In [1]:

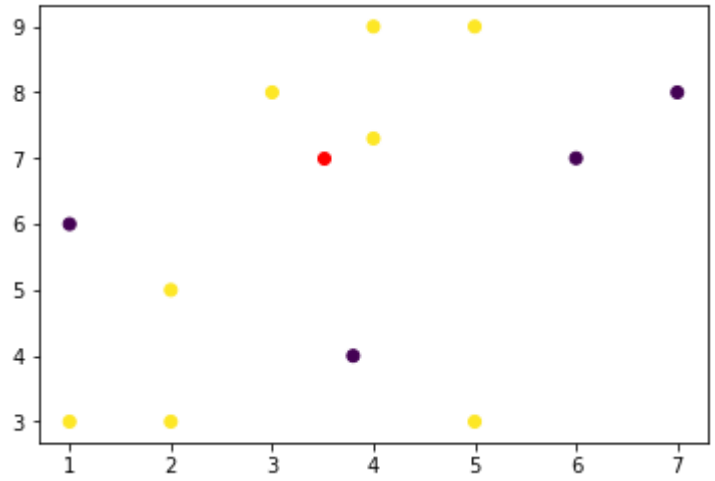
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.datasets import make_classification
```

In [2]:

```
dff = pd.DataFrame({
 'x' : [1,2,3,4,5,6,7,5,2,3.8,4,1],
 'y' : [6,3,8,9,3,7,8,9,5,4,7.3,3],
 'z' : [0,1,1,1,1,0,0,1,1,0,1,1]})
plt.scatter(dff['x'],dff['y'],c=dff['z'])
plt.scatter(3.5,7,color='r')
```

Out[2]:

```
<matplotlib.collections.PathCollection at 0xdfa8c70>
```



In [3]:

```
#(3,8) (4,9)
def ed(x1,x2,z1,z2):
    return np.sqrt((x2-x1)**2 + (z2-z1)**2)
ed(3.5,7,4,7.4)
```

Out[3]:

```
4.879549159502341
```

In [4]:

```
ed(3.5,7,3,8)
```

Out[4]:

```
6.103277807866851
```

In [5]:

```
np.argmin([4.8,6.1])
```

Out[5]:

```
0
```

Customer Purchase

In [11]:

```
df=pd.read_csv('Social_Network_Ads.csv')
df.head()
```

Out[11]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

In [12]:

```
df.drop (columns ='User ID',inplace=True)
```

In [13]:

```
from sklearn.preprocessing import LabelEncoder,StandardScaler
lb = LabelEncoder()
sc = StandardScaler()
df['Gender']=lb.fit_transform(df['Gender'])
df[['Age','EstimatedSalary']] = sc.fit_transform(df[['Age','EstimatedSalary']])
df
```

Out[13]:

	Gender	Age	EstimatedSalary	Purchased
0	1	-1.781797	-1.490046	0
1	1	-0.253587	-1.460681	0
2	0	-1.113206	-0.785290	0
3	0	-1.017692	-0.374182	0
4	1	-1.781797	0.183751	0
...	...	...	...	...
395	0	0.797057	-0.844019	1
396	1	1.274623	-1.372587	1
397	0	1.179110	-1.460681	1
398	1	-0.158074	-1.078938	0
399	0	1.083596	-0.990844	1

400 rows × 4 columns

In [14]:

```
ind = df.iloc[:,:3]
dep = df.iloc[:,-1]
dep
```

Out[14]:

```
0      0
1      0
2      0
3      0
4      0
      ..
395    1
396    1
397    1
398    0
399    1
Name: Purchased, Length: 400, dtype: int64
```

In [15]:

```
ind
```

Out[15]:

	Gender	Age	EstimatedSalary
0	1	-1.781797	-1.490046
1	1	-0.253587	-1.460681
2	0	-1.113206	-0.785290
3	0	-1.017692	-0.374182
4	1	-1.781797	0.183751
...	...	...	...
395	0	0.797057	-0.844019
396	1	1.274623	-1.372587
397	0	1.179110	-1.460681
398	1	-0.158074	-1.078938
399	0	1.083596	-0.990844

400 rows × 3 columns

In [16]:

```
xx = df.iloc
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(ind,dep,test_size=0.2,random_state=0)
```

In [17]:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=2)
knn.fit(x_train,y_train)
```

Out[17]:

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=2, p=2,
           weights='uniform')
```

In [18]:

```
testing = pd.DataFrame({
 'y' : y_test,
 'y_hat': knn.predict(x_test)

})
testing.sample(8)
```

Out[18]:

	y	y_hat
303	1	0
141	0	0
173	0	0
309	0	0
247	1	1
294	0	0
286	0	0
196	0	0

In [19]:

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,knn.predict(x_test))
```

Out[19]:

```
array([[55,  3],
       [ 3, 19]], dtype=int64)
```

In [20]:

```
accuracy = (55+19) / (55+3+3+19)
print(accuracy)
```

```
0.925
```

In [ ]:

```

```