Sri Sivasubramaniya Nadar College of Engineering, Chennai

(Autonomous Institution under Anna University)

Degree & Branch	5 years Integrated M.Tech CSE	**Semester**	V
Subject Code & Name	ICS1512 – Machine Learning Algorithms Laboratory		
Academic Year	2025–2026 (Odd Semester)	**Batch**	2023–2028
Name	Pravin G	**Reg No**	3122237001041

Experiment # 3: Email Spam or Ham Classification using Naive Bayes, KNN, and SVM

Aim:

To classify emails as spam or ham using three classification algorithms—Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM)—and evaluate their performance using accuracy metrics and K-Fold cross-validation.

Libraries used:

- Numpy
- Pandas
- Scipy
- Scikit-Learn
- Matplotlib.pyplot

Description of the objective performed

- **Data Preparation:** Loaded dataset using kagglehub.dataset download() and converted it into a Pandas DataFrame.
- **Exploratory Data Analysis (EDA):**
  - Performed Numerical Column analysis using histogram and pdf
  - Performed Categorical column analysis using One way ANOVA test
  - Visualized Missing Values
  - Visualized distributions and relationships using:
    * plt.hist() for histograms
    * plt.scatter() for 2D scatter plots
    * sns.heatmap() for feature correlation matrix

- **Data Preprocessing :**
  - Handled Missing Values
  - Outlier Treatment.
  - Encoding categorical column values
  - Standardize
- **Modeling**
  - K-Fold cross validation
  - Hyper Parameter Tuning
  - Model Fitting
- **Evaluation and Visualization**
  - Metrics Accuracy, Precision, Recall, F1-score
  - Visualization Confusion Matrix ROC Curve

Mathematical Description

Support Vector Machine (SVM)

Given a set of training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathbf{R}^d$ and $y_i \in \{-1, +1\}$, the primal optimization problem for the soft-margin SVM is formulated as:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{subject to:} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n$$

Here:

- $\mathbf{w}$: weight vector defining the hyperplane
- b: bias term
- ξ_i: slack variables for misclassification
- $C > 0$: regularization parameter

The decision function is:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

—

k-Nearest Neighbors (kNN)

For a given query point $\mathbf{x}$, let $N_k(\mathbf{x})$ denote the set of k training points nearest to $\mathbf{x}$ (measured using a distance metric such as Euclidean distance):

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{d} (x_{i,l} - x_{j,l})^2}$$

The predicted class $\hat{y}$ is obtained by majority voting:

$$\hat{y} = \arg\max_{c \in C} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} \mathbf{1}(y_i = c)$$

where C is the set of possible classes and $\mathbf{1}(\cdot)$ is the indicator function.

—

Naïve Bayes (NB)

Given a feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and class labels $y \in C$, the Naïve Bayes classifier applies Bayes' theorem with the conditional independence assumption:

$$P(y \mid \mathbf{x}) = \frac{P(y) \prod_{j=1}^{d} P(x_j \mid y)}{\sum_{y' \in C} P(y') \prod_{j=1}^{d} P(x_j \mid y')}$$

The predicted class is:

$$\hat{y} = \arg\max_{y \in C} P(y) \prod_{j=1}^{d} P(x_j \mid y)$$

Depending on the feature distribution assumption:

- **Gaussian NB:** $P(x_j \mid y) = \frac{1}{\sqrt{2\pi\sigma_{y,j}^2}} \exp\left(-\frac{(x_j - \mu_{y,j})^2}{2\sigma_{y,j}^2}\right)$
- **Multinomial NB:** suitable for discrete counts
- **Bernoulli NB:** suitable for binary features

Plots Included

Confusion Matrix

Gaussian Naive Bayes

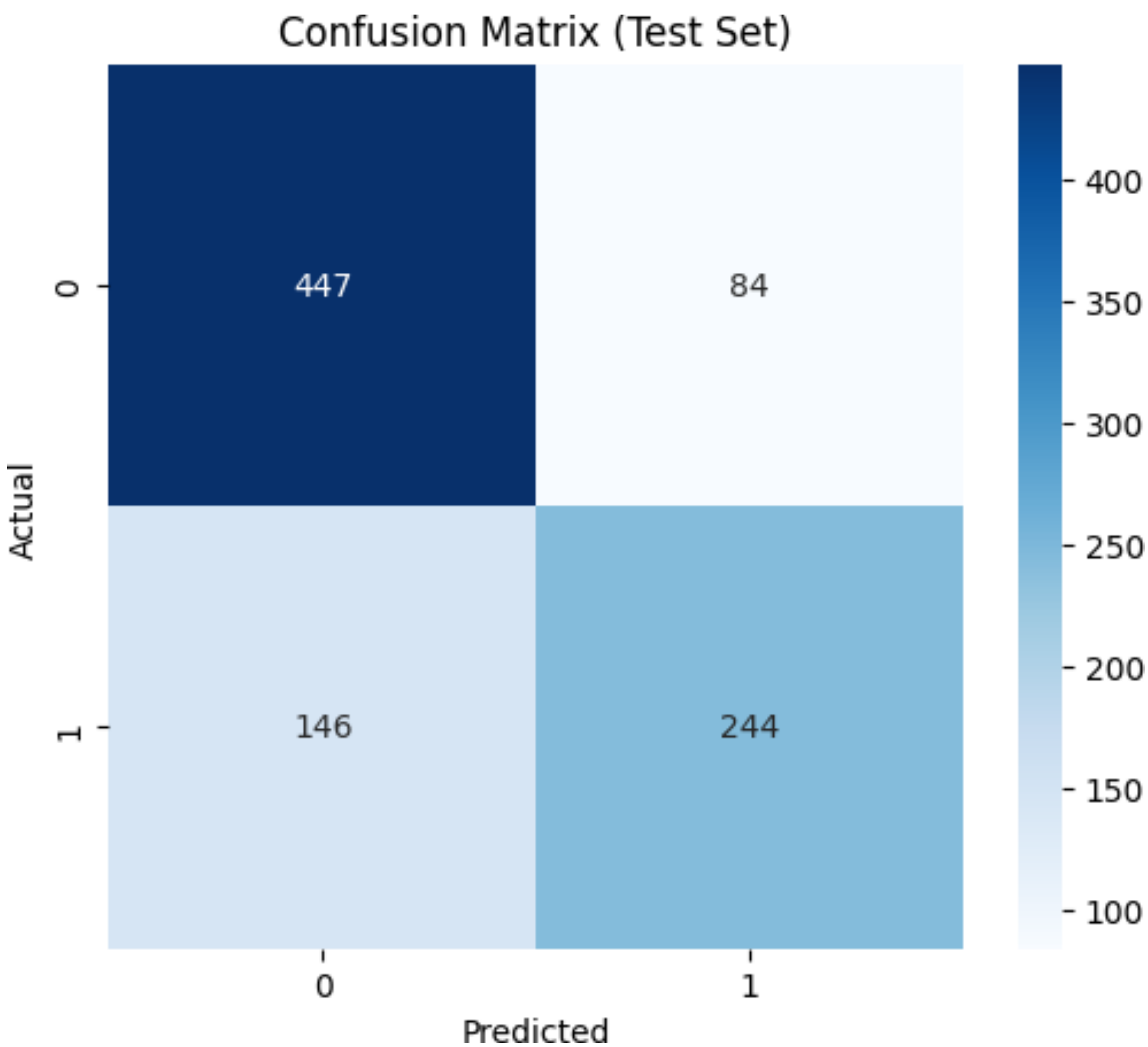

Figure 1: Gaussian Naive Bayes

Multinominal Naive Bayes

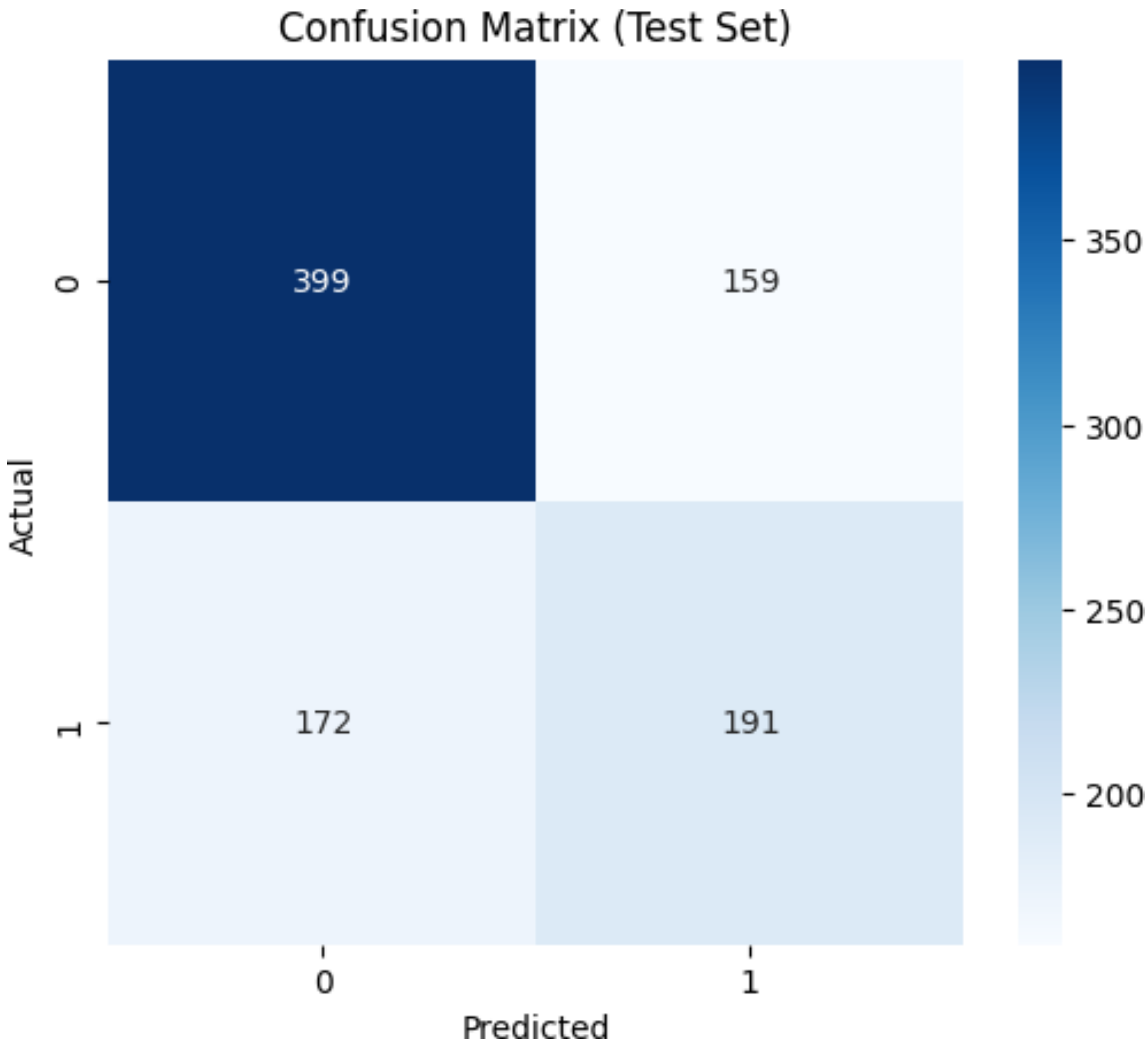

Figure 2: Multinominal Naive Bayes

Bernoulli Naive Bayes

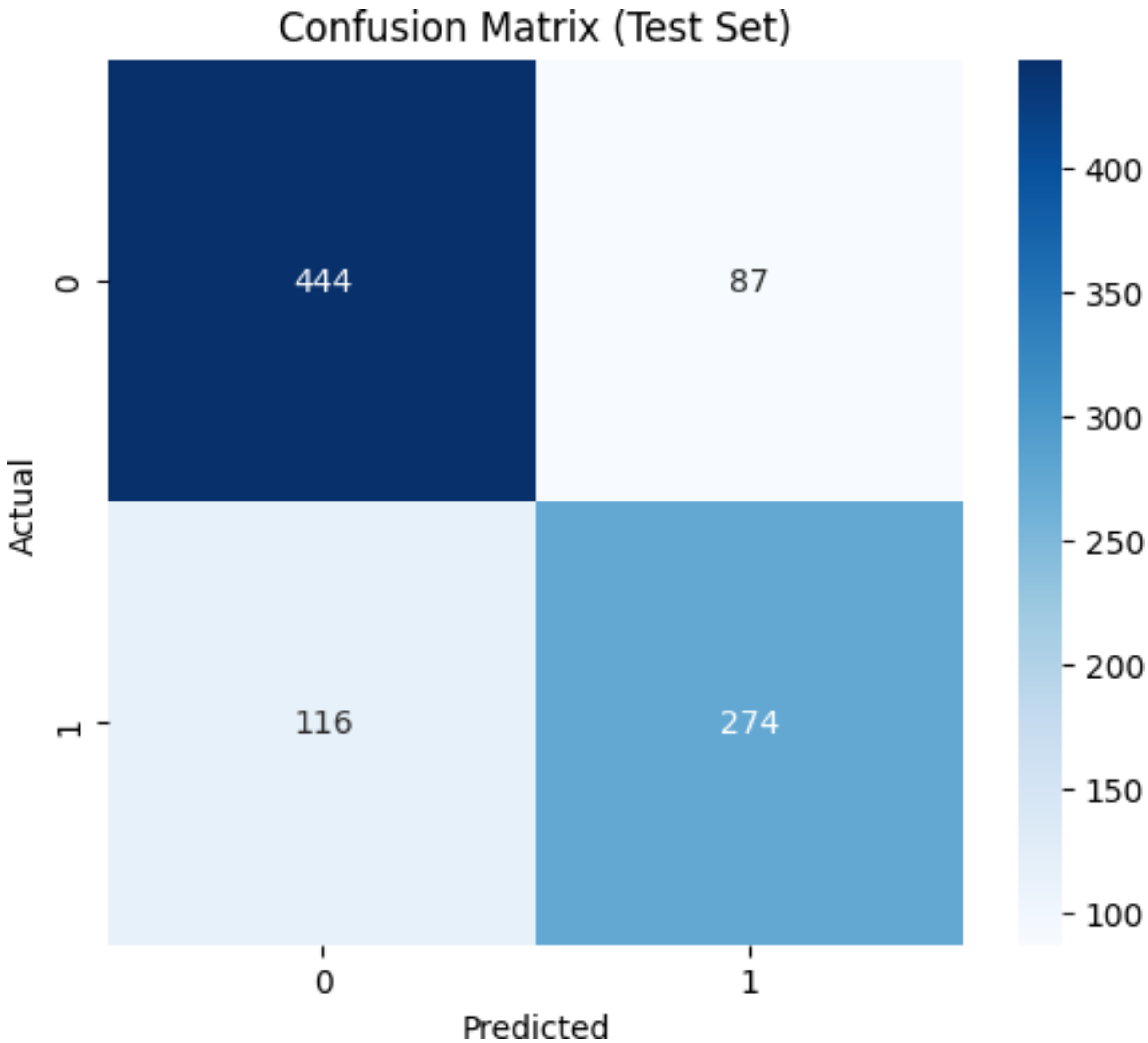

Figure 3: Bernoulli Naive Bayes

KNN

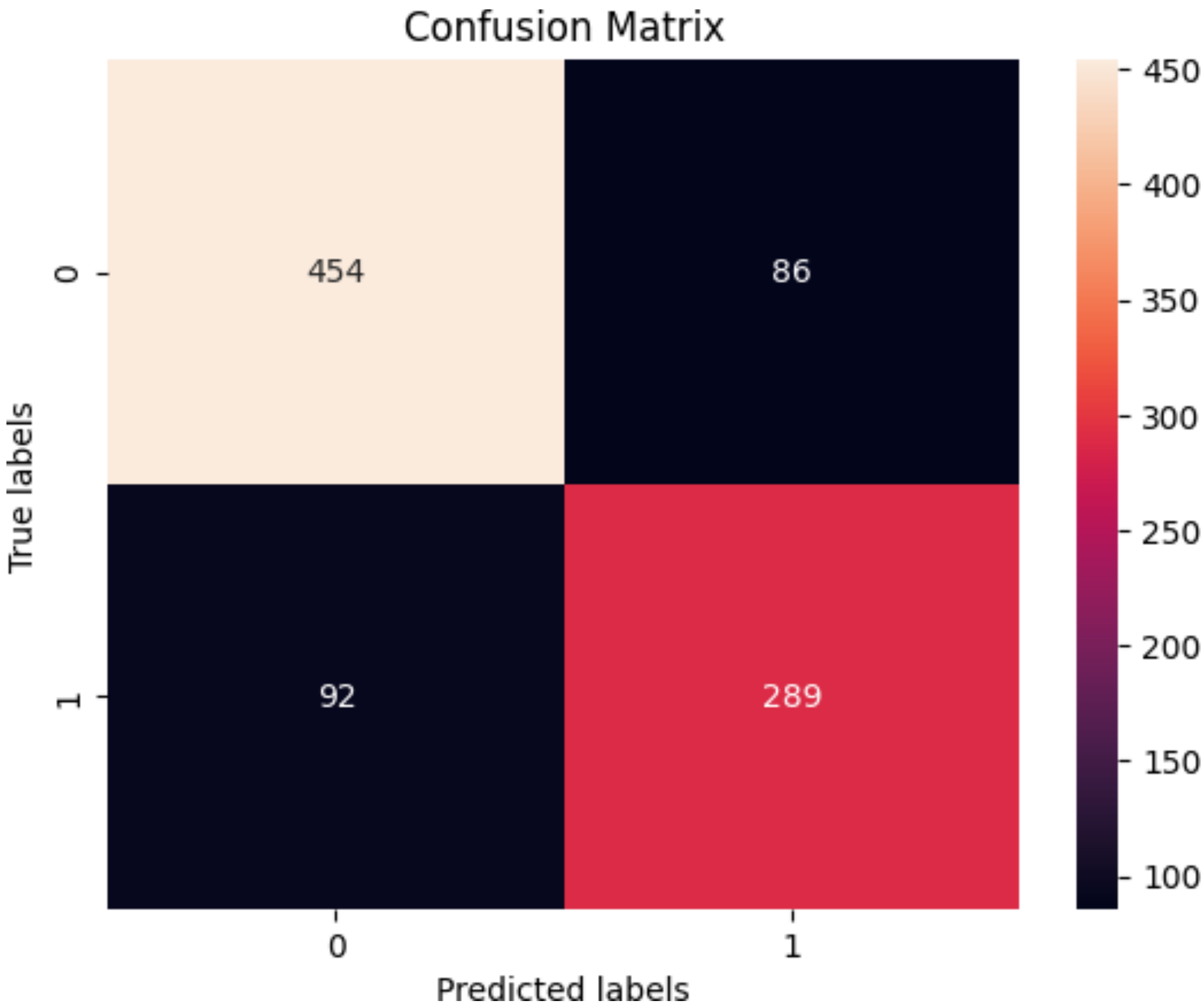

Figure 4: KNN

KDTree

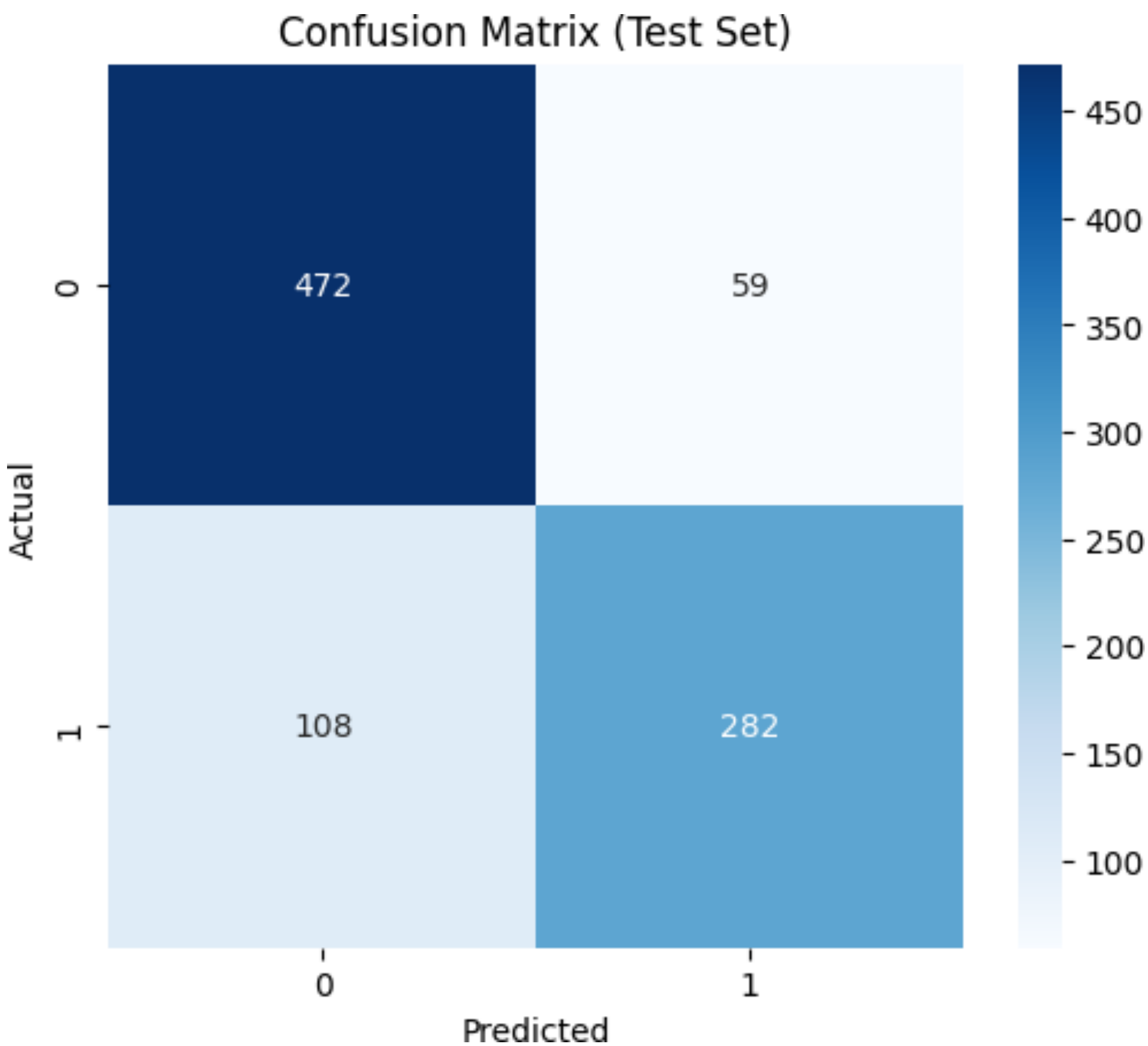

Figure 5: KDTree

Ball Tree

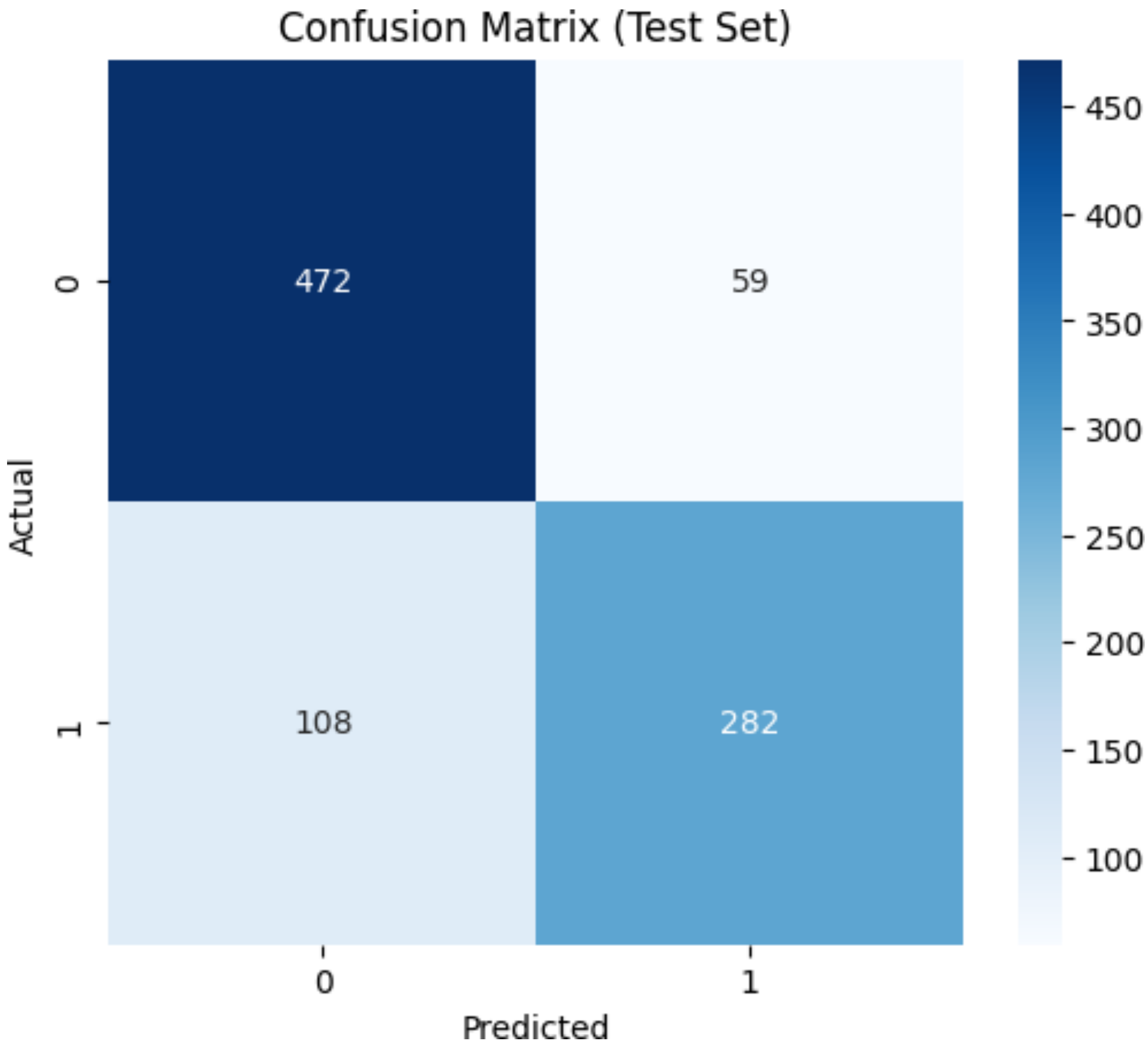

Figure 6: Ball Tree

Linear SVM

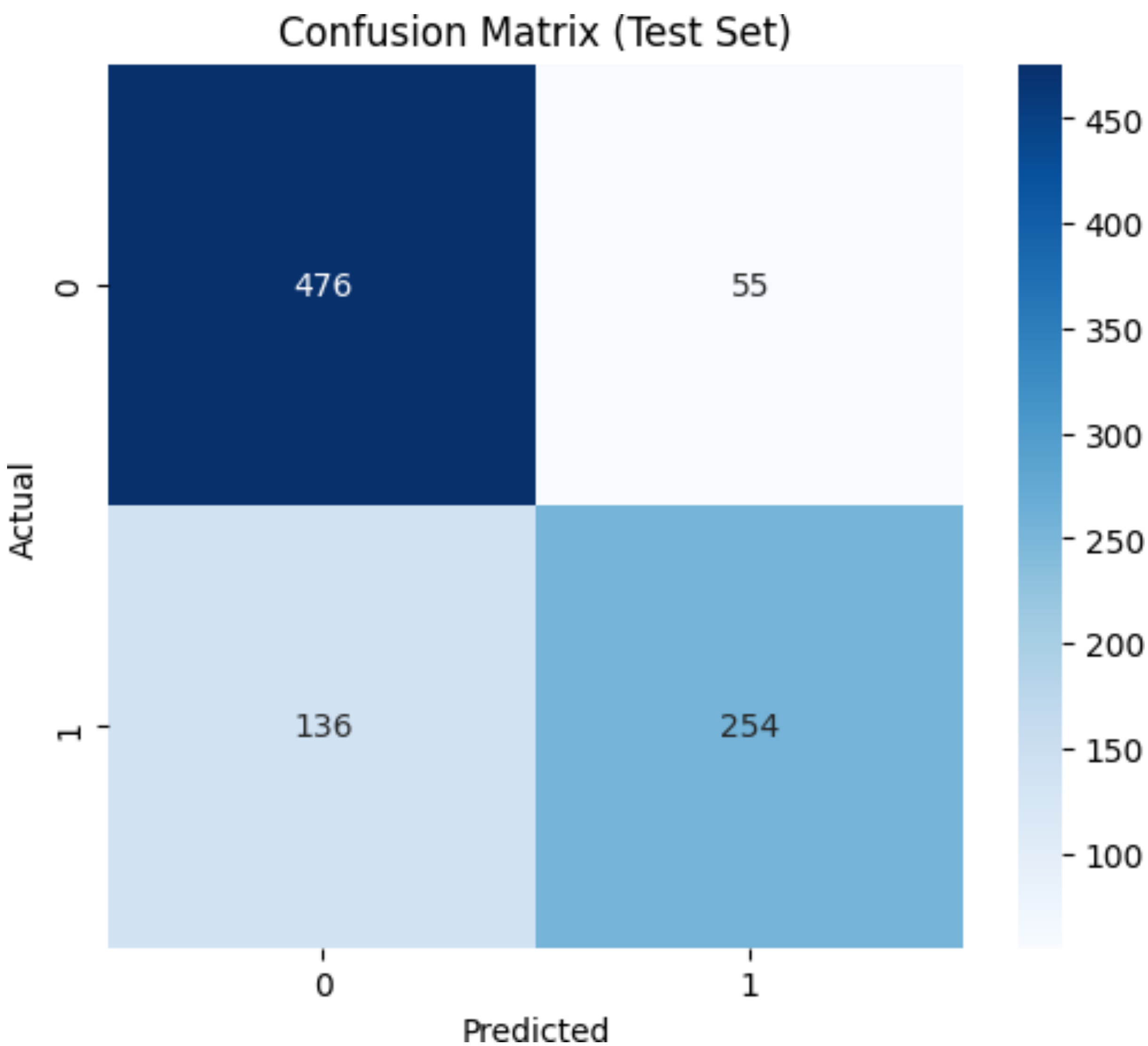

Figure 7: Linear SVM

RBF SVM

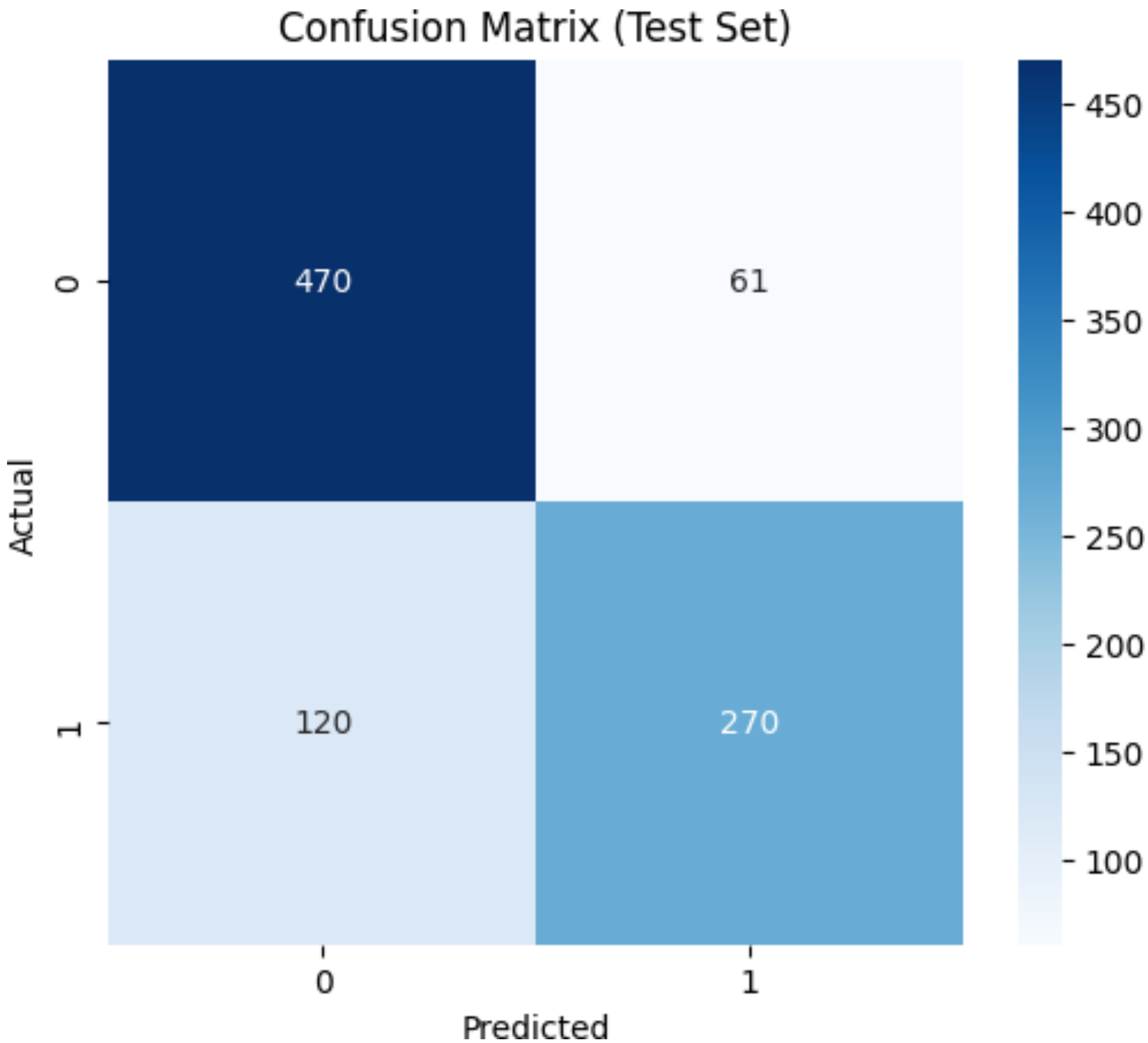

Figure 8: RBF SVM

Sigmoid SVM

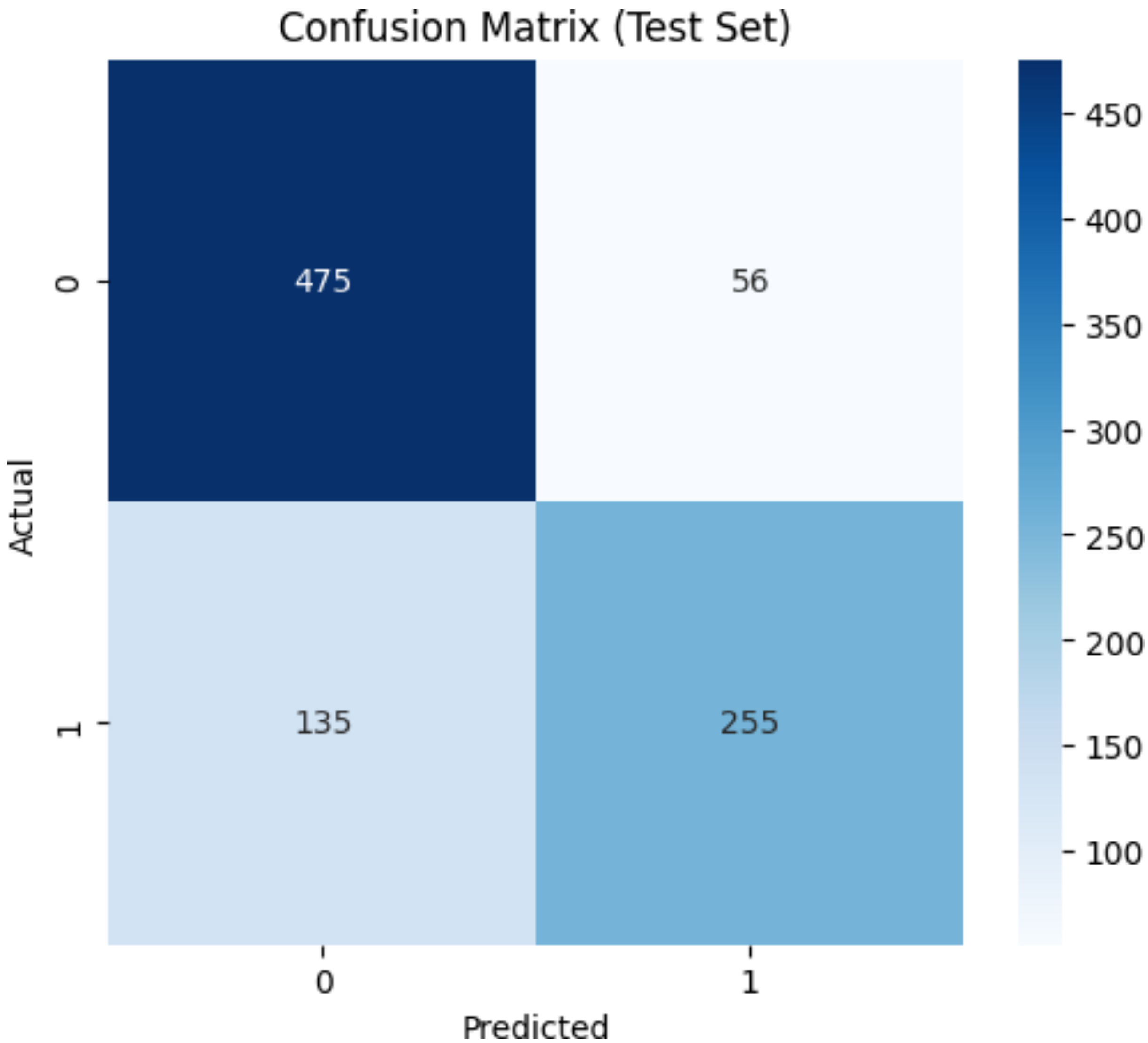

Figure 9: Sigmoid SVM

ROC Curve

Gaussian Naive Bayes

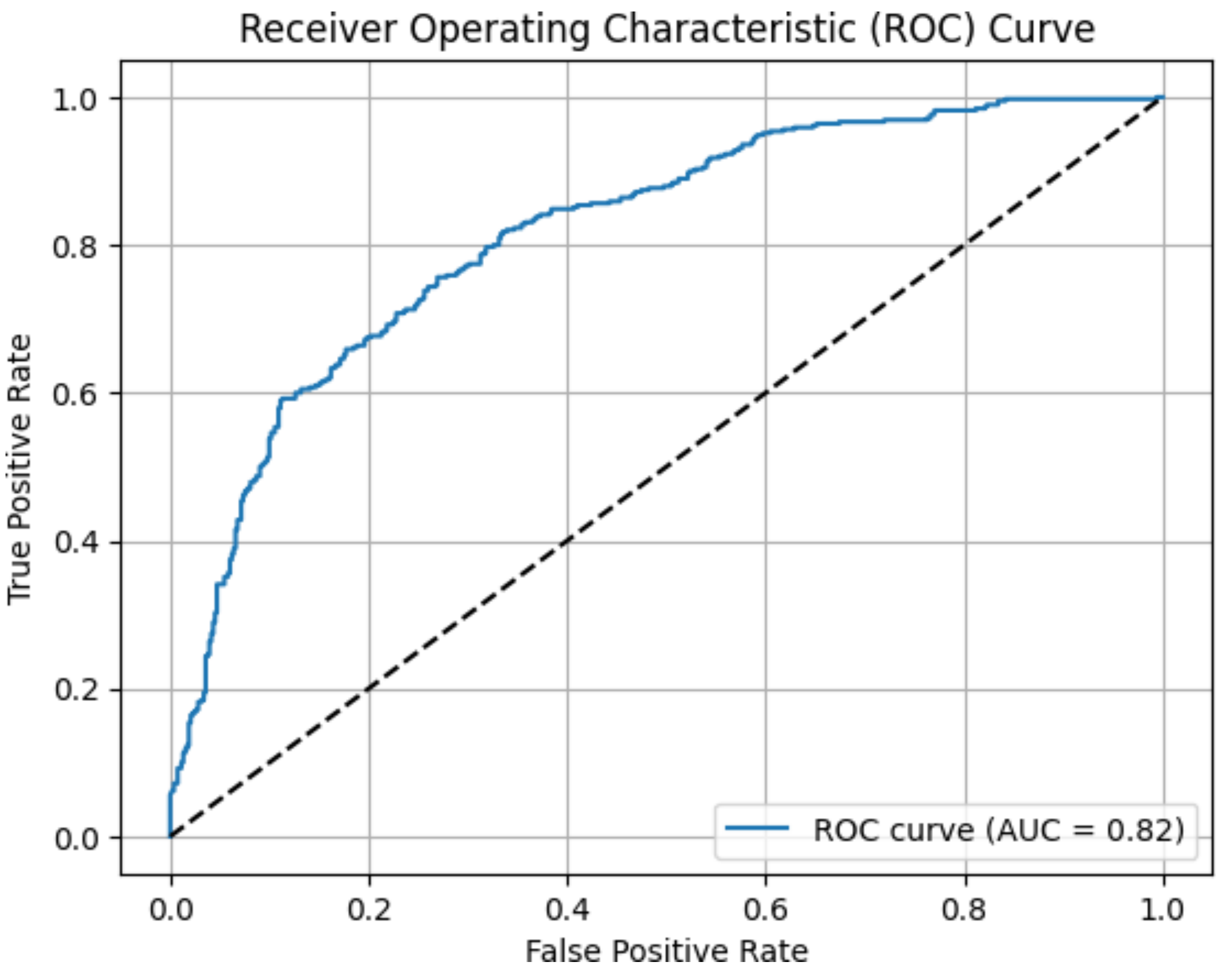

Figure 10: Gaussian Naive Bayes

Multinominal Naive Bayes

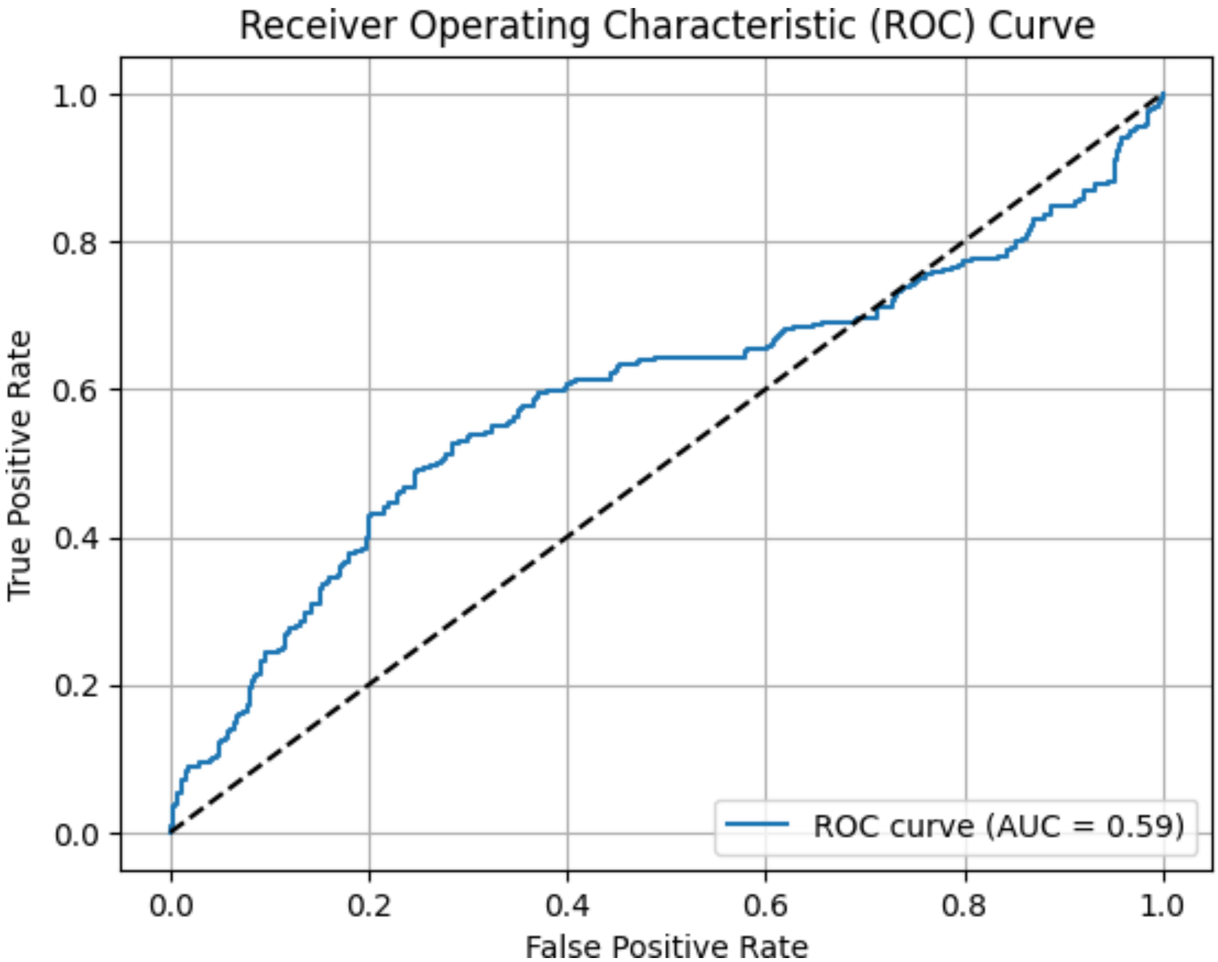

Figure 11: Multinominal Naive Bayes

Bernoulli Naive Bayes

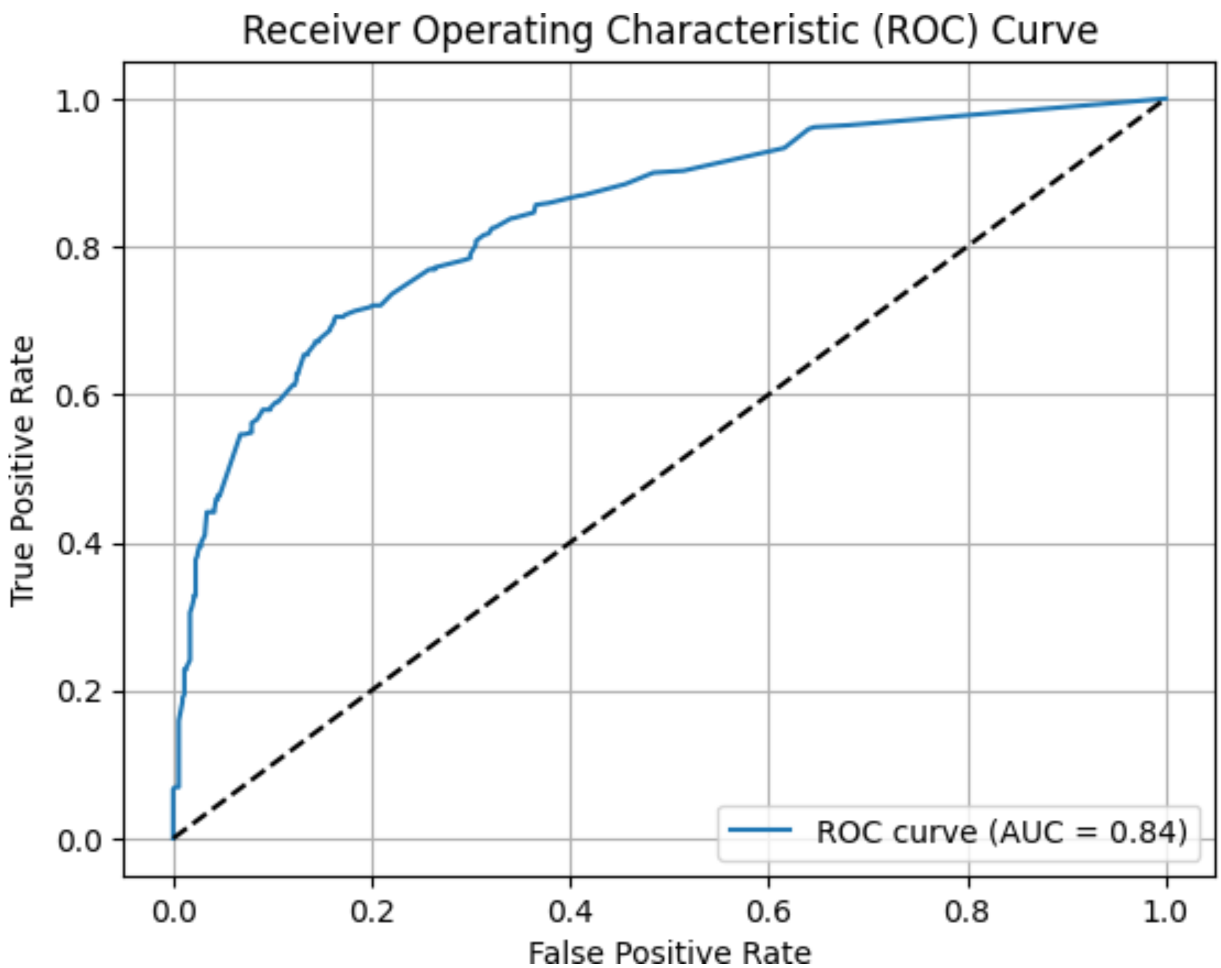

Figure 12: Bernoulli Naive Bayes

KNN

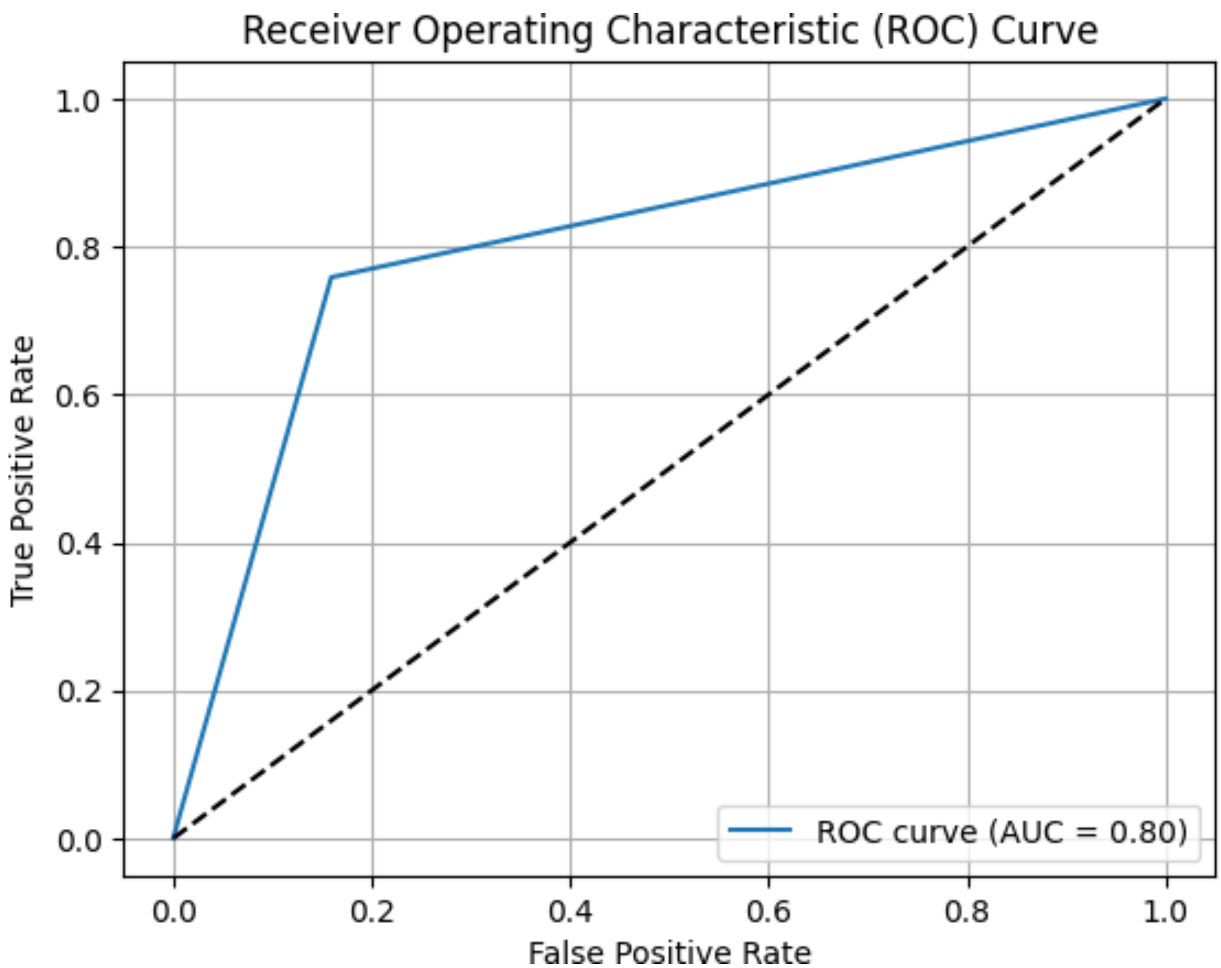

Figure 13: KNN

KDTree

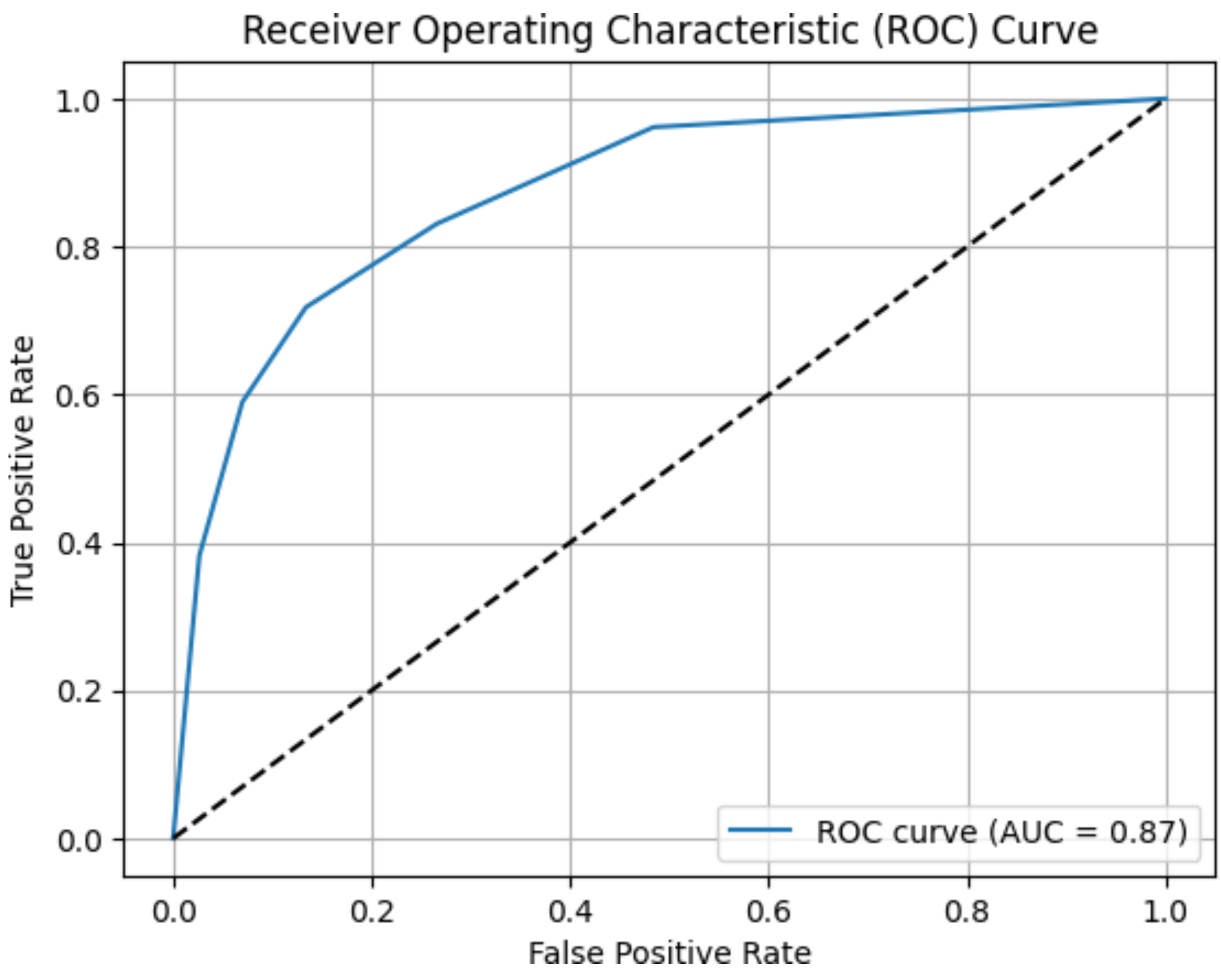

Figure 14: KDTree

Ball Tree

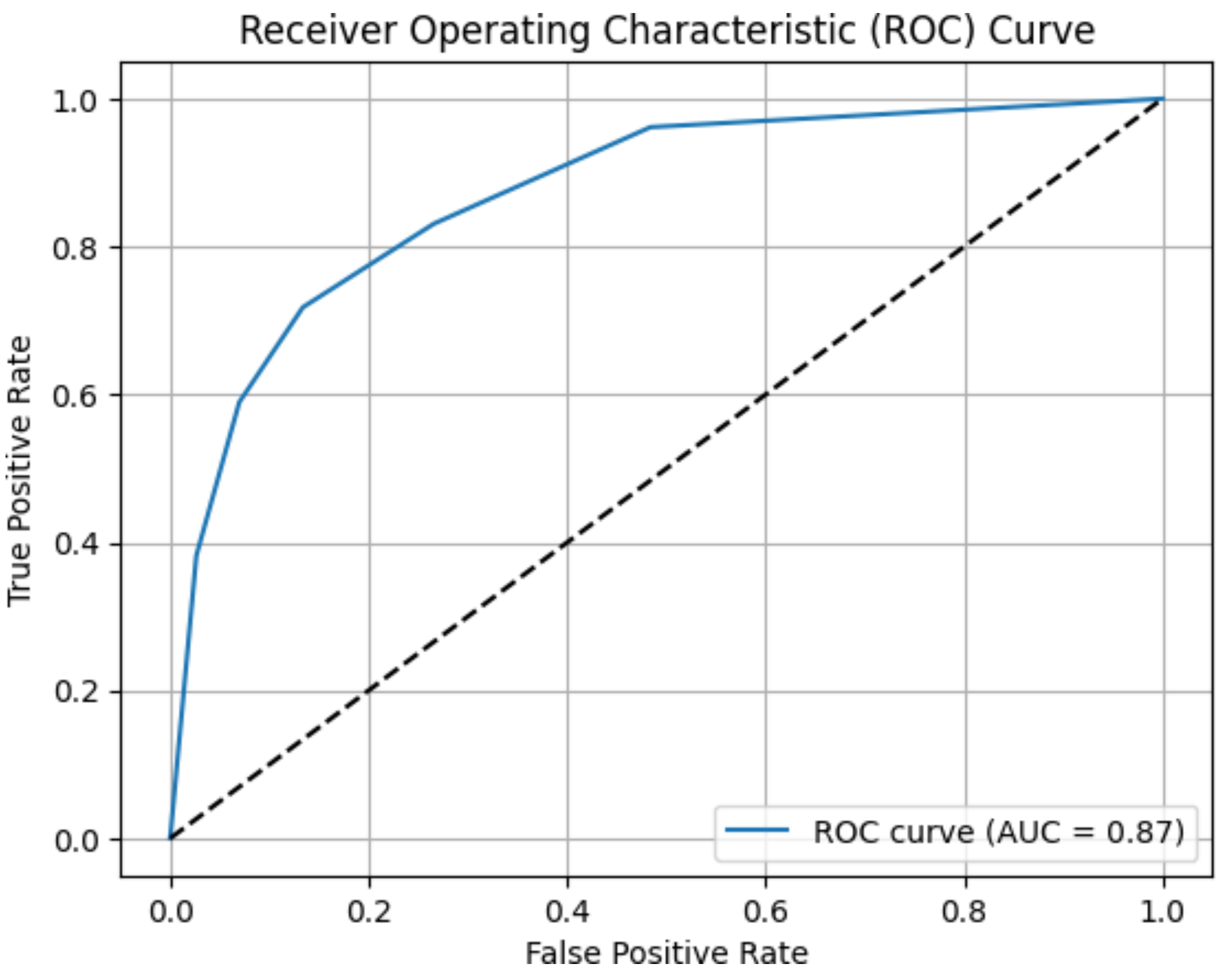

Figure 15: Ball Tree

Linear SVM

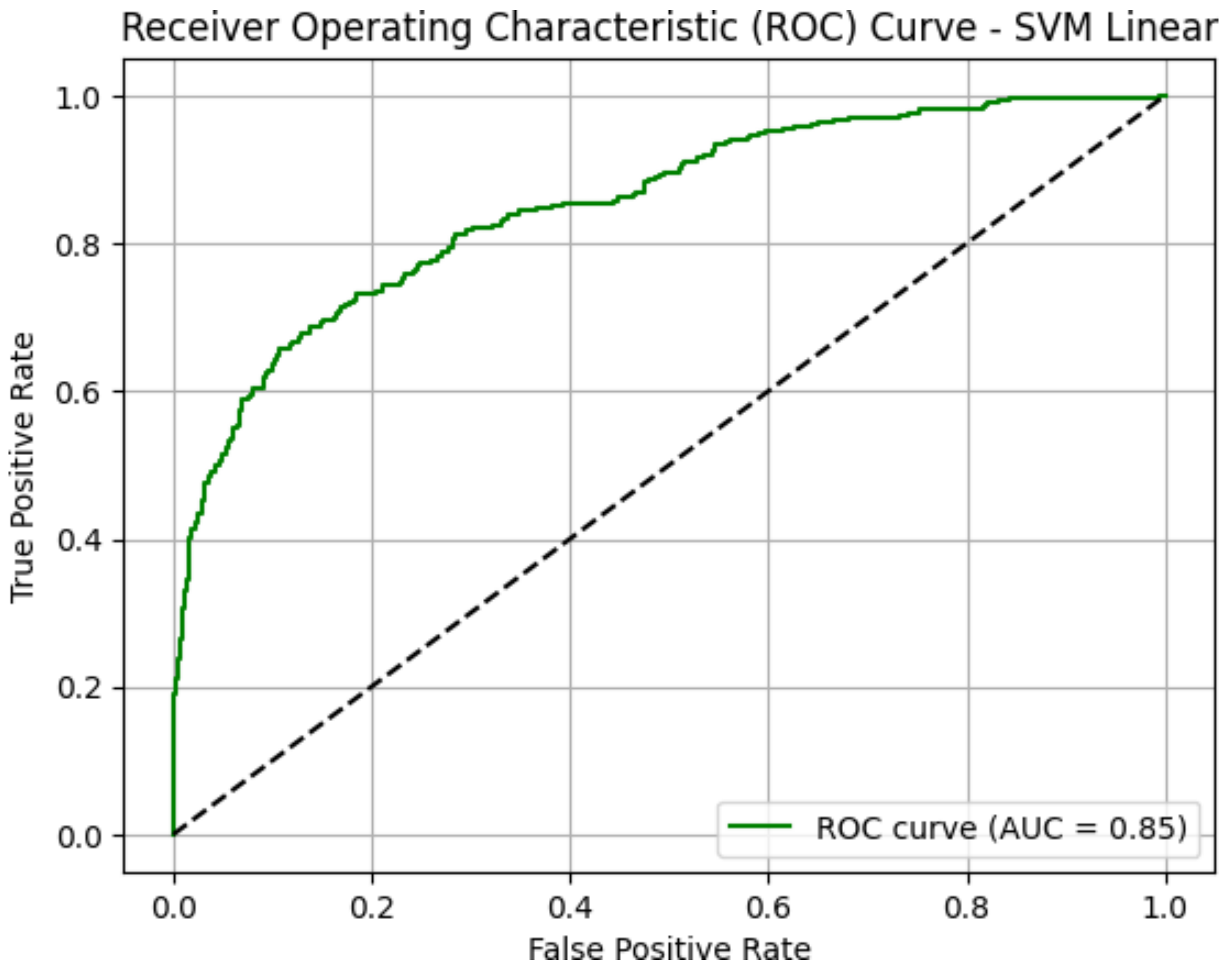

Figure 16: Linear SVM

RBF SVM

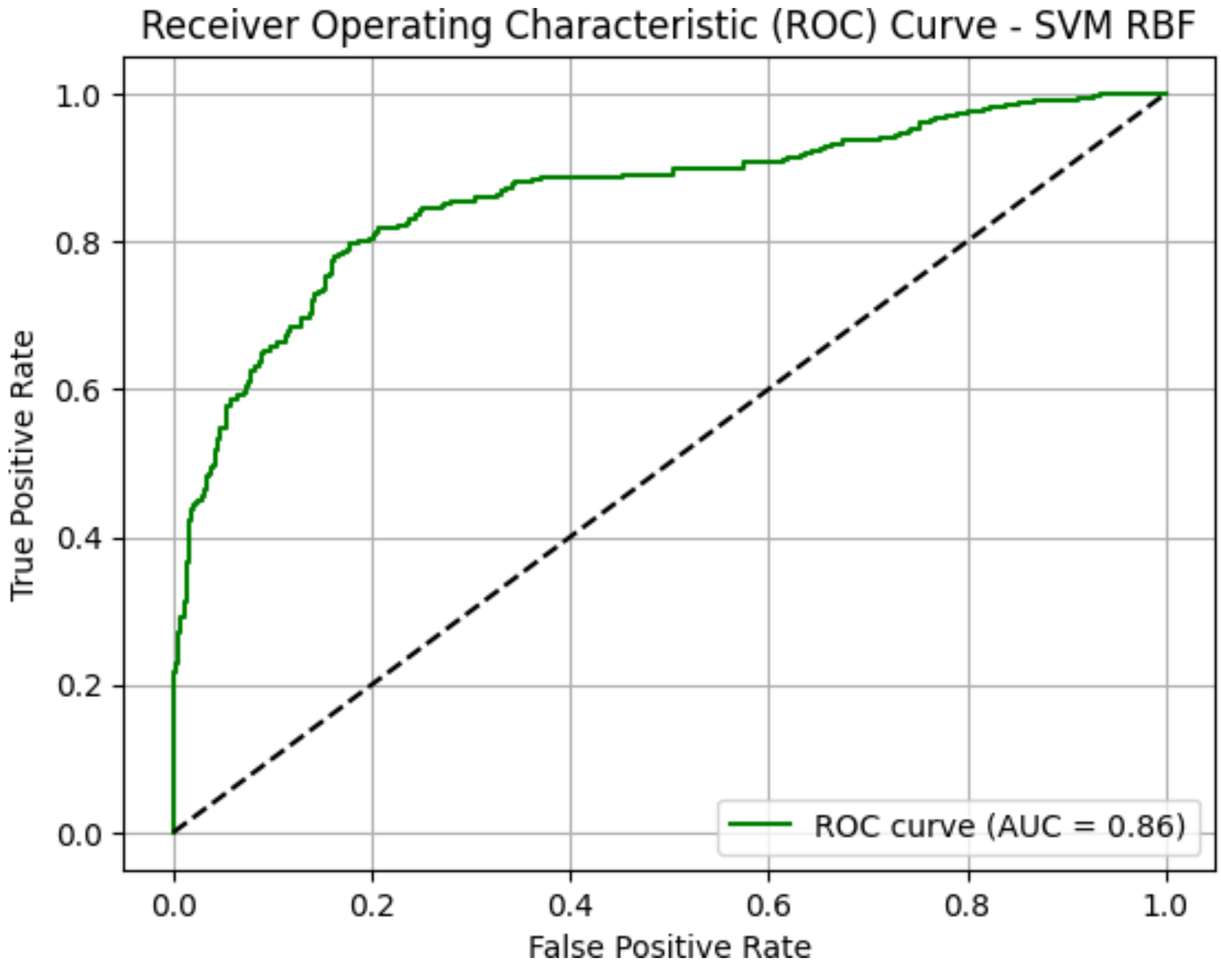

Figure 17: RBF SVM

Sigmoid SVM

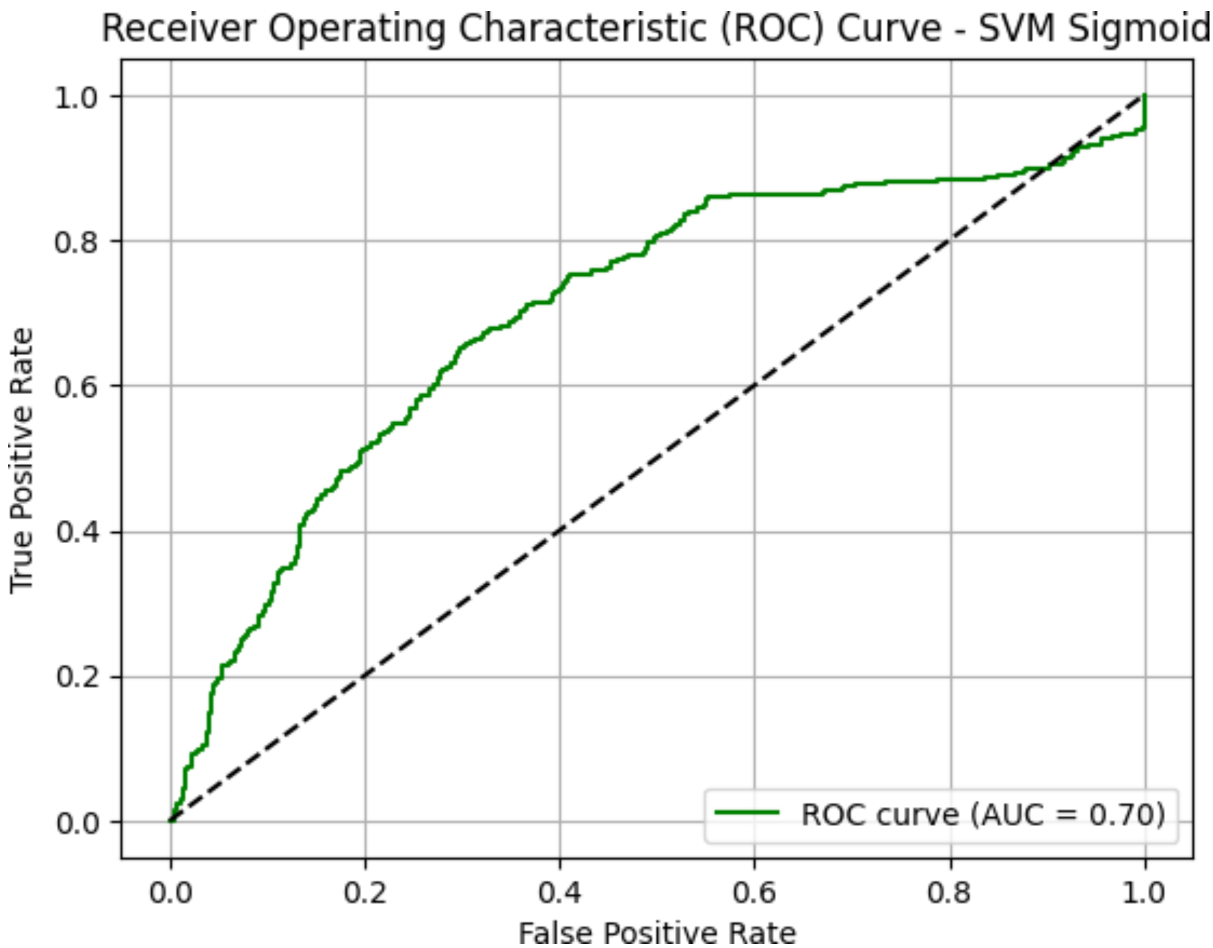

Figure 18: Sigmoid SVM

Result Tables:

Naive Bayes Variant Comparison

Metric	Gaussian NB	Multinomial NB	Bernoulli NB
Accuracy	0.75	0.64	0.80
Precision	0.75	0.64	0.79
Recall	0.75	0.64	0.79
F1 Score	0.75	0.64	0.79

Table 1: Performance Comparison of Na¨ıve Bayes Variants

KNN Performance for Different k Values

k	Accuracy	Precision	Recall	F1 Score
1	0.81	0.81	0.80	0.80
3	0.79	0.80	0.80	0.79
5	0.80	0.79	0.79	0.79
7	0.80	0.80	0.80	0.80

Table 2: KNN Performance for Different k Values

KNN: KDTree vs BallTree

k	KDTree	BallTree
Accuracy	0.819	0.819
Precision	0.819	0.819
Recall	0.819	0.819
F1 Score	0.817	0.816

Table 3: KNN: KDTree vs BallTree

SVM Kernel-wise Results

Kernel	Hyperparameters	Accuracy	F1 Score	Training Time
Linear	C=0.0126	0.793	0.788	0.169
Polynomial	C = 239.502 , degree = 3 , gamma = 0.41	0.80	0.79	553.78
RBF	C = 1.490 , gamma = 0.596	0.82	0.82	0.216
Sigmoid	C = 754.312 , gamma = 2.56	0.79	0.79	0.367

Table 4: SVM Performance with Different Kernels and Parameters

K-Fold Cross-Validation Results

Fold	Naive Bayes Accuracy	KNN (KDTree) Accuracy	SVM (RBF) Accuracy
Fold 1	0.80	0.83	0.82
Fold 2	0.78	0.81	0.83
Fold 3	0.81	0.81	0.83
Fold 4	0.78	0.81	0.81
Fold 5	0.81	0.83	0.82
Average	0.80	0.82	0.82

Table 5: Cross-Validation Scores for Each Model

Learning Outcomes:

- Implementing and comparing Naive Bayes, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) classifiers.
- Using classification metrics such as accuracy, precision, recall, F1-score, and confusion matrix to evaluate the spam detection models.
- Performing hyperparameter tuning to optimize results.