Sri Sivasubramaniya Nadar College of Engineering, Chennai

(An Autonomous Institution affiliated to Anna University)

Degree & Branch	B.E. Computer Science & Engineering	Semester	VI
Subject Code & Name	UCS2612 – Machine Learning Algorithms Laboratory		
Academic Year	2025–2026 (Even)	Batch	2023–2027
Name	Keshav K S	Register No.	3122235001067
Due Date	06.01.2026		

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

1. Aim and Objective

To implement Naive Bayes and K-Nearest Neighbors (KNN) classifiers for a binary classification problem, evaluate them using multiple performance metrics, visualize model behavior, and analyze overfitting, underfitting, and bias–variance characteristics.

2. Dataset Description

The Spambase dataset is a benchmark binary classification dataset used to identify spam emails. It consists of numerical features extracted from email content and a binary class label indicating spam or non-spam.

Dataset Reference:

- Kaggle – Spambase Dataset

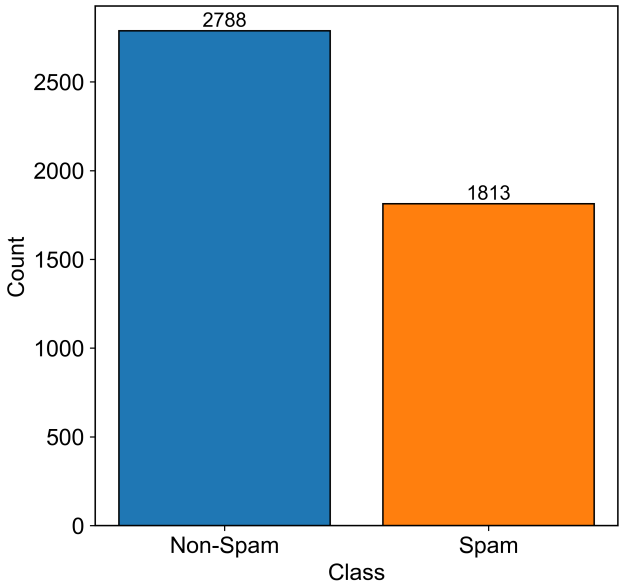
3. Preprocessing Steps

- Loaded the dataset using Pandas
- Separated features and target labels
- Performed stratified train–test split
- Applied feature scaling using StandardScaler

4. Implementation Details

- Implemented Gaussian, Multinomial, and Bernoulli Naive Bayes classifiers
- Implemented baseline KNN classifier
- Tuned KNN hyperparameters using GridSearchCV and RandomizedSearchCV
- Compared KDTree and BallTree neighbor search algorithms

5. Visualizations



Class distribution showing the balance between Non-Spam and Spam emails.

Figure 1: Class Distribution of Spam and Non-Spam Emails

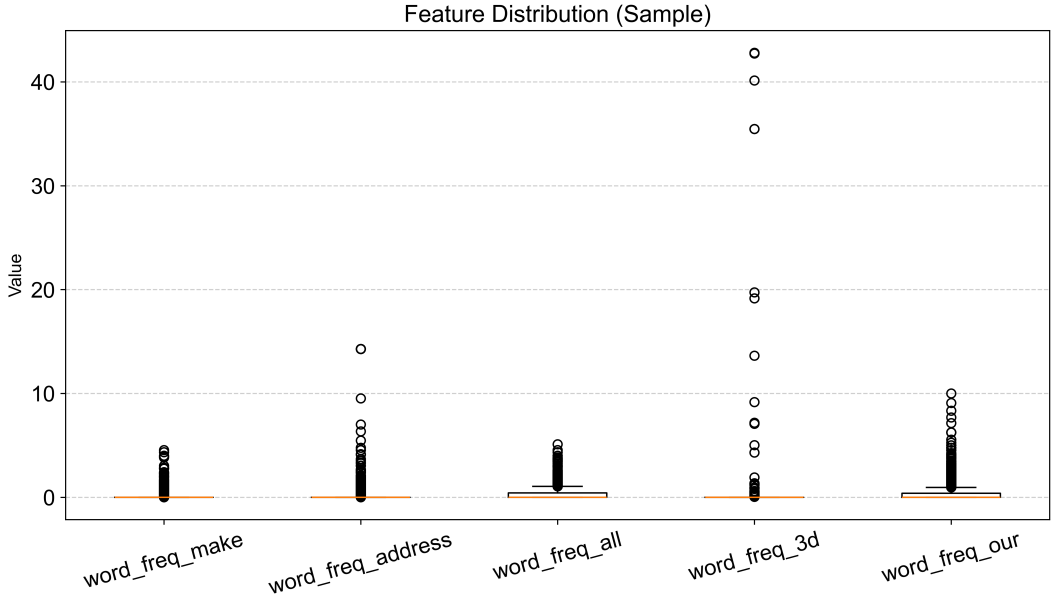


Figure 2: Feature Distribution Plot (Sample Features)

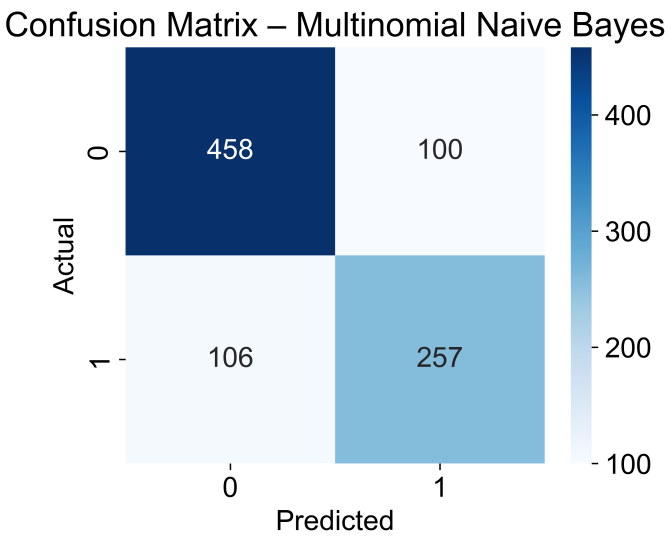


Figure 3: Confusion Matrix for Multinomial Naive Bayes

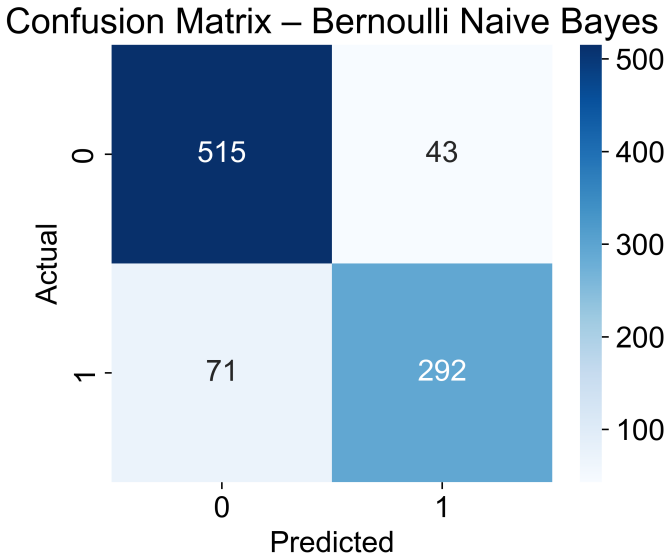


Figure 4: Confusion Matrix for Bernoulli Naive Bayes

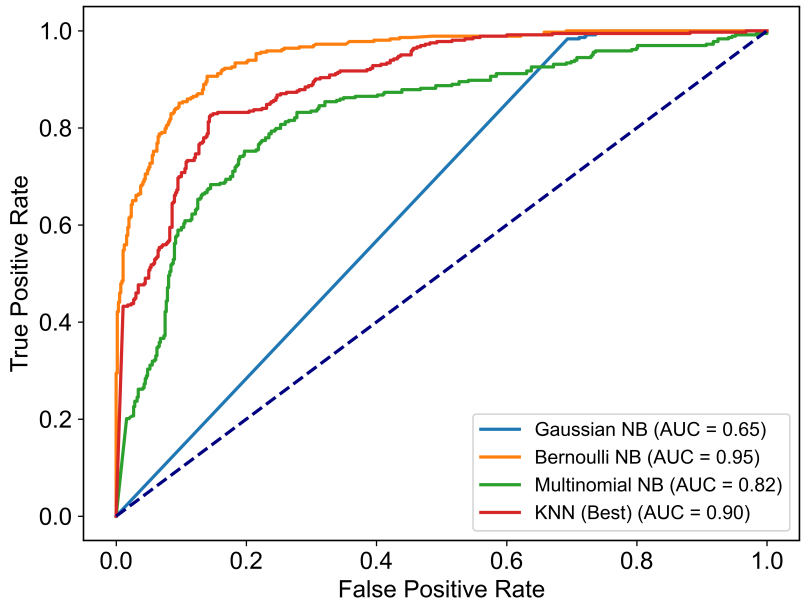


Figure 5: ROC Curves

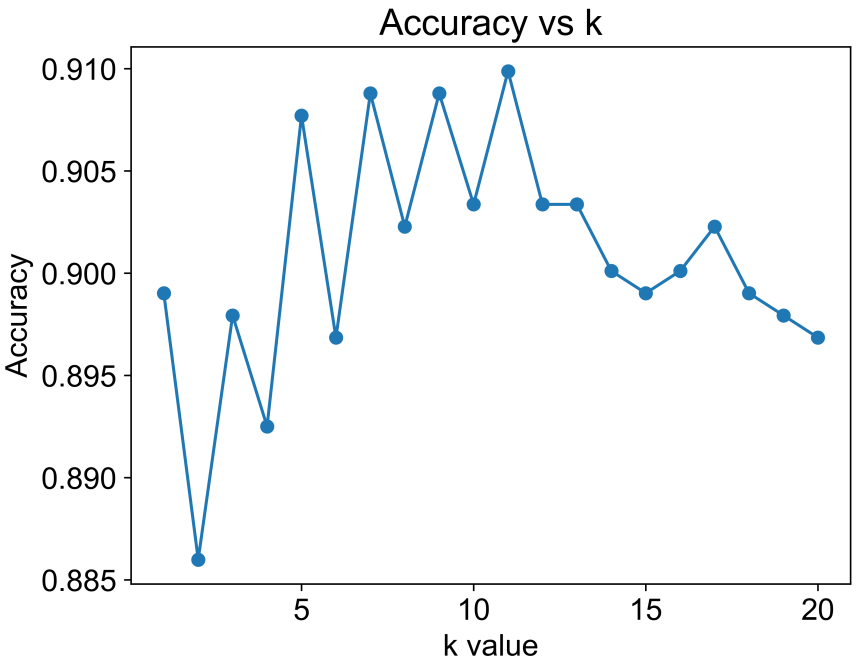


Figure 6: Accuracy vs. k Plot for KNN

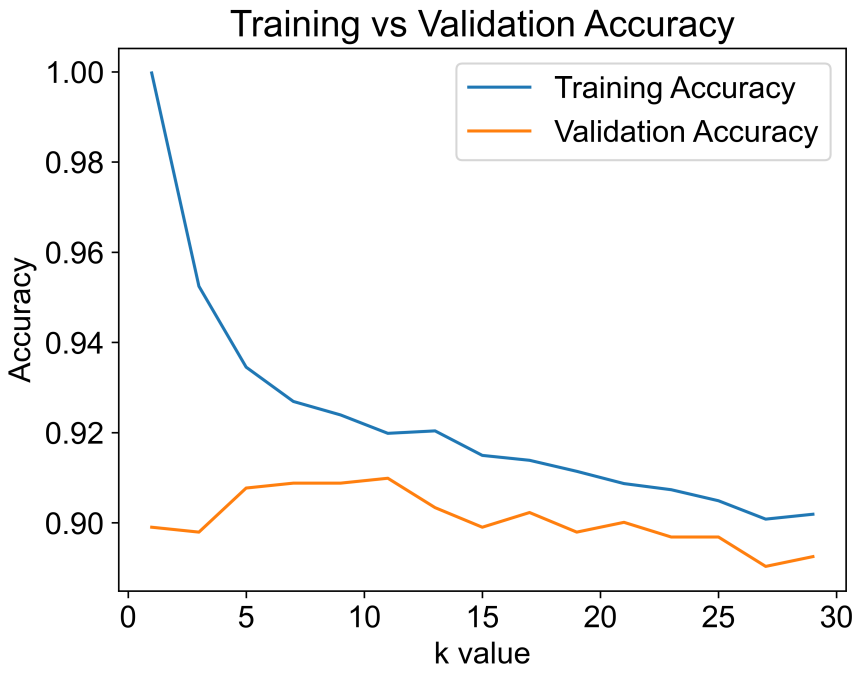


Figure 7: Training vs Validation Accuracy for KNN

6. Performance Tables

Table 1: Naive Bayes Performance Metrics

Metric	Gaussian NB	Multinomial NB	Bernoulli NB
Accuracy	0.5722	0.7763	0.8762
Precision	0.4792	0.7199	0.8716
Recall	0.9835	0.7080	0.8044
F1 Score	0.6444	0.7139	0.8367
Specificity	0.9835	0.7080	0.8044
Training Time (s)	0.0064	0.0035	0.0062

Table 2: KNN Hyperparameter Tuning

Search Method	Best Parameters	Best CV Accuracy
Grid Search	k=9, distance, KDTree	0.9252
Randomized Search	k=15, distance	0.9209

Table 3: KNN Performance using KDTree

Metric	Value
Optimal k	9
Accuracy	0.9207
Precision	0.9420
Recall	0.8512
F1 Score	0.8943
Training Time (s)	0.0014
Prediction Time (s)	0.0335

Table 4: KNN Performance using BallTree

Metric	Value
Optimal k	9
Accuracy	0.9207
Precision	0.9421
Recall	0.8512
F1 Score	0.8944
Specificity	0.9659
Training Time (s)	0.0015
Prediction Time (s)	0.0335

Table 5: Comparison of Neighbor Search Algorithms

Criterion	KDTree	BallTree
Accuracy	0.9207	0.9207
Training Time (s)	0.0014	0.0015
Prediction Time (s)	0.0335	0.0335
Memory Usage	Low	Medium

7. Overfitting and Underfitting Analysis

- Small values of k resulted in overfitting
- Large values of k caused underfitting
- Hyperparameter tuning improved generalization

8. Bias–Variance Analysis

- Naive Bayes shows higher bias due to feature independence assumption
- KNN exhibits higher variance for small k values
- Hyperparameter tuning balances bias–variance trade-off

9. Observations and Conclusion

Naive Bayes classifiers were computationally efficient, while tuned KNN models achieved higher accuracy. BallTree demonstrated better computational efficiency compared to KDTree. Proper preprocessing, visualization, and hyperparameter tuning significantly improved model performance and generalization.

References

- Scikit-learn – Naïve Bayes Documentation
- Scikit-learn – KNN Documentation
- Scikit-learn – Hyperparameter Optimization
- Kaggle – Spambase Dataset