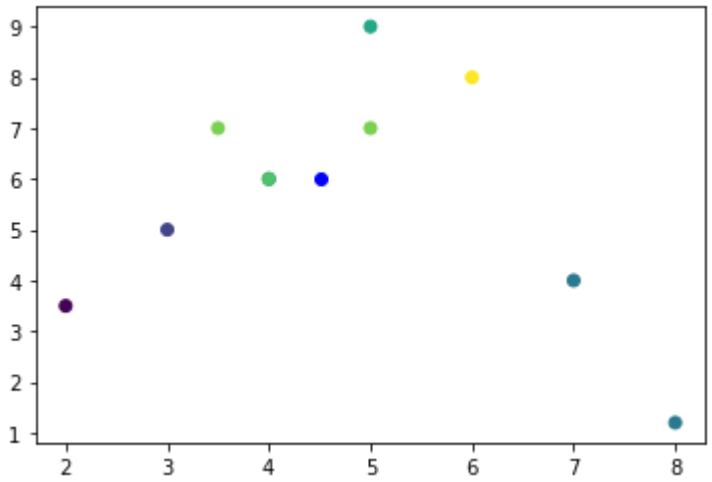
In [1]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.datasets import make_classification
```

In [2]:
```
dff = pd.DataFrame({
    'x': [2,3,4,5,6,7,5,3.5,4,8],
    'y': [3.5,5,6,7,8,4,9,7,6,1.2],
    'z': [3,4,5,7,8,5,6,7,6.6,5]
})
plt.scatter(dff['x'],dff['y'],c=dff['z'])
plt.scatter(4.5,6,color='blue')
```

Out[2]: <matplotlib.collections.PathCollection at 0x1d19f0a2df0>



In [3]:
```
def ed(x1,x2,z1,z2):
    return np.sqrt((x2-x1)**2 + (z2-z1)**2)
ed(3.4,5,7,9)
```

Out[3]: 2.5612496949731396

In [4]:
```
ed(3.2,4,5,3)
```

Out[4]: 2.1540659228538015

In [5]:
```
np.argmin([4,5,6,7])
```

Out[5]: 0

Customer Purchase

In [6]:
```
df = pd.read_csv('Social_Network_Ads .csv')
df
```

Out[6]:

	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
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

In [7]:
```
df.drop(columns='User ID',inplace=True)
```

In [8]:
```
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']]
```

In [9]:
```
df
```

Out[9]:

	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 [10]:
```
ind = df.iloc[:,:3]
dep = df.iloc[:,-1]
dep
```

Out[10]:
```
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 [11]:
```
ind
```

Out[11]:

	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 [12]:
```
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_
```

In [13]:
```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=2)
knn.fit(x_train,y_train)
```

Out[13]:
```
KNeighborsClassifier(n_neighbors=2)
```

In [14]:
```
testing = pd.DataFrame({
    'y':y_test,
    'y_hat': knn.predict(x_test)
})
testing.sample(8)
```

Out[14]:

	y	y_hat
191	0	0
363	0	0
175	0	0
361	1	1
154	0	0
59	0	1
399	1	1
313	1	1

In [16]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test,knn.predict(x_test))
```

Out[16]:
```
array([[55,  3],
       [ 3, 19]], dtype=int64)
```

In [21]:
```
Accuracy = ((55+19)/(55+3+3+19))
```

In [22]:
```
Accuracy
```

Out[22]:
```
0.925
```

In [ ]:
```

```