IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 June 2025, accepted 16 July 2025, date of publication 21 July 2025, date of current version 25 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3590871

RESEARCH ARTICLE

Optimizing Machine Learning-Based Ovarian Cancer Prediction Through Normalization Strategies

ROOPASHRI SHETTY[1], SIDDHANT GUPTA[1], VANSH MEDIRATTA[2], SHWETHA RAI[1], AND M. GEETHA[1]

[1]Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India
[2]Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, Karnataka 576104, India

Corresponding authors: Shwetha Rai (shwetha.rai@manipal.edu), Siddhant Gupta (siddhant051231@gmail.com), and Vansh Mediratta (vanshmedi12@gmail.com)

ABSTRACT Ovarian cancer is one of the most challenging cancers to detect early, often leading to poor survival rates. This study explores supervised and unsupervised machine learning and deep learning approaches to improve predictive performance using clinical and biomarker-based data which was scaled through two popular techniques: Min-Max scaling and Z-Score normalization. The research begins by carefully preprocessing the dataset including feature selection to ensure high-quality inputs. Various baseline and ensemble classifiers, including K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), and Logistic Regression (LR), are tested, for better model efficiency on both datasets. To further boost performance, ensemble methods like Stacking, Bagging, and Gradient Boosting, are incorporated. Additionally, unsupervised models like K-Means and DBSCAN clustering are implemented to study further subgroups of the Ovarian Cancer dataset optimizing results. The effects of different feature selection techniques and the impact of standardization versus normalization are compared on both datasets. The Min-Max normalization technique outperformed Z-Score and it is observed that, the Stacking classifier achieved the highest accuracy of 100%, followed by SVM, Logistic Regression, and Bagging, each recording an accuracy of 97%. Further, DBSCAN, a clustering technique outperformed K-Means with a Silhouette Score of 0.7245 and it is observed that clustering performed well with Min-Max when compared with Z-Score normalization technique. The findings highlight that a well-optimized combination of feature selection, ensemble learning, and clustering significantly enhances ovarian cancer prediction, providing a valuable foundation for early diagnosis and clinical decision support.

INDEX TERMS Clustering, data preprocessing, feature selection, machine learning, ovarian cancer.

I. INTRODUCTION

The human body is a constitution of smaller units called cells that divide and grow to make new cells as our body requires. Usually, older cells die, and new cells take their place. Cancer begins when cells grow uncontrollably, forming a mass called a tumor. The tumor is classified as benign, a non-cancerous that does not spread, and malignant, a cancerous tumor [1].

Ovarian cancer is one of the most common types of cancer that can be seen in the ovaries, with almost 6900 new cases diagnosed every year. In 72% cases, the presence of ovarian cancer was diagnosed in an advanced stage [2]. To ensure that patients with ovarian cancer receive proper treatment in time, an accurate characterization of ovarian pathology before surgery is important [3]. Malignant tumors are tumors that are dangerous cancer cells that have to be treated properly in time [4]. Therefore for any treatment of the patient with the tumor, it is very important to classify the tumor as benign

The associate editor coordinating the review of this manuscript and approving it for publication was Giacomo Fiumara.

or malignant. There exist different ovarian cancer types like epithelial ovarian cancer, metastatic, stromal cell cancer, and germ cell ovarian cancer [5].

Healthcare data mining is a trending area to find useful information/knowledge as well as patterns related to various types of diseases. Data mining is one of the major steps in Knowledge Discovery and Data Mining (KDD) which focuses mainly on the approaches for extracting useful knowledge from data [6]. In databases, the term Knowledge Discovery refers to the process of gathering knowledge in data [7]. There exist several data mining techniques for discovering useful information some of them are clustering, association rule mining, outlier analysis, classification, time series analysis, and prediction, and these can either be supervised or unsupervised [8] technique.

This work aims to create a Machine Learning (ML) model for the prediction of ovarian cancer utilizing the PLCO datasets, as well as to conduct clustering analysis on the datasets.

The methodology encompasses several essential steps as described in Figure 1:

- Investigating the dataset to examine feature distributions, statistical characteristics, and interrelations.
- Executing data pre-processing tasks, which include addressing missing values, normalization, and detecting outliers.
- Performing a comparative analysis of different pre-processing methods to evaluate their influence on model efficacy.
- Training a variety of ML models and assessing their performance through metrics such as accuracy, prevalence, and F1-score to determine the most effective model.
- Implementing clustering techniques and evaluating their effectiveness using suitable assessment metrics.

II. LITERATURE REVIEW

Abramowicz et al. [9] explained the most reliable and simple method to differentiate malignant and benign adnexal masses before surgery, which is the subjective assessment of ultrasound examination. This is an instant diagnosis technique but in the presence of both benign and malignant features and the absence of both benign and malignant features, the result becomes inconclusive and a second type of test is recommended. Hartman et al. [10] used different predictive variables like ultrasound criteria and CA125 to predict ovarian cancer in women with an adnexal mass. 103 women with 110 adnexal masses were included in this study and CA125 was measured in blood sample. Tumors were then classified as benign or malignant using simple ultrasound rules. Xin et al. [11] compared various DL models, including ResNet, DenseNet, Vision Transformer, and Swin Transformer, against expert subjective assessments for evaluating ovarian tumor malignancy using transvaginal ultrasound. Their findings demonstrated that, except for the Vision Transformer, the models performed comparably to expert evaluations. Notably, the Swin Transformer achieved an AUC of 0.92, with a sensitivity of 87.2% and a specificity of 94.3%. Wang et al. [12] investigated the effectiveness of ensemble learning algorithms in predicting ovarian cancer, showing that these models outperformed individual classifiers. Their research, which analyzed 49 risk factors across 349 patients, revealed that ensemble learning techniques improved AUC and accuracy by 19% and 16%, respectively.

Jianing et al. [13] utilized ML models to predict survival outcomes in ovarian cancer patients based on SEER database records. The Random Forest classifier achieved an accuracy of 88.72% and an AUC of 82.38%, while the XGBoost regressor attained an RMSE of 20.61% and an R^2 value of 0.4667. The study used SHAP analysis to identify crucial predictive features, such as histologic type, chemotherapy history, and tumor stage. Zhang et al. [14] developed a predictive model using the LightGBM algorithm for early ovarian cancer diagnosis. Their model achieved an accuracy of 88% and was capable of detecting the disease approximately 17 days before clinical confirmation. The study identified key predictive features, including laboratory test results, imaging findings, demographics, and symptoms, with CA125 levels being one of the most significant indicators. Huang et al. employed the XGBoost classifier on whole-exome sequencing data from 47 ovarian cancer patients to predict driver genes associated with causative mutations. Their model achieved a classification accuracy of 94.6%, identifying 12 potential candidate genes, including LAMA3, LAMC3, COL6A1, and COL5A1, which may serve as clinical biomarkers.

III. METHODOLOGY

Ovarian cancer prediction involves several techniques, and their significance on the results varies based on the type of technique employed. The overall methodology followed in this study is shown in figure 1. The dataset is read into a data frame and irrelevant features are removed as an initial step. The data is pre-processed using data imputation, collinearity reduction, and target variable encoding techniques. Further, to analyze the impact of feature selection techniques, Min-Max and Z-Score scaling are applied to the dataset. The classification and clustering techniques are applied to the resulting datasets and their efficacy is evaluated using the evaluation metrics such as accuracy, precision, recall, F1-score and AUC for classification models and Silhouette Score for clustering.

A. DATASET DESCRIPTION

The dataset used in this study is created by merging four tables containing diagnostic, medical, and screening-related information. Table 1 provides an overview of the individual datasets used in the merging process.

The target variable, **"Canctype"**, which represents the type of cancer diagnosis, is present in the *Medical Complications Table*. Since each table contains multiple records corresponding to the same patient identifier (`plco_id`), the tables are merged to create a consolidated dataset. This

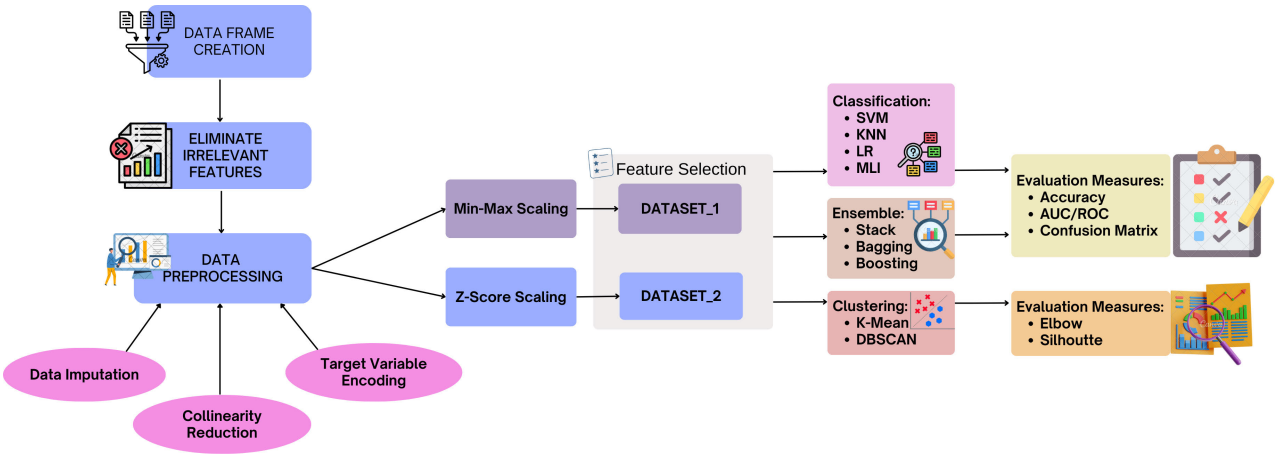


FIGURE 1. Overall methodology for optimization of ML-based ovarian cancer prediction through different normalization strategies.

TABLE 1. Overview of the dataset, showing the distribution of records and features across different tables.

Table Name	Number of Entries	Number of Features
Diagnostic Procedures Table	19,281	16
Medical Complications Table	681	8
Screening Table	150,993	129
Screening Abnormalities Table	20,186	14

serves as a unique identifier for each patient. However, since multiple records existed for the same `plco_id` across different tables, a data compression step is applied to ensure that each patient is represented by a single record in the final dataset. To achieve this, the tables are concatenated and grouped by `plco_id`. During this process:

- For each feature for a particular patient, the **first non-missing (non-NaN) value** is selected.
- If no non-NaN values are available for a particular feature, the value is set to **NaN**.

This approach preserves as much available information as possible while eliminating redundancy. The dataset produced as a result of the merging process is raw and unfit for model training and clustering. Hence, different pre-processing techniques are applied to the dataset to make it fit for model training and clustering.

After preprocessing, the final dataset consists of:

- **30,989 records** (unique `plco_id` entries)
- **134 features**, combining relevant attributes from all four tables

This consolidated dataset serves as the foundation for further analysis in this study.

However, in spite of applying a compression algorithm to the dataset, it is 72.24 % sparse, which may lead to issues with model training since a large number of values are imputed.

B. ELIMINATION OF IRRELEVANT FEATURES

Columns deemed insignificant for model training are removed. This includes columns containing data that could not be converted into numeric form using methods like one-hot encoding, as well as non-unique columns where all records had the same value. This resulted in the number of columns reducing from **155** to **147**.

C. DATA PRE-PROCESSING

- *Data Imputation:* Missing values in the dataset are filled using appropriate imputation techniques based on the data type of each column. Specifically, columns with floating-point data are imputed using the mean, while categorical columns are imputed using the mode. After imputing missing values, duplicate records are also removed to reduce data bias.
- *Target Variable Encoding [15]:* Missing values in the target column 'canctype' are not imputed; instead, records with 'canctype' as NaN are removed. In the raw dataset, 'canctype' ranges from 0 to 4, where 0 indicates the absence of cancer, and values 1 to 4 correspond to different types of ovarian cancer. To simplify the problem into a binary classification task, 'canctype' is encoded such that values 1 to 4 are set to 1.
- *Collinearity Reduction [16]:* Collinearity reduction involves removing columns that exhibit a strong positive or negative correlation with another column. In this study, collinearity is assessed using a correlation matrix, and any column with a correlation coefficient greater than 0.9 is removed. This resulted in the dropping of another 41 features reducing the total count to 106 features.
- *Data Normalization:* To prevent features with varying scales from unduly affecting distance-based models such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM), data standardization is performed using two widely utilized methods [17].: **Min-Max Scaling [18]** and **Z-score (Standard) Scaling [19]**. For further references the dataset transformed using:
  - **Min-Max Scaling** will be referred to as **Dataset 1**.
  - **Z-score Scaling** will be referred to as **Dataset 2**.

  The transformation equations for both scaling techniques are as follows:

Dataset 1:

$$Z = \frac{Y - Y_{\min}}{Y_{\max} - Y_{\min}} \tag{1}$$

where:
- Y stands for the original value of the feature,
- $Y_{\min}$ and $Y_{\max}$ indicate the lowest and highest values of the feature, respectively,
- Z is the normalized value, reduced to the range [0, 1].

Dataset 2:

$$Z = \frac{Y - \mu_Y}{s_Y} \tag{2}$$

where:
- Y represents the original feature value,
- μ_Y is the mean of the feature,
- s_Y is the standard deviation of the feature,
- Z is the standardized value with mean 0 and variance 1.

- *Outliers Detection:* Outliers are defined as extreme values that fall outside a specified threshold, potentially compromising the accuracy of a model [20]. In this research, the identification of outliers is conducted using the Interquartile Range (IQR) method [21]. The lower and upper limits are established in the following manner:

$$\text{Lower Threshold} = q_L - 1.5 \times R_Q \tag{3}$$

$$\text{Upper Threshold} = q_U + 1.5 \times R_Q \tag{4}$$

where:
- q_L represents the lower quartile (25th percentile),
- q_U represents the upper quartile (75th percentile),
- $R_Q = q_U - q_L$ denotes the interquartile range (IQR).

For the dataset considered, the outliers are capped at the defined bounds rather than being removed to preserve data integrity. Values falling below the lower limit are restricted to the lower limit, while those exceeding the upper limit are confined to the upper limit to reduce the influence of outliers.

D. FEATURE SELECTION

Feature selection involves identifying and retaining the most significant features while eliminating redundant or irrelevant ones [22]. In this study, a Random Forest classifier [23] is used to assess feature importance [24]. It constructs multiple decision trees on different subsets of the data and aggregates their outputs to determine the importance of each feature. The importance of a feature X_i is calculated as:

$$I(X_i) = \sum_{t=1}^{T} \sum_{s \in S_t} \Delta G_s \cdot 1(X_s = X_i) \tag{5}$$

where:
- $I(X_i)$ represents the importance score of feature X_i,
- T is the total number of trees in the forest,
- S_t is the set of all splits in tree t,
- ΔG_s The reduction in impurity, such as Gini impurity or entropy, as a result of the split. s,
- $1(X_s = X_i)$ is an indicator function that equals 1 if feature X_i is used at split s, and 0 otherwise.

Feature selection is performed separately for both scaled datasets:
- **Dataset 1**: After feature selection, the dataset consists of **37 selected features** and **399 records**.
- **Dataset 2**: After feature selection, the dataset consists of **28 selected features** and **399 records**.

E. MODEL TRAINING

The cleaned dataset is split into training and testing sets in an 80:20 ratio to adjust hyperparameters, followed by cross-validation of K-means on the dataset.

Multiple models are trained on the dataset, including:
- **K-Nearest Neighbors (KNN)** [25], **Logistic Regression [26]** and **Support Vector Machine (SVM)** [27], which rely on distance-based and margin-based classification techniques, respectively.
- A **Multilayer Perceptron (MLP)** neural network, designed to capture complex nonlinear patterns in the data [28].
- Ensemble learning techniques [29], [30], including:
  - **Bagging** to reduce variance and improve stability [31],
  - **Boosting** to enhance weak classifiers by sequential training [32],
  - **Stacking** to leverage multiple models by training a meta-learner on their outputs [33].

The evaluation of these models is conducted using the following metrics:
- **Classification Report:** This summarizes the model's performance in distinguishing between different classes. It includes Precision, Recall, F1-Score and Accuracy [34].
- **Confusion Matrix:** A confusion matrix is a summary of predictions made by a classification model. It provides a clear picture of how many instances are correctly and incorrectly classified [35]. Table 2 shows the general confusion matrix.

TABLE 2. Confusion matrix.

Actual \ Predicted	Positive (1)	Negative (0)
Positive (1)	TP	TN
Negative (0)	FP	FN

  - **True Positive (TP):** Model correctly predicts the positive class.
  - **False Positive (FP):** Model incorrectly predicts positive when it is negative (Type I Error).
  - **False Negative (FN):** Model incorrectly predicts negative when it is positive (Type II Error).
  - **True Negative (TN):** Model correctly predicts the negative class.
- **K-Fold Cross-Validation [36]:**A method employed to enhance the dependability of model prediction involves

partitioning the dataset into k subsets, known as folds. The model undergoes training and evaluation k times, with each iteration utilizing a distinct fold as the test set while the other $k-1$ folds serve as the training data. The overall performance metric is calculated as the average of the results from all iterations.

$$\text{CV Score} = \frac{1}{k}\sum_{i=1}^{k}\text{Metric}_i$$

  - Reduces variance in model performance estimation.
  - Helps in preventing overfitting.
  - Common values: $k = 5$ or $k = 10$.
- **ROC Curve and AUC:** The ROC curve visualizes the trade-off between true positive rate (Recall) and false positive rate at different classification thresholds [37]. The AUC quantifies overall model performance.

$$\text{True Positive Rate (TPR)} = \frac{TP}{TP + FN}$$
$$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

  - The ROC curve is plotted with TPR on the y-axis and FPR on the x-axis.
  - AUC ranges from 0 to 1:
    * AUC = 0.5: Model performs no better than random guessing.
    * AUC > 0.7: Good model performance.
    * AUC = 1: Perfect classifier.

IV. RESULTS

This section outlines the outcomes of several models and algorithms tested on both datasets. The performance of each model is assessed using conventional metrics such as accuracy and F1-score. This analysis aims to evaluate the efficacy of various methods in comprehending and forecasting data patterns while investigating the impact of different pre-processing techniques. A comprehensive summary of the results for each model is presented in the following subsections.

A. K-NEAREST NEIGHBORS

KNN is a supervised ML algorithm widely used for classification and regression tasks and it classifies a data point based on the majority class of its k nearest neighbors in the feature space. The choice of k, significantly impacts the model's performance: smaller values of k make the model sensitive to noise, while larger values lead to smoother decision boundaries but may cause overgeneralization. Despite this, KNN remains a powerful baseline model, especially when paired with feature scaling techniques like Min-Max scaling or normalization to improve distance calculations.

The following are the results obtained by training the KNN model on both datasets:

- *Hyperparameters:* Table 3 presents the optimal hyperparameters for the KNN model on Dataset 1 and Dataset 2. Both use the Minkowski metric [38] with $p = 2$ (Euclidean distance). Dataset 1 showed negligible changes in accuracy with uniform or distance-based weights even with changes in K value. It is observed that with the varying K value, Dataset 2 is 1.33% effeicient when distance weights are used indicating that normalization alters distance distributions, making weighted neighbors relevant. The auto algorithm selects the most appropriate method based on the dataset size and dimensionality.

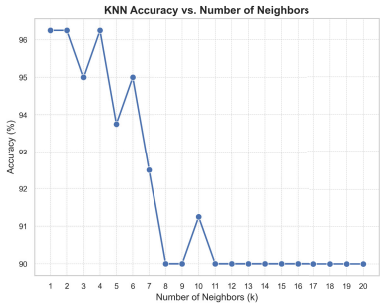
TABLE 3. KNN: Optimal Hyperparameters.

Hyperparameter	Dataset 1	Dataset 2
Algorithm	auto	auto
Metric	minkowski	minkowski
p (Minkowski Power)	2	2
Weights	uniform	distance

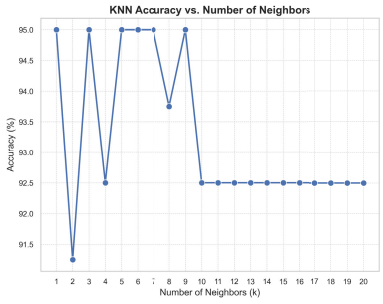
- *K Value:* The value of K is selected by training a model using the best set of hyperparameters on various values of K and choosing the K with the highest accuracy. The accuracy trends for different values of K are shown in Figure 2.
  For Dataset 1 shown in Figure 2a, the highest accuracy is observed at lower values of K (around $K = 3$), but accuracy declines sharply for larger values, suggesting that a small K provides better performance while larger K leads to underfitting.
  For Dataset 2 shown in Figure 2b, accuracy is more stable at higher K values (around $K = 7$) due to the effect of normalization and distance-based weighting. This suggests that distance-based weighting compensates for the altered scale of distances, making KNN more robust to variations in K.
  Overall, the choice of scaling method influences the sensitivity of KNN to K, with min-max scaling favoring small K and normalization benefiting from larger K values.
  Ultimately, to optimize training time and computational resources, $K = 1$ was chosen as the final value.
- *Classification Report:*
  The classification performance for both datasets is summarized in Table 4.
  Dataset 1 achieves a slightly higher accuracy (**96%**) than **Dataset 2** (**95%**), indicating that **Min-Max scaling** (used for Dataset 1) led to a marginally better classification performance compared to **Normalization** (Dataset 2).
  For **Class 0**, recall is **1.00** in both datasets, meaning all instances of Class 0 were correctly classified. For **Class 1**, Dataset 2 has a recall of **0.91**, slightly lower than Dataset 1 (**0.93**), suggesting that normalization caused some misclassifications in this class. The difference in performance suggests that **Min-Max scaling** (**Dataset 1**) retains better separability between classes, leading to slightly higher recall and accuracy than **Normalization** (**Dataset 2**). Both datasets have very close **precision**,

TABLE 4. KNN: Classification Report for Dataset 1 and Dataset 2.

Class	Dataset 1				Dataset 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	0.92	1.00	0.96	36	0.90	1.00	0.95	36
1	1.00	0.93	0.96	44	1.00	0.91	0.95	44
Accuracy	0.96				0.95			
Macro avg	0.96	0.97	0.96	80	0.95	0.95	0.95	80
Weighted avg	0.97	0.96	0.96	80	0.96	0.95	0.95	80



(a) Dataset 1



(b) Dataset 2

FIGURE 2. KNN: K vs Accuracy plots for Dataset 1 and Dataset 2.

recall, and **F1-score** values across all metrics, reinforcing that the choice of data scaling has only a minor impact on **KNN classification**.

Table 5 presents the confusion matrices for Dataset 1 and Dataset 2. Both datasets exhibit high classification accuracy, with no misclassification of Class 0 (**Predicted 0**). However, a slightly higher misclassification of Class 1 is observed in Dataset 2 (4 misclassified instances) compared to Dataset 1 (3 misclassified instances). This suggests that the model performs slightly better on Dataset 1 in distinguishing between the two classes.

TABLE 5. KNN: Confusion Matrix for Dataset 1 and Dataset 2.

Actual Class	Dataset 1		Dataset 2	
	Predict 0	Predict 1	Predict 0	Predict 1
Actual 0	36	0	36	0
Actual 1	3	41	4	40

The table 6 presents the 5-fold cross-validation accuracy scores for /textbfDataset 1 and **Dataset 2**. Each fold's accuracy is reported, along with the mean accuracy and standard deviation, to assess the model's performance consistency across different data splits.

- From the results observed in 6, Dataset 2 has a higher mean accuracy (0.959) compared to Dataset 1 (0.940), indicating slightly better model performance on Dataset 2.
- The standard deviation for Dataset 1 (0.032) is higher than that of Dataset 2 (0.012), suggesting that Dataset 2 provides more consistent accuracy across different folds.
- Fold 3 shows the lowest accuracy for Dataset 1 (0.891), whereas Dataset 2 performs better in the same fold (0.938). This might indicate variations in data distribution or model generalization issues in Dataset 1.
- Fold 4 has the highest accuracy for Dataset 1 (0.984), while Dataset 2 also performs well in this fold (0.969), though slightly lower.
- Fold 2 has identical accuracy (0.953) for both datasets, suggesting that the model exhibits similar behavior on both datasets for this fold.

TABLE 6. KNN: Cross-Validation Scores for Dataset 1 and Dataset 2.

Fold	Dataset 1 Accuracy	Dataset 2 Accuracy
1	0.953	0.969
2	0.953	0.953
3	0.891	0.938
4	0.984	0.969
5	0.921	0.968
Mean Accuracy	**0.940**	**0.959**
Standard Deviation	**0.032**	**0.012**

- ROC Curve with AUC: The ROC curves in Figure 3 indicate that both models perform well, with AUC values above 90%. Dataset 1 (left) shows a slightly higher AUC (96.50%) compared to Dataset 2 (96.45%). Both curves exhibit strong classifier performance, as the curves are close to the top-left corner. The small difference in AUC suggests that the model generalizes well across both datasets.

KNN is a versatile and easy-to-implement classification algorithm that relies on the proximity of data points to make predictions. The performance of KNN is highly dependent on the choice of hyperparameters, particularly the number of neighbors (k), the distance metric, and the weighting scheme. Through techniques like hyperparameter tuning and cross-validation, the accuracy and robustness of KNN models can be significantly improved.

Despite its simplicity, KNN has proven effective for a wide range of tasks, though it is best suited for smaller datasets with fewer dimensions. For larger datasets or high-dimensional data, optimizations such as dimensionality reduction or efficient data indexing methods can help overcome KNN's computational limitations. Ultimately, KNN

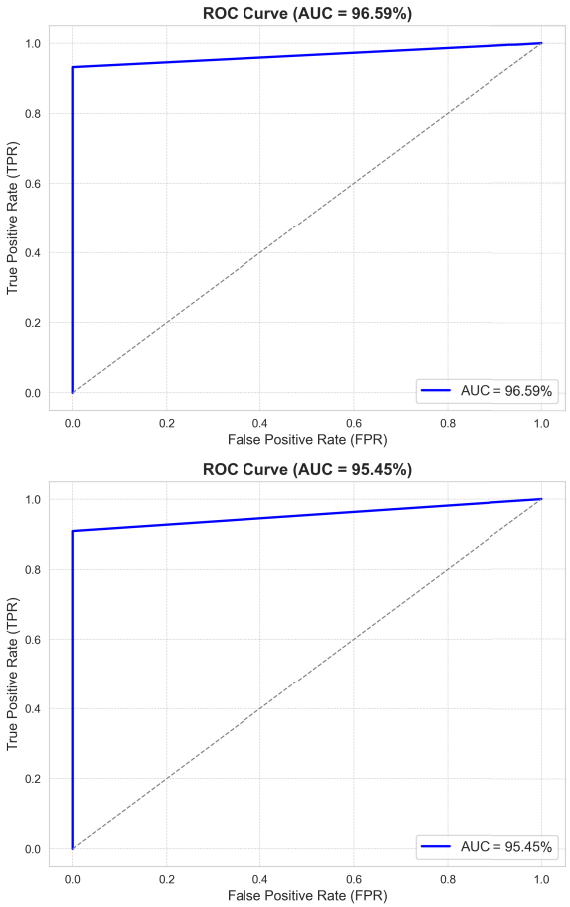


FIGURE 3. KNN: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

remains a powerful and intuitive tool in machine learning, providing valuable insights into data patterns when applied appropriately.

B. SUPPORT VECTOR MACHINE

SVM is highly effective in high-dimensional feature spaces and can handle both linear and non-linear classification problems using various kernel functions: linear, polynomial, radial basis function (RBF), and sigmoid [39]. The performance of SVM model depends on the choice of kernel function and the regularization parameter C [40]. A higher value of C focuses on correctly classifying all training samples, whereas a lower C allows for a wider margin but tolerates some misclassification. The selected kernel function determines the complexity of the decision boundary, significantly influencing the model's ability to capture non-linear relationships.

- *Hyperparameters:*
  - The best hyperparameters for the SVM model are presented in Table 7.
  - Both datasets use the RBF kernel, indicating that a non-linear decision boundary is optimal.
  - The optimal regularization parameter (C) differs, with Dataset 1 favoring a higher value (10) compared to Dataset 2 (1), suggesting different levels of margin flexibility.
  - The gamma parameter is set to 'scale' for Dataset 1 and 'auto' for Dataset 2, which affects how the kernel function influences classification.
  - The polynomial degree is set to 3 in both cases, though it is only relevant for polynomial kernels.

TABLE 7. SVM: Optimal Hyperparameters.

Hyperparameter	Dataset 1	Dataset 2
Kernel	RBF	RBF
C	10	1
Gamma	scale	auto
Degree	3	3

- *Classification Report:* The classification reports for both datasets are shown in Table 8. Both datasets achieve high accuracy (97%), indicating strong model performance. The precision, recall, and F1-score values are nearly identical across both datasets. The model shows a perfect recall for class 0 and perfect precision for class 1, suggesting a strong distinction between the two classes. The macro and weighted averages confirm a balanced classification performance across classes.
- *Confusion Matrix:* The confusion matrices for both datasets are shown in Table 9. The model achieves perfect classification for class 0, with no false positives. There are only two misclassified samples in class 1 for both datasets, showing strong performance. The overall classification is highly accurate, reinforcing the high precision and recall scores observed in the classification report.

  The cross-validation results for both datasets are summarized in Table 10. Dataset 1 achieves a mean accuracy of 0.978 with a standard deviation of 0.008, indicating high performance with slight variability across folds. Dataset 2 has a mean accuracy of 0.969 with an almost negligible standard deviation of 0.0001, suggesting consistent performance across folds. The small standard deviation values for both datasets highlight the model's stability and generalization capability. The accuracy values remain consistently high across all folds, reinforcing the reliability of the SVM classifier.
- ROC Curve with AUC: The ROC curves in Figure 4 illustrate the performance of the SVM model on both datasets. The AUC values indicate high classification performance, with Dataset 1 achieving near-perfect separation and Dataset 2 showing slight variability.

SVM is a robust classification algorithm adept at managing both linear and non-linear decision boundaries. The selection of hyperparameters, such as the regularization parameter C and the kernel function, is crucial in influencing the model's effectiveness. SVM demonstrates particular strength in high-dimensional feature spaces and is capable of addressing intricate classification challenges. Employing hyperparameter optimization and cross-validation methods can enhance the model's accuracy and its ability to generalize.

TABLE 8. SVM: Classification Report for Dataset 1 and Dataset 2.

Class	Dataset 1				Dataset 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	0.95	1.00	0.97	36	0.95	1.00	0.97	36
1	1.00	0.95	0.98	44	1.00	0.95	0.98	44
Accuracy	0.97				0.97			
Macro avg	0.97	0.98	0.97	80	0.97	0.98	0.97	80
Weighted avg	0.98	0.97	0.98	80	0.98	0.97	0.98	80

TABLE 9. SVM: Confusion Matrices for Dataset 1 and Dataset 2.

Actual Class	Dataset 1		Dataset 2	
	Predict 0	Predict 1	Predict 0	Predict 1
Actual 0	36	0	36	0
Actual 1	2	42	2	42

TABLE 10. SVM: Cross-Validation Scores for Dataset 1 and Dataset 2.

Fold	Dataset 1 Accuracy	Dataset 2 Accuracy
1	0.969	0.969
2	0.984	0.969
3	0.969	0.969
4	0.984	0.969
5	0.984	0.968
Mean Accuracy	**0.978**	**0.969**
Standard Deviation	**0.008**	**0.0001**

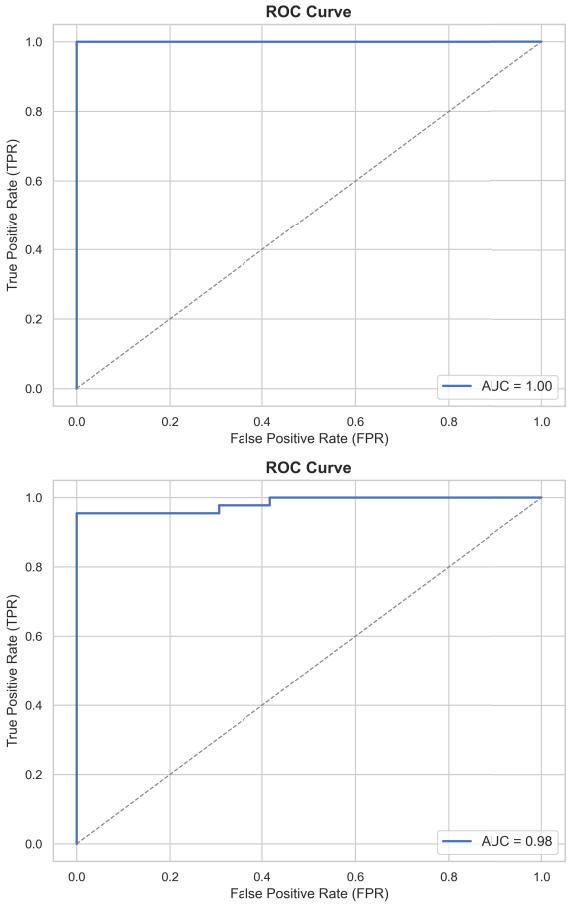


FIGURE 4. SVM: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

However, it is important to note that SVM can be resource-intensive and may not be optimal for extremely large datasets. In contrast, for smaller datasets or those exhibiting distinct margins of separation, SVM proves to be a dependable and efficient option, frequently outperforming other classification methods.

C. LOGISTIC REGRESSION

LR is a statistical technique widely employed for binary classification tasks. It predicts the probability that a given input belongs to a specific class by applying the logistic (sigmoid) function [41].

- *Hyperparameters:* The optimal hyperparameters selected for the LR model are listed in Table 11. The regularization strength **C** is set to **0.1** for both datasets, ensuring a balanced trade-off between bias and variance. **L2 penalty** is preferred, suggesting that the model benefits from ridge regularization to prevent overfitting. The **lbfgs solver** is optimal, making it suitable for convex optimization problems and handling small datasets efficiently.

TABLE 11. LR: Optimal Hyperparameters.

Hyperparameter	Dataset 1	Dataset 2
C	0.1	0.1
Penalty	l2	l2
Solver	lbfgs	lbfgs

- *Classification Report:*
the classification reports for both datasets are given in Table 12. It indicates high precision, recall, and F1-scores, confirming strong model performance. Dataset 1 achieves a slightly higher accuracy (**0.97**) than Dataset 2 (**0.96**), though the difference is minimal. The recall for class 1 in Dataset 2 (**0.93**) is slightly lower than in Dataset 1 (**0.95**), suggesting some positive instances were misclassified. The macro and weighted averages remain balanced, ensuring consistency across class distributions.
The confusion matrices in Table 13 illustrate the performance of the LR model on Dataset 1 and Dataset 2. These matrices show the number of correct and misclassified instances for both classes. The model demonstrates high accuracy, as most predictions are correct. Dataset 1 has a lower misclassification rate compared to Dataset 2. Dataset 2 shows slightly more misclassification in class 1 compared to Dataset 1. The model correctly predicts all instances of class 0 in both datasets.
The cross-validation results in Table 14 provide insights into the model's performance consistency across different folds for both datasets. These scores indicate

TABLE 12. LR: Classification Reports for Dataset 1 and Dataset 2.

Class	Dataset 1				Dataset 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	0.95	1.00	0.97	36	0.92	1.00	0.96	36
1	1.00	0.95	0.98	44	1.00	0.93	0.96	44
Accuracy	0.97				0.96			
Macro avg	0.97	0.98	0.97	80	0.96	0.97	0.96	80
Weighted avg	0.98	0.97	0.98	80	0.97	0.96	0.96	80

TABLE 13. LR: Confusion Matrices for Dataset 1 and Dataset 2.

Actual	Dataset 1		Dataset 2	
	Predict 0	Predict 1	Predict 0	Predict 1
Actual 0	36	0	36	0
Actual 1	2	42	3	41

the robustness and stability of the Logistic Regression model. The model achieves high and consistent accuracy across all folds. Dataset 1 has a slightly higher mean accuracy compared to Dataset 2. The standard deviation for Dataset 1 is lower, indicating more stability in performance. Dataset 2 has slightly more variation in accuracy across folds.

TABLE 14. LR: Cross-Validation Scores for Dataset 1 and Dataset 2.

Fold	Dataset 1 Accuracy	Dataset 2 Accuracy
1	0.969	0.969
2	0.969	0.969
3	0.969	0.938
4	0.969	0.969
5	0.984	0.968
Mean Accuracy	**0.972**	**0.962**
Standard Deviation	**0.006**	**0.012**

- ROC Curve with AUC: The ROC curves in Figure 5 illustrate the performance of the LR model for both datasets. The AUC values indicate the model's ability to distinguish between classes.
  - Both datasets exhibit a high AUC score, suggesting excellent classification performance.
  - The ROC curves remain close to the top-left corner, reinforcing strong model reliability.
  - The minimal difference in AUC between datasets implies consistent performance across different data distributions.

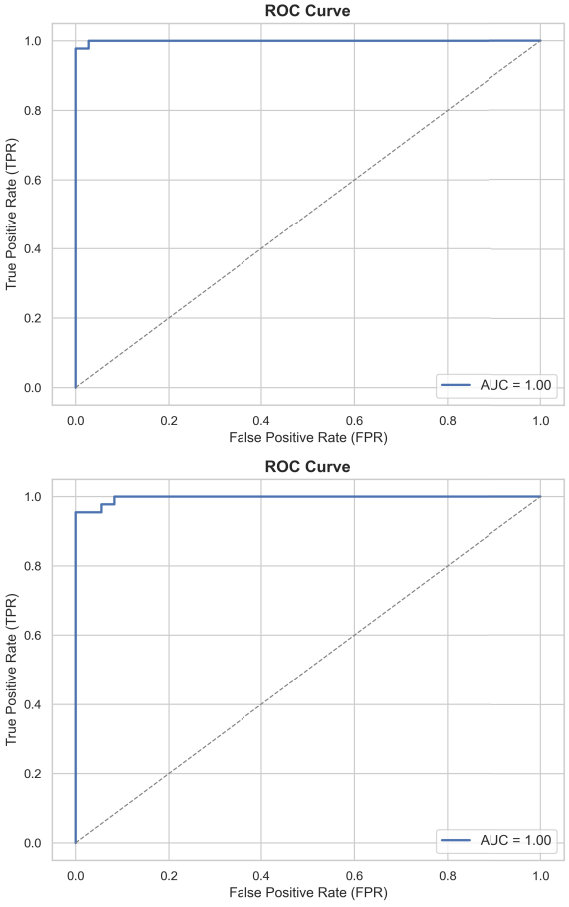


FIGURE 5. LE: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

D. MULTI-LAYER PERCEPTRON (MLP)

MLP is a type of Artificial Neural Network (ANN) commonly used for classification tasks. It consists of multiple layers, employs activation functions such as **ReLU** or **Sigmoid** and is trained using **backpropagation** [42].

- *Hyperparameters:* The optimal hyperparameters for the MLP model, obtained through hyperparameter tuning, are listed in Table 15. These parameters significantly influence the model's learning behavior and performance.
  - The hidden layer architecture remains consistent across both datasets.
  - The `tanh` activation function ensures non-linearity, enhancing learning capabilities.
  - The Adam optimizer is used for adaptive learning rate adjustments.
  - The learning rate is set to a constant value for stable convergence.

TABLE 15. MLP: Optimal Hyperparameters.

Hyperparameter	Dataset 1	Dataset 2
Hidden Layers	(100, 100)	(100, 100)
Activation	tanh	tanh
Solver	Adam	Adam
Alpha	0.0001	0.0001
Learning Rate	Constant	Constant

- *Classification Report:* Table 16 presents the classification performance metrics for both Dataset 1 and Dataset 2. Dataset 1 achieved a higher accuracy (96%) compared to Dataset 2 (93%). Precision, recall, and F1-scores are consistently high for both datasets, indicating balanced

TABLE 16. MLP: Classification Report for Dataset 1 and Dataset 2.

Class	Dataset 1				Dataset 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	0.97	0.94	0.96	36	0.89	0.94	0.92	36
1	0.96	0.98	0.97	44	0.95	0.91	0.93	44
Accuracy	0.96				0.93			
Macro avg	0.96	0.96	0.96	80	0.92	0.93	0.92	80
Weighted avg	0.96	0.96	0.96	80	0.93	0.93	0.93	80

classification performance. Class 0 in Dataset 2 has a slightly lower precision (0.89) compared to Dataset 1 (0.97), suggesting more misclassifications for that class. The macro and weighted averages suggest that the model maintains consistent performance across both datasets.

Table 17 presents the confusion matrices for both datasets using the MLP model. Dataset 1 using MLP has a lower number of misclassifications compared to Dataset 2. The false positive rate for both datasets is low, but Dataset 2 has more false negatives (4) than Dataset 1 (1). Dataset 1 shows slightly better classification performance with a higher count of correctly classified instances.

TABLE 17. MLP: Confusion Matrices for Dataset 1 and Dataset 2.

Actual Class	Dataset 1		Dataset 2	
	Predict 0	Predict 1	Predict 0	Predict 1
Actual 0	32	2	34	2
Actual 1	1	43	4	40

Table 18 summarizes the cross-validation accuracy scores for both datasets using the MLP model. The mean accuracy for Dataset 1 (91.2%) is slightly higher than Dataset 2 (90.9%), suggesting Dataset 1 generalizes slightly better. The standard deviation is lower for Dataset 2 (0.011), indicating more stable performance across different folds.

TABLE 18. MLP: Cross Validation Scores for Dataset 1 and Dataset 2.

Fold	Dataset 1	Dataset 2
1	0.897	0.897
2	0.934	0.906
3	0.906	0.925
Mean Accuracy	**0.912**	**0.909**
Standard Deviation	**0.016**	**0.011**

- *ROC Curve with AUC:*
  Figure 6 displays the ROC curves for both datasets using the MLP model, demonstrating their classification performance. The AUC is higher for Dataset 1 compared to Dataset 2, indicating slightly better classification performance. Both curves are close to the top-left corner, suggesting strong model discrimination ability for both datasets.

E. STACKING CLASSIFIER

Stacking is an ensemble learning technique that combines multiple base models and the outputs of base models are used by the meta-model, to make the final prediction. In this study, KNN and SVM are used as base models, while LR serves as the meta-model.

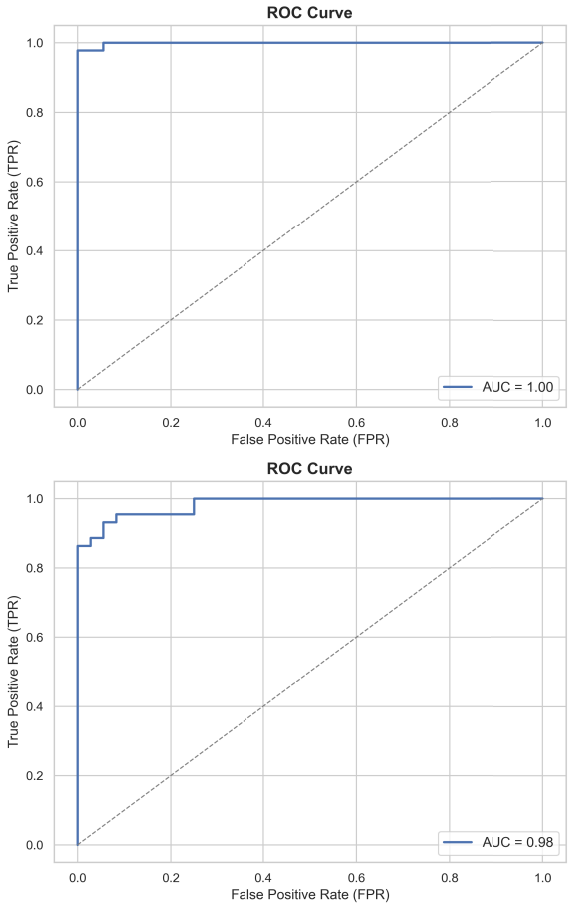


FIGURE 6. MLP: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

- *Hyperparameters:* The hyperparameters used in the model are shown in Table 19

TABLE 19. Stacking: Optimal Hyperparameters.

Hyperparameter	Dataset 1	Dataset 2
KNN	3	3
SVM C	10.0	0.1
Meta-Model C	10	0.1

- *Classification Report:*
  As seen in Table 20, Dataset 1 achieves perfect classification results with an accuracy of 1.00, indicating that the model performs flawlessly on this dataset. Dataset 2 has a slightly lower accuracy of 0.97, suggesting that some misclassifications occurred. The precision and recall values show that Dataset 2 misclassified some instances of class 1, leading to a drop in recall for

TABLE 20. Stacking: Classification Report for Dataset 1 and Dataset.

Class	Dataset 1				Dataset 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	1.00	1.00	1.00	36	0.95	1.00	0.97	36
1	1.00	1.00	1.00	44	1.00	0.95	0.98	44
Accuracy	1.00				0.97			
Macro avg	1.00	1.00	1.00	80	0.97	0.98	0.97	80
Weighted avg	1.00	1.00	1.00	80	0.98	0.97	0.98	80

this class. The difference in performance suggests that Dataset 1 has a more distinguishable feature space or better training conditions compared to Dataset 2.

As per the confusion matrix shown in Table 21, Dataset 1 has a perfect classification with no misclassified instances. In Dataset 2, two instances of class 1 were misclassified as class 0, reducing recall for class 1. The misclassification in Dataset 2 indicates potential overlap in feature space or a less optimal decision boundary for class 1. The overall performance of Dataset 2 remains high, but improvements in distinguishing class 1 could be beneficial. As per the 5-Fold cross validation result

TABLE 21. Stacking: Confusion Matrix for Dataset 1 and Dataset 2.

Actual	Dataset 1		Dataset 2	
	Predicted 0	Predicted 1	Predicted 0	Predicted 1
Actual 0	36	0	36	0
Actual 1	0	44	2	42

shown in Table 22, Both Dataset 1 and Dataset 2 achieve a mean accuracy of 0.972, indicating a similar level of performance. The standard deviation of 0.006 for both datasets suggests stable performance across folds. The variation in accuracy across folds is minimal, indicating that the model generalizes well across different subsets of the data. The consistency in accuracy highlights that the model maintains reliable classification performance regardless of the training and validation split.

TABLE 22. Stacking: Cross Validation Scores for Dataset 1 and Dataset 2.

Fold	Dataset 1 Accuracy	Dataset 2 Accuracy
1	0.984	0.969
2	0.969	0.969
3	0.969	0.969
4	0.969	0.969
5	0.968	0.984
Mean Accuracy	**0.972**	**0.972**
Standard Deviation	**0.006**	**0.006**

- *ROC Curve with AUC:* As per the AUC-ROC curves shown in Figure 7, Dataset 1 has an AUC of 99.95%, indicating near-perfect classification performance.Dataset 2 has an AUC of 91.95%, suggesting a strong but slightly weaker classification ability compared to Dataset 1. The drop in AUC for Dataset 2 implies a reduced ability to separate the classes effectively, aligning with the misclassifications observed in the confusion matrix.Overall, both models perform well, but Dataset 2 may require further optimization to improve class separation.

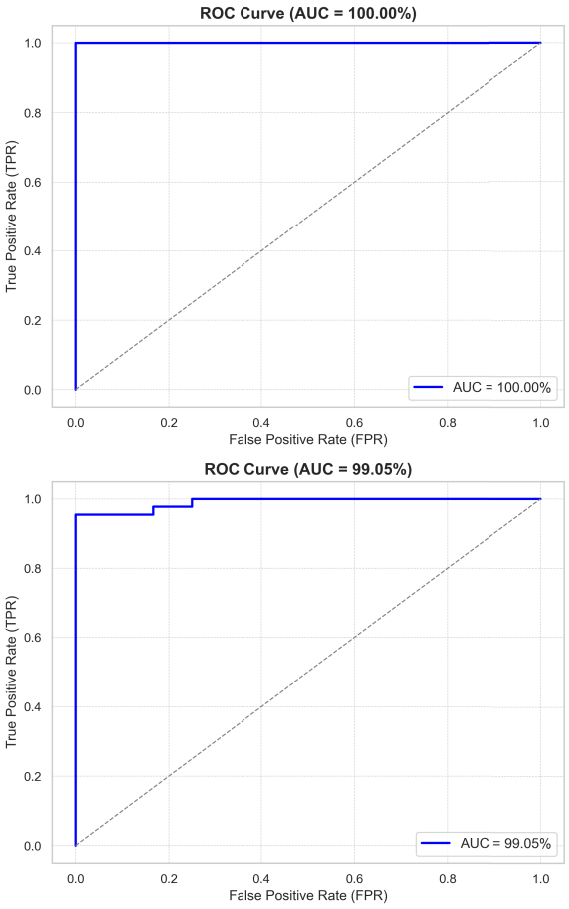


FIGURE 7. Stacking: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

F. BAGGING CLASSIFIER

Bagging, or Bootstrap Aggregating, is an ensemble learning technique that enhances both stability and accuracy by training several base models on various bootstrap samples and subsequently averaging their predictions. In this research, we employed Random Forest, Bagged Support Vector Machine (SVM), and Bagged K-Nearest Neighbors (KNN) as our base models. The model that yielded the highest performance was Bagged SVM, and its results are discussed comprehensively. Table 23 shows the accuracy of different bagging models used.

- *Classification Report:* Table 24 presents the classification reports for Dataset 1 and Dataset 2. The Bagging model demonstrates excellent classification performance on both datasets, achieving a high accuracy

TABLE 23. Accuracy comparison of bagging models.

Model	Dataset 1 Accuracy	Dataset 2 Accuracy
Random Forest	0.95	0.95
Bagged SVM (Best)	**0.975**	**0.975**
Bagged KNN	0.95	0.95

of 97%. Precision and recall values for both classes are consistently high, indicating a well-balanced classifier with minimal false positives and false negatives. The recall for class 0 is 1.00 across both datasets, meaning all instances of class 0 were correctly classified. Similarly, the precision for class 1 is 1.00, ensuring that every predicted instance of class 1 was indeed correct. The macro and weighted averages are nearly identical across both datasets, confirming the model's robustness and generalization ability.

Table 25 presents the confusion matrices for Dataset 1 and Dataset 2. The matrices provide a detailed breakdown of true positives, false positives, true negatives, and false negatives, offering insights into the model's classification performance. The model achieves a perfect classification for Class 0 in both datasets, as there are no false positives or false negatives for this class. In both datasets, there are only two misclassified instances for Class 1, indicating that the model performs well but has a slight tendency to misclassify a few cases. The high number of true positives and true negatives in both datasets suggests strong overall classification performance. The confusion matrices show nearly identical performance across the datasets, supporting the model's generalizability.

Table 26 presents the 5-fold cross-validation accuracy scores for Dataset 1 and Dataset 2. The results provide insight into the model's stability and generalization performance across different subsets of the data. The model achieves a high mean accuracy of approximately 97.18% for both datasets, indicating strong predictive performance. The standard deviation is low (0.0063 for Dataset 1 and 0.0062 for Dataset 2), demonstrating that the model's accuracy remains consistent across different folds. Both datasets show similar performance trends, suggesting that the model generalizes well to variations in the data. The slight variations in individual fold scores indicate minimal performance fluctuations, ensuring robust classification. The highest accuracy in Dataset 1 (0.9844) and Dataset 2 (0.9841) suggests that the model performs optimally on certain subsets, maintaining reliable classification results.

- ROC Curve with AUC: The ROC curves presented in Figure 8 illustrate the model's performance for both datasets. Both ROC curves exhibit a near-perfect shape, indicating excellent classification performance. The Area Under the Curve (AUC) values are close to 1, suggesting that the model effectively distinguishes between the classes. The sharp rise in TPR at very low FPR values shows that the model achieves high sensitivity with minimal false positives. The similarity in ROC curves for both datasets suggests that the model generalizes well across different data distributions.

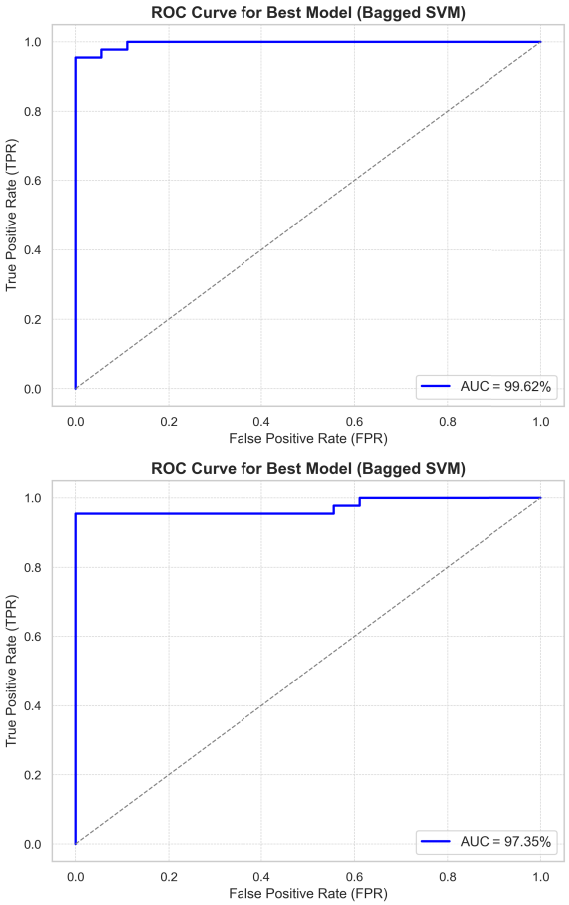


FIGURE 8. Bagging: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

G. GRADIENT BOOSTING CLASSIFIER (GBC)

GBC is an ensemble learning technique that builds a series of weak learners, typically represented as decision trees, and improves their accuracy through a process of iterative enhancement focused on minimizing the residual errors from previous iterations. This method utilizes gradient descent for optimization and is proficient in tackling both regression and classification tasks.

- *Hyperparameter:* Table 27 presents the best hyperparameters chosen for the GBC model based on model performance. The optimal number of estimators differs between datasets, with Dataset 1 requiring 150 while Dataset 2 performs best with only 50, indicating possible differences in dataset complexity. The learning rate remains consistent across both datasets at 0.01, suggesting a stable step size for model optimization. The maximum tree depth is set to 3 for both datasets, ensuring balanced model complexity while preventing overfitting.

TABLE 24. Bagging: Classification Report for Dataset 1 and Dataset 2.

Class	Dataset 1				Dataset 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	0.95	1.00	0.97	36	0.95	1.00	0.97	36
1	1.00	0.95	0.98	44	1.00	0.95	0.98	44
Accuracy	0.97				0.97			
Macro avg	0.97	0.98	0.97	80	0.97	0.98	0.97	80
Weighted avg	0.98	0.97	0.98	80	0.98	0.97	0.98	80

TABLE 25. Bagging: Confusion Matrices for Dataset 1 and Dataset 2.

Class	Dataset 1		Dataset 2	
	Pred 0	Pred 1	Pred 0	Pred 1
Actual 0	36	0	36	0
Actual 1	2	42	2	42

TABLE 26. Bagging: Cross-Validation Scores for Dataset 1 and Dataset 2.

Fold	Accuracy	
	Dataset 1	Dataset 2
1	0.9844	0.9688
2	0.9688	0.9688
3	0.9688	0.9688
4	0.9688	0.9688
5	0.9683	0.9841
Mean Accuracy	**0.9718**	**0.9718**
Standard Deviation	**0.0063**	**0.0062**

TABLE 27. GBC: Optimal Hyperparameters.

Hyperparameter	Dataset 1	Dataset 2
Number of Estimators	150	50
Learning Rate	0.01	0.01
Max Depth	3	3

- *Classification Report:* Table 28 presents the classification reports for both datasets, summarizing the model's precision, recall, F1-score, and overall accuracy. The classification model performs well in both classes, with high precision, recall, and F1-scores. Dataset 1 achieves an overall accuracy of 0.96, indicating reliable classification performance. The recall for class 1 (0.95) suggests a slight under-detection of positive instances compared to class 0. The macro and weighted averages remain balanced, ensuring consistent performance across classes.
  Table 29 presents the confusion matrices for both datasets, illustrating the classification model's performance in terms of correctly and incorrectly classified instances. The model achieves high accuracy, with minimal misclassifications for both datasets. In Dataset 1, only three misclassifications occur—one false positive (class 0 predicted as class 1) and two false negatives (class 1 predicted as class 0). Dataset 2 has a slightly higher number of false negatives (three instead of two), indicating a minor drop in sensitivity for class 1. The low false positive rate across both datasets confirms the model's reliability in identifying class 0 correctly.
  Table 30 presents the cross-validation scores for both datasets, showing the model's stability across multiple folds. The model achieves a high mean accuracy for both datasets, with Dataset 1 scoring 0.991 and Dataset 2 scoring 0.959. Dataset 1 exhibits a lower standard deviation (0.008) compared to Dataset 2 (0.021), indicating more consistent performance across different folds. Dataset 2 has slightly more variability, as seen in its accuracy fluctuations across different folds. The consistently high accuracy across all folds confirms the model's robustness and reliability.
- *ROC Curve with AUC:*
  Figure 9 presents the ROC curves for Dataset 1 and Dataset 2, illustrating the model's classification performance. The AUC for Dataset 1 is 96.35%, indicating strong classifier performance. The AUC for Dataset 2 is 96.02%, slightly lower but still highly effective. Both ROC curves are close to the top-left corner, confirming that the model maintains high sensitivity while minimizing false positives. The minimal difference in AUC values suggests that the model performs similarly across both datasets.

V. COMPARATIVE STUDY: DATASET 1 VS DATASET 2

This section presents an exhaustive comparison of two feature scaling approaches applied to the PLCO dataset:

- **Dataset 1:** Feature values transformed using *Min-Max Scaling*.
- **Dataset 2:** Feature values standardized using *Z-Score Normalization*.

The study evaluates how these transformations impact model performance, decision boundaries, feature interactions, and overall classification accuracy.

After feature selection techniques are applied to both datasets, Dataset 1 is reduced to 37 features, and Dataset 2 is reduced to 28 features where 20 features are common to both the datasets, including the target variable. The comparison of accuracy values for various models used is given in Table 31

The comparison between models trained on datasets with different normalization techniques reveals key differences in performance, feature selection, and computational efficiency. While both datasets retained a significant number of common features, differences in selected attributes suggest that normalization impacts feature importance and model interpretability. Model accuracies showed variations, indicating that normalization affects predictive power, with one method potentially leading to better generalization.

This study highlights a crucial but frequently overlooked aspect of clinical machine learning pipelines which is the choice of normalization technique. A comparative evaluation ML algorithms including KNN, SVM, Logistic Regression, MLP, Stacking, Bagged SVM, and Gradient

TABLE 28. GBC: Classification Report for Dataset 1 and Dataset 2.

Class	Image 1				Image 2			
	Precision	Recall	F1-Score	Support	Precision	Recall	F1-Score	Support
0	0.95	0.97	0.96	36	0.92	0.97	0.95	36
1	0.98	0.95	0.97	44	0.98	0.93	0.95	44
Accuracy	0.96				0.95			
Macro avg	0.96	0.96	0.96	80	0.95	0.95	0.95	80
Weighted avg	0.96	0.96	0.96	80	0.95	0.95	0.95	80

TABLE 29. GBC: Confusion Matrices for Dataset 1 and Dataset 2.

Actual Class	Dataset 1		Dataset 2	
	Predict 0	Predict 1	Predict 0	Predict 1
Actual 0	35	1	35	1
Actual 1	2	42	3	41

TABLE 30. GBC: Cross Validation Scores for Dataset 1 and Dataset 2.

Fold	Dataset 1 Accuracy	Dataset 2 Accuracy
1	1.000	0.915
2	0.984	0.924
3	0.984	0.936
4	0.984	0.920
5	1.000	0.920
Mean Accuracy	**0.991**	**0.959**
Standard Deviation	**0.008**	**0.021**

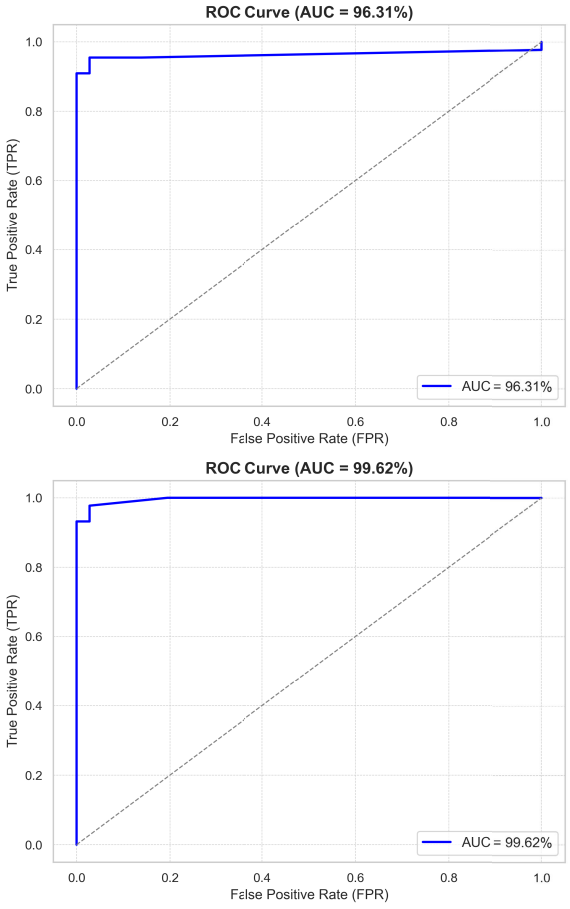


FIGURE 9. GBC: ROC Curves for Dataset 1 (top) and Dataset 2 (bottom).

Boosting Classifier demonstrates that performance metrics vary significantly between Min-Max scaling and Z-score normalization techniques. The Min-Max scaling consistently yielded marginally higher accuracies across most models, with the Stacking model achieving a highest 100% accuracy under this approach, compared to 97% with Z-score normalization. These findings highlights the importance of preprocessing decisions on model outcomes, particularly in sensitive to clinical prediction of ovarian cancer detection.

TABLE 31. Comparison of model accuracy.

Model Accuracy	Dataset 1	Dataset 2
K-NN	0.96	0.95
SVM	0.97	0.97
LR	0.97	0.96
MLP	0.96	0.93
Stacking	1.00	0.97
Bagged SVM	0.97	0.97
GBC	0.96	0.95

A LIME plot provides a visual breakdown of the contribution of each feature to the predicted outcome for a particular instance. Positive values in the plot indicate features that pushed the prediction towards the positive class (e.g., presence of cancer), while negative values indicate features that pushed the prediction away from it. In binary classification problems such as cancer detection, the positive class (Class 1) typically represents the presence of the condition being studied (e.g., malignancy). Therefore, LIME explanations are usually generated for Class 1 to understand which features contributed the most to the model predicting the presence of cancer.

The Figure 10 demonstrates the prediction of Ovarian Cancer on various Baseline classifiers on Datasets 1 and 2 using LIME plot. The most influential features for classifying this instance into class 1 are: scr_link $<=$ 0.00 for Dataset 1 and scr_link $<=$ 0.97 for Dataset 2 which is considered to be the strongest, followed by intstatusumm_cat_link $>$ 0.40 for Dataset 1 and intstatusumm_cat_link $>$ 0.48 for Dataset 2. The other features have minimal or no contribution, suggesting the prediction is heavily driven by scr_link and intstatusumm_cat_link features. It can also be observed that, tvu_ref has no contribution towards class 1 prediction except in LR model.

The Figure 11 demonstrates the prediction of Ovarian Cancer on various Ensemble classifiers on Datasets 1 and 2 using LIME plot. The most influential features for classifying this instance into class 1 are: scr_link which is considered to be the strongest for Datasets 1 and 2, followed by intstatusumm_cat_link on Stacking and Bagging classifier. Further, intstatusumm_cat_link is found to be the major contributor in predicting Ovarian Cancer as seen in

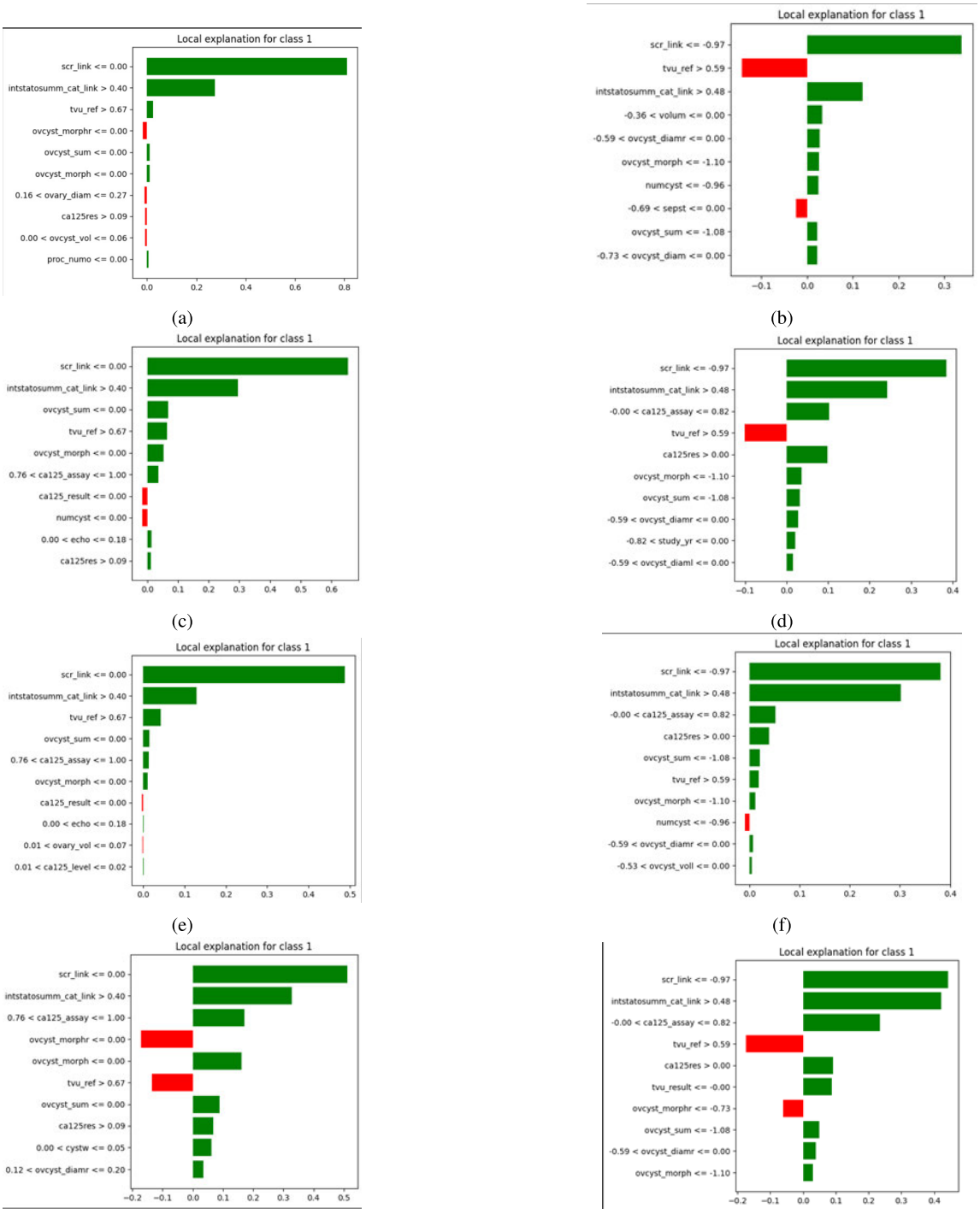


FIGURE 10. Prediction of Ovarian Cancer using Baseline models, 10a: KNN on Dataset 1, 10b: KNN on Dataset 2, 10c: SVM on Dataset 1, 10d: SVM on Dataset 2, 10e: LR on Dataset 1, 10f: LR on Dataset 2, 10g: MLP on Dataset 1, 10h: MLP on Dataset 2.

Figure 11e in contrast to its negative significance is observed in Figure 11f.

A. STATISTICAL ANALYSIS

Table 32 presents the results of the statistical analysis conducted on various machine learning models using t-statistics and their corresponding p-values. These values indicate whether the performance difference of each model, when compared to a population mean, is statistically significant.

The K-Nearest Neighbors (KNN) model shows a t-statistic of -1.5276 and a p-value of 0.2013, suggesting that its performance difference is not statistically significant at common confidence level of 0.05. Support Vector Machine (SVM) has a t-statistic of 2.4466 and a p-value of 0.0707.

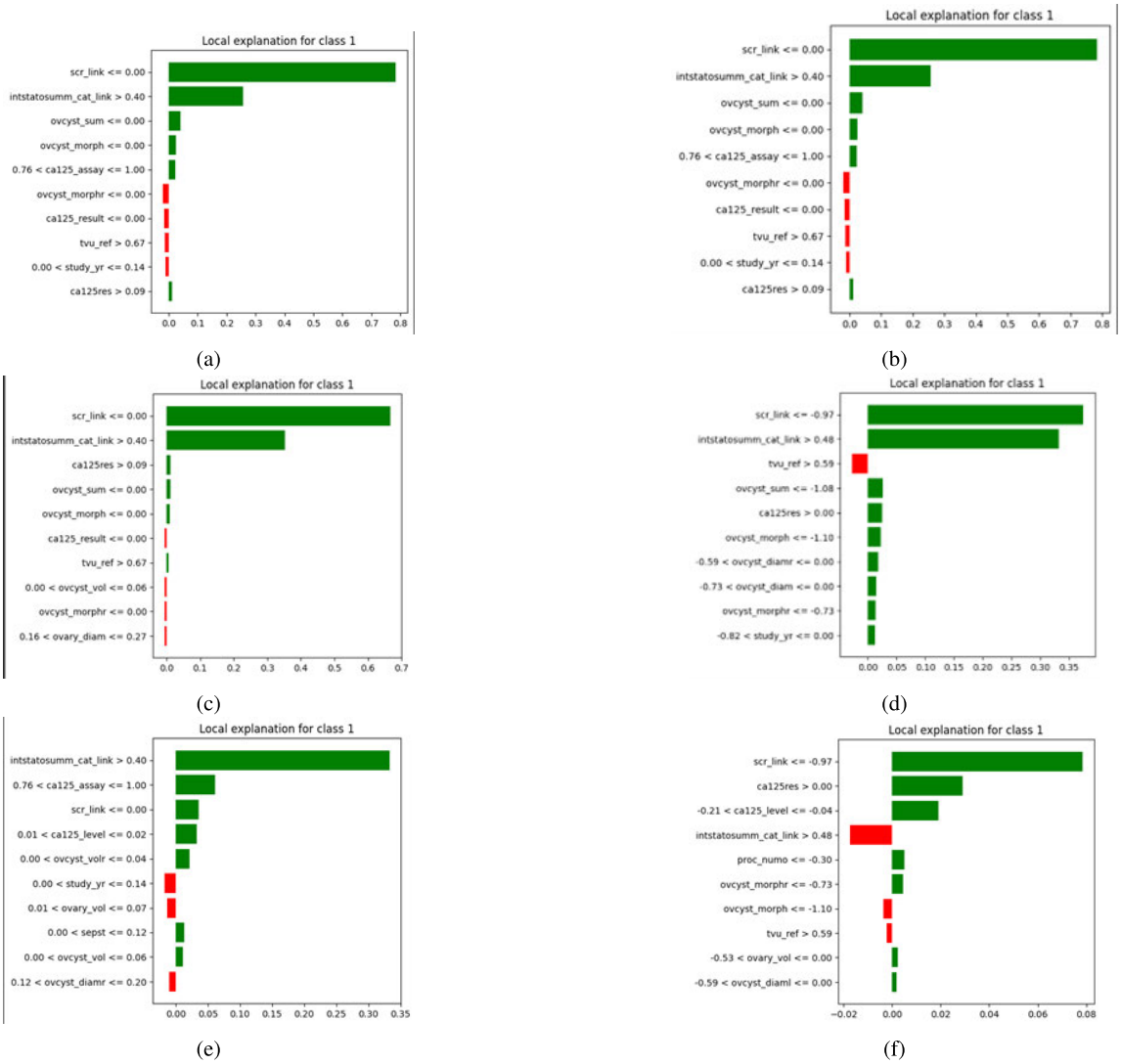


FIGURE 11. Prediction of Ovarian Cancer using Ensemble models, 11a: Stacking on Dataset 1, 11b: Stacking on Dataset 2, 11c: Bagging on Dataset 1, 11d: Bagging on Dataset 2, 11e: GBC on Dataset 1, 11f: GBC on Dataset 2.

TABLE 32. Statistical analysis of ML Models.

Models	t-statistics	p-value
KNN	-1.527595	0.201329
SVM	2.446600	0.070703
LR	1.509857	0.205591
MLP	0.219773	0.846439
Stacking	-0.040790	0.969417
Bagged SVM	-0.008056	0.993957
GBC	9.985459	0.000565

While this value is closer to the 0.05 significance threshold, it still does not reach statistical significance, though it might be considered marginally significant at a 0.10 level. Logistic Regression (LR) shows a t-statistic of 1.5099 and a p-value of 0.2056, which is also not statistically significant. The Multilayer Perceptron (MLP) and Stacking models show high p-values (0.8464 and 0.9694, respectively), indicating no statistically significant difference in their performance. Bagged SVM performs similarly with a p-value of 0.9940, indicating extremely low evidence against the null hypothesis. The Gradient Boosting Classifier (GBC) stands out with a t-statistic of 9.9855 and a highly significant p-value of 0.0006, well below the 0.01 threshold. This indicates a statistically significant improvement in performance compared to the baseline.

VI. CLUSTERING

To uncover hidden patterns in datasets, clustering groups related data points [43]. It aids in the identification of several cancer subtypes in the prediction of ovarian cancer. Clusters identify malignancy tendencies by examining a variety of features, which helps with early detection, individualised treatment, and improved prognosis prediction.

A. K-MEANS CLUSTERING

K-Means clustering is a widely used unsupervised learning technique that partitions a dataset into k clusters by minimizing intra-cluster variance [44]. It is a centroid-based

clustering method that iteratively refines cluster assignments to achieve compact and well-separated groupings.

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with n observations, K-Means aims to partition the data into k clusters $C = \{C_1, C_2, \ldots, C_k\}$, where each cluster C_i is represented by a centroid μ_i. The objective function to minimize is the **Within-Cluster Sum of Squares (WCSS) [45]:**

$$WCSS = \sum_{i=1}^{k} \sum_{x_j \in D_i} ||x_j - \mu_i||^2 \tag{6}$$

where:

- D_i represents the cluster i.
- μ_i is the centroid of the cluster i.
- $||x_j - \mu_i||^2$ is the squared Euclidean distance between a data point x_j and its assigned cluster centroid μ_i.

The standard K-Means algorithm consists of the following steps:

1) **Initialization:** Randomly initialize k centroids $\{\mu_1, \mu_2, \ldots, \mu_k\}$. A commonly used improvement over random initialization is the *K-Means++ algorithm*, which selects initial centroids in a way that improves convergence.
2) **Cluster Assignment:** Each data point x_j is assigned to the nearest cluster based on Euclidean distance:
$$D_i = \{x_j \mid ||x_j - \mu_i||^2 \leq ||x_j - \mu_l||^2, \forall l \neq i\} \tag{7}$$
3) **Centroid Update:** After assigning all points, the centroids are updated as the mean of all points in each cluster:
$$\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \tag{8}$$
4) **Convergence Check:** Steps 2 and 3 are repeated until the centroids no longer change significantly or a predefined number of iterations is reached.

Selecting the appropriate number of clusters is crucial to ensuring high-quality clustering results. In this study, two primary methods were employed to determine the optimal k:

The *Elbow Method* [46] evaluates the *Within-Cluster Sum of Squares (WCSS)* for different values of k. The optimal k is chosen at the "elbow point," where the rate of decrease in WCSS slows down:

$$WCSS = \sum_{i=1}^{k} \sum_{x_j \in D_i} ||x_j - \mu_i||^2 \tag{9}$$

A plot of WCSS against k typically shows an inflection point that indicates the best cluster count.

The *Silhouette Score* [47] evaluates clustering quality by measuring intra-cluster cohesion and inter-cluster separation:

$$S = \frac{d - c}{\max(c, d)} \tag{10}$$

where:

- c is the average distance from a point to other points in its own cluster.
- d is the average distance from a point to points in the nearest-neighbor cluster.

The optimal k maximizes the Silhouette Score, ensuring compact and well-separated clusters.

To enhance clustering performance, several refinements were applied:

a: INITIALIZATION STRATEGY

- The standard K-Means algorithm was improved using *K-Means++ initialization*, which selects diverse and well-spread initial centroids to avoid poor local minima.

b: STOPPING CRITERIA

- The algorithm was terminated when the *centroids no longer changed significantly* or after reaching a predefined *maximum iteration limit*.

c: DISTANCE METRIC

- *Euclidean distance* was used to measure similarity between points.

d: FINAL MODEL CONFIGURATION

- The *Elbow Method and Silhouette Score* were used to determine the optimal k.
- The final model was trained with the selected k, ensuring stable clustering performance.

B. DBSCAN CLUSTERING

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based unsupervised learning algorithm that clusters data points based on the density of their neighborhoods [48]. Unlike K-Means, DBSCAN does not require the number of clusters (k) to be specified in advance and is particularly effective in identifying clusters of arbitrary shape while handling noise effectively.

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with n observations, DBSCAN partitions data into clusters based on two primary parameters:

- *Epsilon (ϵ)*: The maximum radius within which points are considered part of the same cluster.
- *MinPts*: The minimum number of points required to form a dense region.

DBSCAN classifies points into three categories [49]:

- **Core Points**: Points with at least MinPts neighbors within radius ϵ.
- **Border Points**: Points within ϵ of a core point but with fewer than MinPts neighbors.
- **Noise Points (Outliers)**: Points that are neither core nor border points.

The DBSCAN algorithm follows these steps:

1) **Select an unvisited point** x_j.
2) **Determine the neighborhood** of x_j by counting points within distance ϵ.
3) **Classify the point:**
   - If x_j has at least MinPts neighbors, it is a **core point**, and a new cluster is created.

- If x_j is within ϵ of a core point but does not meet the MinPts criterion, it is classified as a **border point** and assigned to the nearest cluster.
- If x_j does not belong to any cluster, it is labeled as **noise**.

4) **Expand the cluster**: Recursively add all reachable core and border points to the cluster.
5) **Repeat until all points are visited**.

Selecting appropriate values for ϵ and MinPts is crucial for DBSCAN's effectiveness. In this study, the following techniques were used for parameter tuning:

The optimal ϵ was determined using the *k-distance plot*, which ranks each data point's distance to its k-th nearest neighbor and plots these values in ascending order. The optimal ϵ is identified at the *knee point*, where the distance rapidly increases.

The *MinPts* value was chosen using the heuristic:

$$\text{MinPts} = 2 \times d \tag{11}$$

where d is the dimensionality of the dataset. This ensures that clusters contain enough points to be considered significant.

To optimize clustering performance, several refinements are applied:

a: NEIGHBORHOOD DEFINITION
- The *k-distance plot* was analyzed to determine the appropriate ϵ.
- The heuristic MinPts $= 2d$ was used to set the minimum density threshold.

b: NOISE HANDLING
- DBSCAN inherently detects outliers as *noise points*, making it robust to datasets with high variance.

c: FINAL MODEL CONFIGURATION
- The final ϵ and MinPts values are selected based on empirical analysis.
- DBSCAN was applied to detect clusters with *arbitrary shapes*, ensuring optimal separation between noise and valid clusters.

VII. CLUSTERING IMPLEMENTATION: K-MEANS AND DBSCAN

This study applied two clustering algorithms—**K-Means** and **DBSCAN**—to identify patterns within the PLCO dataset (Dataset 1 and Dataset 2). Each method is tuned to optimize clustering performance and interpretability.

A. K-MEANS IMPLEMENTATION

1) OPTIMAL CLUSTER SELECTION
- The optimal number of clusters (k) is determined using:
  - **Elbow Method:** Plotted Within-Cluster Sum of Squares (WCSS) against different k values.
  - **Silhouette Score:** Measured intra-cluster cohesion and inter-cluster separation.
- The final k value is selected based on these evaluations.

2) DISTANCE METRIC AND CENTROID INITIALIZATION
- Euclidean distance is used as the similarity metric.
- K-Means++ initialization is used to improve convergence and avoid poor local minima.

3) FINAL MODEL CONFIGURATION
- The final clustering model is run with the selected k.
- Iterations continued until centroids stabilized or the maximum iteration limit is reached.

B. DBSCAN IMPLEMENTATION

1) PARAMETER TUNING
- The two main parameters—*Epsilon* (ϵ) and *MinPts*—are tuned using:
  - **k-Distance Plot:** to identify the optimal ϵ value at the "knee point."
  - **Heuristic Rule:** MinPts is set as $2d$, where d is the dataset's dimensionality.

2) CLUSTERING PROCESS
- Each data point is categorized into one of the following groups:
  - **Core Points:** Points having at least MinPts neighbors within a radius of ϵ.
  - **Border Points:** Points that are within ϵ of a core point but do not meet the MinPts requirement.
  - **Noise Points:** Points that do not belong to any cluster.
- Clusters are generated by progressively linking core points and their neighboring points.

3) FINAL MODEL CONFIGURATION
- The final ϵ and MinPts values are chosen based on empirical analysis.
- DBSCAN successfully identified clusters with arbitrary shapes and filtered out noise points.

VIII. CLUSTERING RESULTS

This section presents the clustering results for both **K-Means** and **DBSCAN** applied to the two Dataset 1 and Dataset 2.

Each dataset is clustered separately, and the results are evaluated based on clustering performance, key metrics, and resulting cluster structures.

A. K-MEANS CLUSTERING RESULTS

1) OPTIMAL NUMBER OF CLUSTERS

The optimal number of clusters (k) is determined using the *Elbow Method* and *Silhouette Score* for both datasets and the same is shown in Table 33.

TABLE 33. The optimal value of k and silhouette score.

Metric	Dataset 1	Dataset 2
Optimal k	19	16
Silhouette Score	0.5082	0.4335

2) CLUSTER VISUALIZATION

Figures 12 to 15 illustrate the clustering outcomes for Dataset 1 and Dataset 2 using K-Means. The left subfigures in each case represent the final cluster assignments, while the right subfigures show the inertia values across different cluster counts.

Lower inertia values indicate tighter cluster formations and the elbow point in the inertia graphs helped determine the optimal number of clusters. The differences suggest that normalization techniques impact how K-Means identifies distinct clusters.

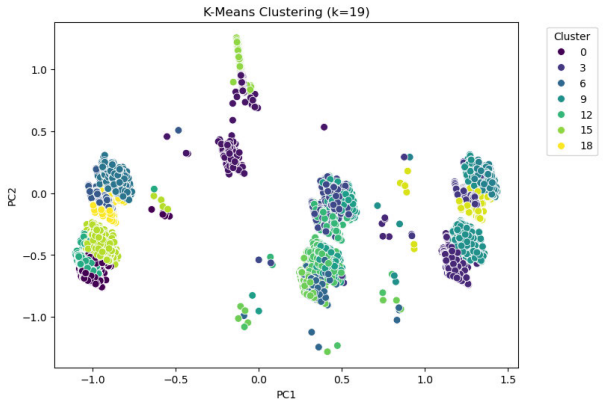


FIGURE 12. K-Means Cluster Assignments: Dataset 1.

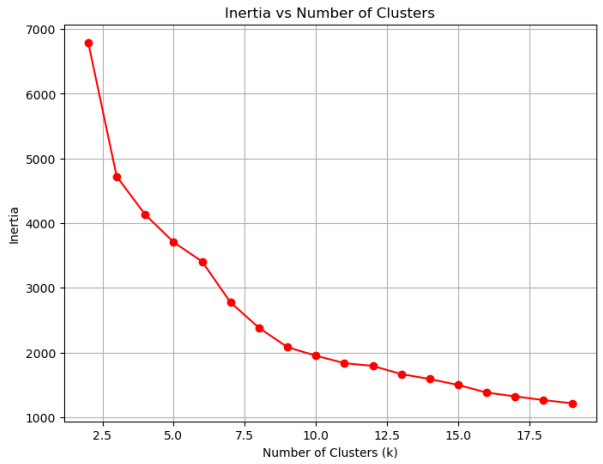


FIGURE 13. Inertia vs Number of Clusters: Dataset 1.

B. DBSCAN CLUSTERING RESULTS

The *DBSCAN* clustering algorithm is applied to both datasets. The key steps included:

- Selecting the optimal *epsilon* (ϵ) and *min_samples* using *k-distance plots*.
- Evaluating cluster formations and noise points.
- Comparing pre- and post-hyperparameter tuning results.

1) DBSCAN PARAMETER SELECTION

Optimal parameters for DBSCAN were selected based on hyperparameter tuning, maximizing the *Silhouette Score* while minimizing the number of noise points which is shown in Table 34.

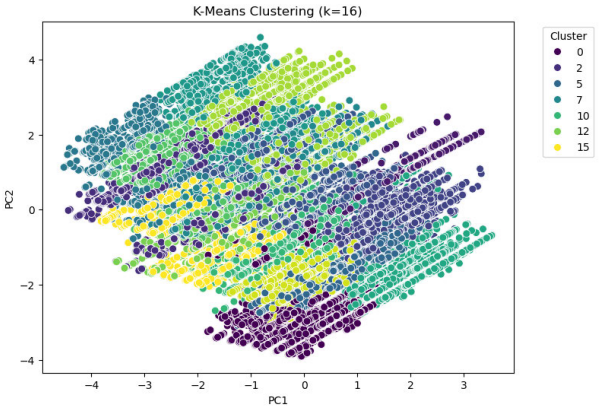


FIGURE 14. K-Means Cluster Assignments: Dataset 2.

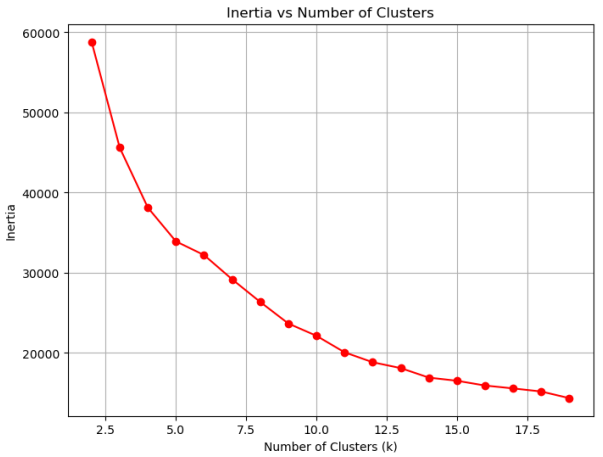


FIGURE 15. Inertia vs Number of Clusters: Dataset 2.

TABLE 34. Optimal DBSCAN Parameters.

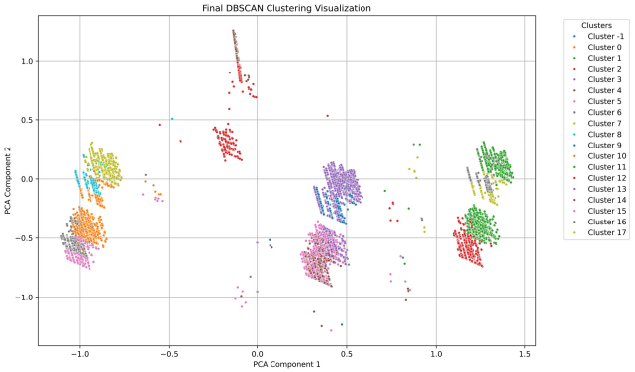
Parameter	Dataset 1 (Min-Max)	Dataset 2 (Z-Score)
ϵ	1.0556	1.1667
MinPts	50	50
Number of clusters	18	72
Noise points	2	615

2) FINAL CLUSTERING PERFORMANCE

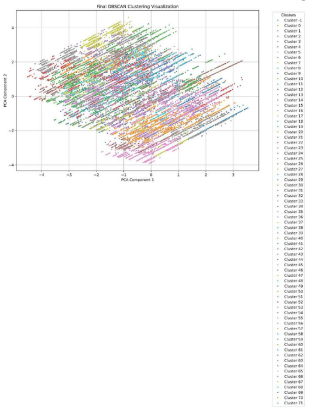
Table 35 shows that DBSCAN with Min-Max Scaling (Dataset 1) produces fewer clusters with minimal noise and a higher silhouette score. In contrast, Z-Score Scaling (Dataset 2) yields more clusters, higher noise, and a lower silhouette score, indicating weaker separation.

TABLE 35. DBSCAN Clustering Performance Metrics after Hyperparameter Tuning.

Metric	Dataset 1 (Min-Max)	Dataset 2 (Z-Score)
Number of Clusters	18	72
Number of Noise Points	1	518
Silhouette Score	0.7245	0.4736



(a) Dataset 1 - Min-Max Scaling



(b) Dataset 2 - Z-Score Scaling

FIGURE 16. **DBSCAN Clustering Visualization for Both Datasets.**

Figures 16a and 16b present the DBSCAN clustering outcomes for Dataset 1 (Min-Max Scaling) and Dataset 2 (Z-Score Scaling). These visualizations demonstrate the identified clusters based on density, with points labeled as noise where applicable.

Dataset 1 formed **18 clusters** with only **1 noise point**, whereas Dataset 2 resulted in **72 clusters** with **518 noise points** (Table 35). The higher noise level in Dataset 2 suggests that Z-Score Scaling may lead to sparser data distributions, impacting cluster density detection. Further, it can be observed that the clusters based on min-max scaling are more clinically interpretable and likely to align with known subtypes or diagnosis stages. However, Z-score scaling resulted in 72 clusters capturing subtle cleanical variations due to which it may be clinical insignificant or the clusters might have formed due to noise. It may identify rare patient subgroups and hence difficult to validate clinically without an expert input.

TABLE 36. **Comparison of feature selection under different scaling techniques.**

Feature Selection Aspect	Dataset 1 (Min-Max Scaling)	Dataset 2 (Z-Score Scaling)
Feature Count After Selection	37	28
Effect on Feature Importance Scores	Minimal	Slight Variations
Sensitivity to Outliers	High	Low
Preservation of Original Distributions	Yes	No
Influence of Feature Magnitude	Retained	Equalized

Table 36 highlights how Min-Max Scaling (Dataset 1) retains more features and preserves original distributions but is sensitive to outliers. In contrast, Z-Score Scaling (Dataset 2) reduces feature count, equalizes feature magnitudes, and minimizes outlier influence.

IX. CONCLUSION

This study explores supervised and unsupervised ML and DL approaches to improve predictive performance using clinical and biomarker-based data which was scaled through 2 popular techniques: Min-Max scaling and Z-Score normalization. Various baseline classifiers, including KNN, SVM, MLP, and LR, and ensemble methods like Stacking, Bagging, and Gradient Boosting, are tested, to analyze their efficacy on both datasets. Further,, unsupervised techiques like K-Means and DBSCAN clustering are implemented to study the subgroups in Ovarian Cancer dataset to optimize the diagnosis. The key takeaways of this study:

- **Random Forest feature selection is robust to normalization techniques**, as most of the selected features were same, however, feature scales also have an impact on feature importance
- **Feature importance scores showed variations** due to the rescaling effects of Z-score normalization.
- **Min-Max Scaling preserves original feature distributions**, making it useful when ability to interpret is a priority.
- **Z-Score Scaling reduces the influence of extreme values**, making it beneficial when handling datasets with a wide range of feature magnitudes.
- There is no outright best way to process the data. Although, model accuracies were higher on *Dataset 1, Dataset 2* had a lower standard deviation for k fold clustering.
- **SVM** offers similar results on both the datasets suggesting that the type of scaled data doesn't affect the model accuracy.
- **MLP** achieved 96% accuracy for *Dataset 1* however the network doesn't converge whereas it achieved 93% accuracy for *Dataset 2* and lead to convergence suggesting that *Dataset 2* is more suited for *MLP*.
- Data seems to be *biased* for **Stacking type ensembling** since 100% accuracy is achieved for *Dataset 2*.
- **Bagging** was found to be 97% accurate for both datasets suggesting their resilience to data scaling.
- In general, **Z-score** isn't suitable for most applications including clustering and is thus not used as the industry standard.
- Although, *Dataset 2* has a higher feature count, 37 after feature selection, model accuracies indicate a minor increase in model performance, with higher training times, especially if the number of records is higher.
- Imputing a large number of values, to the dataset may introduce bias, which may explain the perfect accuracy of the stacking model on dataset 1.

DATA AVAILABILITY

The data used in the study is a secondary dataset obtained from Cancer Data Access System (CDAS) and has to be collected after ethical clearance from National Cancer Institute (NCI).

REFERENCES

[1] A. S. Abdullah, A. G. Ahmed, S. N. Mohammed, A. A. Qadir, N. M. Bapir, and G. M. Fatah, "Benign tumor publication in one year (2022): A cross-sectional study," *Barw Med. J.*, vol. 2, no. 1, pp. 20–25, Nov. 2023.

[2] A. S. O'Shea, "Clinical staging of ovarian cancer," in *Methods in Molecular Biology*. Cham, Switzerland: Springer, 2022, pp. 3–10.

[3] V. Rojas, K. Hirshfield, S. Ganesan, and L. Rodriguez-Rodriguez, "Molecular characterization of epithelial ovarian cancer: Implications for diagnosis and treatment," *Int. J. Mol. Sci.*, vol. 17, no. 12, p. 2113, Dec. 2016.

[4] P. Bisoyi, "Malignant tumors—As cancer," in *Understanding Cancer*. Amsterdam, The Netherlands: Elsevier, 2022, pp. 21–36.

[5] P. Gaona-Luviano, L. A. Medina-Gaona, and K. Magaña-Pérez, "Epidemiology of ovarian cancer," *Chin. Clin. Oncol.*, vol. 9, no. 4, p. 47, 2020.

[6] X. Shu and Y. Ye, "Knowledge discovery: Methods from data mining and machine learning," *Social Sci. Res.*, vol. 110, Feb. 2023, Art. no. 102817.

[7] S. Zhang, C. Zhang, and X. Wu, *Knowledge Discovery in Multiple Databases*. Cham, Switzerland: Springer, 2004.

[8] M. W. Berry, A. Mohamed, and B. W. Yap, *Supervised and Unsupervised Learning for Data Science*. Cham, Switzerland: Springer, 2020.

[9] J. S. Abramowicz and D. Timmerman, "Ovarian mass-differentiating benign from malignant: The value of the international ovarian tumor analysis ultrasound rules," *Amer. J. Obstetrics Gynecol.*, vol. 217, no. 6, pp. 652–660, Dec. 2017.

[10] C. A. Hartman, C. R. T. Juliato, L. O. Sarian, M. C. Toledo, R. M. Jales, S. S. Morais, D. D. Pitta, E. F. Marussi, and S. Derchain, "Ultrasound criteria and CA 125 as predictive variables of ovarian cancer in women with adnexal tumors," *Ultrasound Obstetrics Gyneco.*, vol. 40, no. 3, pp. 360–366, Sep. 2012.

[11] X. He, X.-H. Bai, H. Chen, and W.-W. Feng, "Machine learning models in evaluating the malignancy risk of ovarian tumors: A comparative study," *J. Ovarian Res.*, vol. 17, no. 1, p. 219, Nov. 2024.

[12] C.-W. Wang, Y.-C. Lee, Y.-J. Lin, C.-C. Chang, A.-K.-O. Sai, C.-H. Wang, and T.-K. Chao, "Ensemble biomarkers for guiding anti-angiogenesis therapy for ovarian cancer using deep learning," *Clin. Transl. Med.*, vol. 13, no. 1, p. 1162, Jan. 2023.

[13] J. Fan, Y. Jiang, X. Wang, and J. Lyv, "Development of machine learning prognostic models for overall survival of epithelial ovarian cancer patients: A SEER-based study," *Expert Rev. Anticancer Therapy*, vol. 25, no. 3, pp. 297–306, Mar. 2025.

[14] Y. Tan, W.-H. Zhang, Z. Huang, Q.-X. Tan, Y.-M. Zhang, C.-Y. Wei, and Z.-B. Feng, "AI models predicting breast cancer distant metastasis using LightGBM with clinical blood markers and ultrasound maximum diameter," *Sci. Rep.*, vol. 14, no. 1, p. 15561, Jul. 2024.

[15] Y. Sun, J. Li, Y. Xu, T. Zhang, and X. Wang, "Deep learning versus conventional methods for missing data imputation: A review and comparative study," *Expert Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120201.

[16] J. Y.-L. Chan, S. M. H. Leow, K. T. Bea, W. K. Cheng, S. W. Phoong, Z.-W. Hong, and Y.-L. Chen, "Mitigating the multicollinearity problem and its machine learning approach: A review," *Mathematics*, vol. 10, no. 8, p. 1283, Apr. 2022.

[17] A. Kodipalli, V. S. Devi, S. Guruvare, and T. Ismail, "Explainable AI-based feature importance analysis for ovarian cancer classification with ensemble methods," *Frontiers Public Health*, vol. 13, Mar. 2025, Art. no. 1479095.

[18] S. Sinsomboonthong, "Performance comparison of new adjusted min-max with decimal scaling and statistical column normalization methods for artificial neural network classification," *Int. J. Math. Math. Sci.*, vol. 2022, pp. 1–9, Apr. 2022.

[19] H. Henderi, "Comparison of min-max normalization and Z-score normalization in the K-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer," *IJIIS: Int. J. Informat. Inf. Syst.*, vol. 4, no. 1, pp. 13–20, Mar. 2021.

[20] J. Yang, S. Rahardja, and P. Fränti, "Outlier detection: How to threshold outlier scores?" in *Proc. Int. Conf. Artif. Intell., Inf. Process. Cloud Comput.*, Dec. 2019, pp. 1–6.

[21] H. P. Vinutha, B. Poornima, and B. M. Sagar, "Detection of outliers using interquartile range technique from intrusion dataset," in *Proc. 6th Int. Conf. Ficta Inf. Decis. Sci.*, 2018, pp. 511–518.

[22] M. A. Islam, M. Z. H. Majumder, M. S. Miah, and S. Jannaty, "Precision healthcare: A deep dive into machine learning algorithms and feature selection strategies for accurate heart disease prediction," *Comput. Biol. Med.*, vol. 176, Jun. 2024, Art. no. 108432.

[23] R. Iranzad and X. Liu, "A review of random forest-based feature selection methods for data science education and applications," *Int. J. Data Sci. Anal.*, pp. 1–15, Feb. 2024, doi: 10.1007/s41060-024-00509-w.

[24] M. A. M. Hasan, M. Nasser, S. Ahmad, and K. I. Molla, "Feature selection for intrusion detection using random forest," *J. Inf. Secur.*, vol. 7, no. 3, pp. 129–140, 2016.

[25] F. H. Juwono, W. K. Wong, H. T. Pek, S. Sivakumar, and D. D. Acula, "Ovarian cancer detection using optimized machine learning models with adaptive differential evolution," *Biomed. Signal Process. Control*, vol. 77, Aug. 2022, Art. no. 103785.

[26] T. L. Octaviani, Z. Rustam, and T. Siswantining, "Ovarian cancer classification using Bayesian logistic regression," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 546, no. 5, Jun. 2019, Art. no. 052049.

[27] V. V. P. Wibowo, Z. Rustam, S. Hartini, F. Maulidina, I. Wirasati, and W. Sadewo, "Ovarian cancer classification using K-nearest neighbor and support vector machine," *J. Phys., Conf. Ser.*, vol. 1821, no. 1, Mar. 2021, Art. no. 012007.

[28] A. A. Safar, D. M. Salih, and A. M. Murshid, "Pattern recognition using the multi-layer perceptron (MLP) for medical disease: A survey," *Int. J. Nonlinear Anal. Appl.*, vol. 14, no. 1, pp. 1989–1998, 2023.

[29] S. Das, S. P. Nayak, B. Sahoo, and S. C. Nayak, "Machine learning in healthcare analytics: A state-of-the-art review," *Arch. Comput. Methods Eng.*, vol. 31, no. 7, pp. 3923–3962, Apr. 2024.

[30] S. Arukonda and R. Cheruku, "Nested genetic algorithm-based classifier selection and placement in multi-level ensemble framework for effective disease diagnosis," *Comput. Methods Biomechanics Biomed. Eng.*, vol. 28, no. 4, pp. 487–510, Mar. 2025.

[31] A. Arfiani and Z. Rustam, "Ovarian cancer data classification using bagging and random forest," *AIP Conf. Proc.*, vol. 2168, Feb. 2019, Art. no. 020046.

[32] N. Abuzinadah, S. Kumar Posa, A. A. Alarfaj, E. A. Alabdulqader, M. Umer, T.-H. Kim, S. Alsubai, and I. Ashraf, "Improved prediction of ovarian cancer using ensemble classifier and shaply explainable AI," *Cancers*, vol. 15, no. 24, p. 5793, Dec. 2023.

[33] A. S. Azar, S. B. Rikan, A. Naemi, J. B. Mohasefi, H. Pirnejad, M. B. Mohasefi, and U. K. Wiil, "Application of machine learning techniques for predicting survival in ovarian cancer," *BMC Med. Informat. Decis. Making*, vol. 22, no. 1, p. 345, Dec. 2022.

[34] A. de la Cruz Huayanay, J. L. Bazán, and C. M. Russo, "Performance of evaluation metrics for classification in imbalanced data," *Comput. Statist.*, vol. 40, no. 3, pp. 1447–1473, Aug. 2024.

[35] M. F. Amin, "Confusion matrix in binary classification problems: A step-by-step tutorial," *J. Eng. Res.*, vol. 6, no. 5, pp. 1–12, Dec. 2022.

[36] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1586–1594, Aug. 2020.

[37] A. Tharwat, "Classification assessment methods," *Appl. Comput. Informat.*, vol. 17, no. 1, pp. 168–192, Jan. 2021.

[38] R. C. de Amorim and B. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering," *Pattern Recognit.*, vol. 45, no. 3, pp. 1061–1075, Mar. 2012.

[39] X. Ding, J. Liu, F. Yang, and J. Cao, "Random radial basis function kernel-based support vector machine," *J. Franklin Inst.*, vol. 358, no. 18, pp. 10121–10140, Dec. 2021.

[40] F. Nie, Z. Hao, and R. Wang, "Multi-class support vector machine with maximizing minimum margin," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, 2024, pp. 14466–14473.

[41] Y. Geng, Q. Li, G. Yang, and W. Qiu, "Logistic regression," in *Practical Machine Learning Illustrated with KNIME*. Cham, Switzerland: Springer, 2024, pp. 99–132.

[42] A. Awad, T. J. En, T. M. Umadevi, K. Susan, J. N. C. Yi, and Z. A. A. Salam, "A comparison review of optimizers and activation functions for convolutional neural networks," *J. Appl. Technol. Innov.*, vol. 7, no. 3, p. 37, 2023.

[43] A. Malik and B. Tuckfield, *Applied Unsupervised Learning with R: Uncover Hidden Relationships and Patterns with K-Means Clustering, Hierarchical Clustering, and PCA*. Birmingham, U.K.: Packt Publishing, 2019.

[44] M. Chaudhry, I. Shafi, M. Mahnoor, D. L. R. Vargas, E. B. Thompson, and I. Ashraf, "A systematic literature review on identifying patterns using unsupervised clustering algorithms: A data mining perspective," *Symmetry*, vol. 15, no. 9, p. 1679, Aug. 2023.
[45] M. J. Brusco and D. Steinley, "A comparison of heuristic procedures for minimum within-cluster sums of squares partitioning," *Psychometrika*, vol. 72, no. 4, pp. 583–600, Dec. 2007.
[46] M. Cui, "Introduction to the K-means clustering algorithm based on the elbow method," *Accounting*, vol. 1, no. 1, pp. 5–8, 2020.
[47] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2020, pp. 747–748.
[48] P. Bhattacharjee and P. Mitra, "A survey of density based clustering algorithms," *Frontiers Comput. Sci.*, vol. 15, no. 1, pp. 1–27, Feb. 2021.
[49] M. Fuchs and W. Höpken, "Clustering: Hierarchical, k-means, dbscan," in *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications*. Cham, Switzerland: Springer, 2022, pp. 129–149.

ROOPASHRI SHETTY received the M.Tech. degree in computer science from Manipal Institute of Technology, Manipal, India. She is currently an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education. She has presented several papers in national and international conferences, and her work has been published in various international journals. Her current research interests include data mining and machine learning.

SIDDHANT GUPTA is currently pursuing the B.Tech. degree with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal. His current research interest includes data mining.

VANSH MEDIRATTA is currently pursuing the B.Tech. degree with the Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal. His current research interest includes data mining.

SHWETHA RAI received the M.Tech. degree in computer science from Manipal Institute of Technology, Manipal, India, and the Ph.D. degree in data mining from Manipal Academy of Higher Education, Manipal. She is currently an Assistant Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education. She has presented several papers in national and international conferences, and her work has been published in various international journals. Her current research interests include data mining and parallel computing.

M. GEETHA received the Ph.D. degree from NITK Surathkal, in 2010. She is currently a Professor with the Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India. She has presented several papers in national and international conferences, and her work has been published in several international journals. Her research interests include data mining, text mining in the healthcare, and financial sectors.