ROC curve

import library

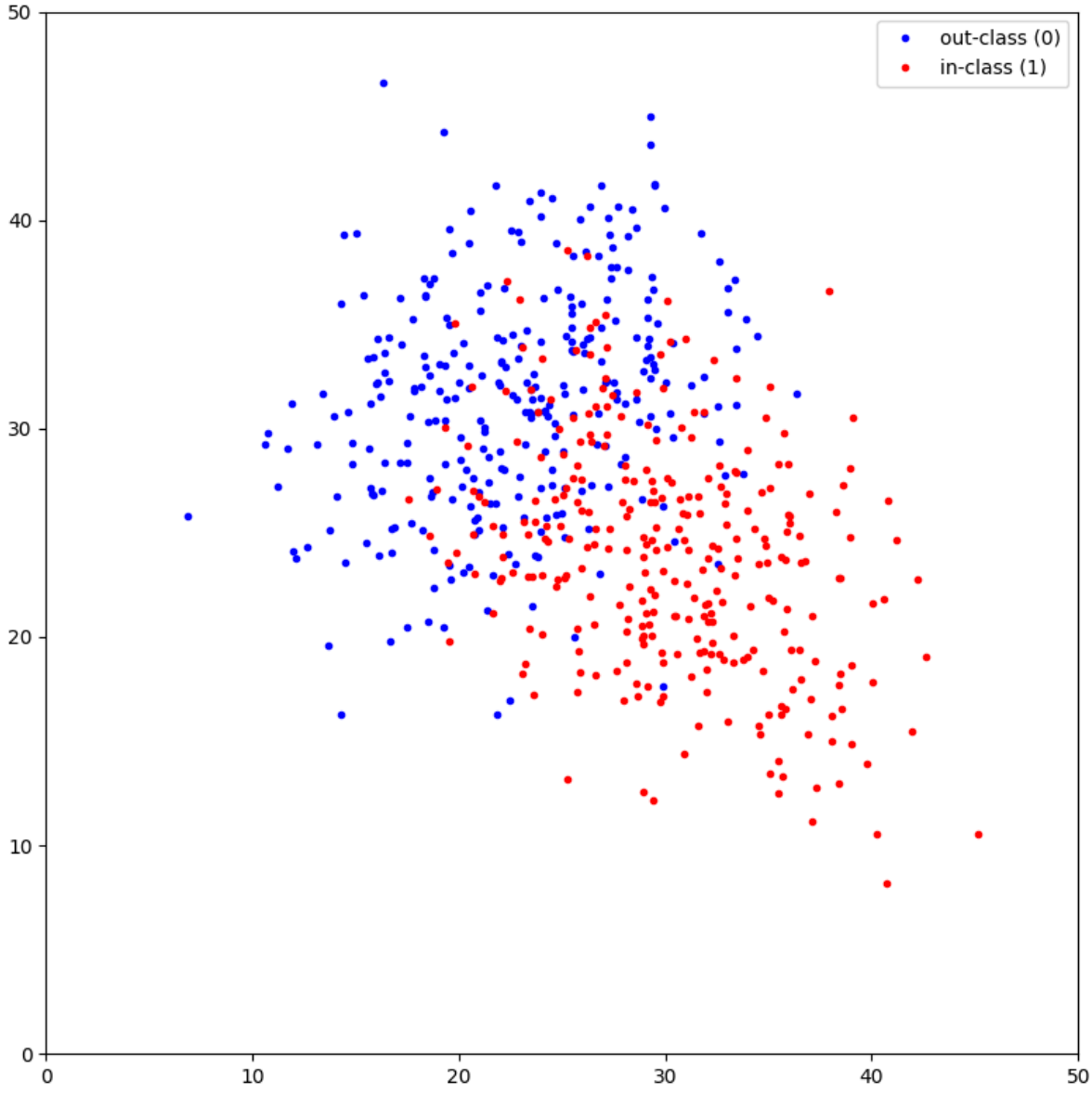
In [ ]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import time
import util
```

load training data

In [ ]:
```
fname_data  = '10_data.csv'
data        = np.genfromtxt(fname_data, delimiter=',')
num_data    = data.shape[0]

x       = np.zeros(num_data)
y       = np.zeros(num_data)
label   = np.zeros(num_data)

for i in range(num_data):
    x[i]        = data[i,0]
    y[i]        = data[i,1]
    label[i]    = 1-data[i,2]

x = np.reshape(x, (num_data, 1))
y = np.reshape(y, (num_data, 1))
label = np.reshape(label, (num_data, 1))
```

plot the data

In [ ]:
```
x0 = x[label == 0]
y0 = y[label == 0]
x1 = x[label == 1]
y1 = y[label == 1]

plt.figure(figsize=(8,8))
plt.plot(x0, y0, '.', color='blue', label='out-class (0)')
plt.plot(x1, y1, '.', color='red', label='in-class (1)')
plt.xlim(0, 50)
plt.ylim(0, 50)
plt.legend()
plt.tight_layout()
plt.show()
```



In [ ]:
```
theta1  = 0.1
pred1   = util.compute_prediction(x, y, theta1)
fp1     = util.compute_false_positive(label, pred1)
tp1     = util.compute_true_positive(label, pred1)
```

In [ ]:
```
def plot_01():
    print('theta =', theta1, ', fp =', fp1, ', tp =', tp1)
```

In [ ]:
```
plot_01()
```

```
theta = 0.1 , fp = 900 , tp = 900
```

In [ ]:
```
theta2  = 1
pred2   = util.compute_prediction(x, y, theta2)
fp2     = util.compute_false_positive(label, pred2)
tp2     = util.compute_true_positive(label, pred2)
```

In [ ]:
```
def plot_02():
    print('theta =', theta2, ', fp =', fp2, ', tp =', tp2)
```

In [ ]:
```
plot_02()
```

```
theta = 1 , fp = 900 , tp = 900
```

In [ ]:
```
theta3  = 100
pred3   = util.compute_prediction(x, y, theta3)
fp3     = util.compute_false_positive(label, pred3)
tp3     = util.compute_true_positive(label, pred3)
```

In [ ]:
```
def plot_03():
    print('theta =', theta3, ', fp =', fp3, ', tp =', tp3)
```

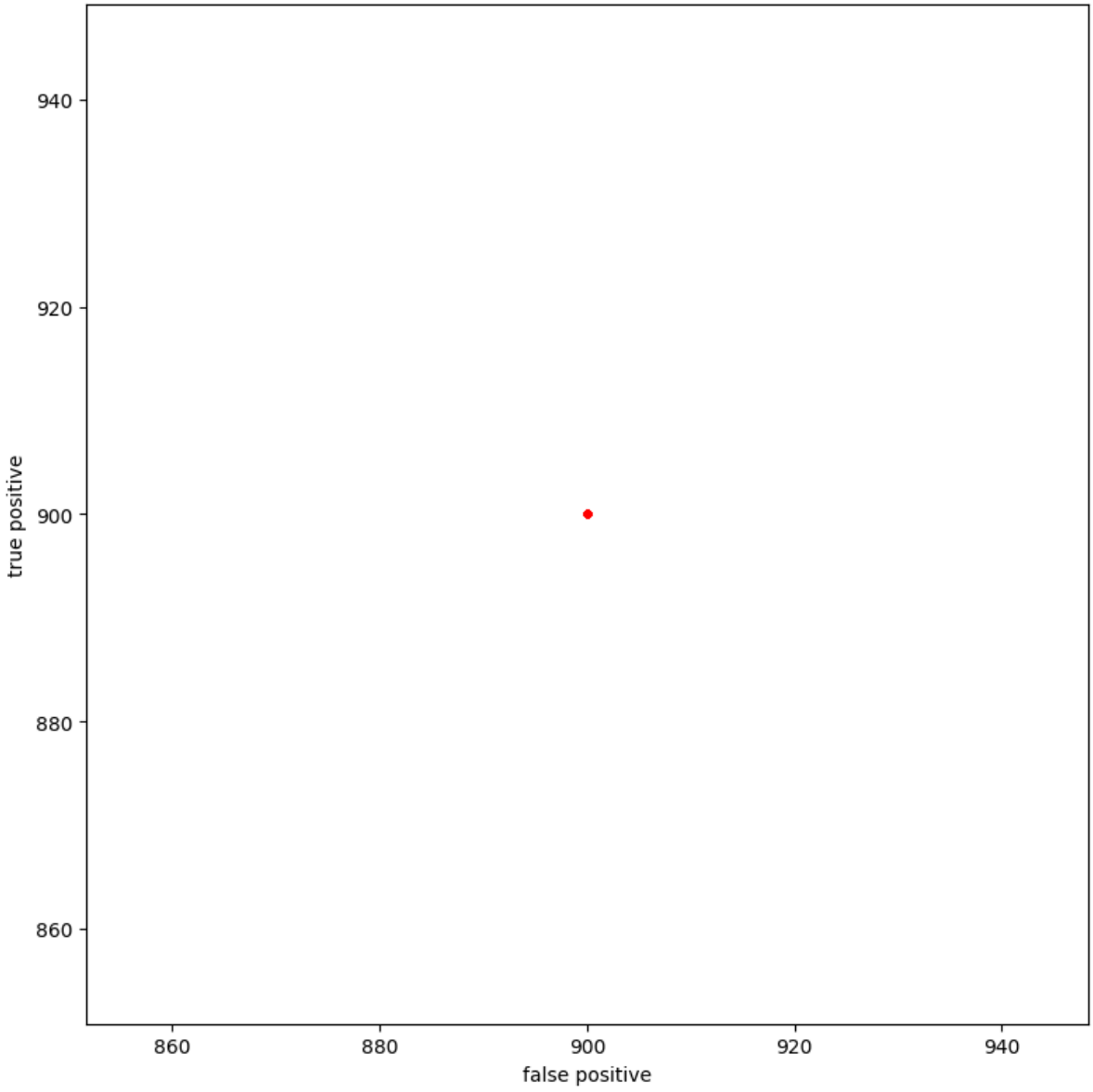
In [ ]:
```
plot_03()
```

```
theta = 100 , fp = 900 , tp = 900
```

In [ ]:
```
tp_list = []
fp_list = []

for theta in np.arange(0.1, 100, 0.01):
    pred    = util.compute_prediction(x, y, theta)
    fp      = util.compute_false_positive(label, pred)
    tp      = util.compute_true_positive(label, pred)
    tp_list.append(tp.item())
    fp_list.append(fp.item())
```

In [ ]:
```
def plot_04():
    plt.figure(figsize=(8,8))
    plt.plot(fp_list, tp_list, '.', color='red')
    plt.xlabel('false positive')
    plt.ylabel('true positive')
    plt.axis('equal')
    plt.tight_layout()
    plt.show()
```

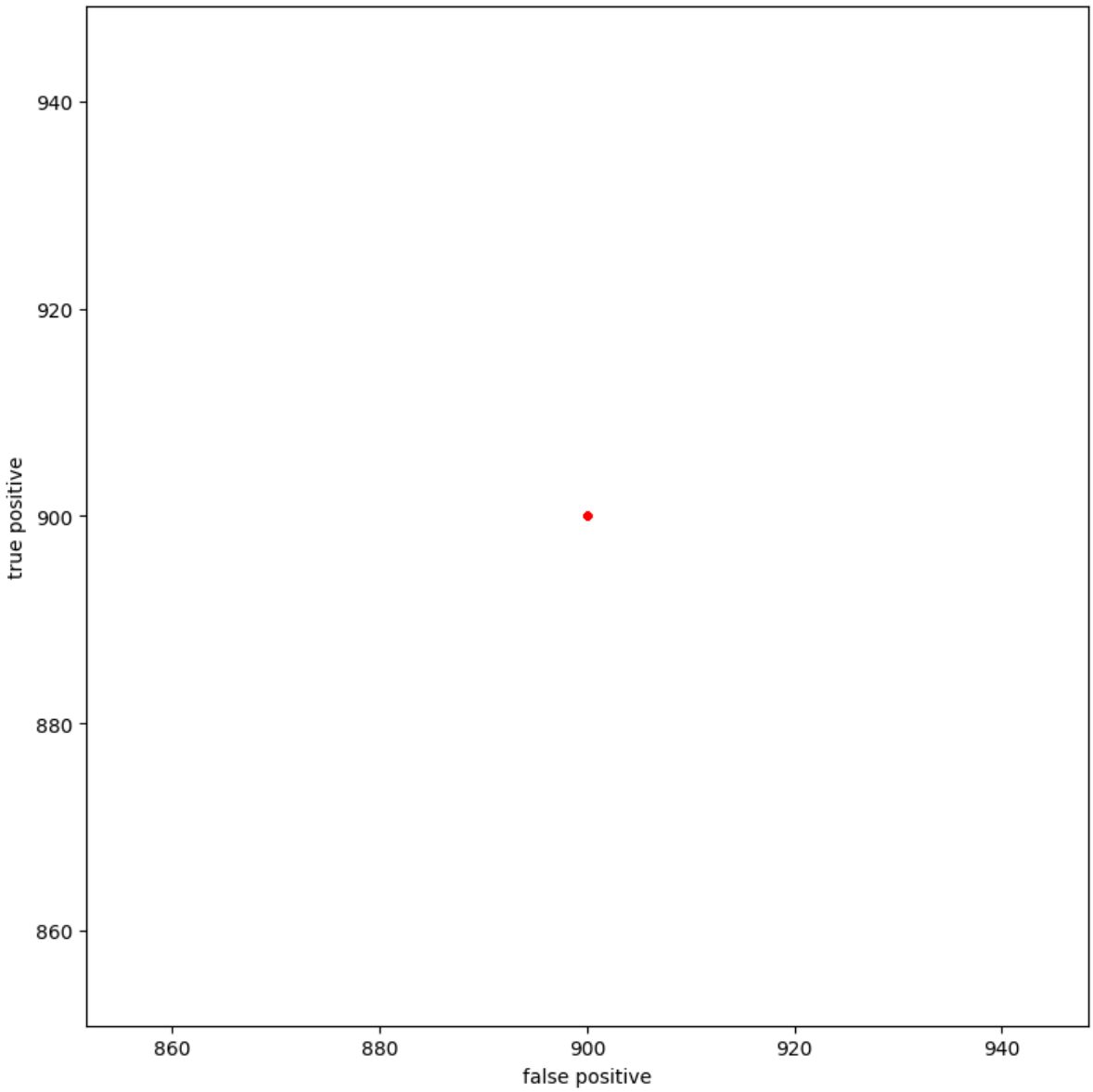
In [ ]:
```
plot_04()
```



results

In [ ]:
```
number_result = 4

for i in range(number_result):
    title           = '# RESULT # {:02d}'.format(i+1)
    name_function   = 'plot_{:02d}()'.format(i+1)

    print('')
    print('#########################################################################################')
    print(title)
    print('#########################################################################################')
    print('')

    eval(name_function)
```

```
##########################################################################################
# RESULT # 01
##########################################################################################

theta = 0.1 , fp = 900 , tp = 900

##########################################################################################
# RESULT # 02
##########################################################################################

theta = 1 , fp = 900 , tp = 900

##########################################################################################
# RESULT # 03
##########################################################################################

theta = 100 , fp = 900 , tp = 900

##########################################################################################
# RESULT # 04
##########################################################################################
```



In [ ]: