Exercise 4:

Implement optimization techniques like L2 regularization and retrain the model. Evaluate the per

[13]:

```python
# Instantiate the model
input_dim = X_train.shape[1]
model = LogisticRegression(input_units=input_dim)
# Define the loss function and optimizer
optimizer = optim.SGD(model.parameters(), lr=0.01, weight_decay=0.01)
epochs = 1000
for epoch in range(epochs):
    # Training phase
    model.train()
    optimizer.zero_grad()
    outputs = model(X_train)
    loss = criterion(outputs, y_train)
    loss.backward()
    optimizer.step()
    if (epoch + 1) % 100 ==0:
        print(f'Epoch [{epoch + 1}], Loss value: {loss.item()}')
# Evaluation phase on test set
model.eval()
with torch.no_grad():
    #get predictions of train and test:
    train_x_pred = model(X_train)
    test_x_pred = model(X_test)
    #use threshold 0.5:
    train_accu = ((train_x_pred > 0.5) == y_train).float().mean()
    test_accu = ((test_x_pred > 0.5) == y_test).float().mean()
    print(f'Training accuracy: {train_accu} and Test accuracy: {test_accu}')
```

```
Epoch [100], Loss value: 0.7349382638931274
Epoch [200], Loss value: 0.7165544033050537
Epoch [300], Loss value: 0.7048165798187256
Epoch [400], Loss value: 0.6974416971206665
Epoch [500], Loss value: 0.692847728729248
Epoch [600], Loss value: 0.6899960041046143
Epoch [700], Loss value: 0.6882261633872986
Epoch [800], Loss value: 0.6871259808540344
Epoch [900], Loss value: 0.6864399909973145
Epoch [1000], Loss value: 0.6860107183456421
Training accuracy: 0.5550000071525574 and Test accuracy: 0.5099999904632568
```