Manual Solution:

1. Tokenization:
  - Split each sentence into individual words:
    [['data', 'science', 'is', 'one', 'of', 'the', 'most', 'important', 'courses', 'in', 'computer', 'science'],
     ['this', 'is', 'one', 'of', 'the', 'best', 'data', 'science', 'courses'],
     ['the', 'data', 'scientists', 'perform', 'data', 'analysis']]

2. Bag of Words (BoW):
  - Count the occurrences of each term in each sentence.
    [[1, 2, 1, ..., 0, 0, 0], # Row 1
     [0, 1, 1, ..., 0, 0, 0], # Row 2
     [0, 2, 1, ..., 1, 1, 0]] # Row 3

3. Term Frequency (TF):
  - For each sentence, calculate the Term Frequency (TF) for each term using the formula: `tfi,d = frequency of term i in document d / total number of terms in document d`.
    [[1/12, 1/12, 1/12, ..., 0, 0, 0], # Row 1
     [0, 1/9, 1/9, ..., 0, 0, 0],   # Row 2
     [0, 2/6, 1/6, ..., 1/6, 1/6, 0]] # Row 3

4. Inverse Document Frequency (IDF):
  - For each term, calculate the Inverse Document Frequency (IDF) using the formula: `idf(t) = log(N / df(t))`, where N is the total number of documents and df(t) is the number of documents containing term t.
    [[0, 0, 0, ..., 0, 0, 0], # Row 1
     [0, 0, 0, ..., 0, 0, 0], # Row 2
     [0, 0, 0, ..., 0, 0, 0]] # Row 3

5. Term Frequency-Inverse Document Frequency (TF.IDF):
  - Multiply the TF matrix by the IDF matrix element-wise.
    [[0, 0, 0, ..., 0, 0, 0], # Row 1
     [0, 0, 0, ..., 0, 0, 0], # Row 2
     [0, 0, 0, ..., 0, 0, 0]] # Row 3

6. Similarity Calculations:
  - Use cosine, Manhattan, and Euclidean distances between the TF.IDF vectors of the sentences.
    [[0, 0.478, 0.681], # Row 1
     [0.478, 0, 0.666], # Row 2
     [0.681, 0.666, 0]] # Row 3