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


dfx =  pd.read_csv("https://raw.githubusercontent.com/coding-blocks-archives/machir
dfy = pd.read_csv("https://raw.githubusercontent.com/coding-blocks-archives/machine
print(dfx.shape)
print(dfy.shape)
```

```
(399, 3)
(399, 2)
```

```
dfx.head(n=4)
```

	0	3.4821055700010444	2.4203917879404533
0	1	3.516279	3.719750
1	2	4.606496	3.414379
2	3	4.677011	3.047244
3	4	4.157725	3.138677

```
X = dfx.values
Y = dfy.values
print(type(X))
print(type(Y))
```

```
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

```
print(X)
```

```
[[ 1.00000000e+00  3.51627900e+00  3.71975047e+00]
 [ 2.00000000e+00  4.60649561e+00  3.41437943e+00]
 [ 3.00000000e+00  4.67701056e+00  3.04724443e+00]
 ...
 [ 3.97000000e+02  7.89625683e-01  9.31669806e-01]
 [ 3.98000000e+02 -1.00253042e+00 -6.76162472e-01]
 [ 3.99000000e+02  2.92386778e-01  5.61177823e-01]]
```

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

```
(399, 2)
(399,)
```

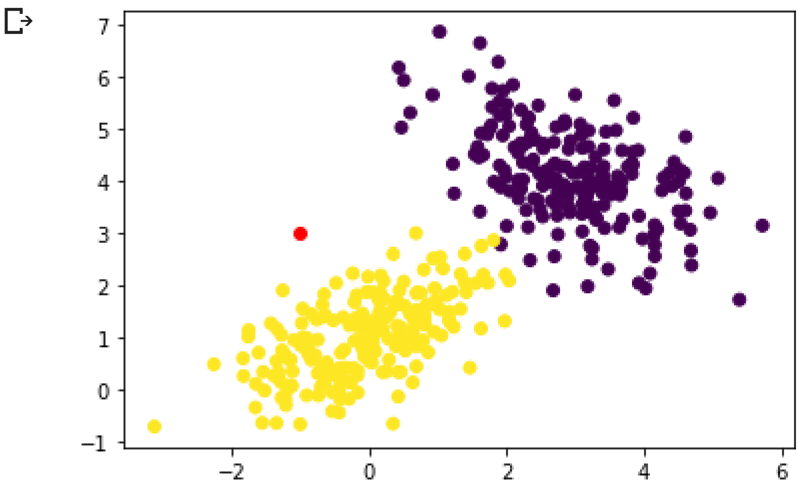
```
print(Y)
```

```
[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

```
plt.scatter?
```

```
query_pt = np.array([6,7])
```

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




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

```
def knn(X,Y,q_pt,k=5):
  vals = []
  m = X.shape[0]
  d = 0.0
  for i in range(m):
    #print(X[i])
    d = dist(X[i],q_pt)
    vals.append((d,Y[i]))
  #print(vals)
  vals = sorted(vals)
```

```
    vals = vals[:k]
    #print(vals)
    vals = np.array(vals)
    #print(vals)
    new_vals = np.unique(vals[:,1],return_counts=True)
    #print(new_vals)
    #print(new_vals[0])
    #print(new_vals[1])
    index = new_vals[1].argmax()
    return new_vals[0][index]


pred = knn(X,Y,query_pt)
print(pred)
```

```
0.0
```

```
#print(X,Y)
```