Roll no - 23053649 Name - Rishika Kumari

```
AD    lab - 10
```

In [ ]:
```
code for generating dataset
```

In [2]:
```
import numpy as np
import pandas as pd

np.random.seed(42)

# Generate synthetic binary classification data
n = 200
x1 = np.random.randn(n) * 2
x2 = np.random.randn(n) * 2

# Decision boundary: x1 + x2 > 0 -> class 1 else 0
y = (x1 + x2 > 0).astype(int)

df = pd.DataFrame({"feature1": x1, "feature2": x2, "target": y})
df.to_csv("logistic_regression_dataset.csv", index=False)

print("Dataset created: logistic_regression_dataset.csv")
df.head()
```

```
Dataset created: logistic_regression_dataset.csv
```

Out[2]:

	feature1	feature2	target
0	0.993428	0.715575	1
1	-0.276529	1.121569	1
2	1.295377	2.166102	1
3	3.046060	2.107604	1
4	-0.468307	-2.755339	0

In [3]:
```
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# Load dataset
df = pd.read_csv("logistic_regression_dataset.csv")

X = df[["feature1", "feature2"]]
y = df["target"]

model = LogisticRegression()
model.fit(X, y)

y_pred = model.predict(X)
```

```python
print("Accuracy:", accuracy_score(y, y_pred))
print("\nConfusion Matrix:\n", confusion_matrix(y, y_pred))
print("\nClassification Report:\n", classification_report(y, y_pred))
```

```
Accuracy: 0.995

Confusion Matrix:
 [[ 89   1]
 [  0 110]]

Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.99      0.99        90
           1       0.99      1.00      1.00       110

    accuracy                           0.99       200
   macro avg       1.00      0.99      0.99       200
weighted avg       1.00      0.99      0.99       200
```

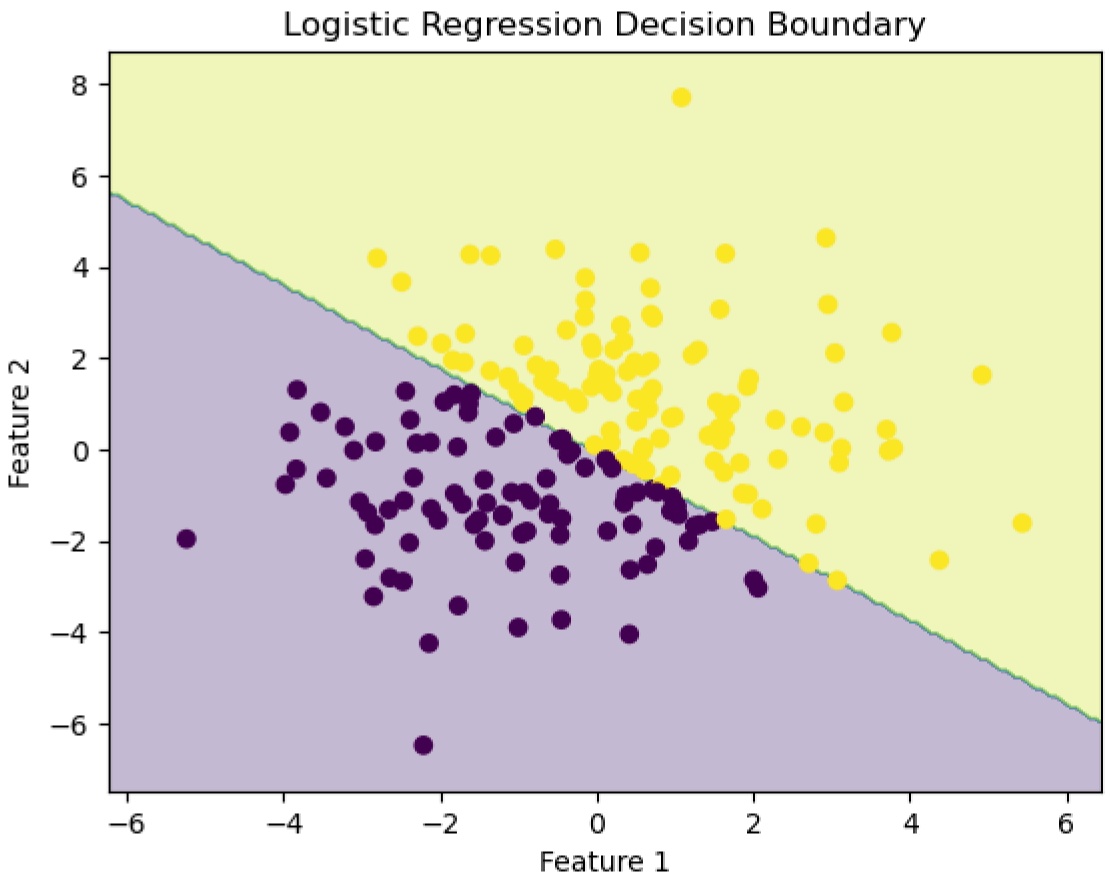
In [5]:
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

x_min, x_max = X["feature1"].min()-1, X["feature1"].max()+1
y_min, y_max = X["feature2"].min()-1, X["feature2"].max()+1

xx, yy = np.meshgrid(np.linspace(x_min, x_max, 200),
                     np.linspace(y_min, y_max, 200))

grid = pd.DataFrame(np.c_[xx.ravel(), yy.ravel()], columns=["feature1", "feature2"]
Z = model.predict(grid)
Z = Z.reshape(xx.shape)

plt.figure()
plt.contourf(xx, yy, Z, alpha=0.3)
plt.scatter(X["feature1"], X["feature2"], c=y)
plt.xlabel("Feature 1")
plt.ylabel("Feature 2")
plt.title("Logistic Regression Decision Boundary")
plt.show()
```


Logistic Regression Decision Boundary
Feature 2
Feature 1
8
6
4
2
0
−2
−4
−6
−6
−4
−2
0
2
4
6

In [ ]: