Project Explanation

The Naive Bayes algorithm is a probabilistic classification algorithm based on Bayes' theorem. It assumes that the presence of each word in a message is independent of the presence of other words, which simplifies the calculations and makes it computationally efficient.

To implement Naive Bayes for spam SMS detection, you would typically follow these steps:

1. Data Preprocessing: Start by preparing your dataset, which should include a collection of labeled SMS messages, indicating whether each message is spam or legitimate. The dataset should be representative of the messages encountered in real-world scenarios. Clean the dataset by removing any irrelevant information, such as special characters or punctuation, and perform preprocessing steps like tokenization (splitting the text into individual words or tokens), stop-word removal (eliminating common words like "the" or "and"), and stemming (reducing words to their root form, e.g., "running" to "run").

2. Training Phase: Divide your dataset into two subsets: a training set and a test set. The training set will be used to calculate the probabilities necessary for classification. In Naive Bayes, you calculate the prior probabilities of spam and legitimate messages by counting the number of occurrences of each class in the training set. You also compute the conditional probabilities of each word given each class, which represents the likelihood of a specific word occurring in a spam or legitimate message. These probabilities are calculated by counting the occurrences of each word within each class.

3. Classification Phase: Once the probabilities have been calculated during the training phase, you can apply the Naive Bayes algorithm to classify new, unseen SMS messages. Given a new message, the algorithm calculates the probability of it belonging to the spam class and the legitimate class separately. This is done by multiplying the prior probability of each class by the conditional probabilities of the words present in the message. The final classification is determined by comparing these probabilities and assigning the message to the class with the higher probability.

4. Evaluation: After classifying the test set, you can evaluate the performance of the algorithm using various metrics such as accuracy, precision, recall, and F1 score. Accuracy measures the overall correctness of the classifications, while precision measures the proportion of correctly classified spam messages. Recall measures the proportion of correctly classified spam messages out of all the actual spam messages. The F1 score combines precision and recall into a single metric to provide a balanced evaluation.

It's worth noting that the performance of the Naive Bayes algorithm can be influenced by the quality and representativeness of the training data, the selection of features, and the handling of any assumptions made (such as the independence assumption). It's important to fine-tune the algorithm and optimize its parameters based on the evaluation metrics to achieve the best possible results. Additionally, the algorithm should be continuously monitored and updated as spamming techniques evolve over time.

Overall, the manually implemented Naive Bayes algorithm for spam SMS detection offers an effective approach to identify and filter unwanted messages by leveraging the probabilistic nature of the algorithm and the independence assumption.