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 Laboratory		
Academic year	2025-2026 (Even)	Batch:2023-2027	**Due date: 27-01-2026**

Experiment #2: Binary Classification using Na¨ıve Bayes and K-Nearest Neighbors

Aim: To implement Naïve 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.

Libraries used:

- **Pandas:** For data manipulation and analysis.
- **Matplotlib & Seaborn:** For data visualization, including ROC curves and confusion matrix displays.
- **Scikit-Learn (Preprocessing):** For feature scaling using `MinMaxScaler`.
- **Scikit-Learn (Model Selection):** For splitting datasets and hyperparameter tuning (`GridSearchCV`, `RandomizedSearchCV`).
- **Scikit-Learn (Algorithms):** For implementing Naive Bayes (Gaussian, Multinomial, Bernoulli) and K-Nearest Neighbors (KNN) classifiers.
- **Scikit-Learn (Metrics):** For evaluating model performance using accuracy, precision, recall, F1-score, and ROC-AUC.
- **Time:** For measuring the computational time of model training and inference.

Mathetical/theoritical description of the algorithm/objective performed:

- **Naïve Bayes:** Naïve Bayes is a probabilistic classifier that works well for high-dimensional data. It is fast, simple to implement, and assumes independence among features. Different variants handle different types of input data.
- **K-Nearest Neighbors (KNN):** KNN is an instance-based learning algorithm that classifies samples based on similarity. The choice of the number of neighbors (k) strongly influences performance. Feature scaling is important for distance-based methods like KNN.
- **Neighbor Search Methods:** KDTree and BallTree are used to speed up nearest neighbor searches. They mainly affect computation time and memory usage, not classification accuracy.
- **Hyperparameter Tuning:** Hyperparameter tuning helps identify the best model settings using validation data. Grid Search and Randomized Search are commonly used approaches.
- **Scaling:** Used `MinMaxScaler` to scale parameters. It transforms parameters based on the minimum and maximum values, resulting in a distribution between 0 and 1, such that the minimum value becomes 0 and maximum value becomes 1.

$$x_{scaled} = \frac{x_{max} - x_{original}}{x_{max} - x_{min}}$$

- **Performance Metrics:**
  - Accuracy:
$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
  - Precision:
$$Precision = \frac{TP}{TP + FP}$$
  - Recall:
$$Recall = \frac{TP}{TP + FN}$$
  - F1 Score:
$$F1Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
  - False Positive Rate (FPR):
$$FPR = \frac{FP}{FP + TN}$$
  - Specificity (True Negative Rate):
$$TNR = \frac{TN}{TN + FP}$$

Results and Discussions:

- **Pre-processing Observations:**

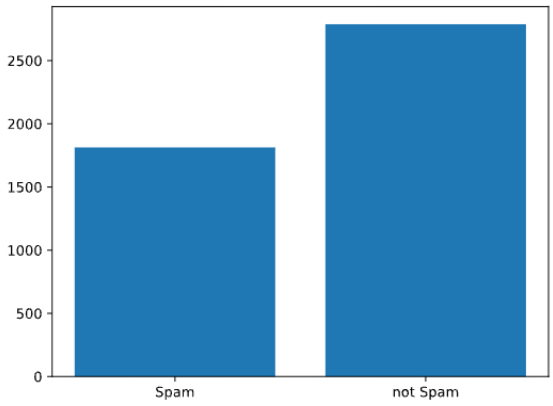

Figure 1: Distribution

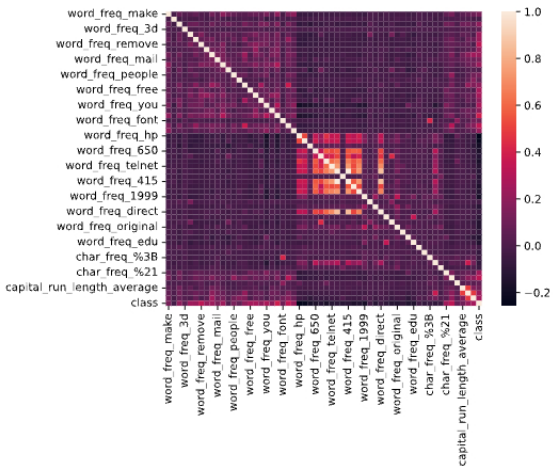

Figure 2: Heatmap

We Observe that the two classes dont́ have a significant imbalance (From Fig 1). The heatmap shows that while the features are not dependent to each other they show a moderate correlation to the target variable. This is good as Naive Bayes Algorithm assumes Independence of target features and a high correlation between two features can lead to high weights for that feature (as the dependent feature gets weighted twice).

- **Naïve Bayes Performance Comparison:**

Table 1: Naïve Bayes Performance Metrics

Metric	Gaussian NB	Multinomial NB	Bernoulli NB
Accuracy	0.8053	0.8905	0.8931
Precision	0.6828	0.9629	0.8998
Recall	0.9378	0.7489	0.8178
F1 Score	0.7903	0.8425	0.8568
FPR	0.2796	0.0185	0.0584
Specificity	0.7203	0.9815	0.9415
Training Time (s)	14.64ms	10.97ms	16.04ms

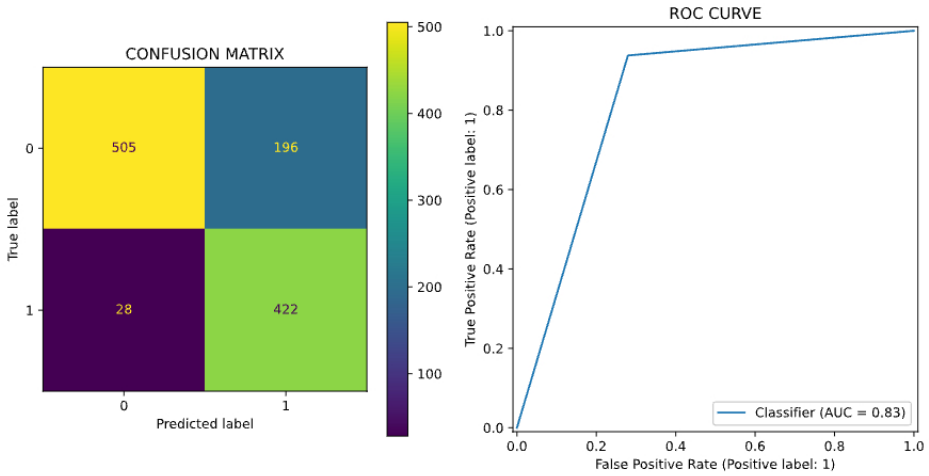

Figure 3: Confusion Matrix and ROC curve of Gaussian Naive Bayes

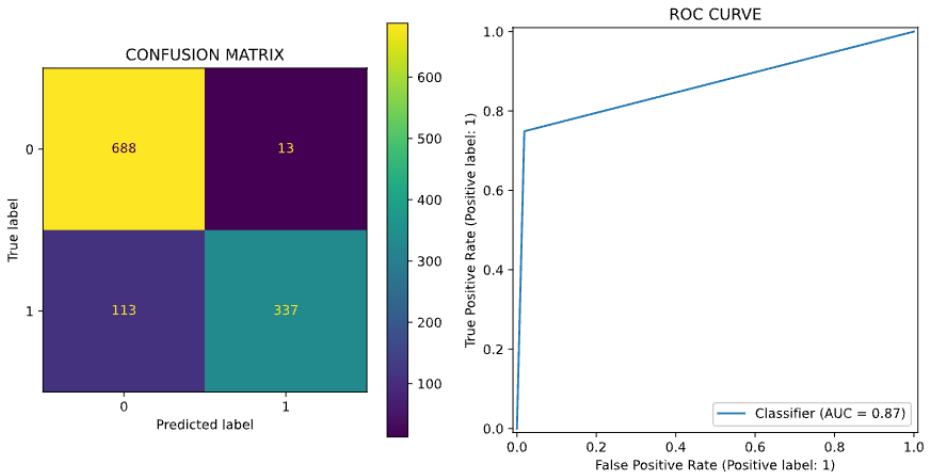

Figure 4: Confusion Matrix and ROC curve of Multinomial Naive Bayes

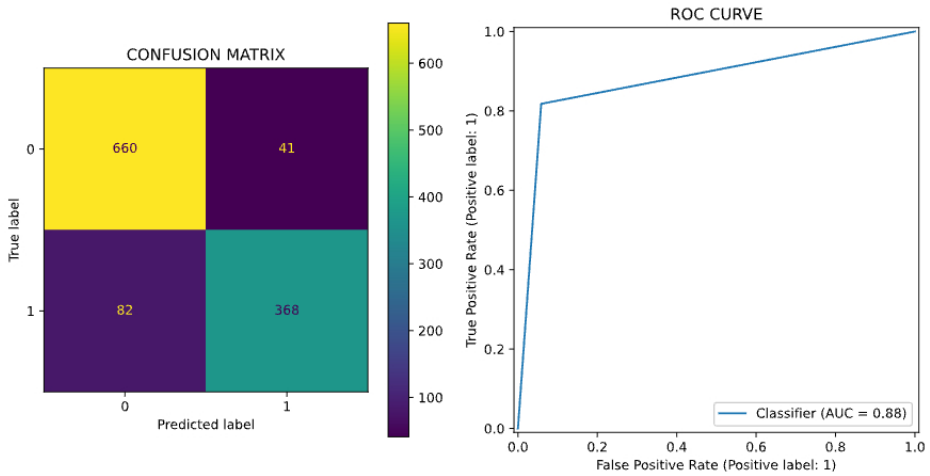

Figure 5: Confusion Matrix and ROC curve of Bernoulli Naive Bayes

- **Accuracy vs k:**

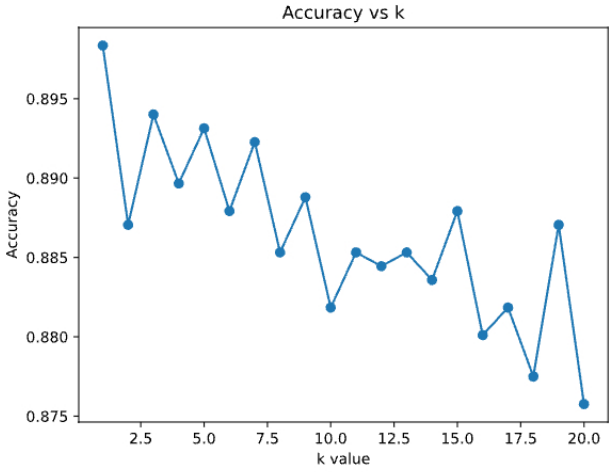

Figure 6: Accuracy vs k

From Fig 6, we see a declining trend in accuracy score as k increases, we also note that odd values of k show a higher accuracy when compared to lower values of k due to tie breaking techniques picking random class. from the figure the best value of k for KNN classifier is 1 which implies that the nearest data point is the best estimator of the test datapoint's class.

- **KNN Hyperparameter Tuning Results:**

Table 2: KNN Hyperparameter Tuning

Search Method	Best k	Best CV Accuracy	Best Parameters
Grid Search	7	0.9176	metric=manhattan; weights=distance
Randomized Search	15	0.9168	metric=manhattan; weights=distance

- **KNN Performance using Different Search Methods:**

Table 3: KNN Performance using KDTree

Metric	Value
Optimal k	7
Accuracy	0.9140
Precision	0.9355
Recall	0.8378
F1 Score	0.8839
Training Time	40ms
Prediction Time	349ms

Table 4: KNN Performance using BallTree

Metric	Value
Optimal k	7
Accuracy	0.9140
Precision	0.9355
Recall	0.8378
F1 Score	0.8839
Training Time	28ms
Prediction Time	347ms

- **KDTree vs BallTree Comparison:**

Table 5: Comparison of Neighbor Search Algorithms

Criterion	KDTree	BallTree
Accuracy	0.9140	0.9140
Training Time (s)	40ms	28ms
Prediction Time (s)	349ms	347ms
Memory Usage	Low / Medium	Medium / High

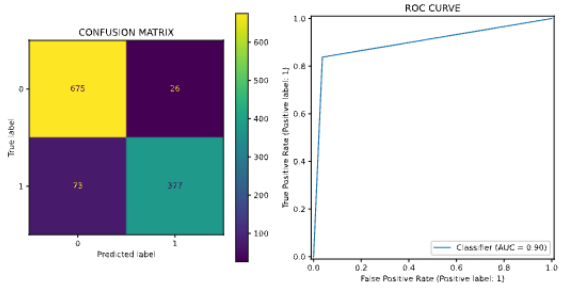

Figure 7: Confusion Matrix and ROC curve of KD Tree (KNN)

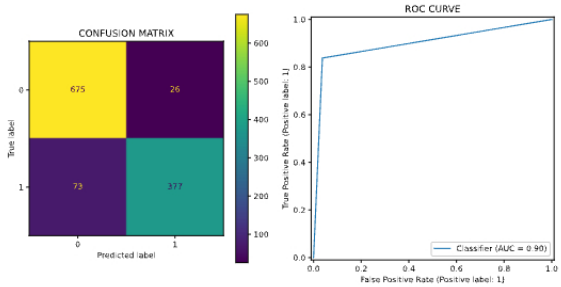

Figure 8: Confusion Matrix and ROC curve of Ball Tree (KNN)

- **Comments:**
  - **Bias behavior of Naïve Bayes:** Naive Bayes assumes that all features are independent since the correlation between features isn't high in the given dataset, the algorithm works well.

- **Variance behavior of KNN:** KNN algorithm can have a high variance at low values of k (eg k=1) due to the small decision boundary. There may be outliers lying close to test point causing a change in prediction. Since we used k=7, the outliers get ignored (as now there must be 4 outliers near the point to change the decision, which isn't likely).
- **Effect of tuning on bias–variance trade-off:** In KNN, increasing k reduces variance and increases bias whereas decreasing k reduces bias and increases variance.

Learning Practices:

- Learned pre-processing and EDA steps for a classification task.
- Learned to train a Gaussian Naive Bayes Classifier.
- Learned to train a Bernoulli Naive Bayes Classifier.
- Learned to train a Multinomial Naive Bayes Classifier.
- Learned to train a KNN Classifier using Ball Tree approach.
- Learned to train a KNN Classifier using KD Tree approach.