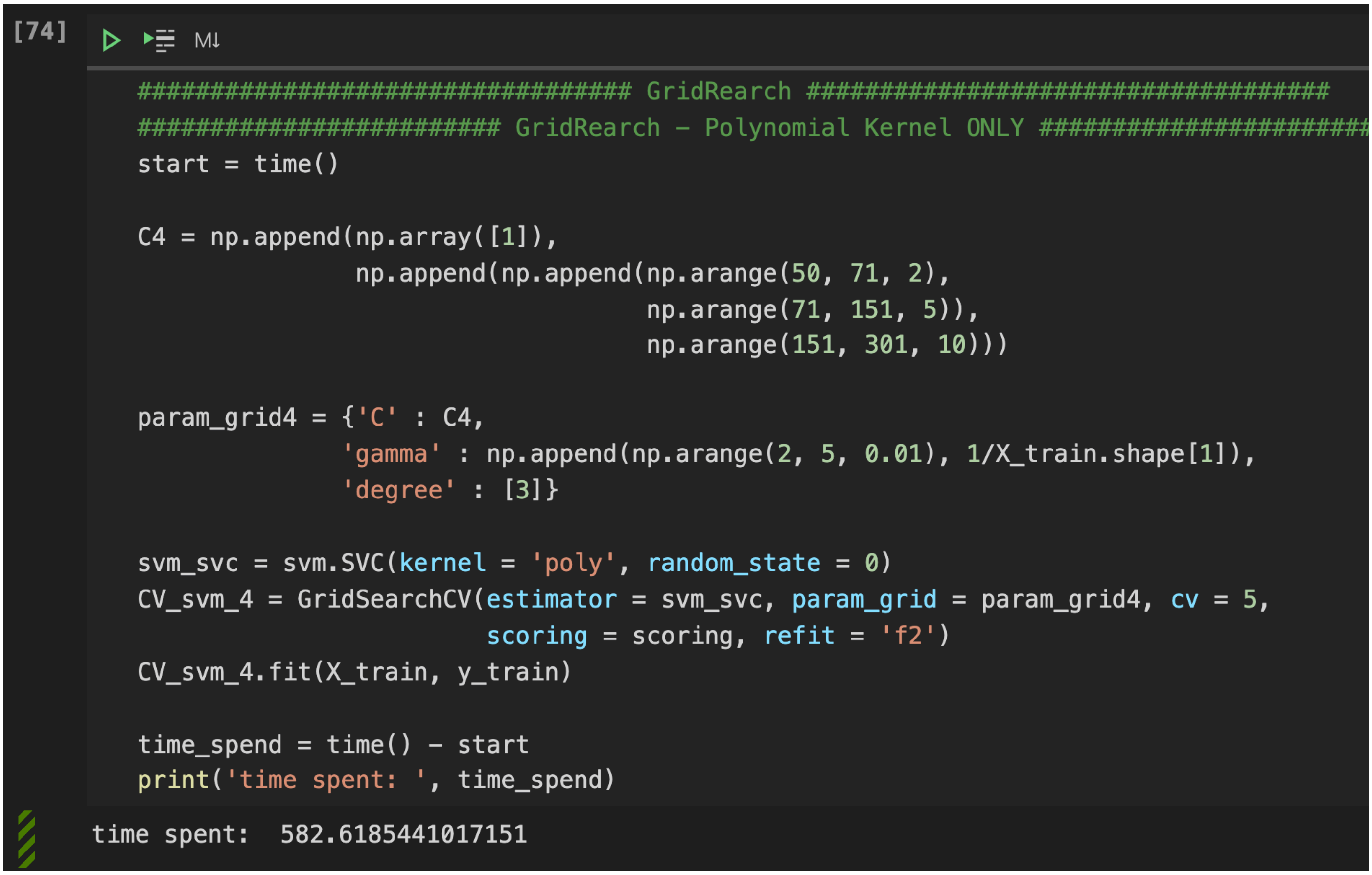
```
######################### GridRearch - Genral search #########################

start = time()

param_grid = {'kernel' : ['poly', 'rbf'],
              'C' : [0.1, 1, 10, 50, 100, 150],
              'gamma' : [1/X_train.shape[1], 0.001, 0.01, 0.1, 1, 2],
              'degree' : [2, 3, 4, 5]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC() #random_state = 0)
CV_svm_2 = GridSearchCV(estimator = svm_svc, param_grid = param_grid, cv = 5,
                      scoring = scoring, refit = 'f2')
CV_svm_2.fit(X_train, y_train)


time_spend = time() - start
print('time spent: ', time_spend)
```

```
time spent:  965.6200728416443
```



```
################################## GridRearch ####################################
######################### GridRearch - Gaussian Kernel ONLY #########################
start = time()

C3 = np.append(np.array([1]),
               np.append(np.append(np.arange(0, 51, 1),
                                   np.arange(51, 101, 5)),
                                   np.arange(101, 201, 10)))

param_grid3 = {'C' : C3,
               'gamma' : np.append(np.arange(0.01, 3, 0.01), 1/X_train.shape[1])}

svm_svc = svm.SVC(kernel = 'rbf', random_state = 0)
CV_svm_3 = GridSearchCV(estimator = svm_svc, param_grid = param_grid3, cv = 5,
                        scoring = scoring, refit = 'f2')
CV_svm_3.fit(X_train, y_train)

time_spend = time() - start
print('time spent: ', time_spend)
```

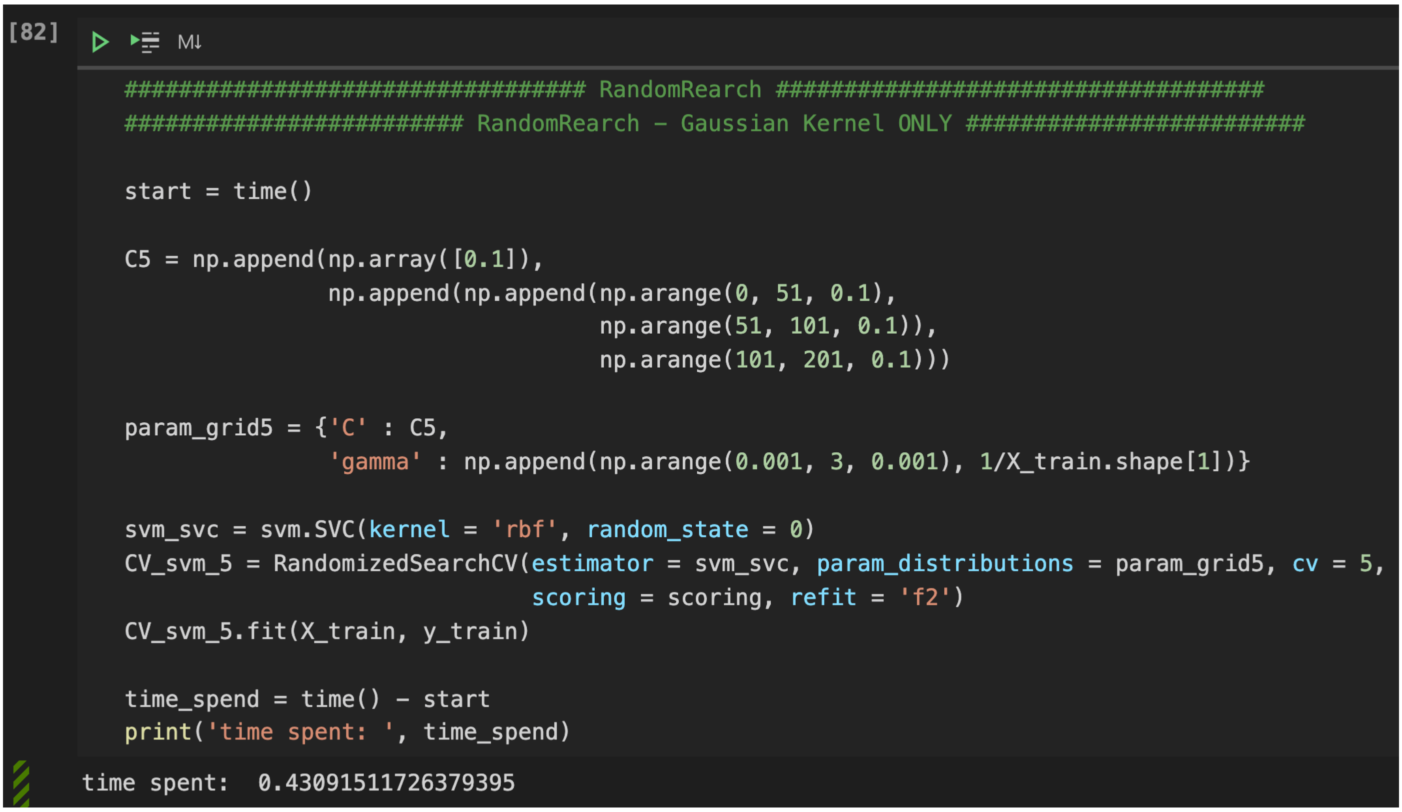
```
time spent:  1120.5987989902496
```



[74]

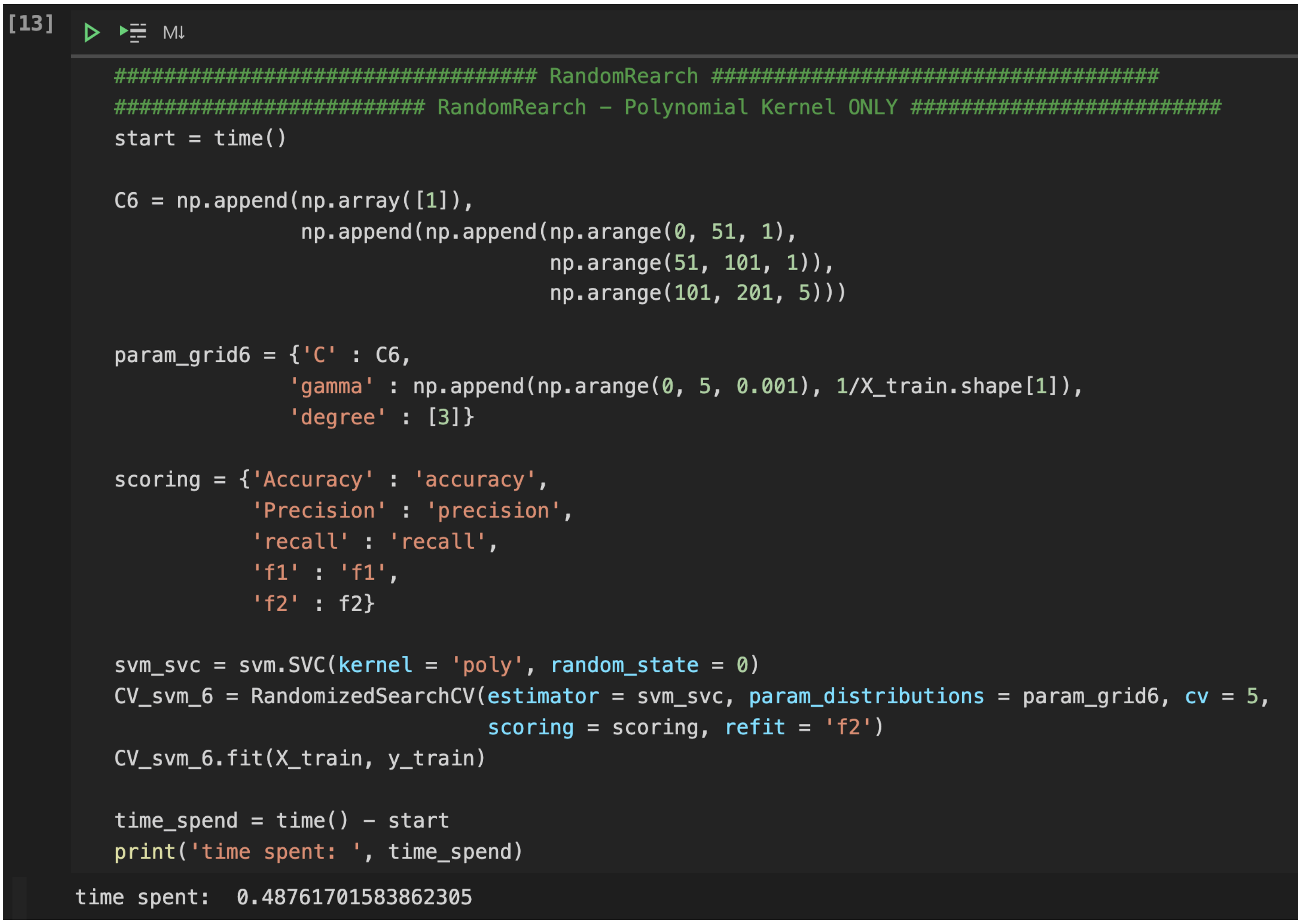
```
################################## GridRearch ####################################
######################### GridRearch - Polynomial Kernel ONLY ######################
start = time()

C4 = np.append(np.array([1]),
               np.append(np.append(np.arange(50, 71, 2),
                                   np.arange(71, 151, 5)),
                                   np.arange(151, 301, 10)))

param_grid4 = {'C' : C4,
              'gamma' : np.append(np.arange(2, 5, 0.01), 1/X_train.shape[1]),
              'degree' : [3]}

svm_svc = svm.SVC(kernel = 'poly', random_state = 0)
CV_svm_4 = GridSearchCV(estimator = svm_svc, param_grid = param_grid4, cv = 5,
                        scoring = scoring, refit = 'f2')
CV_svm_4.fit(X_train, y_train)

time_spend = time() - start
print('time spent: ', time_spend)
```

```
time spent:  582.6185441017151
```



```
################################## RandomRearch ####################################
######################### RandomRearch - Gaussian Kernel ONLY #########################

start = time()

C5 = np.append(np.array([0.1]),
               np.append(np.append(np.arange(0, 51, 0.1),
                                   np.arange(51, 101, 0.1)),
                                   np.arange(101, 201, 0.1)))

param_grid5 = {'C' : C5,
               'gamma' : np.append(np.arange(0.001, 3, 0.001), 1/X_train.shape[1])}

svm_svc = svm.SVC(kernel = 'rbf', random_state = 0)
CV_svm_5 = RandomizedSearchCV(estimator = svm_svc, param_distributions = param_grid5, cv = 5,
                              scoring = scoring, refit = 'f2')
CV_svm_5.fit(X_train, y_train)

time_spend = time() - start
print('time spent: ', time_spend)
```

```
time spent:  0.43091511726379395
```



```
################################## RandomRearch ####################################
######################### RandomRearch - Polynomial Kernel ONLY #########################
start = time()

C6 = np.append(np.array([1]),
               np.append(np.append(np.arange(0, 51, 1),
                                   np.arange(51, 101, 1)),
                                   np.arange(101, 201, 5)))

param_grid6 = {'C' : C6,
              'gamma' : np.append(np.arange(0, 5, 0.001), 1/X_train.shape[1]),
              'degree' : [3]}

scoring = {'Accuracy' : 'accuracy',
           'Precision' : 'precision',
           'recall' : 'recall',
           'f1' : 'f1',
           'f2' : f2}

svm_svc = svm.SVC(kernel = 'poly', random_state = 0)
CV_svm_6 = RandomizedSearchCV(estimator = svm_svc, param_distributions = param_grid6, cv = 5,
                              scoring = scoring, refit = 'f2')
CV_svm_6.fit(X_train, y_train)

time_spend = time() - start
print('time spent: ', time_spend)
```

```
time spent:  0.48761701583862305
```