Pseudocode

```
Algorithm: LogisticRegression

Input:
    X  (training data)
    y  (class labels)
    α  (learning rate)
    T  (number of iterations)

Output:
    w  (weights)
    b  (bias)

Initialize w = 0
Initialize b = 0

for i = 1 to T do
    z = X * w + b
    y_hat = 1 / (1 + exp(-z))

    error = y_hat - y

    w = w - α * (Xᵀ * error)
    b = b - α * sum(error)
end for

return w, b
```

Python Code

```python
import numpy as np

def logistic_regression(X, y, alpha=0.01, iterations=1000):
    n, d = X.shape
    w = np.zeros(d)
    b = 0

    for _ in range(iterations):
        z = np.dot(X, w) + b
        y_hat = 1 / (1 + np.exp(-z))
```

```
        error = y_hat - y
        dw = (1 / n) * np.dot(X.T, error)
        db = (1 / n) * np.sum(error)

        w = w - alpha * dw
        b = b - alpha * db

    return w, b
```