Name - Soumyadeep Ganguly

Register no. - 24MDT0082

Lab_Assessment_5

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.datasets import make_classification
```

In [2]:
```
X, y = make_classification(n_classes=3, n_features=2, n_redundant=0, n_clusters_per_class=1,
random_state=42)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 0)
pd.DataFrame(data = X).head()
```

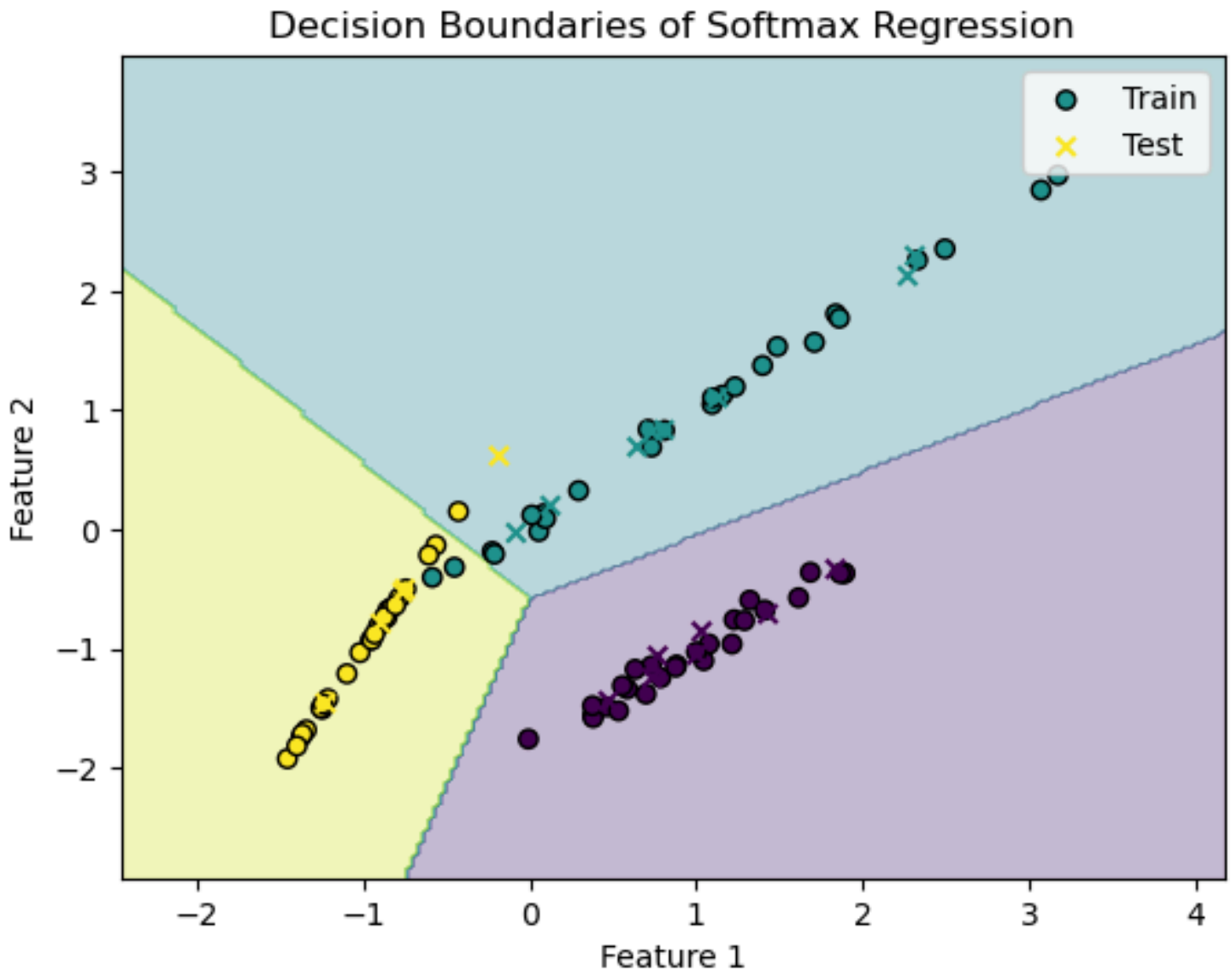
Out[2]:

	0	1
0	0.698409	-1.380295
1	-0.008323	-1.757614
2	1.129916	1.102361
3	1.228312	-0.757178
4	-1.372983	-1.738339

In [3]:

```
clf = LogisticRegression(multi_class='multinomial')
clf.fit(X_train, y_train)
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
xx, yy = np.meshgrid(np.linspace(x_min, x_max, 200), np.linspace(y_min, y_max, 200))

Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)
plt.contourf(xx, yy, Z, alpha=0.3)
plt.scatter(X_train[:, 0], X_train[:, 1], c=y_train, edgecolors='k', label='Train')
plt.scatter(X_test[:, 0], X_test[:, 1], c=y_test, marker='x', label='Test')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.title('Decision Boundaries of Softmax Regression')
plt.legend()
plt.show()
```



In [5]:
```
df = pd.read_csv("train_set_label.csv")
df.head()
```

Out[5]:

	N	P	K	temperature	humidity	ph	rainfall	crop
0	17.0	136.0	196.0	23.871923	90.499390	5.882156	103.054809	apple
1	49.0	69.0	82.0	18.315615	15.361435	7.263119	81.787105	chickpea
2	74.0	49.0	38.0	23.314104	71.450905	7.488014	164.497037	jute
3	104.0	35.0	28.0	27.510061	50.666872	6.983732	143.995555	coffee
4	23.0	72.0	84.0	19.020613	17.131591	6.920251	79.926981	chickpea

In [13]:
```
mms = preprocessing.MinMaxScaler(feature_range=(0,1))
f = mms.fit_transform(df.iloc[:,:-1])
x = f
y = df.iloc[:,-1].values
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
l_reg = LogisticRegression(multi_class='multinomial')
l_reg.fit(x_train,y_train)
```

Out[13]:

LogisticRegression

```
LogisticRegression(multi_class='multinomial')
```

In [15]:
```
predict = l_reg.predict(x_test)
accuracy = accuracy_score(predict, y_test)
accuracy
```

Out[15]:
```
0.9424242424242424
```

In [ ]: