In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc
```

In [2]:
```
=np.asarray([[0,0,1],[0,1,0],[1,0,0],[0,0,1],[1,0,0],[0,1,0],[0,1,0],[0,1,0],[0,0,1],
np.asarray([[0.1,0.2,0.7],[0.1,0.6,0.3],[0.5,0.2,0.3],[0.1,0.1,0.8],[0.4,0.2,0.4],[0.
```

In [3]:
```
n_classes=len(y_true[1,:])
n_classes
```

Out[3]: 3

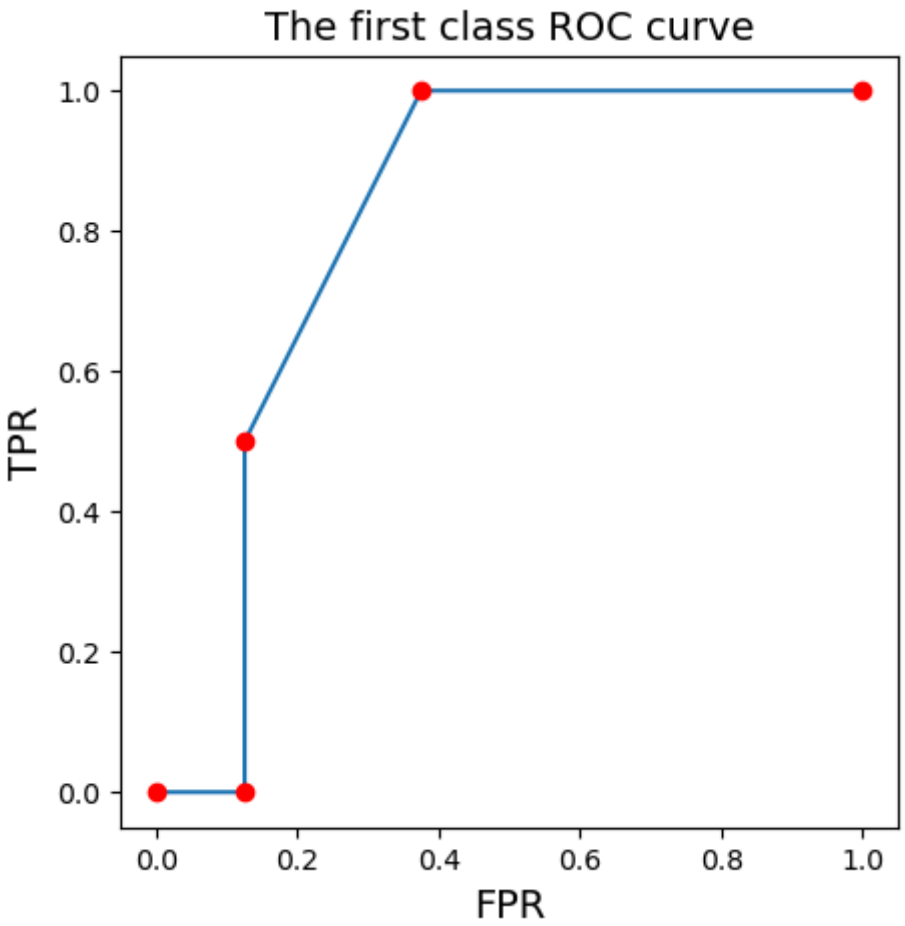
In [4]:
```
fpr=dict()
tpr=dict()
roc_auc=dict()
```

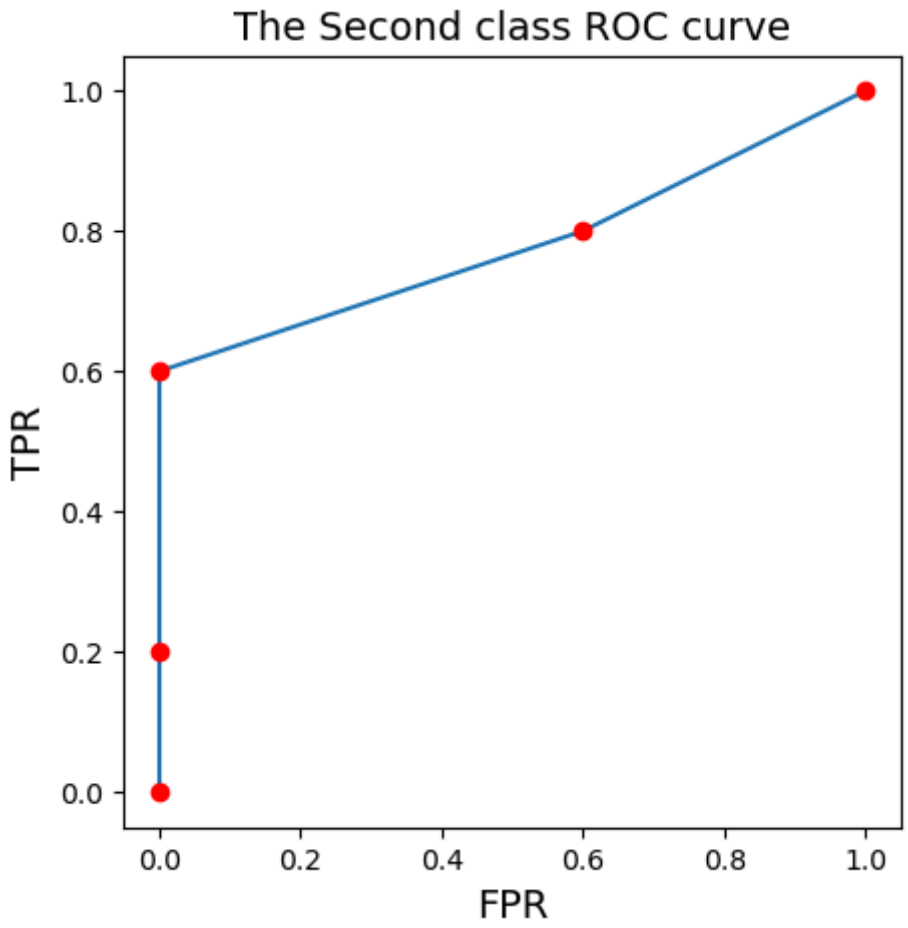
In [5]:
```
for i in range(n_classes):
    fpr[i],tpr[i],_=roc_curve(y_true[:,i],y_pre[:,i])
    roc_auc[i]= auc(fpr[i],tpr[i])
    print(f"Class{i+1} FPR:{fpr[i]} TPR:{tpr[i]}")
```

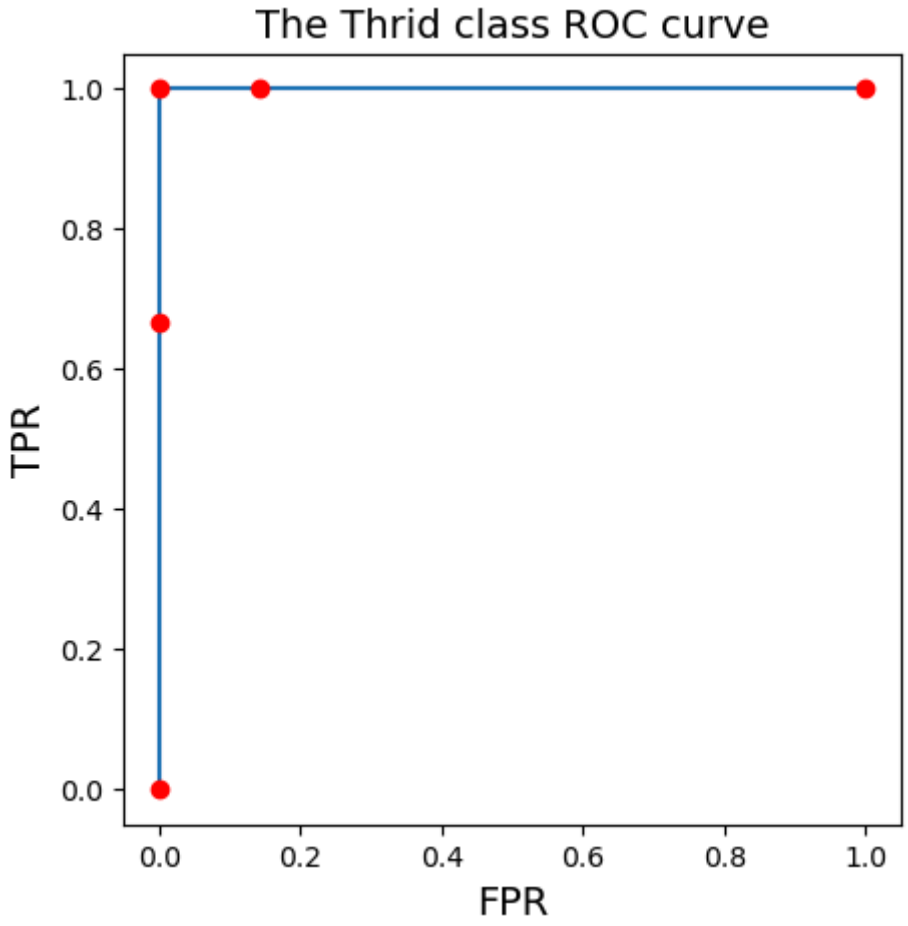
```
Class1 FPR:[0.     0.125 0.125 0.375 1.    ] TPR:[0.   0.   0.5 1.   1. ]
Class2 FPR:[0.   0.   0.   0.6 1. ] TPR:[0.   0.2 0.6 0.8 1. ]
Class3 FPR:[0.         0.         0.         0.14285714 1.        ] TPR:[0.
0.66666667 1.         1.         1.        ]
```

In [6]:
```
for i in range(n_classes):
    plt.figure(figsize=(5,5))
    if i==0:
        plt.title('The first class ROC curve',fontsize=14)
    if i==1:
        plt.title('The Second class ROC curve',fontsize=14)
    if i==2:
        plt.title('The Thrid class ROC curve',fontsize=14)

    plt.plot(fpr[i],tpr[i])
    plt.plot(fpr[i],tpr[i],'ro')
    plt.ylabel('TPR',fontsize=14)
    plt.xlabel('FPR',fontsize=14)
```




The Second class ROC curve
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0


The Thrid class ROC curve
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0

In [7]:
```
for i in range(n_classes):
    print(f"Class {i+1} ROC AUC: {roc_auc[i]:.2f}")
```

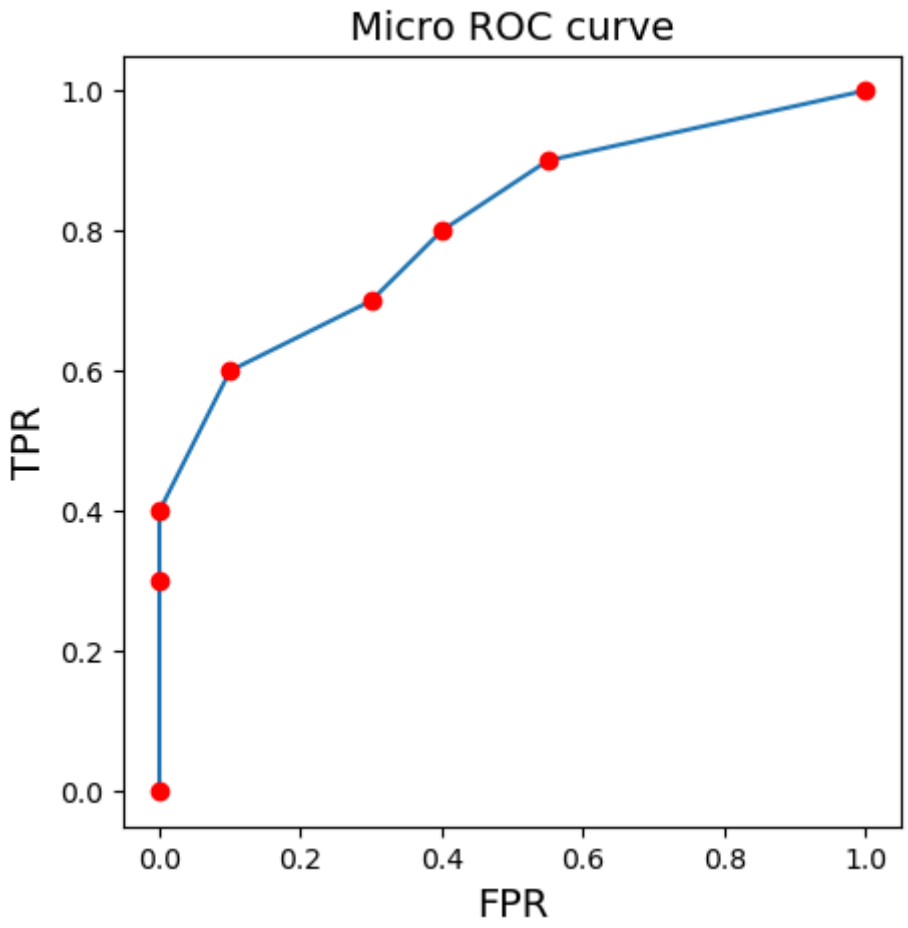
```
Class 1 ROC AUC: 0.81
Class 2 ROC AUC: 0.78
Class 3 ROC AUC: 1.00
```

In [8]:
```
fpr["micro"],tpr["micro"],_=roc_curve(y_true.ravel(),y_pre.ravel())
roc_auc=auc(fpr["micro"],tpr["micro"])
print(f"Micro ROC AUC: {roc_auc:.2f}")
```

```
Micro ROC AUC: 0.81
```

In [9]:
```
plt.figure(figsize=(5,5))
plt.title('Micro ROC curve',fontsize=14)
plt.plot(fpr["micro"],tpr["micro"])
plt.plot(fpr["micro"],tpr["micro"],'ro')
plt.ylabel('TPR',fontsize=14)
plt.xlabel('FPR',fontsize=14)
```

Out[9]:
```
Text(0.5, 0, 'FPR')
```



In [10]:
```
fpr_grid=np.linspace(0.0,1.0,10)
mean_tpr=np.zeros_like(fpr_grid)
for i in range(n_classes):
    mean_tpr+=np.interp(fpr_grid,fpr[i],tpr[i])
```

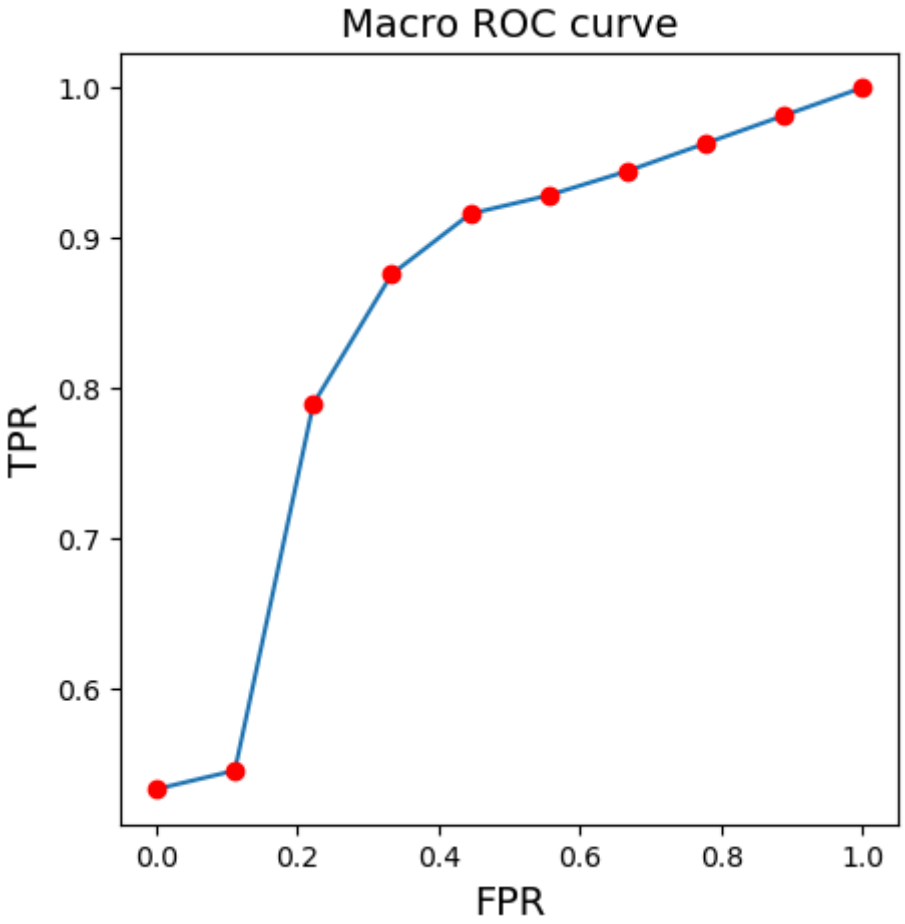
In [11]:
```
mean_tpr/=n_classes
```

In [12]:
```
fpr["macro"]=fpr_grid
tpr["macro"]=mean_tpr
roc_auc=auc(fpr["macro"],tpr["macro"])
print(f"Macro ROC AUC {roc_auc:.2f}")
```

```
Macro ROC AUC 0.86
```

In [13]:
```
plt.figure(figsize=(5,5))
plt.title('Macro ROC curve',fontsize=14)
plt.plot(fpr["macro"],tpr["macro"])
plt.plot(fpr["macro"],tpr["macro"],'ro')
plt.ylabel('TPR',fontsize=14)
plt.xlabel('FPR',fontsize=14)
```

Out[13]:
```
Text(0.5, 0, 'FPR')
```



In [14]:
```
y_true_list=list([tuple(t) for t in y_true])
classNum=dict((a,y_true_list.count(a))for a in y_true_list)
n1=classNum[(1,0,0)]
n2=classNum[(0,1,0)]
n3=classNum[(0,0,1)]
ratio=[n1/(n1+n2+n3),n2/(n1+n2+n3),n3/(n1+n2+n3)]
avg_tpr=np.zeros_like(fpr_grid)
```

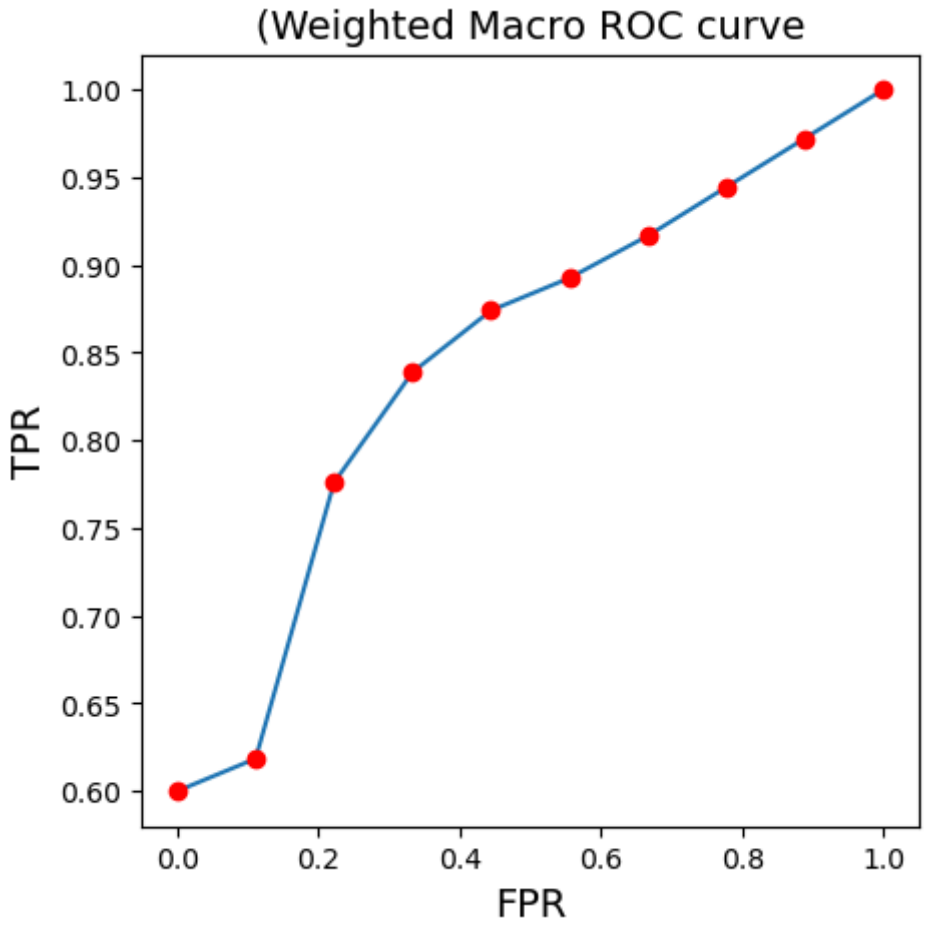
In [15]:
```
for i in range(n_classes):
    avg_tpr+=ratio[i]*np.interp(fpr_grid,fpr[i],tpr[i])
```

In [16]:
```
fpr["weighted"]=fpr_grid
tpr["weighted"]=avg_tpr
roc_auc=auc(fpr["weighted"],tpr["weighted"])
print(f"Weighted Macro ROC AUC {roc_auc:.2f}")
```

```
Weighted Macro ROC AUC 0.85
```

In [17]:
```
plt.figure(figsize=(5,5))
plt.title('(Weighted Macro ROC curve',fontsize=14)
plt.plot(fpr["weighted"],tpr["weighted"])
plt.plot(fpr["weighted"],tpr["weighted"],'ro')
plt.ylabel('TPR',fontsize=14)
plt.xlabel('FPR',fontsize=14)
```

Out[17]:
```
Text(0.5, 0, 'FPR')
```



In [ ]: