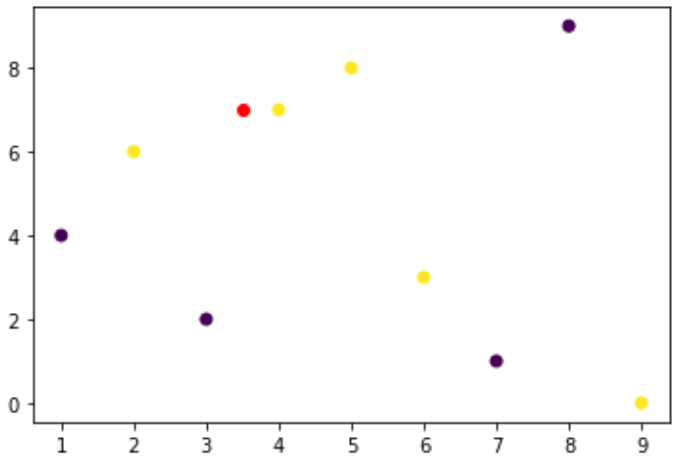
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from sklearn.datasets import make_classification
```

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

Out[2]: <matplotlib.collections.PathCollection at 0x26b04541a90>



In [3]:
```
def ed(x1,y1,x2,y2):
    return np.sqrt(((x2-x1)**2)+(y2-y1)**2)
```

In [4]:
```
ed(3.5,7,5,8)
```

Out[4]: 1.8027756377319946

In [5]:
```
ed(3.5,7,4,7)
```

Out[5]: 0.5

In [6]:
```
np.argmin([1.8,0.5])
```

Out[6]: 1

In [7]:
```
dff=pd.read_csv('Social_Network_Ads.csv')
dff.head()
```

Out[7]:

	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 [8]:
```
dff.drop(columns='User ID',inplace=True)
```

In [9]:
```
from sklearn.preprocessing import LabelEncoder,StandardScaler
lb=LabelEncoder()
```

In [10]:
```
sc=StandardScaler()
```

In [11]:
```
dff['Gender']=lb.fit_transform(dff['Gender'])
dff[['Age','EstimatedSalary']]=sc.fit_transform(dff[['Age','EstimatedSalary']])
```

In [12]:
```
dff
```

Out[12]:

	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 [13]:
```
ind=dff.iloc[:,:3]
dep=dff.iloc[:,-1]
```

In [14]:
```
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(n_neighbors=2)
```

In [18]:
```
testing = pd.DataFrame({
    'y' : y_test,
    'y_hat' : knn.predict(x_test)
})
testing.sample(8)
```

Out[18]:

	y	y_hat
154	0	0
303	1	0
4	0	0
102	0	0
214	0	1
310	0	0
76	0	0
14	0	0

In [19]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
confusion_matrix(y_test,knn.predict(x_test))
```

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

In [20]:
```
accuracy_score(y_test,knn.predict(x_test))
```

Out[20]:
```
0.925
```

In [ ]: