

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
dfx=pd.read_csv("xdata.csv")
dfy=pd.read_csv("ydata.csv")
```

In [8]:

```
X=dfx.values
Y=dfy.values
```

```
X=X[:,1:]
Y=Y[:,1:].reshape((-1,))
```

```
print(X)
print(X.shape)
print(Y.shape)
```

```
[[ 3.516279    3.71975047]
 [ 4.60649561  3.41437943]
 [ 4.67701056  3.04724443]
 [ 4.15772454  3.13867678]
 [ 3.15443087  3.61985805]
 [ 3.41751854  4.59164128]
 [ 1.61124605  3.39835595]
 [ 4.28808719  4.04396943]
 [ 3.19607968  4.95114323]
 [ 1.6258394   4.90239967]
 [ 2.82894521  3.91445026]
 [ 2.28966729  5.06896877]
 [ 2.89206152  3.66529487]
 [ 2.33634919  4.77256166]
 [ 3.67027475  3.79730296]
 [ 2.30717865  4.73262822]
 [ 3.15130599  3.43648588]
 [ 4.14502484  2.94851846]
 [ 2.17253471  3.6488889 ]
 [ 3.39390371  4.13447274]
 [ 1.59644351  4.43875703]
 [ 3.38348549  4.02119623]
 [ 5.08330979  4.03477733]
 [ 1.21644435  4.31174386]
 [ 4.15939179  2.54939222]
 [ 2.2392529   4.33673035]
 [ 2.99918423  5.63946256]
 [ 3.10384641  3.0156258 ]
 [ 2.47612428  4.12508079]
 [ 3.2447355   2.49239435]
 [ 2.03739207  5.03815425]
 [ 0.60211258  5.29511732]
 [ 4.96947578  3.37368252]
 [ 3.21264629  3.83263367]
 [ 4.37032895  4.13833792]
 [ 3.69323485  3.24528272]
 [ 3.09839038  3.35192483]
 [ 3.17307768  3.69712075]
 [ 2.73714175  3.32335845]
 [ 3.67894216  3.97815432]
 [ 2.48794954  3.43911347]
 [ 3.18673695  3.40153958]
 [ 3.18191183  1.97043755]
 [ 4.15883175  2.76019051]
 [ 2.20697823  4.70994226]
 [ 2.84548839  5.14262008]
 [ 4.59079568  4.13744089]
 [ 2.36254478  4.90779038]
 [ 2.67790849  1.89395445]
 [ 1.57605714  4.64044625]
 [ 3.6808431   4.57068586]
 [ 2.92745709  3.82770824]
 [ 3.20588663  3.91087375]
 [ 3.30360182  3.23995077]
 [ 2.20999864  3.8340588 ]]
```

[2.20555004 3.0040000]
[2.323852 5.1822002]
[2.16712307 4.52288264]
[2.39547176 3.646189]
[2.69133673 4.72790838]
[2.34153881 2.47311439]
[3.13307241 4.2486247]
[2.35244208 4.8051259]
[4.61446814 3.74614171]
[3.82434023 4.33027249]
[3.42670212 3.07947486]
[2.41688067 4.77655656]
[2.09213452 5.83065929]
[2.52633631 3.30050546]
[1.98433543 4.0591948]
[3.19426022 4.64349128]
[2.74785473 2.95689444]
[4.68630286 2.65347184]
[2.45409694 3.62006999]
[3.9820543 2.87160599]
[3.10515765 4.61927152]
[3.40675916 4.07355771]
[0.9245533 5.63827035]
[2.28599458 3.42317721]
[4.53695806 3.16695962]
[2.54380538 4.69791807]
[3.25612716 3.35591376]
[3.28681523 3.46930694]
[4.09065723 2.21970997]
[3.65187493 3.71649865]
[3.29565695 4.44706088]
[3.91887995 4.57260046]
[1.94345872 4.85908464]
[3.07424214 5.0737755]
[2.18661739 3.91533977]
[4.44060592 4.34440637]
[3.5954026 4.95697895]
[0.5073436 5.9192382]
[1.78753853 5.40214723]
[1.65929124 4.4842585]
[3.61704845 4.0744272]
[1.81507175 3.96774273]
[1.89446836 5.51508261]
[0.47337102 5.01095642]
[1.87856262 6.26904634]
[2.82616048 5.059888]
[3.93005367 3.31479496]
[2.71837115 4.03463865]
[2.85523865 5.08957878]
[1.91384322 3.86307133]
[2.37340369 4.28116486]
[2.90766062 3.47571456]
[3.10693108 3.53732794]
[1.89776309 5.24708035]
[3.36805877 4.00556566]
[3.83947344 4.29243789]
[4.53045005 4.19310798]
[2.68167384 4.28752483]
[4.60897081 4.83370428]
[3.19558005 4.01577137]
[3.22978254 4.23644138]
[2.73793138 3.53656827]
[1.91524776 2.77081231]
[2.8647964 4.25691454]
[3.44764816 4.92815973]
[4.51019443 3.97253071]
[2.46086159 5.43412805]
[3.64244117 3.09539143]
[3.56701463 5.52961916]
[2.65633903 3.60123719]
[2.3126555 3.09748753]
[2.11295998 4.62970594]
[3.83530819 4.53775661]
[2.71949043 3.52554515]
[1.23563428 3.74363363]
[1.94803141 5.71886393]
[3.05150737 4.14836328]
[1.74035573 4.00355824]

[1.74999975 4.90999024]
[2.79958107 4.26399162]
[3.13871825 3.59740888]
[3.55801893 3.8855882]
[4.19525335 3.05764797]
[1.78582719 5.76128228]
[3.11708136 3.75679897]
[2.53555333 4.61192984]
[3.63713134 4.02121006]
[4.41009022 3.88537402]
[3.25240922 2.69860686]
[3.21587349 2.74229222]
[3.92912952 2.03379345]
[1.89903783 4.28213808]
[3.02993506 3.88041226]
[2.04515476 3.79421602]
[3.31430383 3.26783909]
[4.2635822 3.80169979]
[2.75595723 3.85442476]
[1.98737285 5.24857102]
[4.6949494 2.37725133]
[2.31702188 5.21359801]
[1.97018475 4.18795754]
[2.69550697 2.54698439]
[1.53477598 4.4978563]
[2.59874804 4.18719557]
[3.32667479 3.8889612]
[2.61713671 3.95981993]
[2.92033619 3.37746737]
[2.93282541 4.75517339]
[1.61310107 6.63213118]
[2.90293604 4.59399736]
[4.02871472 1.93268448]
[2.38771937 4.39591838]
[3.13162922 3.51513495]
[5.39435184 1.71535659]
[1.71785836 4.8777556]
[2.56867735 4.17969623]
[3.48002213 2.29931293]
[5.72949356 3.12872352]
[0.43581707 6.15985984]
[1.77674247 5.05791677]
[2.53155437 4.12932503]
[2.52825028 3.89379843]
[2.72292452 5.01829922]
[2.20844219 5.34839973]
[3.42296123 3.9384843]
[2.00120285 3.11937089]
[3.20519271 3.89155738]
[3.82712033 4.12351604]
[3.0858227 4.86586578]
[2.75143147 3.74692602]
[4.50376468 3.40432672]
[3.39733137 3.75755001]
[3.41794219 3.51911197]
[2.23896043 4.00949209]
[2.9659104 4.13566235]
[2.7041812 4.32105397]
[3.76579189 4.2561859]
[3.41383737 4.26329471]
[1.44829095 5.99609402]
[2.23539443 4.66555364]
[3.62699991 3.91953469]
[2.0024338 5.45345399]
[2.74657148 3.77761325]
[2.16214312 4.00014427]
[3.84711149 5.19598101]
[1.02609944 6.849884]
[0.05152467 0.98010834]
[-0.96562169 1.54263818]
[-0.0384046 1.15986311]
[1.32998033 2.20492248]
[0.4267625 -0.14492706]
[-0.9082646 0.75740789]
[-0.74462846 0.94818312]
[-0.01592971 1.26267455]
[0.81631453 1.23617605]
[1.62186002 2.74220650]

[1.05100992 2.14220059]
[1.72932629 2.0364049]
[1.41047037 1.85809335]
[2.03635941 2.08203451]
[-0.17794765 1.80180037]
[0.04376139 0.50793096]
[0.05034174 1.85969371]
[0.84181464 1.60221905]
[0.64295582 1.06322825]
[-0.35719678 0.75745283]
[-0.30936652 0.40486153]
[0.84452019 1.61067407]
[-0.0208738 0.70719916]
[-0.20977721 0.92442059]
[-1.26560254 0.74057482]
[-1.09435589 0.94227959]
[-0.84414463 1.4418423]
[0.44457809 1.89366764]
[-1.19368086 -0.01954525]
[-0.26580221 0.12732931]
[-0.90744502 0.98050483]
[-0.19245619 1.36032203]
[-0.9787596 1.264338]
[0.22287096 1.73082834]
[1.124024 1.80038445]
[-1.28371794 0.1340895]
[0.39200149 0.8244133]
[0.17172657 2.18030806]
[-0.07031094 0.62592476]
[0.16453158 1.85817466]
[-0.70026822 0.04516639]
[-0.51948285 1.31735576]
[-1.60922114 0.69678423]
[1.16824915 1.29119004]
[0.69801618 1.83576263]
[-1.31942684 0.32722143]
[0.4974999 0.83896415]
[-0.37758569 1.37916011]
[1.97232367 1.30481967]
[1.15897124 1.35398136]
[0.03137787 1.21879161]
[-0.1663347 0.26456566]
[-1.27632312 1.03825197]
[1.98442349 2.19557353]
[0.86333341 0.70463669]
[-0.78290686 0.64929574]
[0.68457051 1.51688068]
[0.19990872 0.32525088]
[0.52010418 1.60027593]
[1.03314409 2.53409433]
[-1.75242778 1.12996124]
[-1.25596728 1.89391368]
[1.46417625 0.40852313]
[-0.88561866 0.67429504]
[-1.54315458 0.33101669]
[0.51586645 0.97712412]
[0.20244867 1.15297393]
[0.5945683 1.4484356]
[-1.52070206 -0.03071408]
[0.79966805 2.28957314]
[0.19383241 1.10847749]
[0.09511323 0.72216935]
[-1.83491187 0.58993401]
[-2.26075649 0.4748583]
[-0.61614867 0.51332299]
[0.81072247 1.38205995]
[-0.40825189 0.69247803]
[0.97470632 1.69916866]
[-0.64488028 0.13950991]
[-0.53576697 -0.42725726]
[-0.44673948 0.39442313]
[0.35313046 2.58971674]
[0.52991511 0.73730215]
[-0.68851096 0.44992686]
[-0.43366679 -0.03749477]
[-1.21056762 -0.3103894]
[-0.16276547 0.3482789]
[0.63860610 0.10570076]

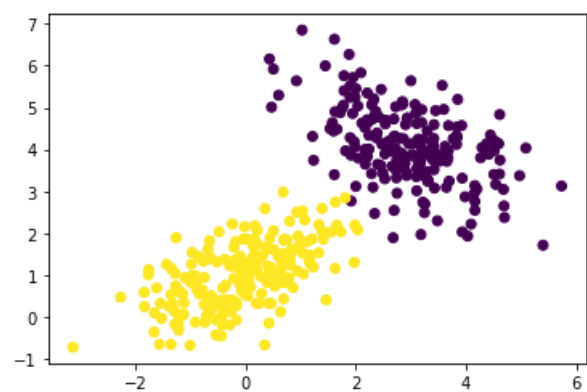
[0.63869619 0.12579976]
[-1.11418101 0.34021499]
[-0.29015181 -0.19005801]
[0.43666721 1.05481581]
[-0.42100349 -0.19149351]
[-0.99230651 0.83544424]
[-0.17874325 1.20829439]
[-0.88861105 0.78937675]
[0.16099552 1.50604104]
[-1.27588892 -0.16940926]
[-0.73552512 -0.12297833]
[-3.1271992 -0.72069566]
[-0.11784564 0.95187713]
[0.56485281 0.90053776]
[0.19901465 1.32580617]
[-0.44147835 0.0083557]
[-1.22127101 0.65153281]
[0.75670456 1.84559747]
[-0.03292948 0.72625097]
[0.4262373 0.32664654]
[-0.66623248 1.38178742]
[1.02802203 1.5292826]
[0.81145086 1.36469983]
[0.74373034 0.8309087]
[0.30999438 1.4711523]
[1.81019139 2.84678918]
[-0.35310715 0.40779954]
[-0.01384928 2.15466352]
[0.43571474 1.64393272]
[0.69174106 0.43429393]
[0.70047505 1.85416258]
[0.46383346 0.32726873]
[0.40644626 1.6764517]
[0.95578276 1.93198428]
[-1.55657749 -0.64968064]
[-1.34096842 1.15580704]
[-0.22928052 0.40511038]
[-0.09663749 1.45001232]
[0.05786182 0.90537313]
[-0.90663136 -0.12080377]
[0.14170214 1.71040743]
[1.65935986 2.19452497]
[-0.47848898 2.03534337]
[0.26187086 0.3389718]
[-1.82889201 0.24918387]
[-0.02490438 0.75287958]
[0.65199215 1.34598421]
[1.0509822 1.02916502]
[0.21453424 2.07330323]
[-0.0349242 1.86136848]
[-1.64871436 0.09975032]
[0.14652598 1.43852602]
[-1.42779987 1.26024357]
[1.20998502 1.86706983]
[1.54807134 2.03164728]
[-1.21477856 -0.20507726]
[1.62625039 1.16129631]
[0.05550105 1.00897311]
[0.40730874 1.21706458]
[-0.09475031 0.78098366]
[-1.38526127 0.25633006]
[0.28422973 0.96592514]
[1.07329112 2.31620201]
[-0.16278352 -0.06822914]
[0.37875618 1.67224333]
[-0.76031838 1.33444967]
[-0.53872398 -0.06106671]
[-1.3480613 0.5391807]
[0.8493378 1.5847846]
[-0.84164408 0.56155873]
[-0.65324807 1.81842374]
[0.39442622 1.62842925]
[0.66997303 0.82487352]
[-0.44509125 0.22864458]
[1.32759496 1.53546323]
[-0.43337466 -0.45197962]
[1.16421902 1.4844649]
[0.46506046 1.10054400]

```
[ 0.46526246  1.18254402]
[-0.23569548  2.22208534]
[-1.65457845 -0.35577968]
[ 0.3508647  -0.66509032]
[ 0.93441431  2.50514637]
[-1.13721476  0.07156738]
[ 0.5582984   0.93354887]
[-0.67741244  1.61881671]
[-1.3477257  -0.6515689 ]
[ 0.01861143  1.30514042]
[ 1.22908985  1.19444233]
[-0.63805939  1.41888598]
[ 0.50820377  2.06225317]
[-0.20977065  1.66794175]
[-1.14772736  0.56508936]
[-0.96111709  0.9932576 ]
[ 0.68478417  2.98360643]
[ 0.65035689  1.15899473]
[-1.7580855   1.00625568]
[ 0.26412506  1.39896241]
[-0.2853563   1.22521862]
[ 1.386236    2.04631046]
[-0.1003743   1.19791665]
[-0.56893866  0.36345923]
[-0.74823522  0.30166248]
[ 0.94091117  1.10141461]
[-1.14704367  0.55017264]
[-0.5628373   0.27017284]
[ 0.00717591  0.53080432]
[-1.31976544  0.21188179]
[ 0.16843331  1.335027 ]
[-1.12207722  0.05917188]
[ 1.38923089  2.59258407]
[-0.35276782  0.13646095]
[ 0.78962568  0.93166981]
[-1.00253042 -0.67616247]
[ 0.29238678  0.56117782]]
```

```
(399, 2)
```

```
(399,)
```

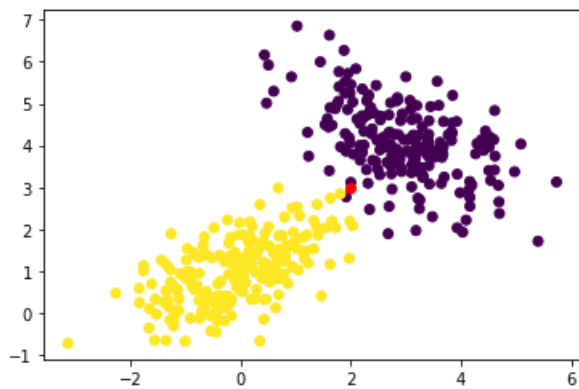
```
plt.scatter(X[:,0],X[:,1],c=Y)
plt.show()
```



```
query_x=np.array([2,3])
plt.scatter(X[:,0],X[:,1],c=Y)
plt.scatter(query_x[0],query_x[1],color="red")
plt.show()
```

In [9]:

In [10]:



In [25]:

```
def dist(x1,x2):
    return np.sqrt(sum((x1-x2)**2))

def knn(X,Y,queryPoint,k=5):

    vals=[]

    m=X.shape[0]

    for i in range(m):
        d=dist(queryPoint,X[i])
        vals.append((d,Y[i]))

    vals=sorted(vals)
    vals=vals[:k]

    vals=np.array(vals)

    # print(vals)

    new_vals=np.unique(vals[:,1],return_counts=True)
    print(new_vals)

    # index=new_vals[1].argmax()
    # pred=new_vals[0][index]

    return vals
```

In [26]:

```
knn(X,Y,query_x)

(array([0., 1.]), array([3, 2], dtype=int64))
```

Out[26]:

```
array([[0.11937695, 0.      ],
       [0.24392799, 1.      ],
       [0.24435617, 0.      ],
       [0.32750158, 0.      ],
       [0.44941874, 1.      ]])
```

MNIST DataSet

In [27]:

```
df=pd.read_csv("train.csv")
print(df.shape)
```

```
(42000, 785)
```

In [28]:

```
print(df.columns)
```

```
Index(['label', 'pixel0', 'pixel1', 'pixel2', 'pixel3', 'pixel4', 'pixel5',
       'pixel6', 'pixel7', 'pixel8',
       ...,
       'pixel774', 'pixel775', 'pixel776', 'pixel777', 'pixel778', 'pixel779',
       'pixel780', 'pixel781', 'pixel782', 'pixel783'],
      dtype='object', length=785)
```

In [29]:

```
df.head()
```

Out[29]:

	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780
0	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	(
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	(
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	(
3	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	(
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	(

5 rows × 785 columns



Create a Numpy Array

In [30]:

```
data=df.values
print(data.shape)
print(type(data))

(42000, 785)
<class 'numpy.ndarray'>
```

In [31]:

```
X=data[:,1:]
Y=data[:,0]

print(X.shape,Y.shape)

(42000, 784) (42000,)
```

In [32]:

```
split=int(0.8*X.shape[0])
print(split)

33600
```

In [33]:

```
X_train=X[:split,:]
Y_train=Y[:split]
X_test=X[split:,:]
Y_test=Y[split:]

print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)

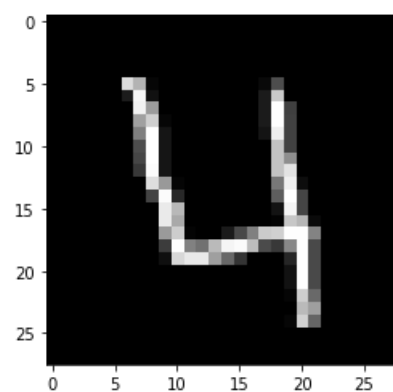
(33600, 784) (33600,)
(8400, 784) (8400,)
```

In [41]:

```
def drawImg(sample):
    img=sample.reshape((28,28))
    plt.imshow(img,cmap='gray')
    plt.show()
```

In [42]:

```
drawImg(X_train[3])
print(Y_train[3])
```

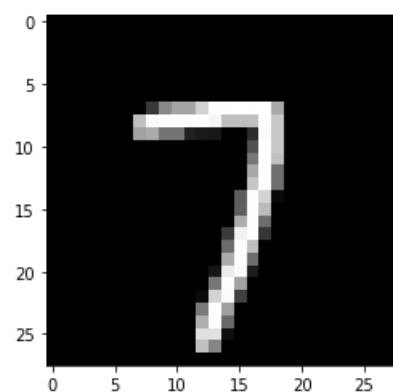
4

Making Prediction

```
pred=knn(X_train,Y_train,X_test[0])  
print(pred)
```

```
(array([0.]), array([5], dtype=int64))  
[[1213.36886395    0.        ]  
 [1235.20322215    0.        ]  
 [1328.63576649    0.        ]  
 [1400.12285175    0.        ]  
 [1411.82612244    0.        ]]
```

```
drawImg(X_test[7])  
print(Y_test[7])
```



7

In [44]:

In [45]:

In []: