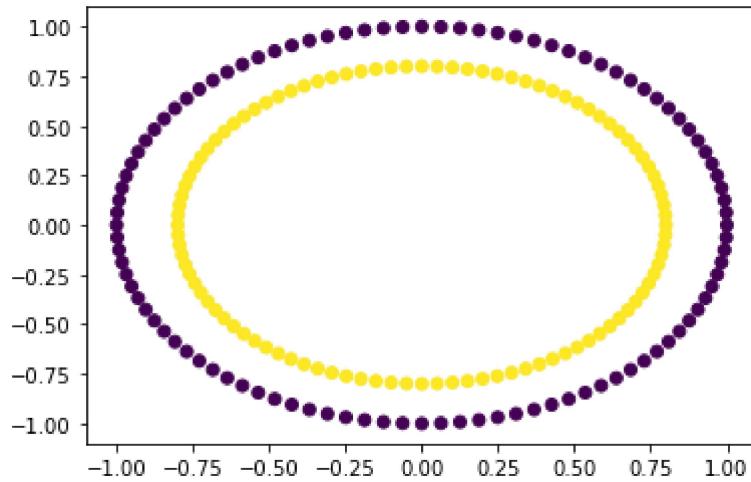


In [5]:

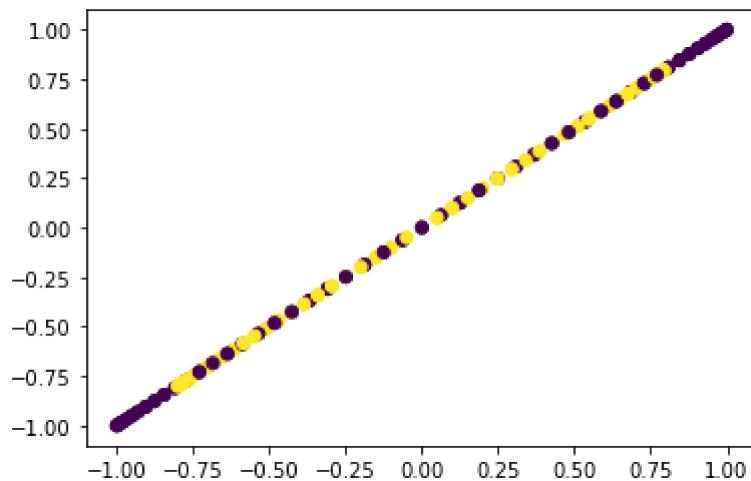
```
# Import necessary libraries
from sklearn.datasets import make_circles
import matplotlib.pyplot as plt

# Generate 2d classification dataset
X, y = make_circles(n_samples=200, shuffle=True,
noise=0, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



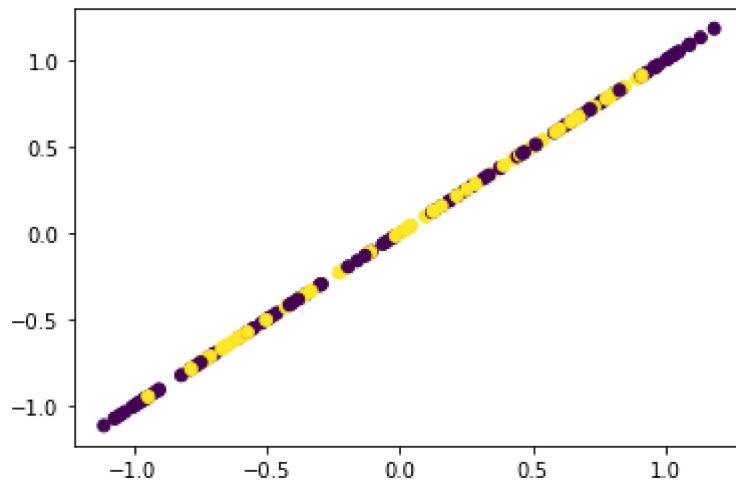
In [6]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=200, shuffle=True,
noise=0, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



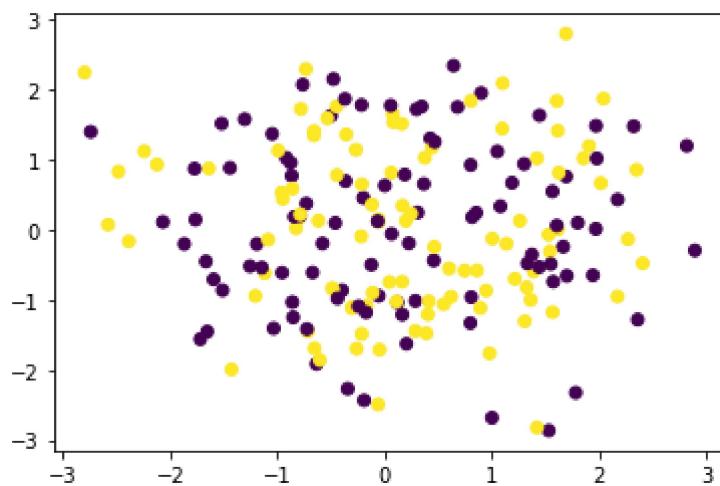
In [7]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=200, shuffle=True,
noise=0.1, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



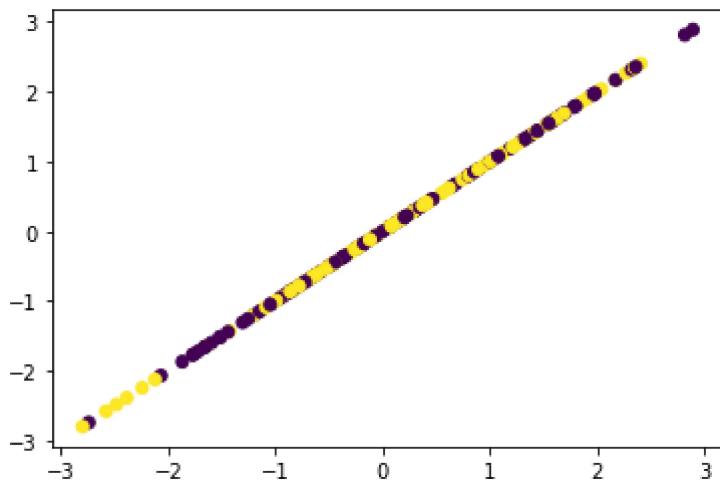
In [10]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=200, shuffle=True,
noise=1, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 1], X[:, 0], c=y)
plt.show()
```



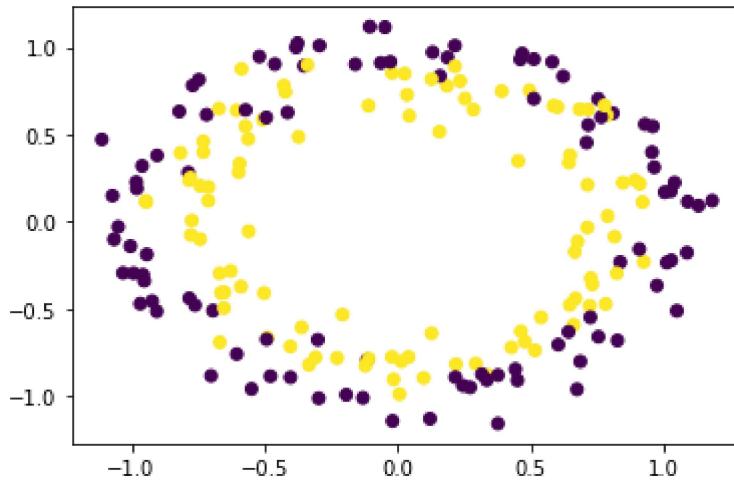
In [11]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=200, shuffle=True,
noise=1, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 1], X[:, 0], c=y)
plt.show()
```



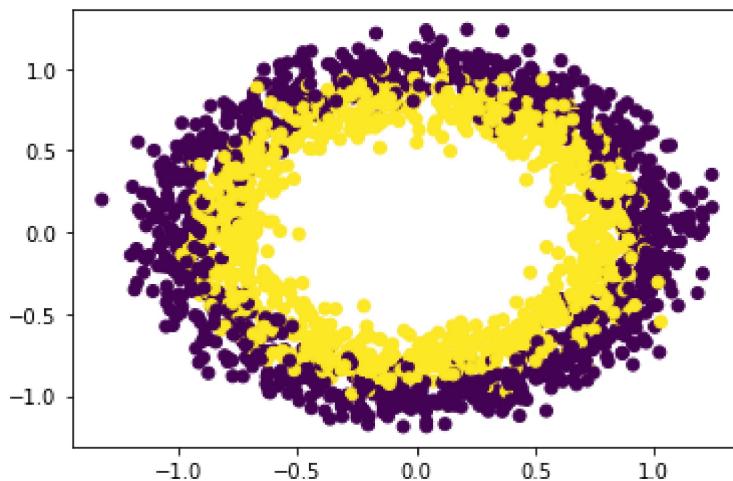
In [12]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=200, shuffle=True,
noise=0.1, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 1], X[:, 0], c=y)
plt.show()
```



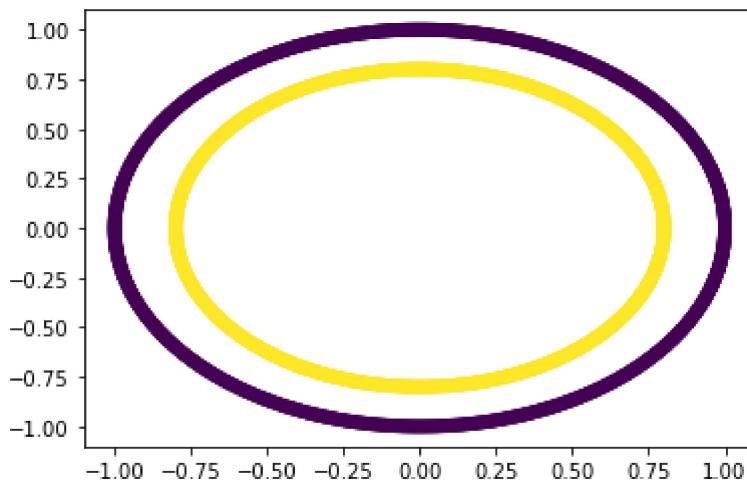
In [14]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=2100, shuffle=True,
noise=0, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 1], X[:, 0], c=y)
plt.show()
```



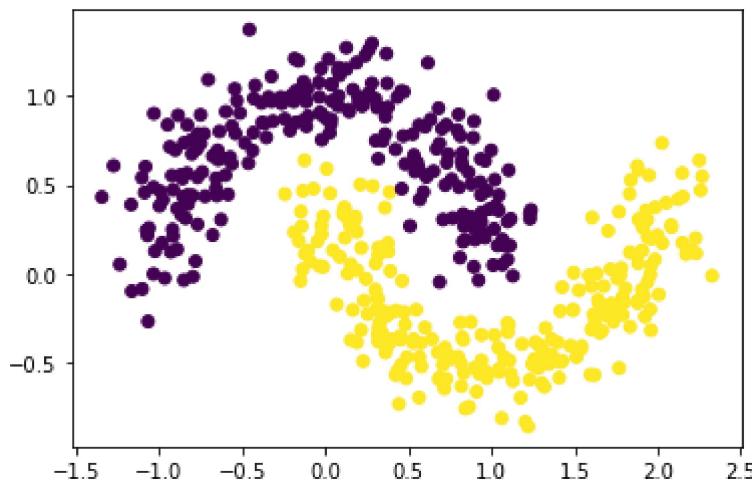
In [15]:

```
# Generate 2d classification dataset
X, y = make_circles(n_samples=2100, shuffle=True,
noise=0, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 1], X[:, 0], c=y)
plt.show()
```



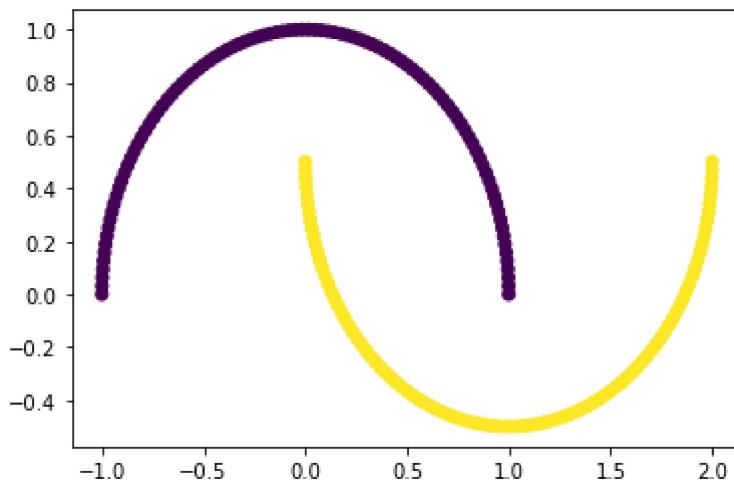
In [16]:

```
#import the necessary libraries
from sklearn.datasets import make_moons
import matplotlib.pyplot as plt
# generate 2d classification dataset
X, y = make_moons(n_samples=500, shuffle=True,
                   noise=0.15, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



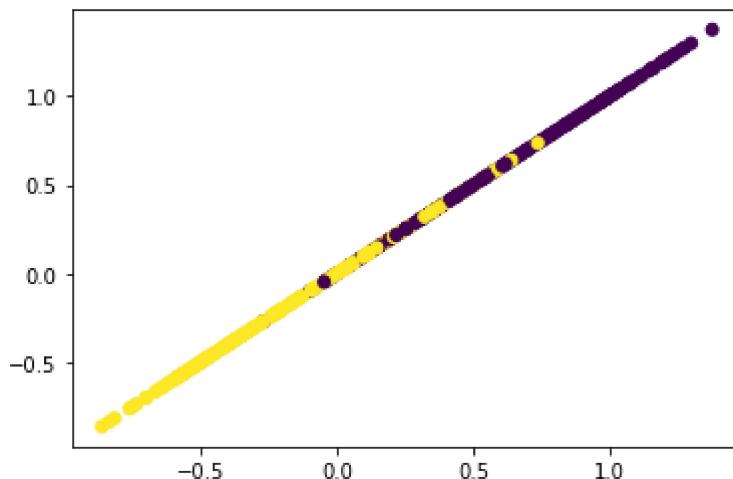
In [18]:

```
# generate 2d classification dataset
X, y = make_moons(n_samples=200, shuffle=True,
noise=0, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



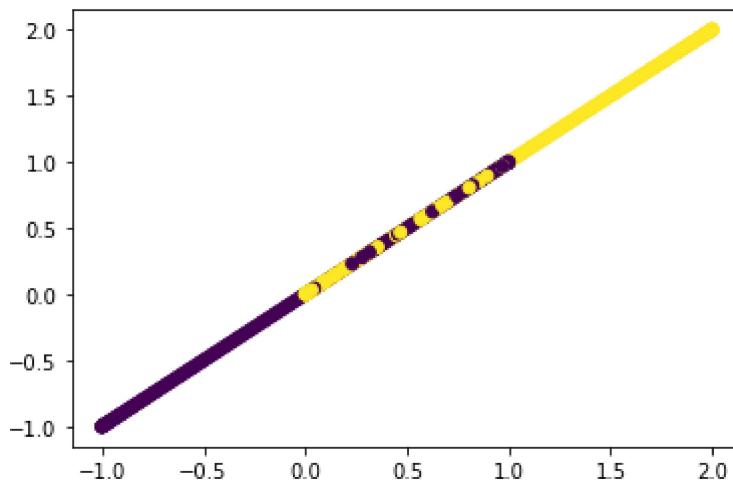
In [19]:

```
# generate 2d classification dataset
X, y = make_moons(n_samples=500, shuffle=True,
noise=0.15, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 1], X[:, 1], c=y)
plt.show()
```



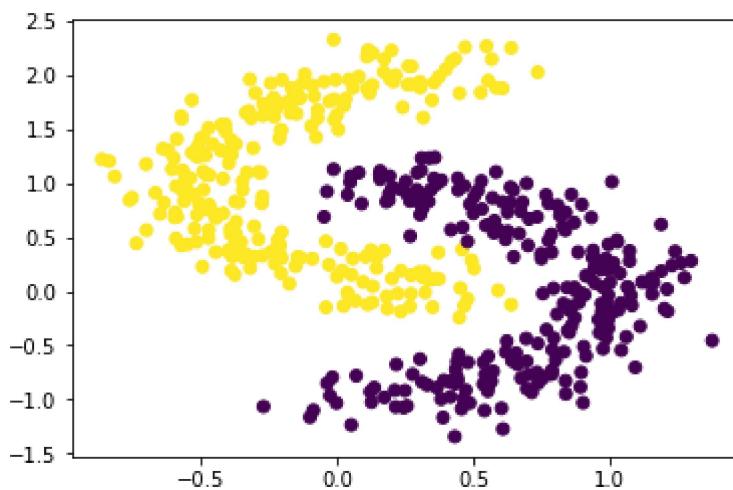
In [20]:

```
# generate 2d classification dataset
X, y = make_moons(n_samples=500, shuffle=True,
noise=0, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [22]:

```
# generate 2d classification dataset
X, y = make_moons(n_samples=500, shuffle=True,
noise=0.15, random_state=42)
# Plot the generated datasets
plt.scatter(X[:, -1], X[:, -0], c=y)
plt.show()
```

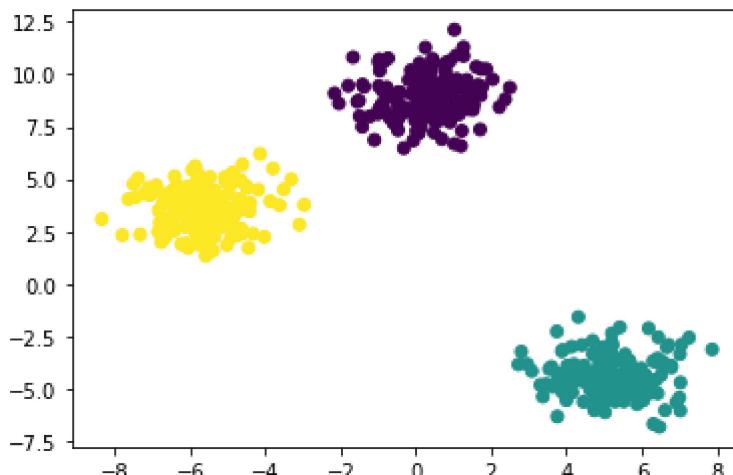


In [23]:

```
#import the necessary Libraries
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt

# Generate 2d classification dataset
X, y = make_blobs(n_samples=500, centers=3, n_features=2, random_state=23)

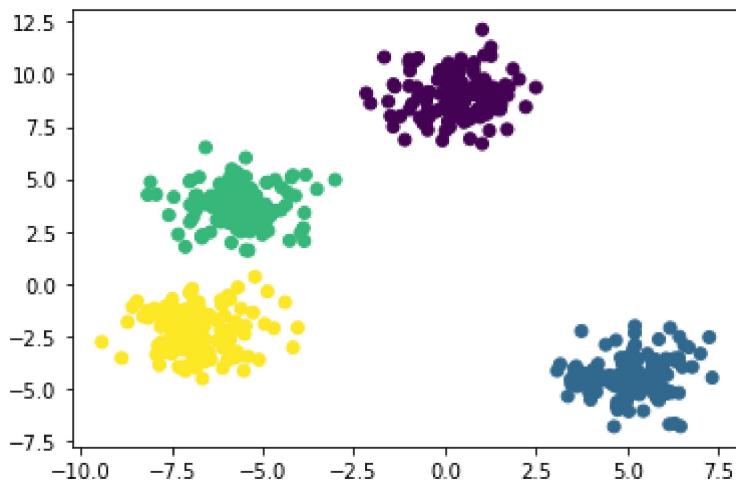
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [24]:

```
# Generate 2d classification dataset
X, y = make_blobs(n_samples=500, centers=4, n_features=2, random_state=23)

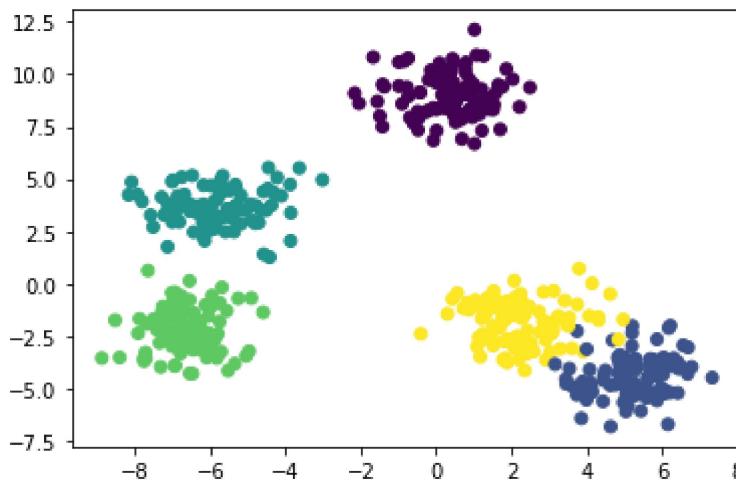
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [25]:

```
# Generate 2d classification dataset
X, y = make_blobs(n_samples=500, centers=5, n_features=2, random_state=23)

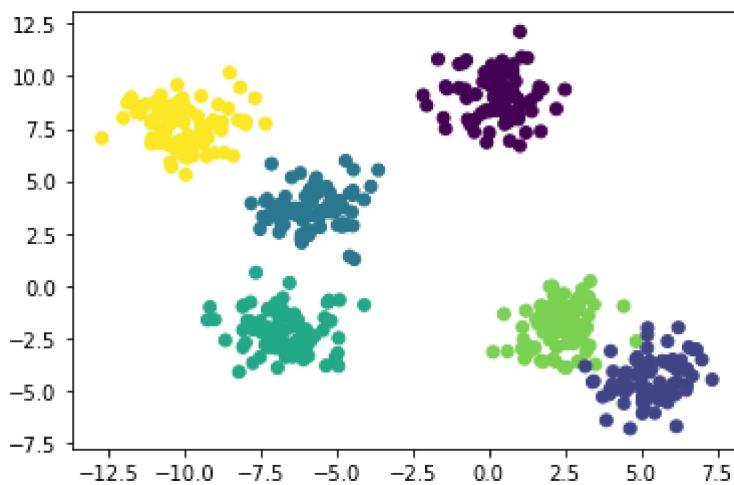
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [26]:

```
# Generate 2d classification dataset
X, y = make_blobs(n_samples=500, centers=6, n_features=2, random_state=23)

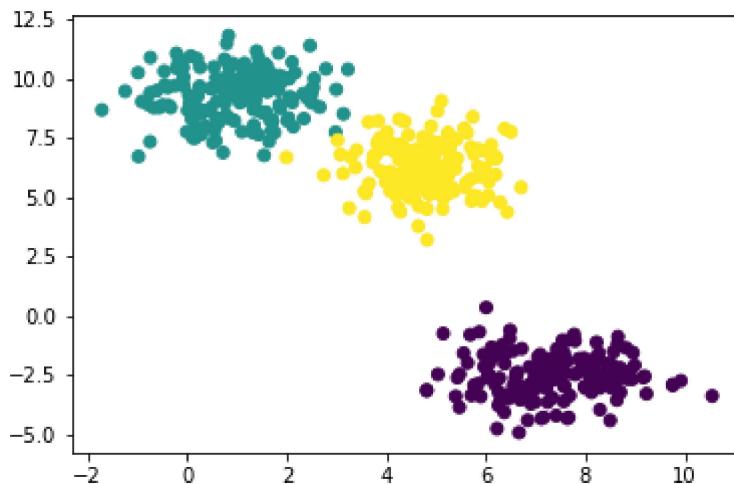
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [28]:

```
# Generate 2d classification dataset
X, y = make_blobs(n_samples=500, centers=3, n_features=2, random_state=32)

# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```

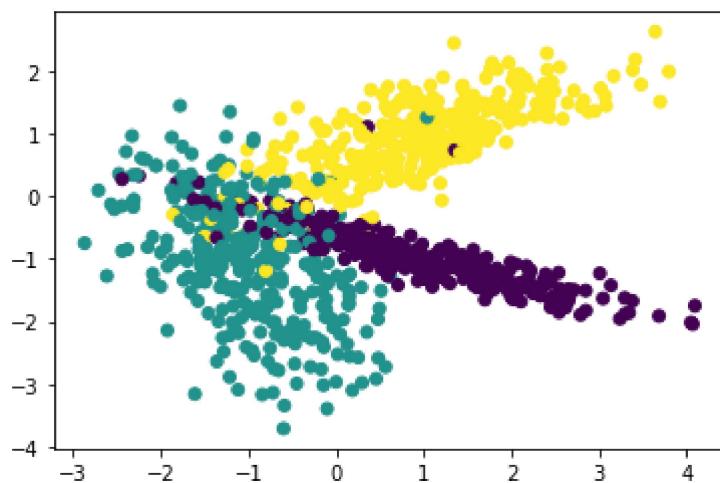


In [36]:

```
#import the necessary libraries
from sklearn.datasets import make_classification
import matplotlib.pyplot as plt

# generate 2d classification dataset
X, y = make_classification(n_samples = 1000,
n_features=2,
n_redundant=0,
n_informative=2,
n_repeated=0,
n_classes = 3,
n_clusters_per_class=1)

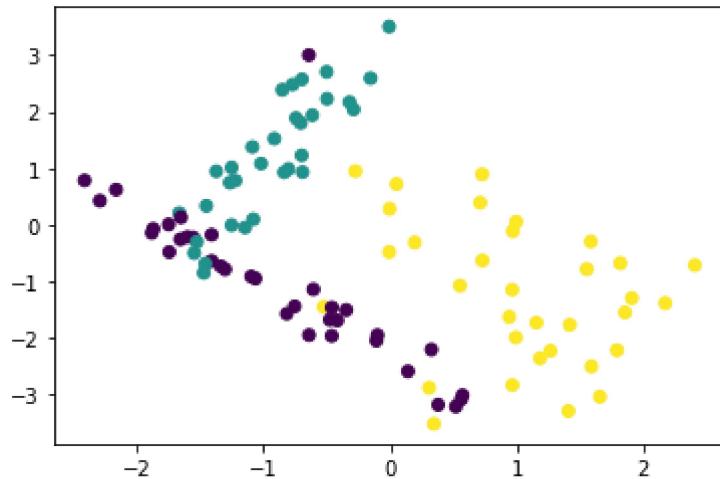
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [38]:

```
# generate 2d classification dataset
X, y = make_classification(n_samples = 100,
n_features=2,
n_redundant=0,
n_informative=2,
n_repeated=0,
n_classes = 3,
n_clusters_per_class=1)

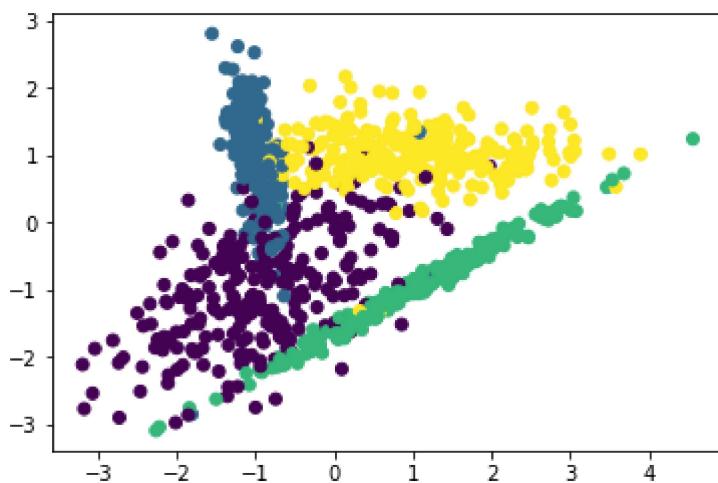
# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [40]:

```
# generate 2d classification dataset
X, y = make_classification(n_samples = 1000,
n_features=2,
n_redundant=0,
n_informative=2,
n_repeated=0,
n_classes = 4,
n_clusters_per_class=1)

# Plot the generated datasets
plt.scatter(X[:, 0], X[:, 1], c=y)
plt.show()
```



In [41]:

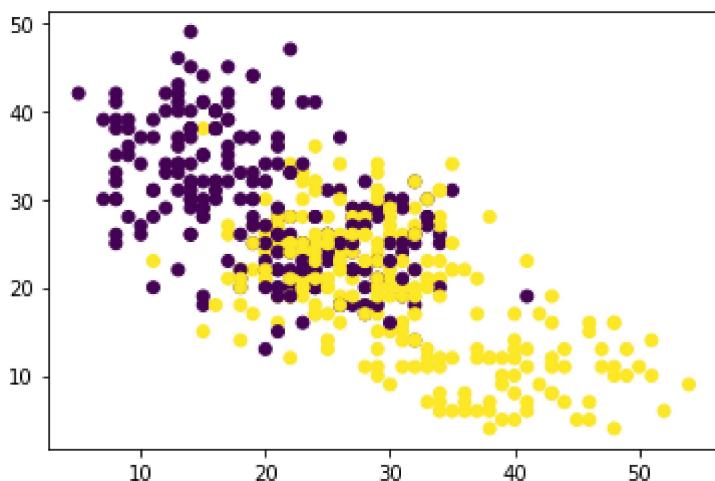
```
# Import necessary libraries
from sklearn.datasets import make_multilabel_classification
import pandas as pd
import matplotlib.pyplot as plt

# Generate 2d classification dataset
X, y = make_multilabel_classification(n_samples=500, n_features=2,
                                       n_classes=2, n_
                                       allow_unlabeled
                                       random_state=23)

# create pandas dataframe from generated dataset
df = pd.concat([pd.DataFrame(X, columns=['X1', 'X2']),
                pd.DataFrame(y, columns=['Label1', 'Label2'])],
               axis=1)
display(df.head())

# Plot the generated datasets
plt.scatter(df['X1'], df['X2'], c=df['Label1'])
plt.show()
```

| | X1 | X2 | Label1 | Label2 |
|---|------|------|--------|--------|
| 0 | 14.0 | 34.0 | 0 | 1 |
| 1 | 30.0 | 22.0 | 1 | 1 |
| 2 | 29.0 | 19.0 | 1 | 1 |
| 3 | 21.0 | 19.0 | 1 | 1 |
| 4 | 16.0 | 32.0 | 0 | 1 |



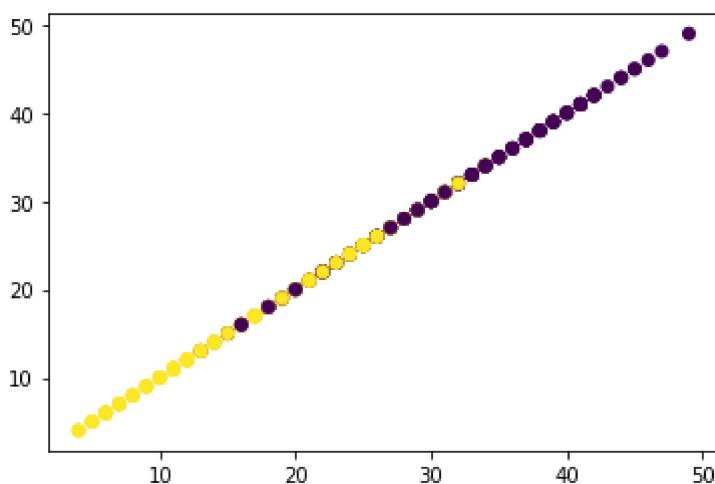
In [42]:

```
# Generate 2d classification dataset
X, y = make_multilabel_classification(n_samples=500, n_features=2,
                                         n_classes=2, n_
                                         allow_unlabeled
                                         random_state=23)

# create pandas dataframe from generated dataset
df = pd.concat([pd.DataFrame(X, columns=['X1', 'X2']),
                 pd.DataFrame(y, columns=['Label1', 'Label2'])],
                axis=1)
display(df.head())

# Plot the generated datasets
plt.scatter(df['X2'], df['X2'], c=df['Label1'])
plt.show()
```

| | X1 | X2 | Label1 | Label2 |
|---|------|------|--------|--------|
| 0 | 14.0 | 34.0 | 0 | 1 |
| 1 | 30.0 | 22.0 | 1 | 1 |
| 2 | 29.0 | 19.0 | 1 | 1 |
| 3 | 21.0 | 19.0 | 1 | 1 |
| 4 | 16.0 | 32.0 | 0 | 1 |



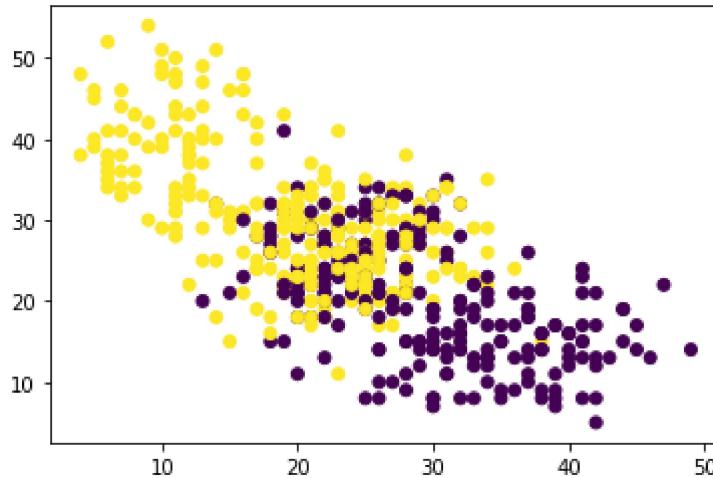
In [43]:

```
# Generate 2d classification dataset
X, y = make_multilabel_classification(n_samples=500, n_features=2,
                                         n_classes=2, n_
                                         allow_unlabeled
                                         random_state=23

# create pandas dataframe from generated dataset
df = pd.concat([pd.DataFrame(X, columns=['X1', 'X2']),
                pd.DataFrame(y, columns=['Label1', 'Label2'])],
               axis=1)
display(df.head())

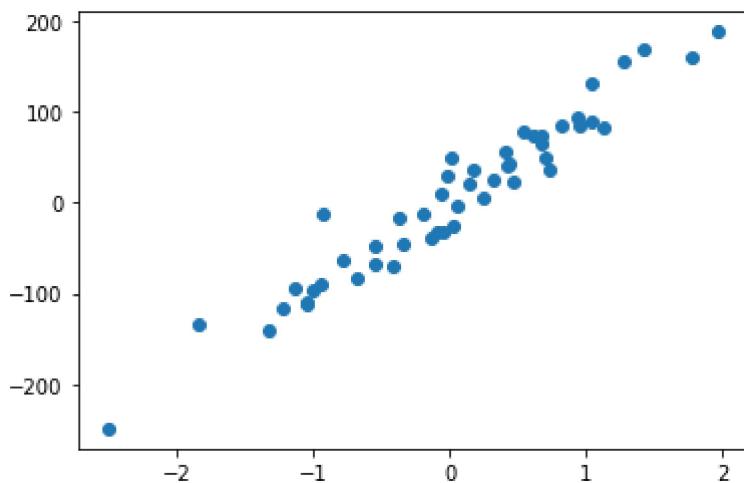
# Plot the generated datasets
plt.scatter(df['X2'], df['X1'], c=df['Label1'])
plt.show()
```

| | X1 | X2 | Label1 | Label2 |
|---|------|------|--------|--------|
| 0 | 14.0 | 34.0 | 0 | 1 |
| 1 | 30.0 | 22.0 | 1 | 1 |
| 2 | 29.0 | 19.0 | 1 | 1 |
| 3 | 21.0 | 19.0 | 1 | 1 |
| 4 | 16.0 | 32.0 | 0 | 1 |



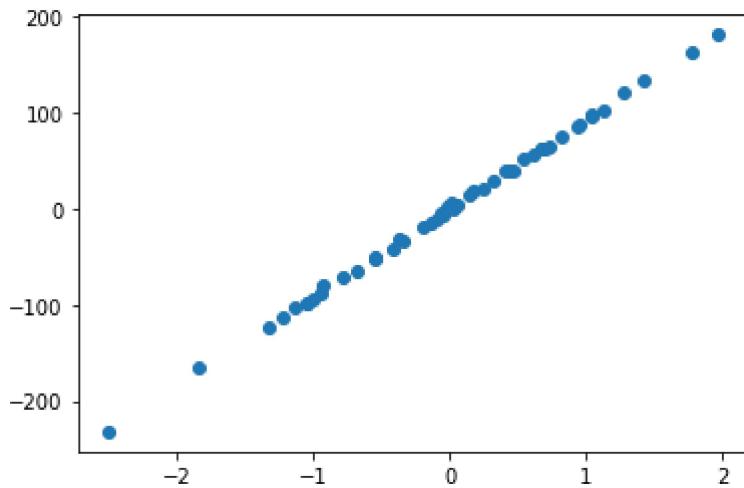
In [44]:

```
# Import necessary libraries
from sklearn.datasets import make_regression
import matplotlib.pyplot as plt
# Generate 1d Regression dataset
X, y = make_regression(n_samples = 50, n_features=1, noise=20, random_state=23)
# Plot the generated datasets
plt.scatter(X, y)
plt.show()
```



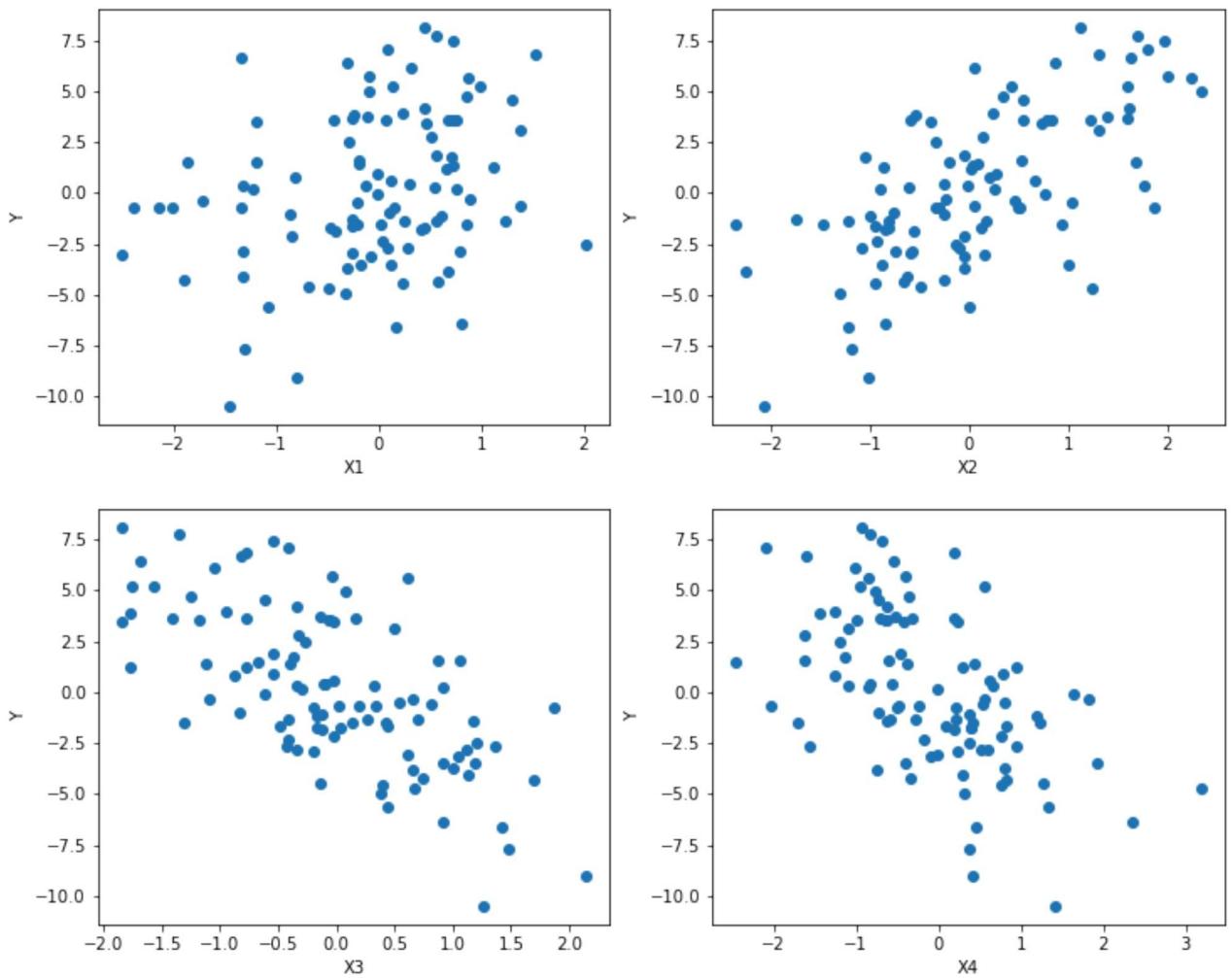
In [47]:

```
# Generate 1d Regression dataset
X, y = make_regression(n_samples = 50, n_features=1, noise=2, random_state=23)
# Plot the generated datasets
plt.scatter(X, y)
plt.show()
```



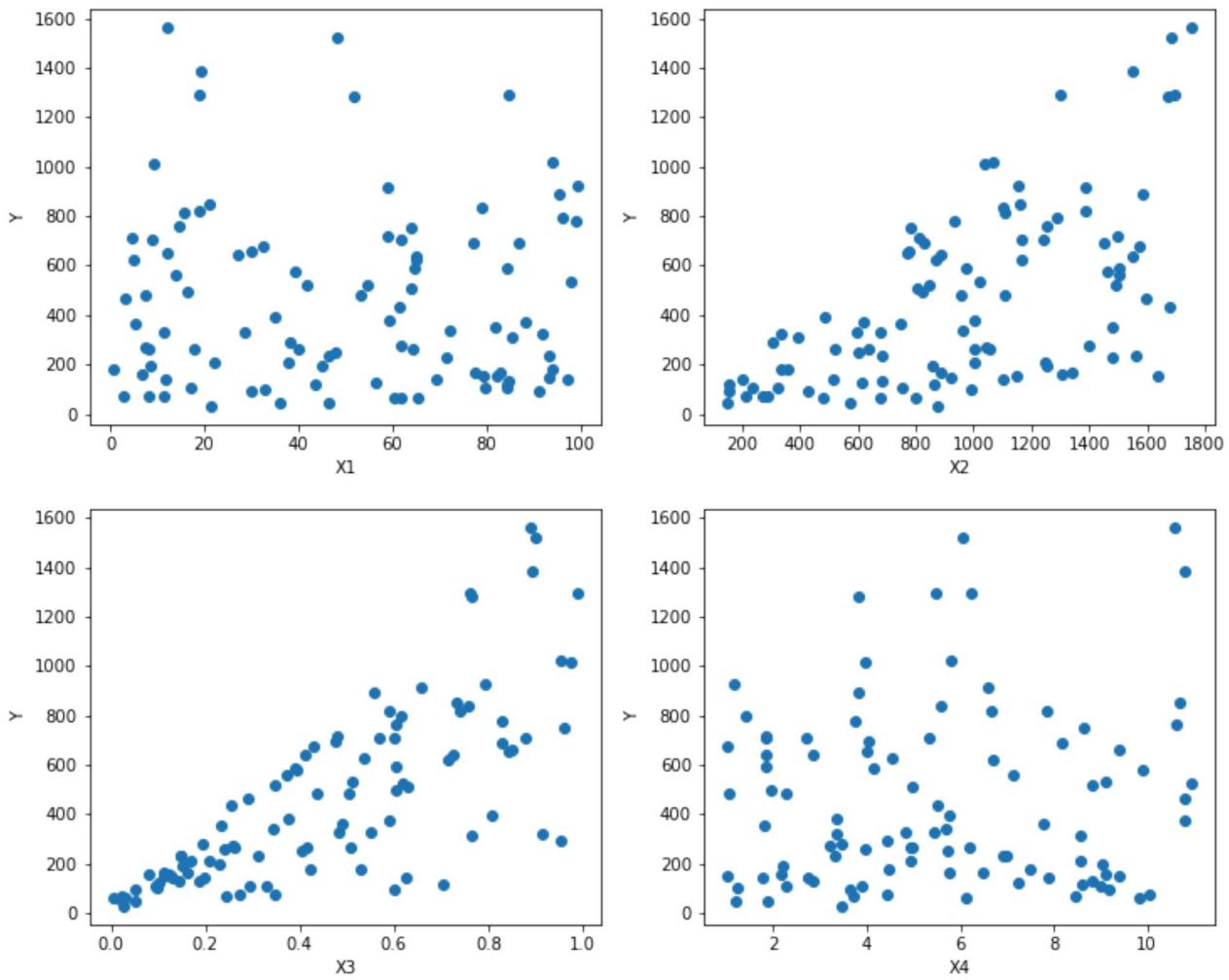
In [48]:

```
# Import necessary libraries
from sklearn.datasets import make_sparse_uncorrelated
import matplotlib.pyplot as plt
# Generate 1d Regression dataset
X, y = make_sparse_uncorrelated(n_samples = 100, n_features=4, random_state=23)
# Plot the generated datasets
plt.figure(figsize=(12,10))
for i in range(4):
    plt.subplot(2,2, i+1)
    plt.scatter(X[:,i], y)
    plt.xlabel('X'+str(i+1))
    plt.ylabel('Y')
plt.show()
```



In [49]:

```
# Import necessary libraries
from sklearn.datasets import make_friedman2
import matplotlib.pyplot as plt
# Generate 1d Regression dataset
X, y = make_friedman2(n_samples = 100, random_state=23)
# Plot the generated datasets
plt.figure(figsize=(12,10))
for i in range(4):
    plt.subplot(2,2, i+1)
    plt.scatter(X[:,i], y)
    plt.xlabel('X'+str(i+1))
    plt.ylabel('Y')
plt.show()
```



In []: