Sri Sivasubramaniya Nadar College of Engineering, Chennai
(An autonomous Institution affiliated to Anna University)

Degree & Branch	B.E. Computer Science & Engineering	Semester	V
Subject Code & Name	UCS2612 & Machine Learning Algorithms Laboratory		
Academic year	2025-2026 (Even)	Batch:2023-2027	**Due date: 26/01/25**

Experiment 2: Binary Classification using Naïve Bayes and K-Nearest Neighbors

Name: Mariya Joevita
Reg.No: 3122235001077
Class: CSE-B

1. Aim and Objective

Aim: To implement and analyze Naïve Bayes and K-Nearest Neighbors classifiers for a binary classification problem.

Objective:

- To perform data preprocessing and exploratory data analysis
- To train Naïve Bayes and KNN classifiers
- To tune KNN hyperparameters using cross-validation
- To compare KDTree and BallTree neighbor search methods
- To evaluate models using multiple performance metrics

Dataset Description

The Spambase dataset contains email message features used to classify emails as spam or non-spam. It consists of word frequency, character frequency, and capital run-length features.

- Total Features: 57
- Target Class: Binary (0 – Non-Spam, 1 – Spam)

Dataset Source: Kaggle – Spambase Dataset

Preprocessing Steps

Before training the machine learning models, the dataset was preprocessed to improve data quality and ensure compatibility with the algorithms.

Loading the Dataset

The dataset contains a total of 4601 email samples. Each sample is represented using 57 numerical features and belongs to one of two classes:

- Spam (1)
- Non-spam (0)

Checking Missing Values

All features were checked for missing values.

$$\text{Number of missing values} = 0$$

Since no missing values were present, no data cleaning was required.

Separating Features and Target

The dataset was divided into input features and output labels.

$$X = \text{All columns except the class label}$$

$$y = \text{Class label column}$$

Here, X represents the feature matrix and y represents the target vector.

Train–Test Split

The dataset was split into training and testing sets to evaluate model performance.

- Training set: 80% of the data (3680 samples)
- Testing set: 20% of the data (921 samples)

Stratified sampling was used to maintain the same class distribution in both sets.

Feature Scaling

Feature scaling was applied to ensure that all features contribute equally to the model.

For each feature, standardization was performed as:

$$\text{Scaled value} = \frac{\text{Value} - \text{Mean}}{\text{Standard Deviation}}$$

After scaling:

- Mean = 0
- Standard deviation = 1

This step is important for distance-based algorithms such as K-Nearest Neighbors.

Result of Preprocessing

After completing the preprocessing steps, the dataset was clean, well-structured, and ready for model training.

Exploratory Data Analysis (EDA)

EDA was performed to understand the dataset characteristics:

- Dataset shape, data types, and statistical summary
- Class distribution analysis
- Feature distribution visualization for word frequency features

Model Implementation

Naïve Bayes

The following Naïve Bayes variants were trained:

- Gaussian Naïve Bayes
- Multinomial Naïve Bayes
- Bernoulli Naïve Bayes

K-Nearest Neighbors

- Baseline KNN classifier trained
- Hyperparameter tuning performed using 5-Fold Cross-Validation
- KDTree and BallTree search algorithms evaluated

Visualizations

The following visualizations were generated to analyze the dataset characteristics and evaluate the performance of the classifiers.

Class Distribution

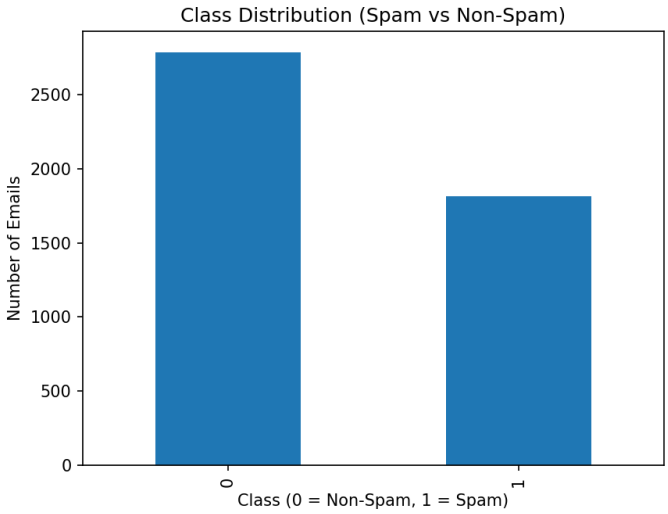

Figure 1: Class distribution of spam and non-spam emails

Feature Distribution Analysis

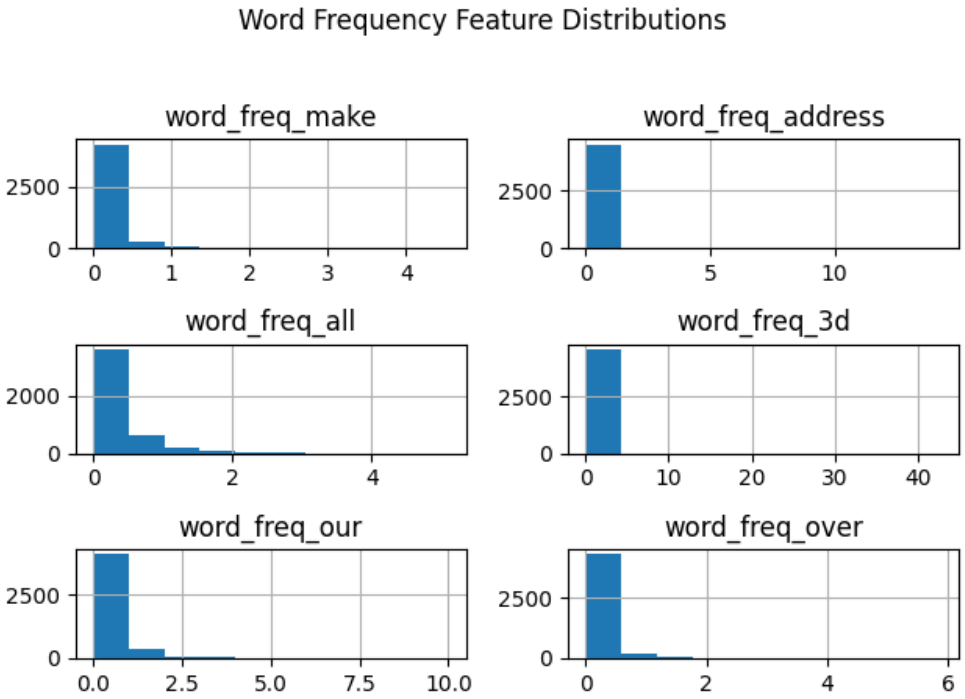

Figure 2: Word frequency feature distributions

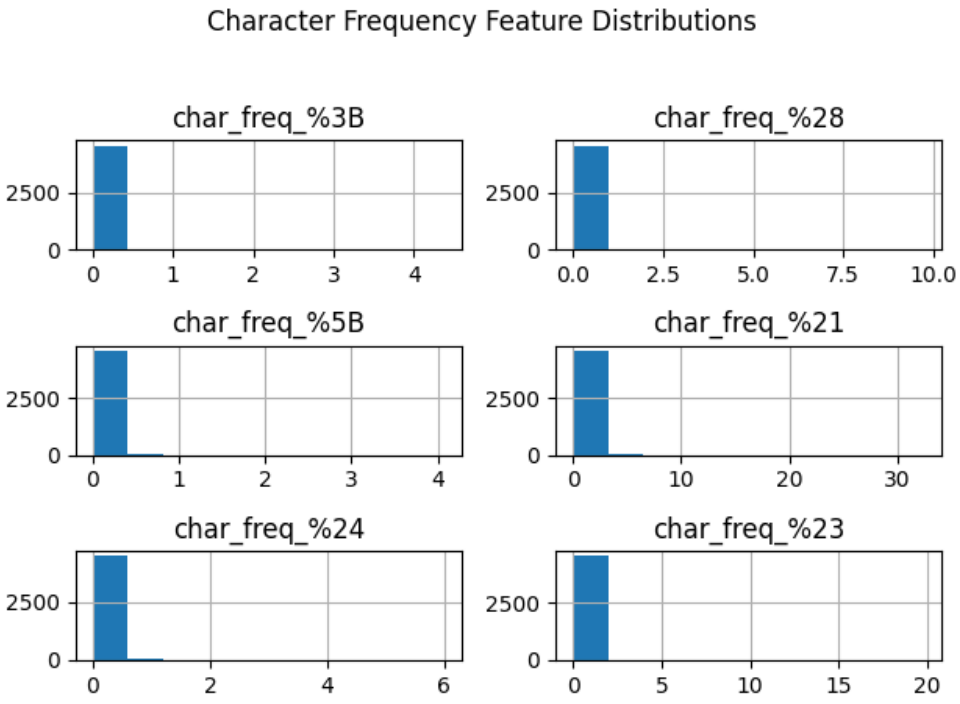

Figure 3: Character frequency feature distributions

Feature Correlation

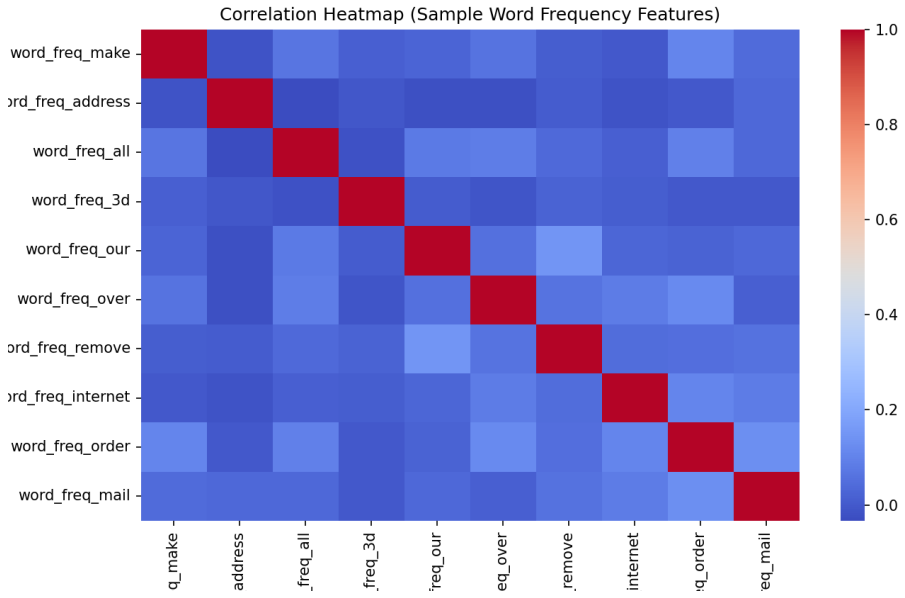

Figure 4: Correlation heatmap of selected features

Confusion Matrices

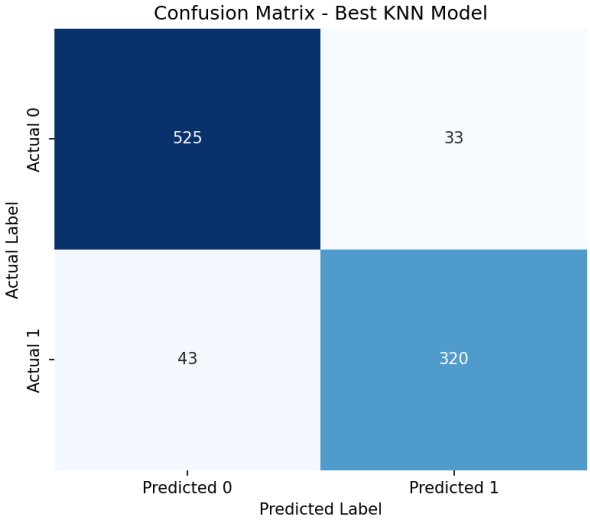

Figure 5: Confusion matrix for best KNN classifier

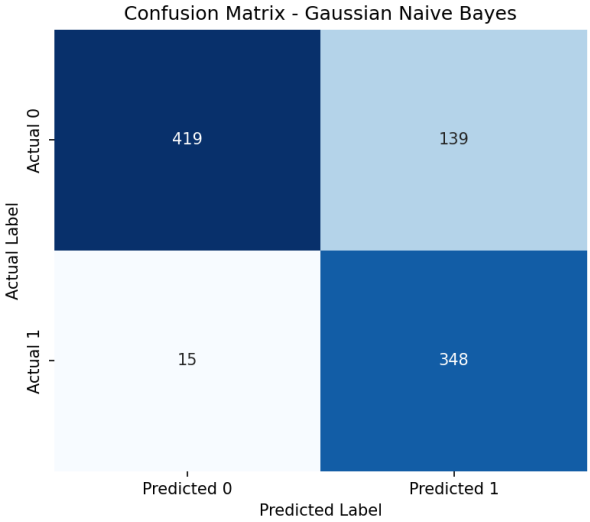

Figure 6: Confusion matrix for Gaussian Naïve Bayes

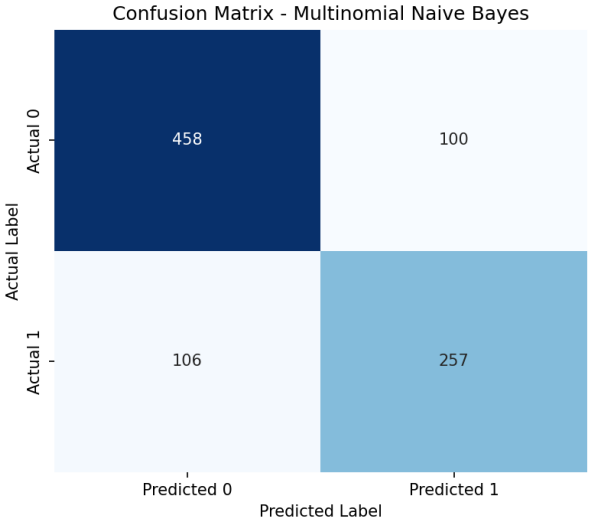

Figure 7: Confusion matrix for Multinomial Naïve Bayes

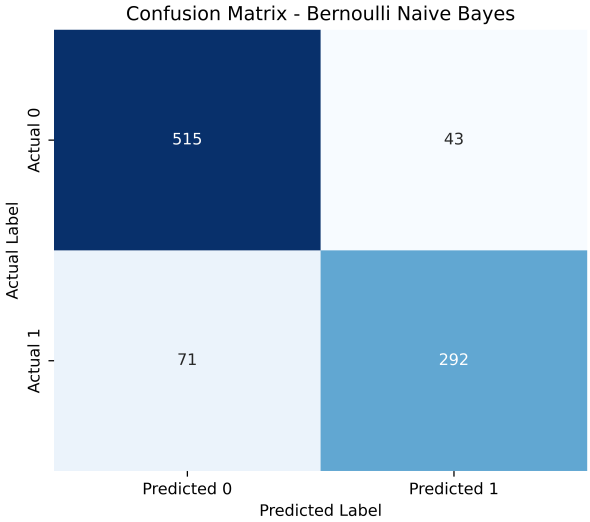

Figure 8: Confusion matrix for Bernoulli Naïve Bayes

ROC Curves

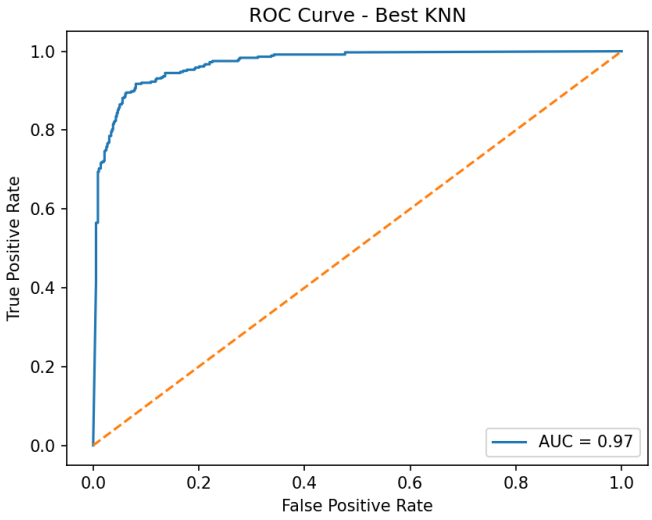

Figure 9: ROC curve for best KNN classifier

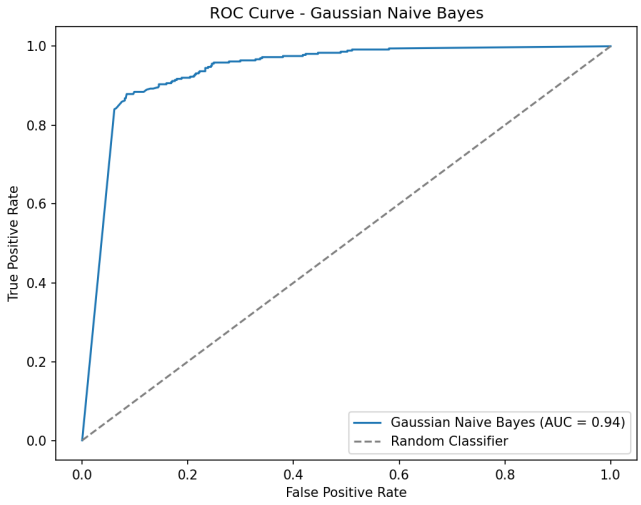

Figure 10: ROC curve for Gaussian Naïve Bayes

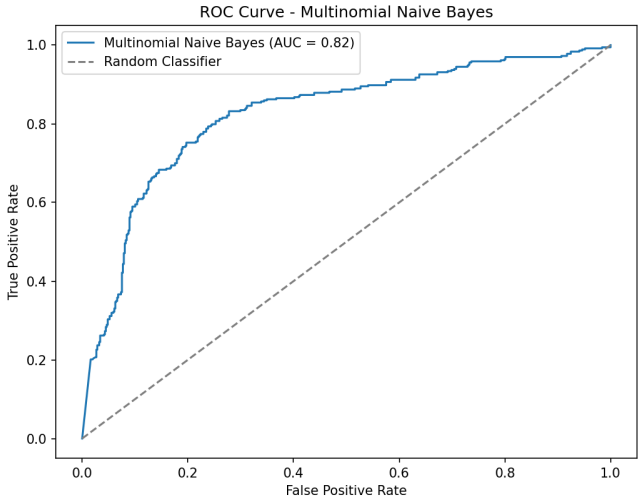

Figure 11: ROC curve for Multinomial Naïve Bayes

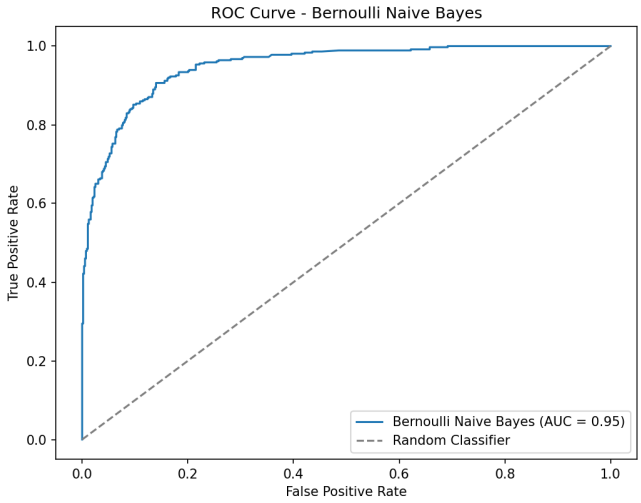

Figure 12: ROC curve for Bernoulli Naïve Bayes

Accuracy vs. k for KNN

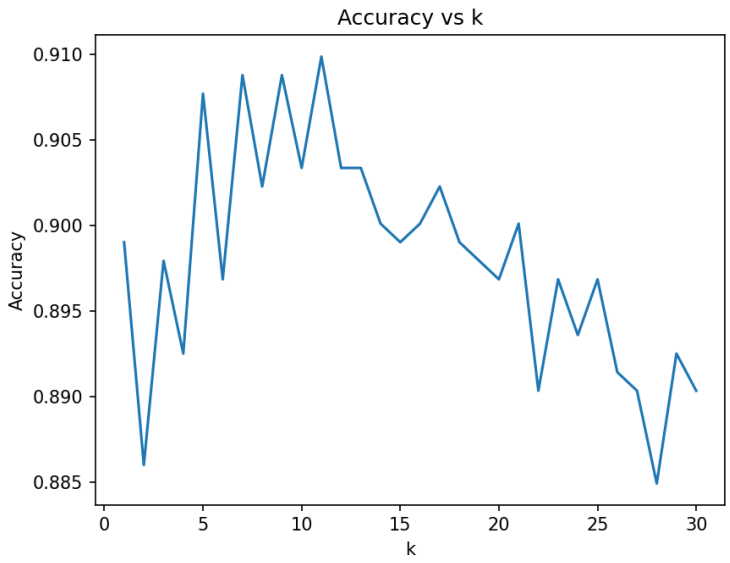

Figure 13: Accuracy vs. number of neighbors (k)

Training vs. Validation Accuracy

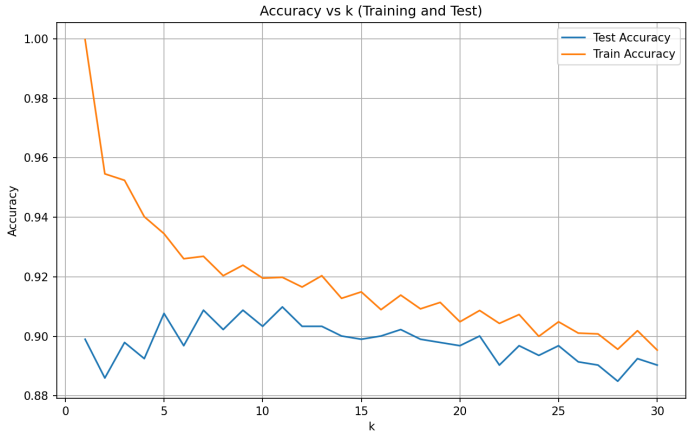

Figure 14: Training and validation accuracy for different k values

Metric	Gaussian NB	Multinomial NB	Bernoulli NB
Accuracy	0.83	0.78	0.88
Precision	0.71	0.72	0.87
Recall	0.96	0.71	0.80
F1 Score	0.82	0.71	0.84
Specificity	0.7509	0.8208	0.9229
Training Time (s)	0.0110	0.0104	0.0132

Naïve Bayes Performance Comparison

KNN Hyperparameter Tuning

Method	Best k	Best CV Accuracy	Best Parameters
Grid Search	15	0.9209	KDTree, uniform weights
Random Search	15	0.9209	Distance weights

KNN Performance Comparison

Metric	KDTree
Optimal k	15
Accuracy	0.9175
Precision	0.9065
Recall	0.8815
F1 Score	0.8939
Training Time (s)	0.0423
Prediction Time (s)	0.4848

Metric	BallTree
Optimal k	15
Accuracy	0.9175
Precision	0.9065
Recall	0.8815
F1 Score	0.8939
Training Time (s)	0.0120
Prediction Time (s)	0.2616

Overfitting and Underfitting Analysis

Based on the KNN hyperparameter tuning results and the Accuracy vs. k plot, the following observations were made:

- For small values of k, the KNN classifier shows high training accuracy but lower test accuracy, indicating overfitting.
- As the value of k increases, the model becomes smoother and generalizes better.
- Very large values of k reduce both training and test accuracy, leading to underfitting.
- The optimal value $k = 15$ provides a good balance between bias and variance.

Bias–Variance Analysis

The bias–variance characteristics of the classifiers were analyzed using the observed performance metrics:

- Naïve Bayes classifiers exhibit high bias due to their strong independence assumption, but have low variance and fast training time.
- KNN classifiers have low bias and high variance for small values of k.
- Increasing k reduces variance but increases bias.
- Hyperparameter tuning using cross-validation helps achieve an optimal bias–variance trade-off.

Conclusion

In this experiment, Naïve Bayes and K-Nearest Neighbors classifiers were successfully implemented for binary email classification. Bernoulli Naïve Bayes achieved better specificity among Naïve Bayes variants, while KNN achieved higher overall accuracy after hyperparameter tuning. The use of feature scaling significantly improved KNN performance, and KDTree and BallTree methods showed comparable accuracy with differences in computational efficiency.

References

- Scikit-learn: Naïve Bayes
- Scikit-learn: KNN
- Spambase Dataset