**Email Classification using Naive Bayes**

**Introduction**

This report explores the implementation of an email classification system using the Naive Bayes algorithm. The primary goal is to automatically categorize emails into spam and non-spam categories, streamlining inbox management and enhancing user experience.

**Tech Stack**

The technology stack for this project includes Python as the programming language, leveraging the scikit-learn library for machine learning tasks. The system utilizes the Multinomial Naive Bayes classifier, which is well-suited for text-based classification. Data preprocessing involves cleaning and transforming raw email text using techniques like lowercasing, removing stop words, and converting text to numerical format through methods such as Bag-of-Words or TF-IDF. The scikit-learn library is also employed for splitting the dataset into training and testing sets, model training, and performance evaluation.

**Implementation Steps**

**1. Data Collection** Gather a labelled dataset of emails.

**2. Preprocessing** Clean and transform text data for machine learning.

**3. Feature Extraction** Convert text into a numerical format suitable for Naive Bayes.

**4. Training** Utilize the Multinomial Naive Bayes classifier to train the model.

**5. Evaluation** Assess model performance using metrics like accuracy, precision, and recall.

**6. Deployment** Deploy the trained model for real-world email classification.

**7. Continuous Improvement** Periodically retrain the model with new data for adaptability.

**Conclusion**

Implementing email classification with Naive Bayes proves to be a powerful solution for automating the categorization of emails. The simplicity and efficiency of Naive Bayes, combined with a carefully chosen tech stack, contribute to a streamlined and effective system for email management. The continuous improvement aspect ensures the model remains accurate and adaptable to changing email patterns over time.