PRML Assignment 3

Model 1: Gaussian Naive Bayes Approach:

Implementation:

1. **Data Preparation:**
   - The dataset is loaded from a CSV file containing email data, where each row represents an email and columns represent features (words).
   - The dataset is split into features (X) and labels (y), with the features excluding the first column (email name) and the last column (labels).

2. **Model Training:**
   - A Gaussian Naive Bayes classifier is implemented using a custom class.
   - The **fit** method of the classifier computes the mean and variance for each class(spam and ham) and features using the training data.
   - The class prior probabilities are also computed based on the frequency of each class in the training data.
3. Model Prediction:
   - The **predict** method of the classifier predicts the class labels for the test data using the computed mean and variance.
   - Posterior probabilities are calculated for each class using the Gaussian probability density function something like below:

     We have two classes: spam (S) and ham (H).
     For a given data point X = (x1, x2, …….. xd), where xi represents the value of the ith feature.
     The posterior probability of class S given the features X is calculated as:

$$P(S|\mathbf{X}) = \frac{P(S) \times \prod_{i=1}^{n} P(x_i|S)}{P(\mathbf{X})}$$

     And the posterior probability of class H given the features X is calculated as:

$$P(H|\mathbf{X}) = \frac{P(H) \times \prod_{i=1}^{n} P(x_i|H)}{P(\mathbf{X})}$$

Where:

- P(S|X) and P(H|X) are the posterior probabilities of classes spam and ham given the features X, respectively.

- P(S) and P(H) are the prior probabilities of classes spam and ham, respectively. These are estimated from the training data.
- P(xi|S) and P(xi|H) are the likelihoods of observing feature xi given classes spam and ham, respectively. These are modeled using the Gaussian probability density function (PDF) for each class.
- P(X) is the evidence, or total probability of observing the features X, and acts as a normalization factor.
- The class with the higher posterior probability (either spam or ham) is then assigned as the predicted class for the given data point X.
-

This is the essence of how Gaussian Naive Bayes classification works, utilizing Bayes' theorem to compute posterior probabilities based on prior probabilities and likelihoods.

4. Evaluation:
The accuracy of the model is evaluated by comparing the predicted labels with the true labels of the test data.
The accuracy score is calculated using the accuracy_score function from scikit-learn.
Formulation:

Gaussian Naive Bayes assumes that the features are continuous and follow a Gaussian distribution within each class.
The likelihood of each feature given the class is modeled as a Gaussian distribution with mean and variance computed from the training data.

Accuracy:

The accuracy of the Gaussian Naive Bayes classifier on the test data is reported as 93.33%.

```
Accuracy: 0.9333333333333333
```

Model 2: Support Vector Machine Approach:

Implementation, Formulation, and Accuracy:

- SVM is implemented using the SVC class from scikit-learn.
- In SVM, the objective is to find the hyperplane that best separates the data into different classes.

- SVM maximizes the margin between the hyperplane and the nearest data points (support vectors).
- The formulation involves minimizing the hinge loss function while incorporating regularization terms to prevent overfitting.
- Accuracy of SVM can vary depending on various factors such as the choice of kernel, regularization parameter, and feature scaling.
- To evaluate the SVM model, the same approach as Gaussian Naive Bayes can be followed, calculating the accuracy score on the test data.

Accuracy: The SVM model gave an accuracy of 95.94%

Summary:
The dataset that I have trained both these models is around 6000 emails, each of them is a feature vector of 3000 features(words)

Both Gaussian Naive Bayes and SVM are popular algorithms for classification tasks. While Gaussian Naive Bayes assumes independence between features and follows a probabilistic approach, SVM aims to find the best decision boundary by maximizing the margin between classes. The choice between these algorithms depends on factors such as the nature of the data, the size of the dataset, and the desired performance metrics.

In my case. Gaussian Naive Bayes gave an accuracy of 93% while SVM gave an accuracy of 96%. So I choose to use SVM model over GNB.