Assignment 12

Task 1

a) Comparison of Dynamic Time Warping (DTW) and Logistic Regression:

Similarities:

1. Both are used in data analysis and machine learning.
2. Both can be applied to time series data.
3. Both can be used for classification tasks in certain scenarios.

Differences:

1. Purpose: DTW is primarily used for measuring similarity between two temporal sequences, while Logistic Regression is used for binary classification.
2. Methodology: DTW aligns time series data to find an optimal match between two sequences, whereas Logistic Regression estimates the probability of an instance belonging to a particular class.
3. Input data: DTW works with sequential data, particularly time series. Logistic Regression typically works with feature vectors.
4. Output: DTW produces a similarity measure or distance between sequences. Logistic Regression outputs probabilities or class labels.
5. Complexity: DTW has a higher time complexity (typically O(n^2)) compared to Logistic Regression (O(n) for prediction).

b) Efficiency after the training phase:

Logistic Regression is generally more efficient after the training phase.

Justification:

1. Prediction time: Once trained, Logistic Regression makes predictions in constant time O(1) or linear time O(n) depending on the number of features. DTW, on the other hand, requires O(n^2) time for each comparison.
2. Memory usage: Logistic Regression stores only the learned coefficients, while DTW requires storing entire reference sequences.
3. Scalability: Logistic Regression can easily handle large datasets for prediction, while DTW's quadratic time complexity can become prohibitive for long sequences.

c)

Template Sequence:

[9, 7, 6, 5, 4, 1, 8, 11, 6, 3, 3, 1, 2]

Input Sequence:

[7, 7, 2, 9, 1, 1]

Step 1: Initialize the DTW Table

The DTW table is initialized to store the cumulative absolute differences.

Step 2: Fill the First Row and Column

First row and column are filled with the cumulative absolute differences:

	9	7	6	5	4	1	8	11	6	3	3	1	2
7	2	2	3	5	8	14	15	19	20	24	28	34	39
7	4	2	3	5	8	14	15	19	20	24	28	34	39
2	11	7	6	6	7	8	14	23	23	21	22	23	23
9	11	9	9	10	11	15	9	11	14	20	26	30	30
1	19	15	14	13	13	11	16	19	16	16	18	18	19
1	27	21	19	17	16	11	18	26	21	18	18	18	19

Step 3: Fill the Rest of the DTW Table

Using the DTW formula:

D(i, j) = |input_seq[i] - template[j]| + min(D(i-1, j), D(i, j-1), D(i-1, j-1))

Completed DTW Table:

	9	7	6	5	4	1	8	11	6	3	3	1	2
7	2	2	3	5	8	14	15	19	20	24	28	34	39
7	4	2	3	5	8	14	15	19	20	24	28	34	39
2	11	7	6	6	7	8	14	23	23	21	22	23	23
9	11	9	9	10	11	15	9	11	14	20	26	30	30
1	19	15	14	13	13	11	16	19	16	16	18	18	19
1	27	21	19	17	16	11	18	26	21	18	18	18	19

Step 4: Identify the Optimal Path

The optimal path is traced back from the bottom-right to the top-left of the DTW table. Here is the path:

	9	7	6	5	4	1	8	11	6	3	3	1	2
7	2	2	3										
7		2		5									
2			6		7								
9						15	9						
1						11		19		18	18		
1						11		18				18	

The optimal path is: [(0, 0), (0, 1), (0, 2), (1, 3), (2, 4), (2, 5), (3, 6), (3, 7), (3, 8), (4, 9), (4, 10), (4, 11), (5, 12)]

Step 5: Specify the Minimal Costs

The minimal cost, which is the value in the bottom-right cell of the DTW table, is 19.0.

Conclusion:

The DTW algorithm for the given template and input sequences results in a minimal cost of 19.0, with the optimal path being [(0, 0), (0, 1), (0, 2), (1, 3), (2, 4), (2, 5), (3, 6), (3, 7), (3, 8), (4, 9), (4, 10), (4, 11), (5, 12)].