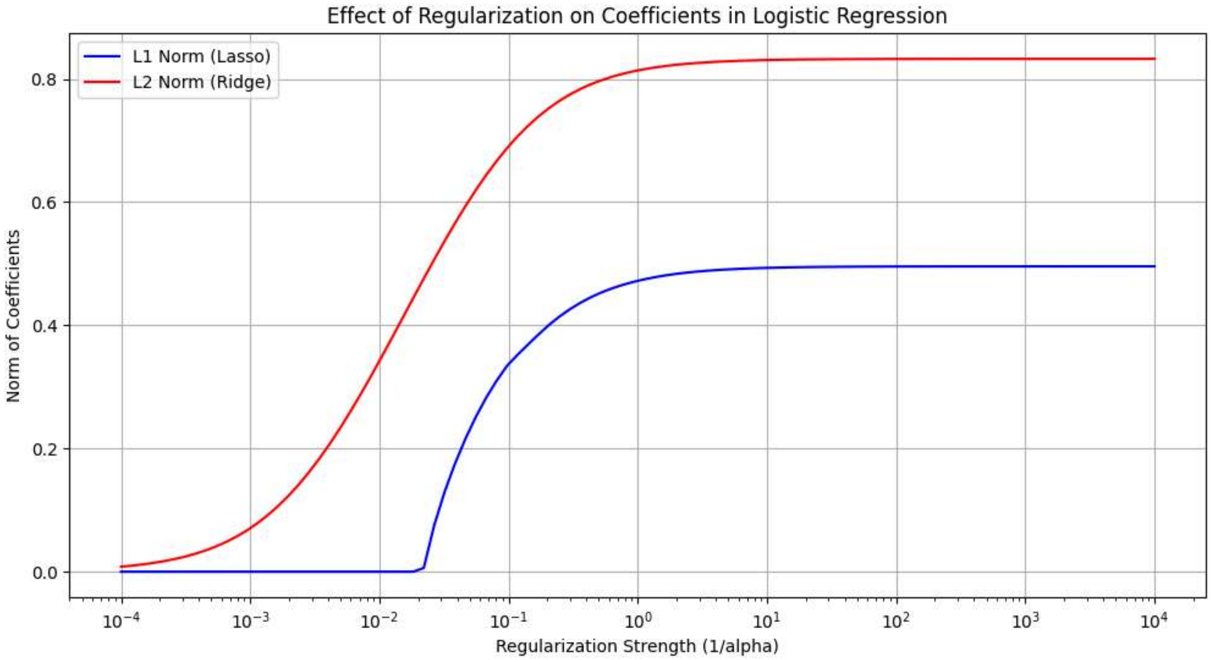
In [ ]:

```
#L1 AND L2 REGULARIZATION EFFECTS IN LOGISTIC REGRESSION
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import make_classification

# Generate synthetic dataset
np.random.seed(42)
X, y = make_classification(
    n_samples=200, n_features=10, n_informative=5,
    n_redundant=0, random_state=42
)

# Range of regularization strengths (inverse of alpha, i.e., C)
reg_strengths = np.logspace(-4, 4, 100)

# Lists to store the norm of weights for each regularization type
l1_norms = []
l2_norms = []

# Apply L1 regularization (Lasso)
for c in reg_strengths:
    model_l1 = LogisticRegression(penalty='l1', solver='saga', C=c, max_iter=10000)
    model_l1.fit(X, y)
    l1_norms.append(np.linalg.norm(model_l1.coef_, ord=1))  # L1 norm

# Apply L2 regularization (Ridge)
for c in reg_strengths:
    model_l2 = LogisticRegression(penalty='l2', solver='lbfgs', C=c, max_iter=10000
    model_l2.fit(X, y)
    l2_norms.append(np.linalg.norm(model_l2.coef_, ord=2))  # L2 norm

# Visualization
plt.figure(figsize=(12, 6))
plt.plot(reg_strengths, l1_norms, label='L1 Norm (Lasso)', color='blue')
plt.plot(reg_strengths, l2_norms, label='L2 Norm (Ridge)', color='red')
plt.xscale('log')
plt.xlabel('Regularization Strength (1/alpha)')
plt.ylabel('Norm of Coefficients')
plt.title('Effect of Regularization on Coefficients in Logistic Regression')
plt.legend()
plt.grid()
plt.show()
```


Effect of Regularization on Coefficients in Logistic Regression
L1 Norm (Lasso)
L2 Norm (Ridge)
Norm of Coefficients
0.0
0.2
0.4
0.6
0.8
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
Regularization Strength (1/alpha)